

DIANA
Finite Element Analysis

User's Manual

Analysis Procedures

Release 10.1

DIANA FEA BV

DIANA – Finite Element Analysis
User's Manual release 10.1
Analysis Procedures

Edited by: Jonna Manie and Wijtze Pieter Kikstra

Published by:
DIANA FEA BV
Delftechpark 19a, 2628 XJ Delft, The Netherlands.

Phone: +31 88 34262 00
Fax: +31 88 34262 99
E-mail: info@dianafea.com
Web page: dianafea.com

Trademarks.

DIANA is a registered trademark of DIANA FEA BV. FEMGV, FEMGEN, and FEMVIEW are trademarks of DIANA FEA BV. CADFIX is a registered trademark of TranscenData Europe Limited. FX⁺ is a registered trademark of Midas IT Corporation. Windows is a registered trademark of Microsoft Corporation. POSTSCRIPT, Adobe Reader is a registered trademark of Adobe Systems, Inc. AUTOCAD is a registered trademark of Autodesk Inc. DXF is a trademark of Autodesk Inc. ACIS is a registered trademark of Spatial Technology Inc. CADDs and PRO/ENGINEER are registered trademarks of Parametric Technology Corporation. CATIA is a registered trademark of Dassault Systemes S.A. IGES is a trademark of IGES Data Analysis, Inc. PARASOLID is software owned by Siemens Product Lifecycle Management Software Inc. © 1986-2016. PATRAN is a registered trademark of MSC Software Corporation. The X Window System is a trademark of M.I.T. UNIX is a registered trademark of UNIX Systems Laboratories, Inc. Intel is a registered trademark of Intel Corporation. HP is a registered trademark of Hewlett-Packard Company. NASTRAN is a registered trademark of the National Aeronautics and Space Administration. All other brand names, product names or trademarks belong to their respective holders.

First edition, February 15, 2017.

Copyright © 2016 by DIANA FEA BV, all rights reserved. No part of this publication may be reproduced in any form by print, photoprint, microfilm or any other means, without the prior written permission of the publisher.

The information in this document is subjected to change without notice and should not be construed as a commitment by DIANA FEA BV. DIANA FEA BV assumes no responsibility for any errors that may appear in this document.

The DIANA system is the sole property of DIANA FEA BV. Software materials made available are solely for use at a single site; they are not to be distributed to others without prior written permission of DIANA FEA BV.

This document was prepared with the L^AT_EX Document Preparation System.

Contents at a Glance

List of Figures	xxi
List of Tables	xxiii
Release Quality Process	xxv
Preface	xxvii
Glossary of Symbols	xxix
I General Input and Commands	1
1 Input of Finite Element Model	3
2 Input for Structural Analysis	15
3 General Commands	47
II Linear Static Analysis	69
4 Regular Linear Static Analysis	71
5 Fatigue Failure Analysis	97
III Dynamic Analysis	101
6 Input for Dynamic Analysis	103
7 Linear Transient Structural Analysis	121
8 Frequency Response Analysis	143
9 Response Spectrum Analysis	169
10 Hybrid Frequency Time Domain Analysis	189
11 Nonlinear Vibration Analysis	199
IV Nonlinear Static and Transient Analysis	207
12 Introduction to Nonlinear Structural Analysis	209
13 Nonlinear Structural Analysis	211
14 Strength Reduction Analysis	281

15 Engineering Liquefaction Analysis	287
16 Engineering Creep Analysis	291
V Stability Analysis	293
17 Introduction to Stability Analysis	295
18 Euler Stability Analysis	299
VI Potential Flow Analysis	311
19 Introduction to Potential Flow Analysis	313
20 Input for Potential Flow Analysis	315
21 Heat Flow Analysis	329
22 Groundwater Flow Analysis	347
23 Beam Cross-section Analysis	365
24 Reynolds Flow – Lubrication Analysis	371
VII Coupled Flow–Stress Analysis	375
25 Introduction to Flow–Stress Analysis	377
26 Staggered Analysis	379
27 Mixture Analysis	385
VIII Phased Analysis	397
28 Phased Structural Analysis	401
29 Phased Potential Flow Analysis	415
IX Solution Procedures	419
30 Solve System of Equations	421
31 Eigenvalue Analysis	425
X Reinforcement Grid Design Checking	441
32 Introduction to Reinforcement Grid Design Checking	445
33 Input for Reinforcement Grid Design Checking	447
34 Reinforcement Grid Design Checking Analysis	451
35 Result Definitions	459

XI	Stiffness Adaptation Analysis	463
36	Introduction to Stiffness Adaptation Analysis	465
37	Input for Stiffness Adaptation Analysis	469
38	Stiffness Adaptation Analysis	481
XII	Parameter Estimation	485
39	Introduction to Parameter Estimation	489
40	Input for Parameter Estimation	493
41	Parameter Estimation Analysis	505
42	Loads Optimization	513
43	Orthotropic Elastic Membrane	517
XIII	Background Theory	527
44	General Concepts of FEM	531
45	Solution Procedures of Sparse Linear Systems	539
46	Solution Procedures for Nonlinear Systems	551
47	Element Results and Nodal Forces	573
48	Dynamic Analysis	585
49	Hybrid Frequency Time Domain Analysis	601
50	Nonlinear Vibration Analysis	607
51	Strength Reduction Method	611
52	Engineering Liquefaction Analysis	613
53	Engineering Creep Analysis	615
54	Stability Analysis	617
55	Potential Flow Analysis	627
56	Soil–Pore Fluid Analysis	631
57	Fracture Mechanics Analysis	639
XIV	Appendix	643
A	Using User-supplied Subroutines	645
B	Available Element Types	671
C	Physical Properties Forms	675

Bibliography	677
--------------	-----

Index	683
-------	-----

Contents

List of Figures	xxi
List of Tables	xxiii
Release Quality Process	xxv
Preface	xxvii
Glossary of Symbols	xxix

I	General Input and Commands	1
1	Input of Finite Element Model	3
1.1	Units	3
1.1.1	Special Cases	3
1.1.2	Syntax	4
1.2	Model Information	6
1.3	Group Specification	7
1.3.1	Introduction	7
1.3.2	Syntax	9
1.4	Node Coordinates	11
1.5	Directions	11
1.6	Elements	12
1.7	Material Properties	12
1.8	Spatial Functions	12
2	Input for Structural Analysis	15
2.1	Rigid Supports	15
2.2	Linear Constraints	16
2.2.1	Nodes and Degrees of Freedom	16
2.2.2	Single- and Multi-point Tyings	16
2.2.3	General Input Syntax	17
2.2.4	Equalities	18
2.2.5	Interconnection	21
2.2.6	Eccentricity	23
2.2.7	General Connection	26
2.2.8	Automatic Tying	28
2.3	Loads	31
2.3.1	Nodal Load	33
2.3.2	Dead Weight	33
2.3.3	Model Equivalent Acceleration Load	34
2.3.4	Centrifugal Load	35
2.3.5	Fixed Displacements or Deformation	35
2.3.6	Mobile Loads	36
2.3.7	Wind and Water Load	45
2.3.8	Load Sets	45

2.3.9	Influence Fields	46
3	General Commands	47
3.1	Control Commands	47
3.2	FILOS File Maintenance	48
3.2.1	Tutorial Introduction	48
3.2.2	Command Syntax	49
3.3	Reading or Remaking Input Data	50
3.4	Element Evaluation	51
3.5	Elements Assembly	54
3.6	Output	55
3.6.1	Output Selection	56
3.6.2	Model Selection	59
3.6.3	Output for Postprocessing with DianaIE	63
3.6.4	Tabular Output	64
3.6.5	Output for Postprocessing with iDIANA	66
3.6.6	Output for Postprocessing with FX+	67
II	Linear Static Analysis	69
4	Regular Linear Static Analysis	71
4.1	Model Evaluation	72
4.2	Output of Analysis Results	73
4.2.1	Loads and Extreme Values	75
4.2.2	Displacements	76
4.2.3	Strains	78
4.2.4	Stresses	81
4.2.5	Hydrostatic Pressure Capacity	87
4.2.6	Nodal Forces	88
4.2.7	Internal Nodal Element Forces	89
4.2.8	Internal Element Forces	91
4.2.9	Model Parameters	91
4.2.10	Fracture Mechanics Parameters	95
5	Fatigue Failure Analysis	97
5.1	Number of Load Cycles to Failure	97
5.1.1	Stress Amplitudes	97
5.1.2	Output Selection	98
III	Dynamic Analysis	101
6	Input for Dynamic Analysis	103
6.1	Mass	103
6.2	Damping	103
6.3	Base Excitation	104
6.4	Modal Pushover Analysis	104
6.5	Transient Analysis	106
6.5.1	Initial Conditions	106
6.5.2	Prescribed Accelerations	107
6.5.3	Time–Load Diagram	108
6.5.4	Ambient Time Dependency	110
6.6	Frequency Domain Analysis	110
6.6.1	Frequency–Load Diagram	111

7	Linear Transient Structural Analysis	121
7.1	Model Evaluation	122
7.2	Type of Transient Analysis	123
7.3	Step Execution	123
7.3.1	Load Steps	124
7.3.2	Time Steps	124
7.3.3	No Equilibrium Iteration	125
7.4	Output of Analysis results	125
7.4.1	Displacements	126
7.4.2	Velocities	128
7.4.3	Accelerations	129
7.4.4	Strains	131
7.4.5	Stresses	133
7.4.6	Status	137
7.4.7	Nodal Forces and Moments	138
7.4.8	Nodal Element Forces	139
7.4.9	Element Forces	140
7.4.10	Dynamic Pressures	142
8	Frequency Response Analysis	143
8.1	Modal Response Analysis	144
8.1.1	Model Evaluation	145
8.1.2	Eigenmodes and Natural Frequencies	145
8.1.3	Response Analysis	146
8.1.4	Example Commands	148
8.2	Direct Response Analysis	149
8.2.1	Model Evaluation	150
8.2.2	Analysis Execution	152
8.3	Output of Analysis Results	152
8.3.1	Displacements	153
8.3.2	Velocities	155
8.3.3	Accelerations	157
8.3.4	Strains	159
8.3.5	Stresses	162
8.3.6	Nodal Forces and Moments	166
8.3.7	Dynamic Pressures	167
9	Response Spectrum Analysis	169
9.1	Model Evaluation	170
9.2	Eigenmodes and Natural Frequencies	170
9.3	Response Analysis	171
9.4	Output of Analysis Results	171
9.4.1	Displacements	174
9.4.2	Velocities	174
9.4.3	Accelerations	175
9.4.4	Strains	176
9.4.5	Stresses	180
9.4.6	Nodal Forces	183
9.4.7	Internal Nodal Element Forces	185
9.4.8	Internal Element Forces	186
9.4.9	Correlation Factors	187
10	Hybrid Frequency Time Domain Analysis	189
10.1	Model Evaluation	190
10.1.1	Element Matrices	190
10.2	Eigenmodes and Natural Frequencies	191
10.3	Response Analysis	192
10.3.1	Reduction of System of Equations	193

10.3.2	Analysis Execution	194
10.3.3	Logging	195
10.3.4	Output of Response Analysis Results	197
11	Nonlinear Vibration Analysis	199
11.1	Model Evaluation	200
11.2	Eigenmodes and Natural Frequencies	200
11.3	Nonlinear Vibration Analysis	201
11.3.1	Perturbation Analysis	201
11.3.2	Continuation Analysis	204
IV	Nonlinear Static and Transient Analysis	207
12	Introduction to Nonlinear Structural Analysis	209
12.1	Input for Nonlinear Analysis	209
12.2	Element Choice for Nonlinear Analysis	209
13	Nonlinear Structural Analysis	211
13.1	Model Evaluation	211
13.2	Type of Nonlinear Analysis	212
13.2.1	Physical Nonlinear Analysis	213
13.2.2	Geometrical Nonlinear Analysis	215
13.2.3	Transient Nonlinear Analysis	217
13.3	Step Execution	219
13.3.1	Initial State Evaluation	220
13.3.2	Load Steps	224
13.3.3	Time Steps	231
13.3.4	Physical Nonlinear Analysis Options	234
13.3.5	Equilibrium Iteration	235
13.3.6	Stop Criteria for Step Execution	240
13.3.7	Logging	243
13.3.8	Initial State References	244
13.3.9	Save/Restore Steps	245
13.4	Output of Analysis Results	246
13.4.1	Step Selection	248
13.4.2	Strains	248
13.4.3	Stresses	253
13.4.4	Status	260
13.4.5	Pore Pressures	268
13.4.6	Temperature, Concentration and Maturity	269
13.4.7	External Forces, Reactions and Residuals	270
13.4.8	Nodal Element Forces	271
13.4.9	Element Forces	272
13.4.10	Model Parameters	274
13.4.11	Fracture Mechanics Parameters	278
13.4.12	Dynamic Pressures	279
14	Strength Reduction Analysis	281
14.1	Step Execution	281
14.1.1	Steps	283
14.1.2	Equilibrium Iteration	284
14.1.3	Stop Criteria for Step Execution	285
15	Engineering Liquefaction Analysis	287
15.1	Steps	288
15.1.1	Iteration Based Adaptive Loading	289
15.1.2	Energy Based Adaptive Loading	289

16 Engineering Creep Analysis	291
V Stability Analysis	293
17 Introduction to Stability Analysis	295
17.1 Euler Stability Analysis	295
17.2 Perturbation Analysis	295
17.3 Continuation Analysis	296
17.4 Input of Initial Imperfections	296
17.5 Applicability of Elements	297
18 Euler Stability Analysis	299
18.1 Model Evaluation	299
18.2 Buckling Modes	300
18.2.1 Stability Analysis	301
18.2.2 Imperfections	302
18.2.3 Output of Buckling Modes and Values	303
18.3 Postbuckling Analysis	305
18.3.1 Perturbation Analysis	305
18.3.2 Continuation Analysis	308
VI Potential Flow Analysis	311
19 Introduction to Potential Flow Analysis	313
20 Input for Potential Flow Analysis	315
20.1 Nodal Potentials	315
20.1.1 Fixed Nodal Potentials	315
20.1.2 Initial Nodal Potentials	316
20.2 Boundary Conditions	317
20.2.1 Nodal Boundary Conditions	317
20.2.2 Transient Boundary Conditions	318
20.3 Linear Constraints	320
20.3.1 Nodes and Degrees of Freedom	320
20.3.2 Single- and Multi-point Tyings	321
20.3.3 General Input Syntax	321
20.3.4 Equalities	322
20.3.5 Interconnection	324
20.3.6 General Connection	325
21 Heat Flow Analysis	329
21.1 General Modelling Aspects	329
21.1.1 Meshing	330
21.1.2 Conductivity and Capacity	330
21.1.3 Heat Sources and Fixed Temperatures	330
21.1.4 Convection and Radiation at Boundaries	330
21.1.5 Conduction and Convection Coefficient	330
21.1.6 Temperature and Flow Fields	331
21.1.7 Special Features	331
21.2 Steady-state Analysis	332
21.2.1 Model Evaluation	333
21.2.2 Linear Steady-state Analysis	333
21.2.3 Nonlinear Steady-state Analysis	334
21.3 Transient Analysis	337
21.3.1 Model Evaluation	337
21.3.2 Initiate Transient Analysis	339
21.3.3 Time Steps Execution	340

21.4	Output of Analysis Results	343
21.4.1	Step Selection	344
21.4.2	Temperatures	344
21.4.3	Fluxes	344
21.4.4	Flows	345
21.4.5	Degrees of Reaction	345
21.4.6	Equivalent Age	346
21.4.7	Internal Temperature of Cooling Pipe	346
22	Groundwater Flow Analysis	347
22.1	Detailed Groundwater Flow	347
22.1.1	Meshing	347
22.1.2	Hydraulic Conductivity and Storativity	347
22.1.3	Sources and Sinks	347
22.1.4	Seepage Faces	348
22.1.5	Hydraulic Head and Flow Fields	348
22.1.6	Special Features	348
22.2	Regional Groundwater Flow	349
22.2.1	Meshing	349
22.2.2	Hydraulic Conductivity and Storativity	349
22.2.3	Sources and Sinks	349
22.2.4	Hydraulic Head and Flow Fields	349
22.2.5	Special Features	349
22.3	Steady-state Analysis	350
22.3.1	Model Evaluation	350
22.3.2	Linear Steady-state Analysis	351
22.3.3	Nonlinear Steady-state Analysis	352
22.4	Transient Analysis	355
22.4.1	Model Evaluation	355
22.4.2	Initiate Transient Analysis	357
22.4.3	Time Steps Execution	358
22.5	Output of Analysis Results	360
22.5.1	Step Selection	361
22.5.2	Hydraulic Heads	361
22.5.3	Fluxes	362
22.5.4	Flows	362
22.5.5	Pore Pressures	363
23	Beam Cross-section Analysis	365
23.1	General Modelling Aspects	365
23.1.1	Meshing	365
23.1.2	Elastic Properties	366
23.2	Analysis Commands	366
23.2.1	Model Evaluation	366
23.2.2	Cross-section Analysis Execution	367
23.3	Output of Analysis Results	368
23.3.1	Cross-sectional Properties	368
23.3.2	Shear Stresses	368
24	Reynolds Flow – Lubrication Analysis	371
24.1	General Modelling Aspects	371
24.1.1	Meshing	371
24.1.2	Viscosity	371
24.1.3	Boundary Conditions	372
24.1.4	Velocity	372
24.2	Analysis Commands	372
24.2.1	Model Evaluation	372
24.2.2	Lubrication Analysis Execution	372

24.3	Output of Analysis Results	373
24.3.1	Nodal Pressures	373
24.3.2	Fluxes	374
VII	Coupled Flow–Stress Analysis	375
25	Introduction to Flow–Stress Analysis	377
25.1	Staggered Solution with Pore Pressure	377
25.2	Staggered Solution with Temperature	378
25.3	Simultaneous Solution with Pore Pressure	378
26	Staggered Analysis	379
26.1	How to Perform Staggered Analysis	379
26.2	Model Definition	380
26.3	Potential Flow Analysis	380
26.4	Conversion to Structural Analysis	382
26.4.1	Heat and Concentration Flow to Structural	382
26.4.2	Groundwater Flow to Structural	382
26.4.3	Checking the Conversion	383
26.5	Structural Analysis	383
27	Mixture Analysis	385
27.1	Model Input	385
27.1.1	Mixture Elements	385
27.1.2	Input of Pore Pressure Potential	387
27.1.3	Input of Weight Load	389
27.2	Preliminary Linear Analysis	390
27.3	Transient and Nonlinear Analysis	390
27.3.1	Initialization	390
27.3.2	Initial Equilibrium	391
27.3.3	Load and Time Steps	392
27.4	Output of Analysis Results	393
27.4.1	Pressures	394
27.4.2	Nodal Discharges	395
27.4.3	Saturation	396
27.4.4	Darcy Flux	396
VIII	Phased Analysis	397
28	Phased Structural Analysis	401
28.1	Input of the Finite Element Model	402
28.1.1	Read First Model	402
28.1.2	Adapt Model	402
28.1.3	Import Deformation	402
28.1.4	Initial Velocity	404
28.2	Phase Initialization	404
28.2.1	Element Activation	406
28.2.2	Reinforcement Activation	406
28.2.3	Support Activation	407
28.2.4	Fixed Temperature Activation	407
28.2.5	Fixed Head Activation	407
28.2.6	Fixed Potential Activation	408
28.2.7	Tying Activation	408
28.2.8	Reinforcement	408
28.2.9	Displacement Constraints	408
28.2.10	Loads	409
28.3	Output of Displacements	409

28.4	Incrementing Analysis Results	409
28.4.1	Unchanged Model	409
28.4.2	Initial Stress	411
28.4.3	Model Changes	414
29	Phased Potential Flow Analysis	415
29.1	Implementation Backgrounds	416
29.1.1	Basic Equations	416
29.1.2	Solution	416
29.2	Input of the Finite Element Model(s)	417
29.3	Phase Initialization	417
29.4	Transient Analysis	417
29.4.1	Initialization	417
29.4.2	Time Steps	418
29.4.3	Aging	418
IX	Solution Procedures	419
30	Solve System of Equations	421
30.1	PARDISO – Parallel Direct Sparse Solver	421
30.2	Sparse Cholesky Based Solution Method	422
30.3	Out-of-Core Direct Solution	422
30.4	Iterative Solution	423
30.5	Substructuring	424
31	Eigenvalue Analysis	425
31.1	Model Evaluation	428
31.2	Type of Eigenvalue Problem	428
31.2.1	Free Vibration	429
31.2.2	Standard Eigenproblem	430
31.2.3	Linearized Buckling	431
31.3	Execute Eigenvalue Analysis	432
31.3.1	Arnoldi Method Based Eigenvalue Analysis	432
31.3.2	FEAST Method Based Eigenvalue Analysis	434
31.4	Calculate Rayleigh Damping Coefficients	435
31.5	Output of Analysis Results	436
31.5.1	Eigenvalues	437
31.5.2	Relative Errors	437
31.5.3	Generalized Mass	437
31.5.4	Participation Factors	438
31.5.5	Effective Mass	438
31.5.6	Modal Mass	438
31.5.7	Equivalent Mass	438
31.5.8	Transformation Factors	438
31.5.9	Modal Damping Factors	438
31.5.10	Eigenmodes	438
X	Reinforcement Grid Design Checking	441
32	Introduction to Reinforcement Grid Design Checking	445
32.1	Background	445
32.2	Results	445

33 Input for Reinforcement Grid Design Checking	447
33.1 Geometrical Properties	447
33.2 Material Properties	448
33.2.1 Reinforcement Steel	448
33.2.2 Concrete	448
34 Reinforcement Grid Design Checking Analysis	451
34.1 Eurocode 2 EN 1992-1-1 Design Checking	451
34.2 Linear Analysis	452
34.2.1 Model Evaluation	452
34.3 Load Case Combination	453
34.4 Load Envelope	454
34.5 Output of Analysis Results	454
35 Result Definitions	459
35.1 Reinforcement Area Checks	459
35.2 Diameter and Spacing Checks	461
35.3 Coverage Check	461
35.4 Shearforce Check	461
XI Stiffness Adaptation Analysis	463
36 Introduction to Stiffness Adaptation Analysis	465
36.1 Background	465
36.2 Stiffness Reduction Method	465
36.3 Load Increments	466
36.4 Comparison with Full Nonlinear Analysis	466
36.5 Element Types	467
37 Input for Stiffness Adaptation Analysis	469
37.1 Tensile Stress-Strain Curves	470
37.2 Compressive Stress-Strain Curves	474
37.3 No-tension Function for Interface Elements	479
38 Stiffness Adaptation Analysis	481
38.1 Model Evaluation	481
38.2 Step Execution	482
38.3 Output of Analysis Results	482
38.3.1 Crack Width	483
38.3.2 Stiffness and Model Parameters	483
XII Parameter Estimation	485
39 Introduction to Parameter Estimation	489
39.1 Some Background Theory	490
39.2 How to Use Module PAREST	491
40 Input for Parameter Estimation	493
40.1 Target Data	493
40.1.1 Material Points	495
40.1.2 Observables	495
40.1.3 Weighting the Observables	496
40.2 Parameter Estimation	496
40.2.1 Parameter Specification	499
40.2.2 Superparameters	500
40.2.3 Weighting the Parameters	501
40.3 User-supplied Subroutines	501

40.3.1	Superparameter USRPAR	501
40.3.2	Bounds USRBOU	503
41	Parameter Estimation Analysis	505
41.1	Preliminary Analysis	505
41.2	Analysis Commands	505
41.2.1	Linear Static Analysis Results	508
41.2.2	Nonlinear Analysis Results	508
41.2.3	Iteration	509
41.3	Output of Analysis Results	509
41.3.1	Job Logging	509
41.3.2	Estimation Report	510
42	Loads Optimization	513
42.1	Input Data	513
42.1.1	Target Data	513
42.2	Analysis Commands	514
42.2.1	Model Evaluation	514
42.2.2	Analysis Execution	514
42.3	Output of Analysis results	515
42.3.1	Estimation Report	515
42.3.2	Optimized Load	516
43	Orthotropic Elastic Membrane	517
43.1	Experimental Setup	517
43.2	Finite Element Model	518
43.2.1	Element Mesh	518
43.2.2	Material Data, Supports and Loads	519
43.3	Preliminary Analysis	519
43.3.1	Deformation	520
43.3.2	Stresses	520
43.4	Parameter Estimation Analysis	520
43.4.1	Additional Data	520
43.4.2	Analysis	525
43.4.3	Estimation Report	526
XIII	Background Theory	527
44	General Concepts of FEM	531
44.1	Global Formulation	531
44.1.1	Displacements	531
44.1.2	Strains and Stresses	531
44.1.3	Equilibrium	532
44.1.4	Principle of Virtual Displacements	532
44.2	Discretization to Elements	533
44.2.1	Displacements	533
44.2.2	Strains and Stresses	534
44.2.3	Element Assembly	534
44.2.4	Virtual Strain Energy	534
44.2.5	Element Stiffness Matrix	535
44.3	Assembling the Load Vector	536
44.4	Equilibrium	536

45	Solution Procedures of Sparse Linear Systems	539
45.1	Direct Solution Methods	539
45.1.1	Factorization and Fill-in	540
45.1.2	Sparse Cholesky Method	541
45.1.3	PARDISO – Parallel Direct Sparse Solver	541
45.2	Eigenvalue Procedures	541
45.2.1	Standard Eigenproblem	541
45.2.2	Generalized Eigenproblem	542
45.2.3	Shifting	544
45.2.4	Eigenvalues and Frequencies	545
45.2.5	Accuracy	545
45.2.6	Solution Techniques	546
45.3	Iterative Solution Methods	546
45.3.1	Conjugate Gradient	547
45.3.2	Generalized Minimal Residual	547
45.3.3	Schwarz Domain Decomposition	547
45.3.4	Preconditioning	548
45.3.5	Termination Criterion	549
45.3.6	Some Remarks on Practical Use	549
45.4	Substructuring	549
46	Solution Procedures for Nonlinear Systems	551
46.1	Incremental-Iterative Solution	551
46.1.1	Iterative Procedures	552
46.1.2	Continuation	557
46.1.3	Line Search	557
46.1.4	Convergence Criteria	558
46.1.5	Incremental Procedures	560
46.2	Geometric Nonlinearity	566
46.2.1	Definitions	567
46.2.2	Total Lagrange	568
46.2.3	Updated Lagrange	568
46.2.4	Nonconservative Loading	570
46.2.5	Contact Analysis	571
47	Element Results and Nodal Forces	573
47.1	Element Strains	573
47.1.1	Equivalent Von Mises Strain	573
47.1.2	Principal Strains	574
47.1.3	Volumetric Strain	574
47.1.4	Distributed Seismic Moment	574
47.2	Element Stresses	575
47.2.1	Equivalent Von Mises Stress	575
47.2.2	Principal Stresses	575
47.2.3	Plastic Yield	575
47.2.4	Pressure	576
47.2.5	Stress Invariants	576
47.2.6	Reinforcement Moments and Forces	576
47.2.7	Stress Gradient in Reinforcement Bar	581
47.2.8	Shear Stress in Reinforcement Mother Element Connection	581
47.2.9	Shear Capacity and Hydrostatic Pressure Capacity	581
47.3	Nodal Forces	583
47.3.1	Internal Nodal Forces	583
47.3.2	External Nodal Loads	583
47.3.3	Reaction Forces and Residual Forces	584

48 Dynamic Analysis	585
48.1 General Phenomena	585
48.1.1 Mass	585
48.1.2 Damping	586
48.1.3 Loading	586
48.1.4 Base Excitation	586
48.2 Frequency Response	588
48.2.1 Mode Superposition	589
48.2.2 Direct Solution	589
48.2.3 Analysis Results	590
48.3 Response Spectrum Analysis	590
48.4 Transient Response	592
48.4.1 Newmark	593
48.4.2 Euler Backward	593
48.4.3 Hilber–Hughes–Taylor	593
48.4.4 Wilson	593
48.4.5 Runge–Kutta	594
48.5 Fluid–Structure Interaction Analysis	595
48.5.1 Solid	596
48.5.2 Fluid	596
48.5.3 Discretized Coupled Equations	597
48.5.4 Solution of Coupled System	598
48.5.5 Simplification for Fixed Fluid Boundaries	599
49 Hybrid Frequency Time Domain Analysis	601
49.1 Hybrid Frequency Time Domain Method	601
49.2 Finite Element Equations	602
49.3 Solution Procedure Sequence	603
49.4 Fourier Transform Period	603
49.5 Time Segmentation	604
49.6 Decaying Functions	605
49.7 Convergence Criteria	606
50 Nonlinear Vibration Analysis	607
50.1 The Perturbation Method	607
50.2 Finite Element Implementation	610
51 Strength Reduction Method	611
51.1 Iterative Procedure for Factor of Safety	611
51.2 Nonlinear Analysis at Each Iteration	611
52 Engineering Liquefaction Analysis	613
53 Engineering Creep Analysis	615
54 Stability Analysis	617
54.1 Linear Buckling (Euler)	617
54.2 Imperfections	620
54.3 Postbuckling Analysis	621
54.4 Extensions of Postbuckling Analysis	622
54.4.1 Functional Notation	622
54.4.2 The Perturbation Method	622
54.5 Effect of Imperfection	623
54.5.1 Finite Element Implementation	624
54.5.2 Dynamic Buckling Analysis	625

55 Potential Flow Analysis	627
55.1 Basic Equations	627
55.1.1 Convection–Diffusion Equation	627
55.1.2 Boundary Conditions	627
55.1.3 Finite Element Formulation	627
56 Soil–Pore Fluid Analysis	631
56.1 Basic Equations and Assumptions	631
56.1.1 Porosity, Saturation and Apparent Density	631
56.1.2 Undrained Compressibility	632
56.1.3 Stress Separation and Pore Pressure Potential	632
56.1.4 Momentum Conservation	633
56.1.5 Mass Conservation	633
56.1.6 Darcy Flow	633
56.1.7 Elastoplastic Stiffness	634
56.2 Finite Element Equations	634
56.2.1 Space Discretization	634
56.2.2 Time Discretization	635
56.2.3 Undrained Behaviour	636
56.2.4 Hydraulic Pore Pressure Load	637
57 Fracture Mechanics Analysis	639
57.1 Stress Intensity Factor	639
57.1.1 Mesh Adaptation	640
57.2 Energy Release Rate	640
57.2.1 Two-dimensional Analysis	640
57.2.2 Three-dimensional Analysis	641
XIV Appendix	643
A Using User-supplied Subroutines	645
A.1 Supplying the Source Code	645
A.1.1 Building Shared Libraries	646
A.2 Available Routines	647
A.3 Programmer’s Service Libraries	648
A.3.1 Matrix Manipulation	648
A.3.2 Vector Manipulation	656
A.3.3 Miscellaneous	660
A.3.4 Database I/O	669
B Available Element Types	671
C Physical Properties Forms	675
C.1 Mobile	675
C.1.1 Dutch VBB Code	675
C.1.2 Dutch VOSB Code	676
C.1.3 European ENV Code	676
C.1.4 General	676
Bibliography	677
Index	683

List of Figures

2.1	Tying degrees of freedom	16
2.2	Multi-point tying	17
2.3	Equality tyings	19
2.4	Application of equality tyings	19
2.5	Multi-point equality	20
2.6	Single-point interconnection	21
2.7	Mesh refinement	21
2.8	Multi-point interconnection	22
2.9	User-specified eccentricity	23
2.10	Eccentricity in two-dimensional model	24
2.11	Multi-point eccentricity	25
2.12	General connections in two-dimensional space	27
2.13	Multi-point general connection	27
2.14	Automatic tying of connected elements	29
2.15	Automatic tying of loose elements	30
2.16	Mobile load definition	36
2.17	Ultimate truck positions	37
2.18	Mobile truck and distributed load	39
2.19	Lanes on a carriage way applied for VOSB code	40
2.20	Influence line in transverse direction	41
2.21	Lanes on a carriage way applied for ENV 1991-3 code	42
2.22	Mobile truck and distributed load according to the ENV 1991-3 code	43
3.1	Element shapes	52
3.2	Result averaging over a region with radius r according to JSCE	62
5.1	Stress Amplitude	97
8.1	Loading multiplication specification	144
9.1	Loading multiplication specification	170
13.1	Result based stop criteria	242
13.2	Crack status – linear tension softening	261
20.1	Tying degrees of freedom	320
20.2	Multi-point tying	321
20.3	Multi-point equality	323
20.4	Single-point interconnection	324
20.5	Mesh refinement	324
20.6	Multi-point interconnection	325
20.7	Multi-point general connection	327
23.1	Cross-section properties	369
27.1	Addition of pore pressure potential to continuum elements	386
27.2	Addition of pore pressure potential to interface elements	387

28.1	Phases with unchanged model	410
28.2	Increasing load F	411
28.3	Phases with unchanged model and initial stress	412
28.4	Increasing load F and initial stress σ^0	413
28.5	Addition of an element	414
36.1	Uniaxial stress-strain curve with stress and stiffness reduction	465
36.2	Flowchart of analysis sequence for stiffness adaptation analysis	467
37.1	Predefined tension softening for stiffness adaptation analysis	471
37.2	Predefined compression behaviour for stiffness adaptation analysis	475
39.1	Diagram of identification approach	489
43.1	Identification experiment	517
43.2	Finite element model	518
43.3	Deformed mesh	520
43.4	Stresses	521
45.1	Krylov-based iterative solution methods.	546
46.1	Iteration process	552
46.2	Regular Newton–Raphson iteration	553
46.3	Modified Newton–Raphson iteration	554
46.4	Quasi-Newton iteration	555
46.5	Linear Stiffness iteration	556
46.6	Continuation method (Linear Stiffness)	557
46.7	Line Search iteration	558
46.8	Norm items	559
46.9	Load and displacement control	560
46.10	Arc-length control	561
46.11	Work increment	564
46.12	Large displacements	566
46.13	Coordinate Frames	567
47.1	Reinforcement directions	577
47.2	Reinforcement moments \perp reinforcement	577
47.3	Forces in reinforcement directions	578
47.4	Shear stresses	579
47.5	Internal beam arm for bending only	580
47.6	Internal beam arm for bending plus compressive force	580
47.7	Internal beam arm for bending plus large tension force	580
48.1	Linear acceleration	594
48.2	Fluid–structure interaction	595
51.1	Flow-chart for the determination of FS	612
51.2	Mohr-Coulomb yield surface after strength reduction	612
52.1	Concept of engineering liquefaction analysis	613
54.1	Arch, loading and deformation	619
54.2	Arch, equilibrium node 3 deformed state	619
57.1	Loading modes	639
57.2	Virtual crack extension in three-dimensional analysis	641

List of Tables

1.1	CONSISTENT SYSTEMS OF UNITS	4
1.2	VALUES FOR CONSTANTS	4
2.1	MOBILE LOADS ACCORDING TO THE VOSB/VBB CODE	40
2.2	LANES AND CARRIAGE WAY ACCORDING TO THE ENV 1991-3 CODE . .	43
2.3	LOAD CLASSES ACCORDING TO THE ENV 1991-3 CODE	43
3.1	SHAPE PARAMETERS	52
4.1	AVAILABILITY OF STRAIN OUTPUT FOR REGULAR LINEAR STATIC ANALYSIS	79
4.2	AVAILABILITY OF STRESS OUTPUT FOR REGULAR LINEAR STATIC ANALYSIS	82
4.3	AVAILABILITY OF MODEL PARAMETER OUTPUT	92
7.1	AVAILABILITY OF STRAIN OUTPUT FOR LINEAR TRANSIENT ANALYSIS . .	132
7.2	AVAILABILITY OF STRESS OUTPUT FOR LINEAR TRANSIENT ANALYSIS . .	134
8.1	AVAILABILITY OF STRAIN OUTPUT FOR FREQUENCY RESPONSE ANALYSIS	160
8.2	AVAILABILITY OF STRESS OUTPUT FOR FREQUENCY RESPONSE ANALYSIS	163
9.1	AVAILABILITY OF STRAIN OUTPUT FOR RESPONSE SPECTRUM ANALYSIS .	177
9.2	AVAILABILITY OF STRESS OUTPUT FOR RESPONSE SPECTRUM ANALYSIS .	180
13.1	COMBINATIONS OF INITIAL CONDITION OPTIONS	223
13.2	AVAILABILITY OF STRAIN OUTPUT FOR NONLINEAR ANALYSIS	249
13.3	AVAILABILITY OF STRESS OUTPUT FOR NONLINEAR ANALYSIS	255
13.4	AVAILABILITY OF MODEL PARAMETER OUTPUT	275
17.1	STABILITY ANALYSIS FOR STRUCTURAL ELEMENTS	297
23.1	OUTPUT OF CROSS-SECTION PROPERTIES	368
26.1	ELEMENTS FOR STAGGERED FLOW-STRESS ANALYSIS	381
34.1	NATIONAL ANNEX PARAMETERS	452

Release Quality Process

This section sets out to illustrate the rigorous and ongoing development cycle patterns with respect to quality at DIANA FEA BV. While the program DIANA has always undergone stringent and regular tests, this outline tries to summarise the process and provide the user with a 'manageable' set of data on which to base risk criteria.

Method statement. DIANA FEA BV continually strives to produce software of the highest quality, reducing the variance of risk on incorrect calculation results to a minimal and meaningful level that gives confidence to end user, and developers alike, that the test suite inherent in the development process captures the various permutations that are possible while modifying, enhancing or adding to source code.

Disclaimer. At no time can DIANA FEA BV warrant that the software produced, like most software, is fully bug-free but what DIANA FEA BV can do is clearly declare that at DIANA FEA BV there is a continuous endeavour to achieve a robust and as near perfect product as possible.

The above statement is one of reality and does not diminish the efforts made at the company, by all employees at all times, to produce the best quality product possible with the minimum number of errors. This is, of course, in the best interest of the customer but is equally important to the efficient running and future of DIANA FEA BV with regard to support, development and future sales.

Current process. The current test suite that is run for each DIANA release contains around 5400 tests and examples. These files are run on all platforms for which the software is delivered, with a sequence of automated checks that report any deviation between anticipated and previous results. Within software development it is clear that a small change in one part of the code can equally modify a result elsewhere and by running this wide range of automated checks, such errors should be highlighted and rectified rapidly. To try and document all these checks for the customer would be unrealistic and the volume of output for each run, benchmarked against a standard test, will produce voluminous output. These tests and examples are available in the distribution of DIANA software. On a regular base DIANA FEA BV checks automatically that all functions in DIANA (element types, material models, loadings, solutions procedures etc.) are covered by the test suite. The user can retrieve relevant tests for specific functions via keywords by using the `Dtest` utility program that is delivered in the DIANA installation.

With consideration for the above, a series of easily identifiable and 'qualified' benchmarks have been set aside with reference to standard data tests such as those of NAFEMS, the *National Agency for Finite Element Methods & Standards* in the United Kingdom in order to define a high degree of clarity against known outcomes for the user. Tests are therefore made available in the distribution of DIANA software.

Release documentation. While release documentation already considers changes made to the software and outlines the new functionality, release documentation will also reference the known incompatibilities and changes that have taken place since the last major release.

Additional documentation. In setting up test documentation, there are essential references to known background theory and benchmark results for the test suite used at DIANA FEA BV. The suite of problems is available in the Verification Manual for the product. All tests used in the product are also available to users within the software distribution.

Preface

Part [I](#) of this volume formally presents the general input for the element model, the additional input for structural analysis and the general analysis commands. Parts [II](#) to [V](#) deal with the execution of the various types of structural analysis. Part [VI](#) and Part [VII](#) respectively tell you how to perform potential flow and coupled flow–stress analysis.

The next two parts describe features that are applicable for structural as well as for flow analysis: Part [VIII](#) tells you how to perform a phased analysis, Part [IX](#) outlines the application of various solution procedures.

The next four parts describe special application modules of the DIANA finite element code:

[X](#) *Reinforcement Grid Design Checking* — describes Module DESIGN for reinforcement grid design checking.

[XI](#) *Stiffness Adaptation Analysis* — describes Module STADAP for stiffness adaptation analysis.

[XII](#) *Parameter Estimation* — describes Module PAREST for parameter estimation, and the special Module BALANC for load optimization [[Ch. 42](#) p. [513](#)].

Part [XIII](#) gives background theory for many of the features described in the previous parts. This part also references publications, listed in the bibliography, for further reading. In the appendices of Part [XIV](#) you will find general information on DIANA like notation conventions in the manuals, how to run a job and programmers service libraries.

Related volumes. Novice users should read Volume *Getting Started* and browse through Volume *Analysis Examples* before going into details of this volume. Volume *Element Library* describes the elements that DIANA offers you to create a finite element model, including the necessary input for these elements like geometric properties, loading and specific options to customize the behaviour of the element.^{[1](#)} Volume *Material Library* is a reference manual for the available material models, it describes the input of these models and their basic background theory.

Cautionary note

Throughout this manual, it will be assumed that the reader has a basic understanding of computational mechanics and the Finite Element Method.^{[2](#)}

¹See [Appendix B](#) for a short overview of all available elements.

²Very informative introductions are the “Guidelines to Finite Element Practice” [[60](#)] and the book “A Finite Element Primer” [[62](#)], both published by NAFEMS.

Glossary of Symbols

Scalars ³

A	Area, cross-section [m ²].	K_p	Penalty conduction coefficient [s ⁻¹].
A	Projected area [m ²].	K_s	Solid compression modulus [N/m ²].
B	Bandwidth of matrix [-].	L	Area coordinates.
B	Boundary [m ²].	M	Bending moment (Bernoulli) [N · m].
C	Concentration.	M	Concentrated mass [kg].
C	Damping coefficient.	M	Concentrated moment [N · m].
C	Hardening constant.	M	Maturity.
C_c	Compression index.	M^R	Reaction moment [N · m].
C_d	Drag coefficient [-].	M^r	Residual moment [N · m].
D	Interface stiffness modulus [N/m ³].	N	Interpolation polynomial [-].
D	Linear stiffness modulus [N/m ²].	N	Normal force (Bernoulli) [N].
D^{II}	Crack shear stiffness [N/m ²].	N	Number of nodes [-].
E	Young's modulus [N/m ²].	N	Order of matrix [-].
E_{har}	Hardening modulus [N/m ²].	$\sigma\kappa\mathcal{R}$	Overconsolidation ratio [-].
E_p	Hardening modulus [N/m ²].	P	Perimeter [m].
F	Concentrated force [N].	P	Potential.
F	R.m.s. wavefront of matrix [-].	P	Pressure [N/m ²].
F^R	Reaction force [N].	P	Profile of matrix.
F^r	Residual force [N].	P_{ex}	External potential (LEFM).
G	Shear modulus [N/m ²].	Q	Discharge [m ³ /s].
G	Energy release rate (LEFM).	Q	Produced heat [J/m ³].
G_c	Compressive fracture energy [N/m].	Q	Second Biot material parameter [N/m ²].
G_f	Fracture energy [N/m ²].	Q	Shear force (Bernoulli) [N].
H	Enthalpy [J/m ³].	RH	Relative humidity [-].
H	Hardening parameter.	Re	Reynolds number [-].
I	Moment of inertia [m ⁴].	S	Boundary [-].
J	Creep function [m ² /N].	S	Degree of saturation [-].
K	Bulk or compression modulus [N/m ²].	S	Second Piola–Kirchhoff stress [N/m ²].
K	Conduction coefficient [W/(m ² · K)].	S	Shear stress correction factor [-].
K	Spring stiffness [N/m].	S	Surface area [m ²].
K	Stress intensity factor (LEFM) [-].	S_A	Spectral acceleration [m/s ²].
K_0	Lateral pressure ratio [-].	S_D	Spectral displacement [m].
K_D	Drained compression modulus [N/m ²].	T	Period of time [s].
K_f	Fluid compression modulus [N/m ²].	T	Temperature [K].
		T	Transmissivity.
		V	Volume [m ³].

³SI units in brackets.

W Maximum wavefront of matrix [-].	h Phreatic level [m].
W Moment of rigidity [m ³].	h_{cr} Crack bandwidth [m].
W Section modulus.	h_t Local thickness [m].
W Strain energy function [J].	i Counter [-].
W Normalized cumulative energy [-].	j Counter [-].
W_{in} Elastic energy (LEFM) [J].	k Conductivity [W/(m · K)].
W_k Specific kinetic energy [J/m ³].	k Creep factor [-].
X First global Cartesian coordinate [m], or axis.	k Interface stiffness modulus [N/m ³].
Y Second global Cartesian coordinate [m], or axis.	k Permeability [m ²].
Z Third global Cartesian coordinate [m], or axis.	k' Modified permeability [m ³ · s/kg].
a Rayleigh damping parameter [1/s].	k_p Conductivity of resistance layer [s ⁻¹].
a Mode amplitude.	l Length [m].
a Crack length (LEFM) [m].	m Distributed moment [N · m/m].
b Rayleigh damping parameter [s].	m' Reinforcement moment.
b Width [m].	n Degree of nonlinear elasticity [-].
c Capacitance [J/(m ³ · K)].	n Distributed in-plane force [N/m].
c Cohesion [N/m ²].	n Node number [-].
c Incompressibility penalty factor.	n Order of system matrix [-].
c Wave speed [m/s].	n Porosity [-].
c_A Arrhenius constant [K].	n Traction axis, normal.
c_e Elastic storativity.	p Isotropic strain invariant [-].
c_p Phreatic storativity [-].	p Polynomial.
co Reinforcement coverage [m].	p Pore pressure [N/m ²].
d Diameter [m].	p Pressure [N/m ²].
d Displacement height [m].	p' Effective pressure [N/m ²].
d Relative thickness [-].	p'_c Preconsolidation stress [N/m ²].
e Deviatoric strain [-].	p_e Excess pore pressure [N/m ²].
e Element number [-].	p_{head} Pressure head [m].
e Emissivity [-].	q Deviatoric strain invariant [-].
e Void ratio [-].	q Deviatoric stress [N/m ²].
f Distributed force [N/m, N/m ²].	q Distributed shear force [N/m].
f Frequency $f = \frac{\omega}{2\pi}$ [1/s].	q Flux.
f Volume source.	q Specific discharge, distributed source.
f_c Compressive strength [N/m ²].	r Degree of reaction [-].
f_{cc} Cube compressive strength [N/m ²].	r Distance from crack tip (LEFM) [m].
f_{ck} Characteristic strength [N/m ²].	r Maturity variable [-].
f_{cm} Mean compressive strength [N/m ²].	r Radius [m].
f_k Characteristic strength [N/m ²].	s Deviatoric stress [N/m ²].
f_n Natural frequency $f_n = \frac{\omega_n}{2\pi}$ [1/s].	s Length along crack front (LEFM) [m].
f_{sp} Proportionality yield stress [N/m ²].	s Traction axis.
f_{sy} Maximum yield stress [N/m ²].	t Thickness [m].
f_t Tensile strength [N/m ²].	t Time [s].
g Acceleration of gravity [m/s ²].	t Traction axis.
h Height [m].	t Traction [N/m ²].
	t_{eq} Equivalent age of concrete [s].
	u Translational displacement [m].
	\dot{u} Velocity [m/s].
	\ddot{u} Acceleration [m/s ²].

v Speed (wind, water) [m/s].	γ Modal participation factor.
v^* Friction wind speed [m/s].	γ Volumetric weight [N/m ³].
v_0 Specific volume [-].	γ_f Volumetric fluid weight [N/m ³].
v_c Convection velocity [m/s ²].	γ^p Deviatoric plastic strain.
x First local Cartesian coordinate [m], or axis.	δ Extension, elongation [m].
y Second local Cartesian coordinate [m], or axis.	δ Time integration parameter.
z Third local Cartesian coordinate [m], or axis.	δ_{ij} Kronecker delta [-].
z_0 Roughness height [m].	δt Change in Δt [s].
z_d Absolute internal beam arm [m].	ϵ Convergence criterion [-].
z_g Global Z coordinate of the ground [m].	ϵ Logarithmic strain [-].
z_r Relative internal beam arm [-].	ϵ Tolerance.
z_w Global Z coordinate of the water level [m].	ϵ Engineering strain [-].
\bar{z}_e Average global Z coordinate [m].	ϵ^C Concentration strain [-].
ΔC Concentration difference.	ϵ^T Thermal strain [-].
ΔM Maturity difference.	ϵ^{cr} Crack strain [-].
ΔP Pressure difference [N/m ²].	ϵ^c Creep strain [-].
ΔT Temperature difference [K].	ϵ^e Elastic strain [-].
Δt Time increment t [s].	ϵ^p Plastic strain [-].
Γ Boundary (surface).	ϵ^{sh} Uniaxial shrinkage strain [-].
Γ Euler gamma function.	ϵ_{vol}^{sh} Volumetric shrinkage strain [-].
Ω Domain.	ϵ_{st} Strain at which decay starts [-].
Ω Excitation frequency [Hz].	ϵ_{su} Strain at ultimate load [-].
Ω Rotation speed [rad/s].	ϵ_{sy} Strain at maximum yield stress [-].
Π Energy potential [J].	ϵ Emission coefficient [-].
Ψ Generalized strain.	ζ Third parametric coordinate.
α First Biot material parameter [-].	η Second parametric coordinate.
α General diffusivity.	η Viscosity.
α Generalized modal displacement.	θ Threshold angle.
α Preconditioning parameter.	κ Compression modulus.
α Shape factor.	κ Curvature (Bernoulli) [1/m].
α Thermal expansion coefficient [1/K].	κ Equivalent plastic strain [-].
α Time integration parameter.	κ Hardening parameter.
β Shear retention factor [-].	κ Swelling index.
γ Compressibility parameter [-].	λ Buckling value.
γ Concentration expansion coefficient.	λ Compression index.
γ Decay factor.	λ Eigenvalue.
γ Deviator strain [-].	λ Loading parameter.
γ Green–Lagrange strain [-].	λ Plastic multiplier [-].
γ Iteration parameter.	λ Relaxation time [s].
γ Shape factor.	μ Dynamic viscosity [N · s/m ²].
γ Shear deformation [-].	μ Friction coefficient (Coulomb).
γ Structural damping factor.	μ Shifting factor [-].
	μ Spring stiffness [N/m].
	ν Kinematic viscosity [m ² /s].
	ν Poisson’s ratio [-].
	ξ Damping ratio [-].
	ξ First parametric coordinate [-].
	ρ Mass density [kg/m ³].

- ρ_{dry} Mass density of dry soil [kg/m³].
 ρ_{f} Fluid density [kg/m³].
 ρ_{red} Reduced mass density [kg/m³].
 ρ_{sat} Saturated mass density [kg/m³].
 σ Stefan–Boltzmann constant [J/(m² · s · K⁴)].
 σ Stress (Cauchy) [N/m²].
 σ' Effective stress [N/m²].
 σ' In-situ stress [N/m²].
 σ^{cr} Crack stress [N/m²].
 σ_{eq} Equivalent stress (Von Mises) [N/m²].
 σ_{y} Yield stress [N/m²].
 τ Dimensionless time [–].
 v Curvature (Bernoulli).
 ϕ Friction angle.
 ϕ Pore pressure potential [N/m²].
 ϕ Potential.
 ϕ Rotational displacement.
 ϕ_1 Wobble factor [1/m].
 ϕ_{head} Hydraulic head [m].
 ϕ_{p} Pressure component of hydraulic head.
 φ Phase angle.
 ψ Dilatancy angle.
 ω Circular frequency $\omega = 2\pi f$ [1/s].
 ω Reinforcement percentage.
 ω Angular velocity (spin) [1/s].
 ω_{n} Natural circular frequency $\omega_{\text{n}} = 2\pi f_{\text{n}}$ [1/s].
 ∇C Concentration gradient.
 ∇M Maturity gradient.
 ∇P Pressure gradient [N/m³].
 ∇T Temperature gradient [K/m].

Vectors, Tensors

- \mathbf{a} Pseudo-acceleration vector [m/s²].
 \mathbf{f} Distributed forces.
 \mathbf{f} Equivalent element forces.
 \mathbf{f} Load vector.
 \mathbf{f}_{ex} External force vector.
 \mathbf{f}_{in} Internal force vector.
 \mathbf{f}_{R} Reaction force vector.
 \mathbf{f}_{r} Residual force vector.
 \mathbf{f}_{u} Force vector [N].
 \mathbf{f}_{w} Dead weight load.
 \mathbf{f}_{ϕ} Discharge vector [m³/s].
 \mathbf{g} Body forces per unit volume.
 \mathbf{g} Gravity acceleration vector [m/s²].
 \mathbf{g} Out-of-balance forces.
 \mathbf{i} Unity vector.
 \mathbf{k} Conductivity tensor.
 \mathbf{m} Distributed moments.
 \mathbf{n} Normal vector.
 \mathbf{n} Generalized element forces.
 \mathbf{p} Pressure vector.
 \mathbf{q} Flux vector [m/s].
 \mathbf{r} Internal forces, reactions.
 \mathbf{r} Residual vector.
 \mathbf{t} Traction [N/m²].
 \mathbf{u} Displacement vector [m].
 $\dot{\mathbf{u}}$ Velocity vector [m/s].
 $\ddot{\mathbf{u}}$ Acceleration vector [m/s²].
 \mathbf{x} Position vector [m].
 \mathbf{y} Result of forward substitution.
 \mathbf{E} Strain tensor.
 β Convective velocity field.
 γ Green–Lagrange strains.
 ε Strain vector.
 σ Total stress vector [N/m²].
 $\bar{\sigma}$ Inter-granular stress vector [N/m²].
 σ' Effective stress vector [N/m²].
 τ 2nd Piola–Kirchhoff stresses.
 ϕ Buckling modes.
 ϕ Eigenvector.
 ϕ Pressure potentials vector [N/m²].
 φ Phase angles.

Matrices

- \mathbf{B} Interpolation matrix.
 \mathbf{B} Strain–displacement relation.
 \mathbf{C} Capacity matrix.
 \mathbf{C} Compliance matrix.
 \mathbf{C} Coupling matrix.
 \mathbf{C} Damping matrix.
 \mathbf{C} Right Cauchy–Green stretch tensor.
 \mathbf{D} Diagonal matrix.
 \mathbf{D} Elasticity matrix.
 \mathbf{D} Material stiffness matrix.
 \mathbf{D} Rigidity matrix.
 \mathbf{E} Green–Lagrange strain tensor.
 \mathbf{F} Deformation gradient.
 \mathbf{H} Modified elastic stiffness matrix.
 \mathbf{I} Identity (unity) matrix.
 \mathbf{J} Jacobian matrix.
 \mathbf{K} Conductivity matrix.

K Permeability matrix.	K_B ...for the boundary.
K Stiffness matrix.	K_F ...for the fluid domain.
L Differential operator.	K_I ...for the interface.
L Lower triangular matrix.	K_S ...for the structural domain.
M Mass matrix.	σ_{123} Principal ...
N Interpolation matrix.	u_{XYZ} ...in global orientation.
P Preconditioning matrix.	u_{xyz} ...in local orientation.
R Rotation matrix.	u^(e) ...in element orientation.
S Second Piola–Kirchhoff stress tensor.	u^(s) ...in system orientation.
S Stress matrix.	σ_n Normal ...
S System directions.	σ_t Tangential ...
T Element directions.	K_e ...for an element.
T Transformation matrix.	S_n ...for a node.
U Upper triangular matrix.	f_{dy} Dynamic ...
Λ Eigenvalue diagonal matrix.	f_{ext} External ...
Accents, sub- and superscripts	
\bar{u} Average of ...	f_{int} Internal ...
\ddot{u} Second time derivative of ...	f_{st} Static ...
\dot{u} First time derivative of ...	K[*] Effective ...
\hat{u} Amplitude of ...	σ^0 Initial ...
\tilde{M} Added ...	$\Im u$ Imaginary part of ...
	$\Re u$ Real part of ...
	φu Phase angle of ...

Part I

General Input and Commands

Chapter 1

Input of Finite Element Model

The input data of a finite element model for a DIANA analysis are usually in a file on disk, the *input file*. This input file is a normal (ASCII) text file with 80 characters per line, which can be made in various ways: for instance via a text editor, or generated by means of a preprocessor program like `¡DIANA` [Vol. *iDIANA*] or `FX+` Volume *FX+ for DIANA*. See also Volume *Getting Started* for the rules of syntax and grammar for the DIANA input file.

1.1 Units

DIANA gives the output in the units that you chose for the input. You must choose a consistent system of units and enter all data in units from that system.

Because of its clear consistency we strongly advise to use the SI (Système International) unit system. However, other systems could also be used, for instance those shown in Table 1.1 on the following page. Because of its unnatural consistency for time, velocity and acceleration the ‘mm-kg-N’ system should be used with great care in time dependent analysis. Similarly, we advise against the use of customary inconsistent unit systems, like ‘US’ and ‘British Imperial’ in time dependent analysis. Table 1.2 on the next page gives some values for constants in the various consistent unit systems. See also the books on unit conversion by Horvath [45] and Cardarelli [15].

1.1.1 Special Cases

In some cases, outlined below, you must specify which system of units you used. You may specify the units of the input data via table ‘UNITS’ [§ 1.1.2]. If you do not, DIANA assumes the SI units system (m, kg, s, K) and angles in radians.

1.1.1.1 Model Codes

In some cases of DIANA input data, the material properties may be specified by the name of an international standard code, for instance `MC1990` for the modelling of concrete creep according to the European CEB-FIP Model Code 1990 as explained in Volume *Material Library*. Normally these codes require certain units to be used, for instance length in millimeters and time in hours. DIANA translates the units of the input data to the units of the code. Unless you use SI units for input data, you must specify your input units via table ‘UNITS’ [§ 1.1.2].

1.1.1.2 Steel Profiles

If you use the library of profiles to indicate the cross-section of beam elements, it is important to realize that DIANA converts the profile data from the units used in the system library files to the units used in the finite element model. It is good practice to specify table ‘UNITS’ whenever you use the profile library. If you do not, DIANA assumes

Table 1.1: CONSISTENT SYSTEMS OF UNITS

Unit	derivation	Unit system			
		SI	mm-kg-N	m-t-kN	CGS
Length	l	m	mm	m	cm
Mass	m	kg	kg	t	g
Time	t	s	s/ $\sqrt{1000}$!!	s	s
Temperature	T	K	K	K	K
Velocity	$l t^{-1}$	m/s	m/($\sqrt{1000} \cdot s$) !!	m/s	cm/s
Acceleration	$l t^{-2}$	m/s ²	m/s ² !!	m/s ²	Gal
Force	$m l t^{-2}$	N	N	kN	dyne
Stress	$m l^{-1} t^{-2}$	Pa	MPa	kPa	dyne/cm ²
Energy	$m l^2 t^{-2}$	J	mJ	kJ	erg
Angle	$m m^{-1}$	rad	rad	rad	rad

Unit	derivation	Unit system			
		mm-t-N	mm-kg-s	mm-N-day	m-N-day
Length	l	mm	mm	mm	m
Mass	m	t	kg	7.46496·10 ¹² kg	7.46496·10 ⁹ kg
Time	t	s	s	day	day
Temperature	T	K	K	K	K
Velocity	$l t^{-1}$	mm/s	mm/s	mm/day	m/day
Acceleration	$l t^{-2}$	mm/s ²	mm/s ²	mm/day ²	m/day ²
Force	$m l t^{-2}$	N	mN	N	N
Stress	$m l^{-1} t^{-2}$	MPa	mPa	MPa	Pa
Energy	$m l^2 t^{-2}$	mJ	μJ	mJ	J
Angle	$m m^{-1}$	rad	rad	rad	rad

Table 1.2: VALUES FOR CONSTANTS

Constant	Unit system				
	SI	mm-kg-N	m-t-kN	CGS	
Acceleration of gravity	9.81	m/s ²	9.81	9.81	981.
Water					
Mass density at 4°C	1.000·10 ³	kg/m ³	1.000·10 ⁻⁶	1.000	1.000
Dyn. viscosity at 20°C	1.005·10 ⁻³	Pa · s	31.78·10 ⁻⁹	1.005·10 ⁻⁶	10.05·10 ⁻³
Bulk modulus	2.174·10 ⁹	Pa	2.174·10 ³	2.174·10 ⁶	21.74·10 ⁹
Iron					
Thermal capacity	3.62·10 ⁶	J/(m ³ · K)	3.62	3.62·10 ³	36.2·10 ⁶
Thermal conductivity	79.	J/(s · m · K)	2.5	79·10 ⁻³	7.9·10 ⁶
Elasticity modulus	220·10 ⁹	Pa	220·10 ³	220·10 ⁶	2.20·10 ¹²

that the model is input in SI units, which may not be correct. See also Chapter *Beam Elements* in Volume *Element Library*.

1.1.2 Syntax

In some cases it is necessary to tell DIANA via table 'UNITS' in which units the input data is specified [§ 1.1.1].

syntax

'UNITS'

1 80
[LENGTH $ulen_w$]

```
[MASS umasw ]
[FORCE uforw ]
[TIME utimw ]
[TEMPER utemw ]
[ANGLE uangw ]
```

LENGTH specifies that the used unit of length is *ulen*: MM for millimeter, CM for centimeter, DM for decimeter, M for meter (the default), KM for kilometer, IN for inch, FT for foot, YD for yard, or MI for statute mile. [M]

MASS specifies that the used unit of mass is *umas*: G for gram, KG for kilogram (the default), T for ton (1000 kg), OZ for ounce (UK), LB for pound (UK), KLB for kilopound (UK) or LTN for ton (UK) or long ton (US). [KG]

Instead of the unit of mass, you may specify the unit of force. Specification of the mass unit overrules specification of the force unit, i.e., if you specify both MASS and FORCE then DIANA ignores the FORCE specification.

FORCE specifies that the used unit of force is *ufor*: N for newton (the default), KN for kilo-newton, DYN for dyne (10^{-5} N), KGF for kilogram-force (kgf), KP for kilo-pond (kgf), MT for metric-ton (MT), PDL for poundal (pdl), P for pond or gram-force (gf), LBF for pound-force (lbf) or KLBF for kilo-pound (klbf). To derive the unit of mass from the specified unit of force, DIANA applies an acceleration of gravity $g = 9.80665 \text{ m/s}^2$. [N]

TIME specifies that the used unit of time is *utim*: SEC for second (the default), MIN for minute, HOUR for hour, DAY for day, or YEAR for year (= 365 days). [SEC]

TEMPER specifies that the used unit of temperature is *utem*: CELSIU for degree Celsius ($0^\circ\text{C} = 273.15 \text{ K}$), KELVIN for kelvin (absolute temperature, the default) or FAHREN for degree Fahrenheit. [KELVIN]

ANGLE specifies that the used unit of angles is *uang*: RAD for radian (the default), or DEGREE for degree. [RAD]

Default (SI)		<i>file.dat</i>
'UNITS'		
LENGTH	M	
MASS	KG	
TIME	SEC	
TEMPER	KELVIN	
ANGLE	RAD	

If you do not specify a particular unit or no table 'UNITS' at all, DIANA assumes the SI units, see Table 1.1, which is equivalent to the example input above.

mm-g-N		<i>file.dat</i>
'UNITS'		
LENGTH	MM	
FORCE	N	

In the example above the unit of mass is set to gram, leading to the *mm-g-N* unit system, see Table 1.1.

m-t-kN		<i>file.dat</i>
'UNITS'		
FORCE	kN	

In the example above leads to the *m-t-kN* unit system, see Table 1.1. The unit of mass is *ton* (1000 *kg*).

CGS *file.dat*

'UNITS'
LENGTH CM
MASS G

In the example above leads to the CGS unit system, see Table 1.1. The unit of force is *dyne*.

mm-t-N *file.dat*

'UNITS'
LENGTH MM
FORCE N

In the example above leads to the *mm-t-N* unit system, see Table 1.1. The unit of mass is *ton* (1000 *kg*).

mm-kg-s *file.dat*

'UNITS'
FORCE kN

In the example above leads to the *mm-kg-s* unit system, see Table 1.1. The unit of force is *kN*.

mm-N-day *file.dat*

'UNITS'
TIME DAY
LENGTH MM
FORCE N

For nonlinear analysis of creep and shrinkage of concrete models, DIANA offers the model codes, e.g. CEB-FIP Model Code 1990, see for instance the *rcbeam* example in Volume *Concrete and Masonry Analysis*. The CEB-FIP regulations are commonly expressed in 'newton', 'millimeter', and 'day' for force, length, and time respectively. These can be conceived as the *extrinsic* set of units. The *intrinsic* set of units in DIANA are 'newton', 'meter', and 'second'. The mass unit is then determined as follows.

The extrinsic set of units for force is $\frac{mass \times length}{time^2}$

The intrinsic set of units for force is $\frac{kg \times m}{s^2}$

This results in

$$\frac{mass \times length}{time^2} \equiv 1 \frac{kg \times m}{s^2} \quad \therefore \quad \frac{mass \times 1 \text{ mm}}{(1 \text{ day})^2} = 1 \frac{kg \times m}{s^2}$$

$$\frac{mass \times 10^{-3} \text{ m}}{86400^2 \text{ s}^2} = 1 \frac{kg \times m}{s^2} \quad \therefore \quad mass = \frac{86400^2}{10^{-3}} \text{ kg} = 7.46496 \times 10^{12} \text{ kg}$$

1.2 Model Information

Table 'MODEL' defines parameters that are set at global level, i.e. for the entire finite element model.

Model information

syntax

'MODEL'

1	5	6	80
[GRAVAC	$gacc_r$		
[GRAVDI	dir_n		
[DENSFL	ρf_r		
[REFHEA	xco_r yco_r zco_r		
[RAYLEI	a_r b_r		

GRAVAC $gacc$ is the acceleration of gravity g .

[$g = -9.81\text{m/s}^2$]

GRAVDI dir is the direction number referring to table 'DIRECT' [§ 1.5 p. 11] for the direction of gravity. The default for axisymmetric, plane strain and other two-dimensional models is GRAVDI 2, i.e. the global Y direction. Other models by default use GRAVDI 3, i.e. the global Z direction.

DENSFL ρf is the fluid density ρ_f . By default the fluid density is the density of water.

($\rho_f \geq 0$)

REFHEA xco , yco , and zco are the XYZ coordinates of the origin \mathbf{x}_{ref} of the position vector. The origin of the position vector is used as the reference point for the total head definition. If you do not specify the origin, then DIANA assumes the origin of the XYZ coordinate system by default.

[$\rho_f = 1000\text{kg/m}^3$]

[$\mathbf{x}_{\text{ref}} = (0, 0, 0)$]

RAYLEI specifies Rayleigh damping: a and b are the coefficients a and b to set up the damping matrices \mathbf{C} according to

$$\mathbf{C} = a \mathbf{M} + b \mathbf{K} \quad (1.1)$$

For Rayleigh damping with lumped matrices, the second coefficient b must be equal to zero.

The gravity parameters in table 'MODEL' are also used for the transformation of the pressure heads to pore pressures in mixture and groundwater flow-stress analysis, and for the K_0 initial stress ratio calculation in a geotechnical analysis.

1.3 Group Specification

Specification of groups in a table of the input data file is an optional but powerful tool to reduce and simplify the input of element sets, reinforcement sets, and node numbers in commands and input tables.

1.3.1 Introduction

A name and a number may be assigned to a group of elements, nodes or reinforcements in input table 'GROUPS'.

Input

file.dat

```
'GROUPS'
ELEMEN
1 ROOF / "Roof left" "Roof right" /
```

This input data gives group number 1 the name ROOF and specifies that this group contains element sets "Roof left" and "Roof right".

Nesting*file.dat*

```

'GROUPS'
ELEMEN
  1  ROOF1  / "Roof1 left" "Roof1 right" /
  2  ROOF2  / "Roof2 left" "Roof2 right" /
  3  ROOFS  / ROOF1 ROOF2 /

```

Groups may be nested like in the example above. Nesting must be 'backward': groups ROOF1 and ROOF2 must be specified prior to group ROOFS.

Automatic numbering*file.dat*

```

'GROUPS'
ELEMEN
  0  FLOORS  / "Floor1" "Floor2" "Floor3" /
  0  WALLS   / "Walls1" "Walls2" "Walls3" /
  0  ROOF    / "Roof left" "Roof right" /

```

A group number zero indicates automatic numbering. The group number is incremented by one (highest group number + 1). In the example, FLOORS will be group one, WALLS group two and ROOF group three.

Reference from commands*file.dcf*

```

BEGIN SELECT
  BEGIN ELEMENTS 1 ROOF 17 20
  ...
  END ELEMENTS
END SELECT

```

Groups may be referenced in commands and input tables by name only. Specification of the group name refers to all the elements in that group.

Reference from input*file.dat*

```

'LOADS'
CASE 1
ELEMEN
  / 235-290(3) ROOF / 1
  LINE
  FORCE 100.
  DIRECT 2
  ...

```

Group reference from input tables is analogous to reference from commands.

Elements to nodes*file.dat*

```

'GROUPS'
ELEMEN
  1  FLOORS  / "Floor1" "Floor2" /

```

Groups of elements may be embedded in sets of nodes. Group FLOORS, although specified with elements may be referred in a node selection command.

file.dcf

```

BEGIN SELECT
  NODES FLOORS
END SELECT

```

This command selects all the nodes attached to the elements in group FLOORS.

1.3.2 Syntax

The syntax of table 'GROUPS' is as follows.

Element groups *syntax*

'GROUPS'			
ELEMEN			
1	5	6	80
<i>grnr</i> _{<i>n</i>}	<i>grnam</i> _{<i>s</i>}	/	/
		<i>elsets</i> _{<i>s...</i>}	
		<i>grnams</i> _{<i>s...</i>}	

ELEMEN is the subtable name for groups of elements. Subtable names may occur in arbitrary order and even more than once.

grnr is the group number. If *grnr* > 0 then it is unique over all subtables. A *grnr* = 0 indicates an automatic number: DIANA adds one to the current highest group number to get the number of the new group. Groups may be specified in arbitrary order and numbers may be skipped.

grnam is the group name, unique over all subtables.

Note that group names also must be different from set names.

The group name must start with a letter and may not have more than twenty characters. The group name END is not allowed. Actual contents of the group must be specified within slashes / /, with an arbitrary mixture of:

elsets element set names as defined in table 'ELEMEN' [Vol. *Element Library*].

grnams group names, referring to previously specified groups in the same subtable.

Reinforcement groups *syntax*

'GROUPS'			
REINFO			
1	5	6	80
<i>grnr</i> _{<i>n</i>}	<i>grnam</i> _{<i>s</i>}	/	/
		<i>resets</i> _{<i>s...</i>}	
		<i>grnams</i> _{<i>s...</i>}	

REINFO is the subtable name for groups of reinforcements. Subtable names may occur in arbitrary order and even more than once.

grnr is the group number. If *grnr* > 0 then it is unique over all subtables. A *grnr* = 0 indicates an automatic number: DIANA adds one to the current highest group number to get the number of the new group. Groups may be specified in arbitrary order and numbers may be skipped.

grnam is the group name, unique over all subtables.

Note that group names also must be different from set names.

The group name must start with a letter and may not have more than twenty characters. The group name END is not allowed. Actual contents of the group must be specified within slashes / /, with an arbitrary mixture of:

resets reinforcement set names as defined in table 'REINFO' [Vol. *Element Library*].

grnams group names, referring to previously specified groups in the same subtable.

Node groups

syntax

'GROUPS'			
<i>subtbl_w</i>			
1	5	6	80
<i>grnr_n</i>	<i>grnam_s</i>	/	/
		<i>nrs_{n...}</i>	
		<i>grnams_{s...}</i>	

NODES is the subtable name for groups of nodes. Subtable names may occur in arbitrary order and even more than once.

grnr is the group number. If *grnr* > 0 then it is unique over all subtables. A *grnr* = 0 indicates an automatic number: DIANA adds one to the current highest group number to get the number of the new group. Groups may be specified in arbitrary order and numbers may be skipped.

grnam is the group name, unique over all subtables.

Note that group names also must be different from set names.

The group name must start with a letter and may not have more than twenty characters. The group name END is not allowed. Actual contents of the group must be specified within slashes / /, with an arbitrary mixture of:

nrs node numbers as defined in table 'COORDI'.

grnams group names, referring to previously specified groups in the same subtable.

- Multiple specified numbers will be filtered out.
- Numbers may be specified in arbitrary order.
- No assumption shall be made of internal ordering of numbers in a group.

file.dat

```

'GROUPS'
ELEMEN
  1  FLOOR1      / "Floor1 left" "Floor1 right" /
  4  "FLOOR 2"   / "Floor2 left" "Floor2 right" /
  5  FLOORS      / FLOOR1 "FLOOR 2" "Floor3 left" "Floor3 right" /
NODES
  2  UPPEREDGE   / 1-10(2) /
  3  LOWEREDGE   / 2-10(2) /
  9  EDGE1        / 11-20(2) /
 10  EDGE2        / 12-20(2) /
REINFO
  6  BAR1         / "Reinfo1-1" "Reinfo1-2" "Reinfo1-3" /
  7  BAR2         / "Reinfo2-1" "Reinfo2-2" "Reinfo2-3" /
 13  GRID1        / "Reinfo3-1" "Reinfo3-2" "Reinfo3-3" /
 14  GRID2        / "Reinfo4-1" "Reinfo4-2" "Reinfo4-3" /

```

Notation convention. In syntax descriptions, the possibility to refer to groups in a range of numbers is indicated with a subscript *g* behind the variable:

syntax

numbrs_{ngs...}

An arbitrary mixture of numbers, group and set references may be specified for variable *numbrs*.

1.4 Node Coordinates

Nodal coordinates are input via table 'COORDI'. The coordinates are expressed in global Cartesian XYZ axes.

syntax

'COORDI' [DI= <i>dimens</i> _{<i>n</i>}]				
1	5	6		80
<i>node</i> _{<i>n</i>}	<i>x</i> _{<i>r</i>}	<i>y</i> _{<i>r</i>}	<i>z</i> _{<i>r</i>}	

DI=*dimens* is an optional parameter which indicates the dimensionality of the nodal coordinate system. Specify DI=3 for a three-dimensional, or DI=2 for a two-dimensional system. The default is three-dimensional.

[DI=3]

node is the node number. Nodes may be input in arbitrary order, numbers may be skipped. Values *x*, *y* and *z* are the model XYZ coordinates. You must omit *z* for a two-dimensional system but specify it for a three-dimensional system.

[*z* = 0]

Two-dimensional coordinates

file.dat

'COORDI' DI=2			
1	1.0	0.5	
3	2.0	0.0	

In the above example, for both nodes $Z = 0$!

Three-dimensional coordinates

file.dat

'COORDI'			
3	2.0	3.1	5.4
1	1.6E-2	0.32	7.4E+2

1.5 Directions

Input table 'DIRECT' contains directions which are specified as vectors in the model coordinate system [§ 1.4]. The directions in table 'DIRECT' have no meaning in themselves. They are used for reference from other tables, for example in order to specify the direction of a support or a nodal loading.

syntax

'DIRECT'				
1	5	6		80
<i>dirnr</i> _{<i>n</i>}	<i>x</i> _{<i>r</i>}	<i>y</i> _{<i>r</i>}	<i>z</i> _{<i>r</i>}	

dirnr is the direction number. Directions may be input in arbitrary order, numbers may be skipped on the condition that they are not referred to from other tables. Values *x*, *y* and *z* are the model XYZ components of the direction vector. It is not necessary that the direction vectors are unit vectors, i.e., they do not have to be normalized.

Model XYZ directions (default)

file.dat

'DIRECT'			
1	1.0	0.0	0.0
2	0.0	1.0	0.0
3	0.0	0.0	1.0

DIANA assumes three default directions: number 1 the model X axis, number 2 the model Y axis and number 3 the model Z axis. This is equivalent to the input shown above. You may overwrite the default directions by specifying other values in table 'DIRECT', like in the following example.

Specified directions					<i>file.dat</i>
'DIRECT'					
1	0.0	1.0	0.0		
2	0.0	2.0	0.0		
4	1.0	1.0	0.0		
5	1.0	0.3	4.0		
6	0.0	-1.3	0.0		

In the above example 1 is the model Y axis (overwrites the default!), 2 is also the model Y axis, 4 is 45° with X and Y axis, 5 is an arbitrary direction in three-dimensional space and 6 is the negative model Y axis. In this case direction 3 still represents the (default) Z axis.

1.6 Elements

For input of the elements see Volume *Element Library*.

1.7 Material Properties

For input of the material properties see Volume *Material Library*.

1.8 Spatial Functions

This section describes the input data and its syntax for spatial functions for material and geometry properties and loads. Spatial functions can be used to define position dependency. Spatial functions can be 1-, 2-, or 3-dimensional, i.e. they dependent on 1, 2, or 3 global coordinate components, and are specified grid wise, i.e. one multiplication factor for each coordinate component combination.

Note that linear interpolation is being used inside the functions to calculate the multiplication factor at a specific position. If the position is outside the range of the function an error message is given.

Spatial functions		<i>syntax</i>
'FUNCTI'		
1		80
NAME <i>funnam_s</i>		
{		
X <i>xvals_{r...}</i> /		
Y <i>yvals_{r...}</i> /		
Z <i>zvals_{r...}</i> /		
FACTOR <i>mfacs_{r...}</i> /		

'FUNCTI' is the table heading for the spatial function input.

NAME *name* is the name for the spatial function and may be used for identification of and reference to the spatial function. A name specification line must follow immediately after the table heading. The name specification may be respecified at the start of a new subtable.

X *xvals* are the global X coordinate values for which the multiplication factors are specified. Note that the global X coordinate values need to be provided in ascending order.

Y *yvals* are the global Y coordinate values for which the multiplication factors are specified. Note that the global Y coordinate values need to be provided in ascending order.

Z *zvals* are the global *Z* coordinate values for which the multiplication factors are specified. Note that the global *Z* coordinate values need to be provided in ascending order.

FACTOR *mfacs* are the multiplication factors for the corresponding global *X*, *Y*, and/or *Z* coordinate component values dependent on the dimensionality of the spatial function. Factors need to be input grid wise. The multiplication factors are applied to the value of the material, geometry, or load parameter to which the function is attached.

1-dimensional*file.dat*

```
'FUNCTI'
NAME "Depth"
Z      0.   1.   2.   3.   5.  10.  20. /
FACTOR 100. 90.  80.  70.  50.  20.   0.           # x1 x2 ... x7
```

2-dimensional*file.dat*

```
'FUNCTI'
NAME "Horizontal"
X      0.  10.  20.  30.  40.  50. /
Y      0.   1.   2.   3. /
FACTOR 1.   2.   3.   4.   5.   6.           # x1y1 x2y1 ... x6y1
        2.   3.   4.   5.   6.   7.           # x1y2 x2y2 ... x6y2
        3.   4.   5.   6.   7.   8.           # x1y3 x2y3 ... x6y3
        4.   5.   6.   7.   8.   9. /         # x1y4 x2y4 ... x6y4
```

3-dimensional*file.dat*

```
'FUNCTI'
NAME "Box"
X      0. 10. 20. 30. /
Y      0. 1.  2. /
Z      10. 20. /
FACTOR 1.  2.  3.  4.           # x1y1z1 x2y1z1 ... x4y1z1
        2.  3.  4.  5.           # x1y2z1 x2y2z1 ... x4y2z1
        3.  4.  5.  6.           # x1y3z1 x2y3z1 ... x4y3z1

        10. 20. 30. 40.           # x1y1z2 x2y1z2 ... x4y1z2
        20. 30. 40. 50.           # x1y2z2 x2y2z2 ... x4y2z2
        30. 40. 50. 60. /         # x1y3z2 x2y3z2 ... x4y3z2
```

Spatial functions may be attached to specific loads [Vol. *Analysis Procedures*, Vol. *Element Library*] and to specific material [Vol. *Material Library*] and geometry properties [Vol. *Element Library*].

Function attachment to loads is defined in table 'LOADS':

Function attachment to load*syntax*

'LOADS'			
1	5	6	12 13
FUNCTI		<i>funnam_s</i>	

FUNCTI describes that a spatial function with name *funnam* as defined in table 'FUNCTI' is being attached to a load, which must be one of the following:

NODAL nodal force and moment loads [§ 2.3.1].

DEFORM prescribed displacements [§ 2.3.5].

ACCELE prescribed accelerations [§ 6.5.2].

ELEMEN LINE distributed force loads on truss elements, infinite shell elements and axisymmetric shell elements; distributed force and moment line loads on beam elements [Vol. *Element Library*].

ELEMEN EDGE distributed force edge loads on plane stress elements, plane strain elements, axisymmetric elements, and solid elements; distributed force and moment edge loads on plate bending, flat shell elements, and curved shell elements [Vol. *Element Library*].

ELEMEN FACE distributed force face loads on plane stress elements, plane strain elements, axisymmetric elements, and solid elements; distributed moment face loads on plate bending elements, distributed force and moment face loads on flat shell elements and curved shell elements [Vol. *Element Library*].

ELEMEN VOLUME distributed force volume loads on solid elements [Vol. *Element Library*].

ELEMEN PRESSU distributed pore pressure load [Vol. *Element Library*].

Note that spatial functions can only be attached to these loads if there is only a single value specified as load value.

Example.

Distributed line load on beam elements

file.dat

```
'LOADS'
CASE 1
NODAL
/ 7-12 / FORCE 2 2.
      FUNCTI NODALX
'FUNCTI'
NAME NODALX
X      0. 1. 4.   5. 10. /
FACTORS 2. 2. 2.25 2.5 2.5. /
```

This example describes a position dependent nodal force load elements by defining a 1-dimensional spatial function named NODALX. This spatial function gives a nonlinear relationship in global X direction and is attached to the force load (FORCE), which is a nodal load in direction 2 acting on nodes 7 to 12. At $X = 0$ the force equals $F(0) = 2. \times 2. = 4.$, at $X = 1$ the force equals $F(1) = 2. \times 2. = 4.$, at $X = 4$ the thickness equals $F(4) = 2.25 \times 2. = 4.5$, etc.

Chapter 2

Input for Structural Analysis

This chapter describes the input data and its syntax for structural analysis. For input of elements in table 'ELEMEN' and reinforcements in table 'REINFO' see Volume *Element Library*. For input of material properties in table 'MATERI' see Volume *Material Library*. For general aspects of input syntax see Volume *Getting Started*.

2.1 Rigid Supports

In table 'SUPPOR' the rigid supports are given. Rigid supports have displacements equal to zero, unless a fixed displacement is specified in subtable DEFORM of table 'LOADS' [§ 2.3.5 p. 35]. Each support is specified by a node number a type (translation or rotation) and a direction.

				<i>syntax</i>
'SUPPOR'				
NAME <i>name</i> _s				
1	5	6		80
<i>node</i> _n	<i>type</i> _w	<i>dirnrs</i> _{n...}	{ <i>type</i> _w <i>dirnrs</i> _{n...} ...}	
1	5	6		80
/ <i>nodes</i> _{ng...} /				
	<i>type</i> _w	<i>dirnrs</i> _{n...}	{ <i>type</i> _w <i>dirnrs</i> _{n...} ...}	

NAME *name* is the name for the support set and may be used for identification of and reference to the support set. A name specification line must follow immediately after the table heading. The name specification may be respecified at the start of a new subtable.

node is a single node number, *nodes* is a series of nodes with the same support type and direction. The series of nodes must be specified between slashes and may comprise numbers and/or groups.

The support type is specified with *type*: TR for translation or RO for rotation (clamped). The support direction is specified with a number referring to table 'DIRECT' [§ 1.5]. The translation in, or the rotation around the specified direction is prevented. Direction numbers *dirnrs* may be input with a single number or with a series of numbers. The series means the same type of support for multiple directions.

Supports input		<i>file.dat</i>
'SUPPORTS'		
NAME	VERTICAL	
12	TR 3	
NAME	LEFT	
24	TR 2 RO 1	

```
NAME SKEW
/ 25-38 / TR 4
/ 50-56 EDGE1 / TR 1 2 RO 3
```

2.2 Linear Constraints

Linear constraints are user specified linear *dependencies* between degrees of freedom of the system of equations (displacements, rotations, temperatures etc.). These dependencies are specified in input table 'TYINGS', and in DIANA terms called *tyings*.

Some examples of the application of tyings are: hinges, sliding connections, symmetry, mesh refinement, connection in case of incompatibility, eccentric connection, keeping edges or planes straight. Before you go into detail on these topics, keep in mind what NAFEMS [60] says on this point:

“Constraint equations or displacement transformations, as required by the analysis system, must be formulated with extreme care and geometric consistency to the full analysis accuracy. It is better to truncate the nodal geometry data to ensure transcription than to make errors in physically meaningless digits. If in any doubt, always seek expert advice!”

In general, you should always note when applying tyings whether they have a physical meaning or not. In particular, tyings depend on the system degrees of freedom that will arise in the nodes during the composition of the finite element model. If you apply tyings that are physically meaningless, this will often become clear from the absence of equilibrium and/or from inexplicable answers.

2.2.1 Nodes and Degrees of Freedom

Generally speaking, a tying consists of a degree of freedom in a *master* node, in the figures of this section drawn as ●, and one or more degrees of freedom in *slave* nodes, drawn as ○. Figure 2.1 shows the notation convention for degrees of freedom for tyings. As for

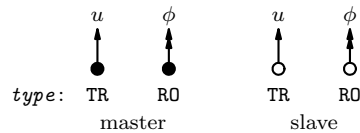


Figure 2.1: Tying degrees of freedom

rigid supports, degrees of freedom for tyings are specified by means of a *type*: TR for translation or RO for rotation, and a direction number *dirnr* referring to table 'DIRECT' [§ 1.5 p. 11].

A slave degree of freedom cannot be a master in another tying, nor can it be supported.

The solution of the system of equations yields the solution for the degrees of freedom of the master. DIANA derives the solution for the slave degrees of freedom from the master's.

2.2.2 Single- and Multi-point Tyings

Tyings may be input in two formats: *single-point* or *multi-point*.

2.2.2.1 Single-point

In single-point format you must explicitly specify all nodes for each tying separately.

2.2.2.2 Multi-point

In multi-point format the tying is specified as a connection between a master edge and a slave edge, where these edges consist of straight line sections [Fig. 2.2]. The principle is that you specify the corner nodes of these edges, the so-called *vertices*, indicated with double circles. You also specify a set of master and slave nodes in arbitrary order, the

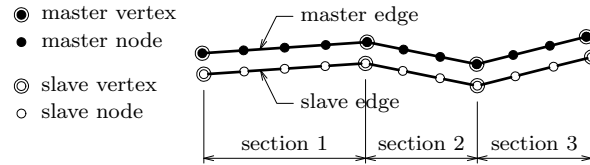


Figure 2.2: Multi-point tying

single circles.

The master and slave nodes also comprise the vertices themselves, so you must specify the vertices twice.

If the master and slave nodes are on the straight edge sections then DIANA determines their sequence [Fig. 2.2]. DIANA ignores specified nodes which are not a straight edge section.

Conditions. For multi-point tyings, you must specify as many master vertices as slave vertices. For each edge section, the amount of (accepted) master and slave nodes depends on the type of tying.

2.2.3 General Input Syntax

Tyings must be input in one single table 'TYINGS', divided in separate subtables for the various types of tyings. These subtables, listed below, may appear repeatedly and in arbitrary order.

syntax

'TYINGS'

NAME *name_s*

EQUAL

EQUMPC

1 5 6 80

Equalities ...

BETWEE

BTWMP

1 5 6 80

Interconnections ...

ECCENT

ECCMPC

1 5 6 80

Eccentric connections ...

FIX

FIXMPC

1 5 6 80

General connections ...

ELEMEN

¹
Automatic tying . . .

80

NAME *name* is the name for the linear constraint set and may be used for identification of and reference to the linear constraint set. A name specification line must follow immediately after the table heading. The name specification may be respecified at the start of a new subtable.

Equalities describe equality of degrees of freedom [§ 2.2.4]. Subtable **EQUAL** is for single-point tyings, **EQUMPC** for multi-point tyings.

Interconnections describe the interconnection of a node (and its degrees of freedom) to two other nodes [§ 2.2.5]. Subtable **BETWEE** is for single-point tyings, **BTWMP** for multi-point tyings.

Eccentricities describe an eccentric connection of one or more nodes (and their degrees of freedom) to another node [§ 2.2.6]. Subtable **ECCENT** is for single-point tyings, **ECCMPC** for multi-point tyings.

General connections describe linear dependency between one degree of freedom and one or more other degrees of freedom [§ 2.2.7]. Subtable **FIX** is for single-point tyings, **FIXMPC** for multi-point tyings.

Automatic tying let DIANA determine the correct tying of loose elements, for instance the connection of shell to solid elements [§ 2.2.8.2]. Subtable **ELEMEN** specifies the master and slave elements for automatic tying.

The input syntax for the various types of tyings is described in the referred sections which also comprise some instructive examples.

2.2.4 Equalities

The most common use of tyings is the equalization of degrees of freedom, or more specifically: the constraint that one or more displacements are equal. Equality tyings may be input in single- or multi-point format.

2.2.4.1 Single-point Equalities

Equalities in single-point format must be specified in subtable **EQUAL** with one or more slave nodes, combined with a single master node.

syntax

```
'TYINGS'
NAME names
EQUAL typew dirnrsn... {typew dirnrsn... . . . }


---


1      5 6
snoden  mnoden


---


1      5 6
/snodesng... /
      mnoden
```

type dirnrs describe one or more degrees of freedom [§ 2.2.1 p. 16].

snode is the number of the slave node.

snodes is a series of slave nodes.

mnode is the number of the master node.

General example. The following input is a general example of the single-point equality tyings as shown in Figure 2.3 on the next page.

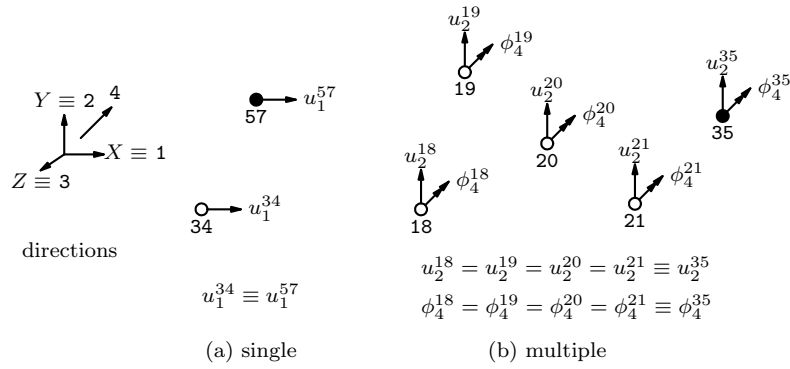


Figure 2.3: Equality tyings

```

file.dat
'DIRECTIONS'
  1  1. 0. 0.
  2  0. 1. 0.
  3  0. 0. 1.
  4  1. 1. 0.
'TYINGS'
NAME SAMETRX
: (a) two nodes with same translation in X direction
EQUAL TR 1
  34  57
NAME SAMEDISP
: (b) range of nodes having the same displacements
EQUAL TR 2 RO 4
  / 18-21 / 35

```

The first tying [Fig. 2.3a] describes that the translation u_1 in direction number 1 is the same for node 34 and 57. In the second tying [Fig. 2.3b], translations in direction 2 (u_2) and rotations around direction 4 (ϕ_4) of nodes 18 to 21 are the same as for node 35.

Applications. Typical applications for equality tyings are straight edges or faces, hinges, and slides. The following input is an example of these applications for the tyings shown in Figure 2.4.

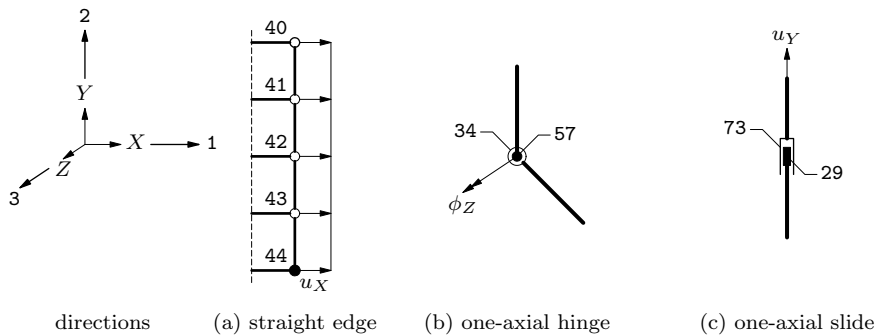


Figure 2.4: Application of equality tyings

```

file.dat
'TYINGS'
NAME EQTRX
: (a) straight edge in X direction
EQUAL TR 1
  / 40-43 / 44

```

```

: (b) hinge, rotation around Z axis free
NAME HINGE
EQUAL TR 1 2 3 RO 1 2
    34    57
: (c) slide, translation in Y direction free
NAME SLIDE
EQUAL TR 1 3 RO 1 2 3
    73    29

```

In these examples, only model *XYZ* directions are used, which by default correspond to direction numbers 1, 2, and 3 respectively. Therefore it is not necessary to specify table 'DIRECTIONS'. The *Straight edge* [Fig. 2.4a] is specified such that the translation in direction number 1 for the edge nodes is equal to the same translation of the master node. For the *One-axial hinge* [Fig. 2.4b] the translations in three directions and the rotation around two axes are kept equal for the two nodes. Consequently, the rotation around the third axis is the only degree of freedom that may differ. The *Slide* [Fig. 2.4c] is specified likewise the hinge, but now the translation in one direction is left free for both nodes. Note that for the hinge as well as for the slide, the two nodes do coincide, i.e., share the same coordinates.

2.2.4.2 Multi-point Equalities

Equalities in multi-point format [§ 2.2.2 p. 16] must be specified in subtable EQUMPC with series of slave vertices and nodes, combined with series of master vertices and nodes [Fig. 2.5]. For each edge section, the amount of master nodes n must be equal to the

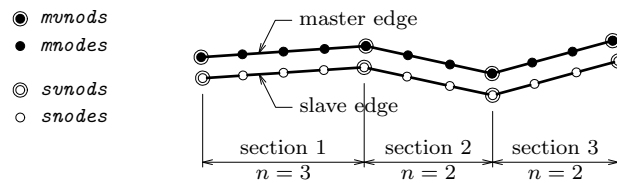


Figure 2.5: Multi-point equality

amount of slave nodes, i.e., each slave node is connected to one master node.

syntax

```

'TYINGS'
NAME names
EQUMPC typew dirnrsn... {typew dirnrsn... ...}
1 5 6 80
/ snodesn... /
      / mnodesn... /
/ snodesng... /
      / mnodesng... /

```

type dirnrs specify one or more degrees of freedom [§ 2.2.1 p. 16].

snodes is a series of nodes indicating the slave vertices.

mnodes is a series of nodes indicating the master vertices.

snodes is a series of slave nodes.

mnodes is a series of master nodes.

file.dat

```

'TYINGS'
NAME MPCEQU
EQUMPC TR 1 2
    / 16 26 /          / 1 11 /
    / 26 23 21 18 16 /  / 11 9 6 4 1 /

```

2.2.5 Interconnection

An interconnection of tying connects one node (the slave), to two others (the masters). Interconnection tyings may be input in single- or multi-point format. An interconnection tying indicates that a slave node lies in between two master nodes.

2.2.5.1 Single-point Interconnection

Interconnections in single-point format must be specified in subtable **BETWEE** with a single slave node and two master nodes [Fig. 2.6]. By default DIANA assumes that the slave node

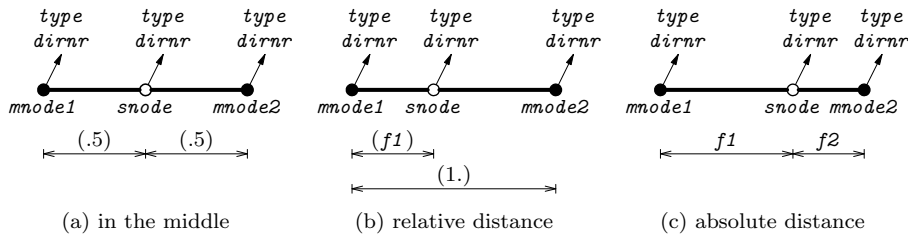


Figure 2.6: Single-point interconnection

is just in the middle of the two masters [Fig. 2.6a]. You may overrule this assumption by specifying the position of the slave node explicitly [Fig. 2.6bc].

syntax

'TYINGS'		
NAME <i>name_s</i>		
BETWEE <i>type_w dirnrs_{n...} {type_w dirnrs_{n...}...}</i>		
1	5	6
<i>snode_n mnode1_n mnode2_n {f1_r {f2_r}}</i>		
80		

type dirnrs specify one or more degrees of freedom [§ 2.2.1 p. 16].

snode is the number of the slave node.

mnode1, *mnode2* are the two master nodes. The optional factors *f1* and *f2* specify the position of the slave node relative to the master nodes. A single factor *f1* specifies the relative distance from the slave node to the first master node [Fig. 2.6b]. Two factors *f1* and *f2* specify the absolute distances from the slave node to the masters [Fig. 2.6c]. If *f1* + *f2* is not equal to the distance between the master nodes, then *snode* is positioned in between them proportionally to *f1* and *f2* to determine the linear dependency. [*f1* = 0.5]
(0 < *f1* < 1)

Mesh refinement. A typical application of interconnection tyings is *Mesh refinement* as shown in the input for the mesh of Figure 2.7.

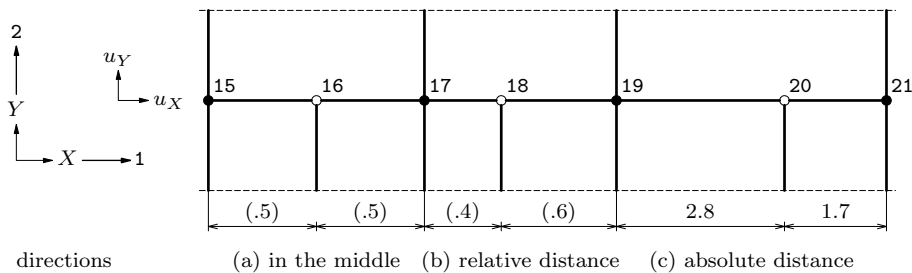


Figure 2.7: Mesh refinement

file.dat

```

'TYINGS'
NAME TYGBTW
BETWEEN TR 1 2
: (a) just in the middle
  16 15 17
: (b) relative distance
  18 17 19 .4
: (c) absolute distance
  20 19 21 2.8 1.7

```

Assuming that the finite element model consists of plane stress elements in the XY plane, only the u_X and u_Y translations have to be tied.

By default, the X and Y directions correspond to direction numbers 1 and 2 respectively. Therefore it is not necessary to specify table 'DIRECTIONS'.

2.2.5.2 Multi-point Interconnection

Interconnections in multi-point format [§ 2.2.2 p. 16] must be specified in subtable BTWMPD with series of slave vertices and nodes, combined with series of master vertices and nodes [Fig. 2.8]. If there are n master nodes between two vertices and k slave nodes between

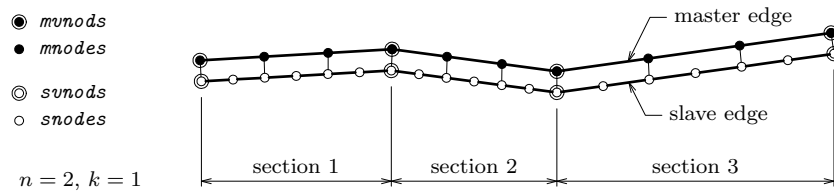


Figure 2.8: Multi-point interconnection

each pair of master nodes then there must be $n + k \times (n + 1)$ nodes on each section of a slave edge. The values of n and k must be the same for each section.

syntax

```

'TYINGS'
NAME name_s
BTWMPD type_w dirnrs_n... { type_w dirnrs_n... }

```

1	5	6	80
/ <i>snods_n...</i> /			
	/ <i>mnods_n...</i> /		
/ <i>snodes_ng...</i> /			
	/ <i>mnodes_ng...</i> /		

type dirnrs specify one or more degrees of freedom [§ 2.2.1 p. 16].

snods is a series of nodes indicating the slave vertices.

mnods is a series of nodes indicating the master vertices.

snodes is a series of slave nodes.

mnodes is a series of master nodes.

file.dat

```

'TYINGS'
NAME MPCBTW
BTWMPD TR 1 2
  / 13 9 /      / 26 28 /
  / 9 10 11 12 13 / / 26 27 28 /

```


2.2.6 Eccentricity

An eccentricity tying indicates that one or more slave nodes are eccentrically connected to a master node. Eccentric connections may be useful if the real physical model comprises eccentricity, whereas in the finite element model the nodes do coincide.

Eccentric connections must be applied with great care, they are a potential source of incorrect behaviour of the finite element model!

Tyings for eccentric connection may be input in single- or multi-point format. You may let DIANA calculate the eccentricities from the locations of the nodes or you may explicitly specify the eccentricities yourself.

2.2.6.1 Calculated vs. User-specified Eccentricity

By default DIANA calculates the eccentricity from the coordinate difference of master and slave node: $xe = X_{mnode} - X_{snode}$, $ye = Y_{mnode} - Y_{snode}$, and $ze = Z_{mnode} - Z_{snode}$. By one or more of the parameters $DX=xe$, $DY=ye$, or $DZ=ze$ (for the eccentricity in x , y , or z direction respectively) you may explicitly specify an eccentricity to overrule the default [Fig. 2.9]. If you specify one or two of these parameters, then DIANA assumes that the

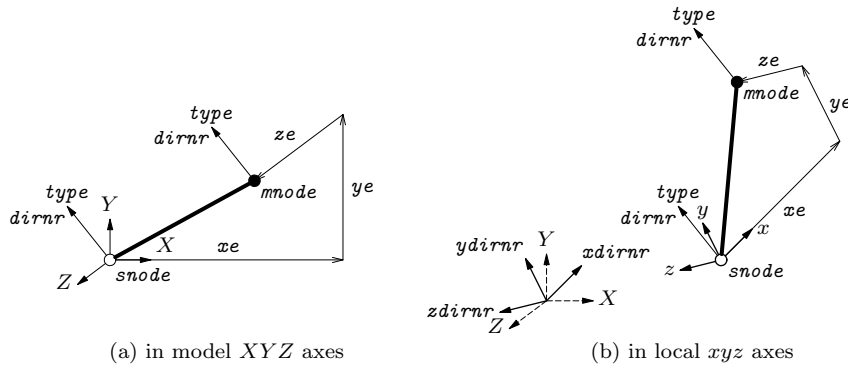


Figure 2.9: User-specified eccentricity

unspecified eccentricities are zero.

[$xe = 0$]

[$ye = 0$]

[$ze = 0$]

2.2.6.2 Axes of Eccentricity

By default user-specified eccentricities are expressed in the global model XYZ coordinate system [Fig. 2.9a]. However, you may specify eccentricities in a local xyz coordinate system with the slave node as origin [Fig. 2.9b]. Therefore you must use the **AXES** input option in the subtable heading, followed by three direction numbers $xdirnr$, $ydirnr$, and $zdirnr$ for the direction of the local x , y and z axis respectively. These direction numbers refer to table 'DIRECT' [§ 1.5 p. 11]. The specified xyz axes must form a right-handed Cartesian coordinate system.

2.2.6.3 Single-point Eccentricity

Eccentricities in single-point format must be specified in subtable **ECCENT** with one or more slave nodes, combined with a single master node.

syntax

```
'TYINGS'
NAME names
ECCENT _____ {AXES xdirnrn ydirnrn zdirnrn}
               typew dirnrsn... ...
```

1	5	6	80
$snode_n$	$mnode_n$	$\{ \frac{\quad}{\quad} \}$	
		$DX = xe_r$	
		$DY = ye_r$	
		$DZ = ze_r$	
1	5	6	80
<hr/>			
$/ snodes_{ng...} /$			
	$mnode_n$		
<hr/>			

type dirnrs specify one or more degrees of freedom [§ 2.2.1 p. 16]. With the **AXES** input option you may specify the local axes [§ 2.2.6.2 p. 23].

snode is the number of the slave node, *snodes* is a series of slave nodes.

mnode is the master node. If this master node is connected to a single slave node, then you may explicitly specify the eccentricity by one or more of the parameters $DX=$, $DY=$, or $DZ=$ [§ 2.2.6.1].

Note that an eccentricity tying can be viewed as an eccentric connection between a slave and a master node by means of a rigid link such that a rotation of the master node will rotate the rigid link along with the slave node attached to the other end of the link. Therefore, in order to account for the effect of eccentricity the appropriate rotational degree of freedom must be specified using *type* and *dirnrs* in addition to the translational degrees of freedom. In the situations where the element does not have any rotational degree of freedom (e.g. plane stress and strain, solid elements) or does not have the appropriate rotational degree of freedom (e.g. out-of-plane/drilling rotation for regular shell elements) the use of eccentricity tying may cause problems in the model evaluation or may yield erroneous results. In general, eccentricity tying is suitable for beam elements.

Two-dimensional example. The following example input is for a finite element model in two-dimensional space, for instance a plane frame with L6BEN elements. This example involves the displacement degrees of freedom u_X , u_Y , and ϕ_Z as shown in Figure 2.10.

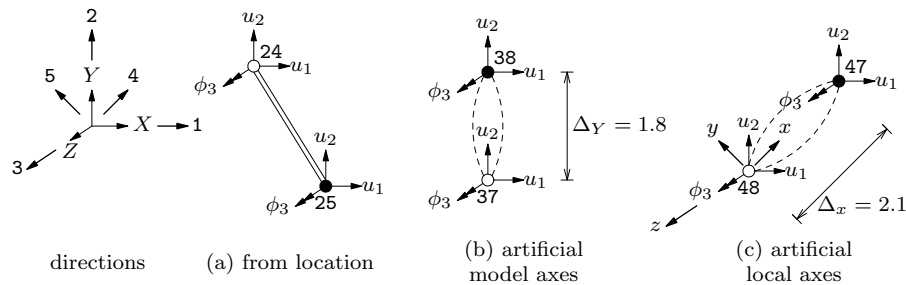


Figure 2.10: Eccentricity in two-dimensional model

file.dat

```
'DIRECTIONS'
1  1.  0.  0.
2  0.  1.  0.
3  0.  0.  1.
4  1.  1.  0.
5 -1.  1.  0.
'TYINGS'
NAME TYGEC1
ECCENT TR 1 2 R0 3
: (a) eccentricity from node coordinates
24 25
: (b) artificial eccentricity in model axes
37 38 DY=1.8
: (c) artificial eccentricity in local axes
```

```
NAME TYGEC2
ECCENT TR 1 2 RO 3 AXES 4 5 3
46 47 DX=2.1
```

In these examples, five directions are used. Figure 2.10a shows the first tying, an eccentric connection between two nodes which do not have the same location in the finite element model. DIANA uses the distance between the two nodes to calculate the dependence of the slave node displacements on the master displacement'. The tying of Figure 2.10b consists of two coinciding nodes. The eccentricity in the real physical model is applied artificially: a distance Δ_Y in the direction of the model Y axis. For the tying of Figure 2.10c the artificial eccentricity Δ_x is specified in a local xyz coordinate system.

2.2.6.4 Multi-point Eccentricity

Eccentricities in multi-point format [§ 2.2.2 p. 16] must be specified in subtable ECCMPC with series of slave vertices and nodes, combined with series of master vertices and nodes [Fig. 2.11]. For each edge section, the amount of master nodes n must be equal to the

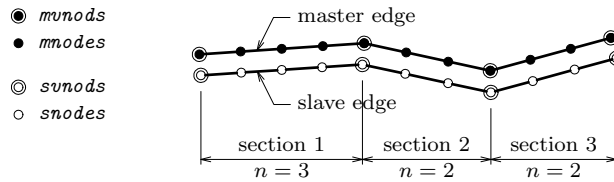


Figure 2.11: Multi-point eccentricity

amount of slave nodes, i.e., each slave node is eccentrically connected to one master node

syntax

```
'TYINGS'
NAME name_s
ECCMPC _____ {AXES xdirnr_n ydirnr_n zdirnr_n}
            type_w dirnrs_n... ...
```

```
1 _____ 5 6 _____ 80
/ svnods_n... /
/ mnodes_n... /
/ snodes_ng... /
/ mnodes_ng... / { _____ }
                  DX=x e_r
                  DY=y e_r
                  DZ=z e_r
```

type dirnrs specify one or more degrees of freedom [§ 2.2.1 p. 16]. With the **AXES** input option you may specify the local axes [§ 2.2.6.2 p. 23].

svnods is a series of nodes indicating the slave vertices.

mnods is a series of nodes indicating the master vertices.

snodes is a series of slave nodes.

mnodes is a series of master nodes. You may explicitly specify the eccentricity by one or more of the parameters **DX=**, **DY=**, or **DZ=** [§ 2.2.6.1]. If you do so, DIANA will apply this eccentricity for each pair of master and slave nodes.

Note that an eccentricity tying can be viewed as an eccentric connection between a slave and a master node by means of a rigid link such that a rotation of the master node will rotate the rigid link along with the slave node attached to the other end of the link. Therefore, in order to account for the effect of eccentricity the appropriate rotational degree of freedom must be specified using *type* and *dirnrs* in addition to the translational degrees

of freedom. In the situations where the element does not have any rotational degree of freedom (e.g. plane stress and strain, solid elements) or does not have the appropriate rotational degree of freedom (e.g. out-of-plane/drilling rotation for regular shell elements) the use of eccentricity tying may cause problems in the model evaluation or may yield erroneous results. In general, eccentricity tying is suitable for beam elements.

file.dat

```
'TYINGS'
NAME MPCECC
ECCMPC TR 1 2 3 RO 3
/ 1 3 / / 9 11 /
/ 1 2 3 / / 9 10 11 /
```

2.2.7 General Connection

If none of the previous types of tying is applicable, you may specify a tying in its most general way: a linear relation between one slave degree of freedom and one or more master degrees of freedom. These types of tyings, called *general connections*, may be input in single- or multi-point format.

2.2.7.1 Single-point General Connection

General connections in single-point format must be specified in subtable FIX with a single slave node connected to one or more master nodes.¹

syntax

```
'TYINGS'
NAME names
FIX stypew sdirnrn
1 5 6 80
snoden
mnoden mtypew mdirnrn {facr} ...
```

stype sdirnr specify one slave degree of freedom [§ 2.2.1 p. 16].

snode is the number of the slave node.

mnode is a master node which must be followed by the description of its degree of freedom [§ 2.2.1 p. 16]. Factor *fac* is an optional multiplication factor: *slave* = *fac* × *master*. You may specify multiple masters for each slave node, each one on a new line!

[*fac* = 1]

Two-dimensional example. The input data below presents two simple examples of general connections in two-dimensional space as shown in Figure 2.12 on the next page.

file.dat

```
'DIRECTIONS'
1 1. 0. 0.
2 0. 1. 0.
3 0. 0. 1.
'TYINGS'
: (a) seesaw
NAME SEESAW
FIX TR 2
```

¹The REMAKE option of Module INPUT [§3.3 p.50] always delivers single-point general connection tyings regardless of the original specification.

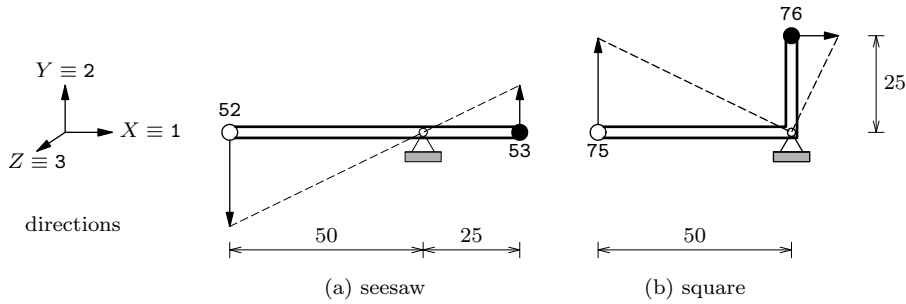


Figure 2.12: General connections in two-dimensional space

```

52 53 TR 2 -2.
: (b) square
NAME SQUARE
FIX TR 2
75 76 TR 1 2.

```

For the *Seesaw* connection [Fig. 2.12a] the Y displacement for the slave node is twice as much as for the master and in the opposite direction (note the minus sign in factor -2). The *Carpenter's square* [Fig. 2.12b] causes the slave's Y displacement to be twice as much as the master's X displacement.

Multiple Masters Example. The input data below keeps the X distance between two pairs of nodes the same. In formula: $u_1 - u_2 = u_3 - u_4$.

file.dat

```

'DIRECTIONS'
1 1. 0. 0.
2 0. 1. 0.
3 0. 0. 1.
'TYINGS'
NAME FIXTRX
FIX TR 1
1 2 TR 1 1.
3 TR 1 1.
4 TR 1 -1.

```

2.2.7.2 Multi-point General Connection

General connections in multi-point format [§ 2.2.2 p. 16] must be specified in subtable `FIXMPC` with series of slave vertices and nodes, combined with series of master vertices and nodes [Fig. 2.13]. For each edge section, the amount of master nodes n must be equal

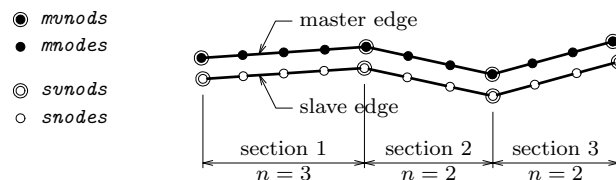


Figure 2.13: Multi-point general connection

to the amount of slave nodes, i.e., each slave node is connected to one master node.

syntax

```

'TYINGS'
NAME name_s

```

FIXMPC $stype_w$ $sdirnr_n$

1	5	6		80
<hr/>				
/ $svnods_{n...}$ /				
/ $mnods_{n...}$ /				
/ $snodes_{ng...}$ /				
/ $mnodes_{ng...}$ / $mtype_w$ $mdirnr_n$ { fac_r }				
<hr/>				

$stype$ $sdirnr$ specify one slave degree of freedom [§ 2.2.1 p. 16].

$svnods$ is a series of nodes indicating the slave vertices.

$mnods$ is a series of nodes indicating the master vertices.

$snodes$ is a series of slave nodes.

$mnodes$ is a series of master nodes. Each series of master nodes must be followed by $mtype$ and $mdirnr$, the description of a degree of freedom [§ 2.2.1 p. 16]. DIANA will apply this degree of freedom for all master nodes in the series. Factor fac is an optional multiplication factor: $slave = fac \times master$. This factor will also be applied for all master nodes in the series.

[$fac = 1$]

file.dat

'TYINGS'				
NAME FIXTRY				
FIXMPC TR 2				
/ 16 26 / / 1 11 /				
/ 16 18 21 23 26 / / 1 4 6 9 11 / TR 2 1.0				

2.2.8 Automatic Tying

In DIANA's library of finite elements for structural analysis we can distinguish two types of elements [Vol. *Element Library*]: elements with *translational* degrees of freedom only, such as truss, regular plane stress, plane strain and solid elements, and elements with both *translational and rotational* degrees of freedom such as beams and shells.

If in a finite element model elements of the first type are connected to elements of the second type, then additional tyings must be applied for reasons of compatibility: the rotational degree of freedom in the connection node must be tied to one or more translational degrees of freedom. Usually, these tyings are of the 'eccentric connection' type and can be input in table 'TYINGS' [§ 2.2.6]. However, DIANA can generate this type of tyings automatically for the connection of beams to plane stress and solid elements and for the connection of shell elements to solid elements.

Restrictions. There are some important restrictions to connection by automatic tyings:

- A beam element can only be connected fully to a plane stress element if both elements are in the model XY plane.
- If multiple shell elements are connected to the same node and this node must be tied automatically to a solid element, then the thickness vectors of the shell elements at that node must coincide. This may be achieved by specification of a suitable **SHAPE** for these shell elements [Vol. *Element Library*].
- DIANA neglects the geometrical dimensions (thickness, cross-section) of beam and shell elements when searching for slave nodes of tyings.
- For beam to plane stress connections, the master node of the beam element must be on the same edge as the slave nodes of the plane stress element. For beam to solid connections, the master node of the beam element must be in the same plane as the slave nodes of the solid element. For shell to solid connections, the master nodes of the shell element must be in the same plane as the slave nodes of the solid element.

- For beam to plane stress connections, the beam element needs to be perpendicular to the edge of the slave nodes of the plane stress element in the master node. For beam to solid connections, the beam element needs to be perpendicular to the plane of the slave nodes of the solid elements in the master node. For shell to solid connections, the shell element needs to be perpendicular to the plane of the slave nodes of the solid elements in the master nodes.

Modes of operation. The Automatic Tying option can operate in two different modes. In the first mode the elements to be tied have common node numbers in their connectivity, i.e., they would also be *connected* without tyings but not correctly, for instance hinged instead of clamped [§ 2.2.8.1]. The second mode may be used if the elements to be tied do not have common node numbers, without tyings they would be *loose*, i.e., not connected at all [§ 2.2.8.2]. For both cases, DIANA will generate tyings automatically if you turn the *automatic tying* option “on” [§ 3.5 p. 54].

2.2.8.1 Connected Elements

Elements connected to a beam or shell element are candidates to deliver slave nodes for the tying. By default DIANA examines all elements of the model. However, you may select parts of the model for which automatic tying must be activated [§ 3.5 p. 54]. For automatic tying of connected elements, it is not necessary to specify input table ‘TYINGS’.

Automatic tying of connected elements is possible for the connections as shown in Figure 2.14 and outlined below.

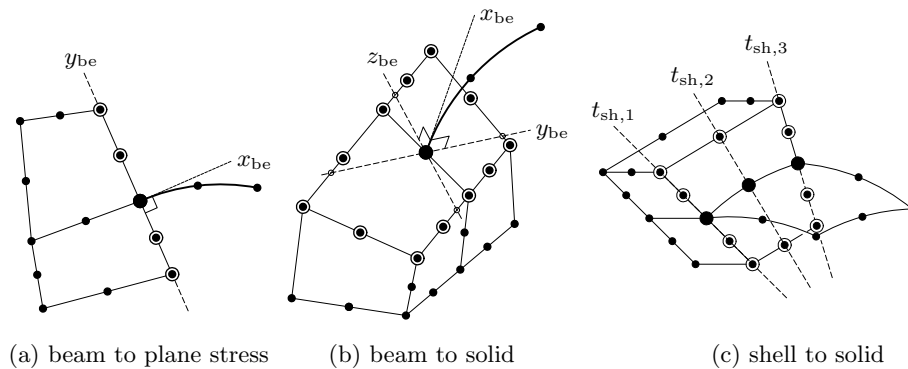


Figure 2.14: Automatic tying of connected elements

Beam to plane stress. [Fig. 2.14a] The nodes of the connected plane stress elements which are on the beam’s y axis will be tied automatically to the beam’s degrees of freedom.

Beam to solid. [Fig. 2.14b] The nodes of the connected solid elements which are in the plane of the beam’s cross-section, i.e., the plane through the yz axes of the beam, will be tied automatically to the beam’s degrees of freedom.

Shell to solid. [Fig. 2.14c] The nodes of the connected solid elements which are on the thickness vectors t of the shell will be tied automatically to the shell’s degrees of freedom.

2.2.8.2 Loose Elements

For loose elements DIANA cannot determine the candidate slave elements for automatic tying, therefore you must specify for each tying the master element and its slave elements in subtable **ELEMEN** of table ‘TYING’.

*Note that when both loose and connected elements are subjected to automatic tyings, the subtable **ELEMEN** of table ‘TYINGS’ must include also the elements relevant for connected elements (in addition to that of loose elements).*

syntax

```

'TYINGS'
NAME names
ELEMEN
1      5 6
/ selemsng... / melemn ...
/ selemsng... /
      / melemsng... / ...

```

selems is a set of slave elements (plane stress or solid), specified by numbers and/or groups, which are candidates to deliver slave nodes for the tying.

melem is a master element (beam or shell) which must be tied automatically to the slave elements.

melems is a set of master elements (beam or shell) which must be tied automatically to the slave elements. Elements of different types are skipped.

file.dat

```

'TYINGS'
NAME LOOSE
ELEMEN
/ 1-1062 / 2113
/ 1063-2097 / 2114

```

Automatic tying of loose elements is possible for the connections as shown in Figure 2.15 and outlined below.

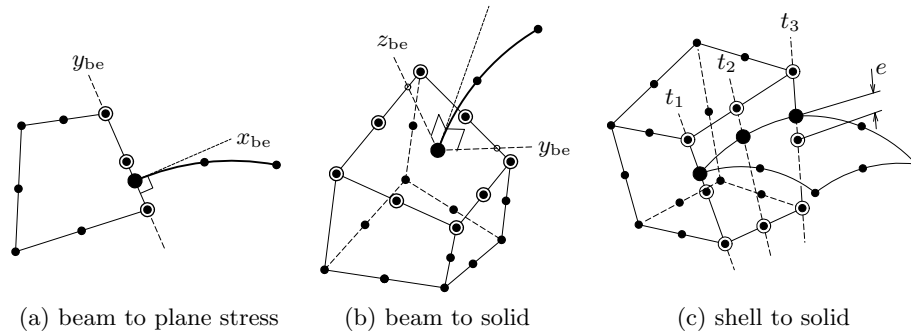


Figure 2.15: Automatic tying of loose elements

Beam to plane stress. [Fig. 2.15a] All nodes of the specified slave plane stress elements which are on the beam's y axis will be tied automatically to the beam's degrees of freedom.

Beam to solid. [Fig. 2.15b] All nodes of the specified slave solid elements which are in the plane of the beam's cross-section, i.e., the plane through the yz axes of the beam, will be tied automatically to the beam's degrees of freedom.

Shell to solid. [Fig. 2.15c] All nodes of the specified slave solid elements which are on the thickness vectors t of the shell will be tied automatically to the shell's degrees of freedom. The nodes of shell and solid element may coincide, i.e., the distance e may be zero.

2.3 Loads

There are various classes of loads input, each of which is specified in a separate subtable of table 'LOADS'. Regular loading input is divided into cases, which may be combined into load sets. Apart from these classes of regular loading, there is a special form of loading input for the determination of *influence fields* [§ 2.3.9 p. 46].

syntax

'LOADS'

CASE *case_n*

[NAME *name_s*]

NODAL

1	5	6	80
---	---	---	----

nodal load

ELEMEN

1	5	6	12	13	80
---	---	---	----	----	----

element load

WEIGHT

1	80
---	----

dead weight

EQUIAC

1	80
---	----

equivalent acceleration load

CENTRI

1	80
---	----

centrifugal load

DEFORM

1	5	6	80
---	---	---	----

fixed displacements

MOBILE

1	5	6	12	13	80
---	---	---	----	----	----

mobile loads

REINFO

1	5	6	12	13	80
---	---	---	----	----	----

reinforcement load

BASE

1	5	6	12	13	80
---	---	---	----	----	----

base excitation load

PUSHOV

1	5	6	12	13	80
---	---	---	----	----	----

modal pushover load

ACCELE

1	5	6	80
---	---	---	----

prescribed accelerations

WIND

1	5	6	12	13	80
---	---	---	----	----	----

wind load

WATER

1	5	6	12	13	80
---	---	---	----	----	----

water wave load

COMBIN

2.3.1 Nodal Load

Nodal loads are input in subtable NODAL. There are three forms of input syntax: (1) one nodal load per line, (2) nodes in a series of numbers or groups or both, with one load value valid for all the nodes in the series, (3) nodes in a series of numbers and/or groups, with a series of load values: one value for each node. A spatial function can be attached to this property [§ 1.8].

syntax

```
'LOADS'
NODAL
1 5 6 80
noden typew dirnrn valuer
1 5 6 80
/ nodesng... /
typew dirnrn valuer
1 5 6 80
/ nodesng... /
typew dirnrn / valuesr... /
```

node is a single node number. **nodes** is a series of nodes, it must be specified between slashes and may comprise numbers or groups or both.

type is the load type: FORCE for force or MOMENT for moment. The direction number **dirnr** refers to table 'DIRECT' [§ 1.5 p. 11]: a force in, or a moment around the specified direction.

value is the load value. For axisymmetric models, this value is the total circumferential load. **values** is a series of load values, one for each node in **nodes**, it must be specified between slashes.

file.dat

```
'LOADS'
CASE 2
NODAL
3 FORCE 2 1.E3
5 MOMENT 1 200.
/ 4-12 WALL 31 /
FORCE 5 250.
/ 20 22 25 27 35 54 68 /
FORCE 1 / 100.0:160.0(10.) /
```

2.3.2 Dead Weight

Load due to dead weight is specified in subtable WEIGHT. The direction and gravity acceleration for dead weight are defined in table 'MODEL' [§ 1.2 p. 6]. To evaluate dead weight loading, DIANA retrieves the mass density for specific materials from the material properties in table 'MATERI' [Vol. Material Library].

In static analysis, the distributed mass, which can be modeled with line and surface distributed translational mass elements [Vol. Element Library], acts not as mass for dead weight.

syntax

```
'LOADS'
WEIGHT
```

WEIGHT indicates a dead weight load.

A weight load may be defined in several load cases, but the direction and acceleration are always equal and defined in table 'MODEL'. When different acceleration loads are applied in different directions and/or with different acceleration values, equivalent acceleration loads can be specified [§ 2.3.3].

file.dat

```
'LOADS'
CASE 5
WEIGHT
```

In this example, a weight load is defined in case 5.

2.3.3 Model Equivalent Acceleration Load

Loading due to an equivalent acceleration on the entire model is specified in subtable EQUIAC with direction and acceleration. To evaluate equivalent acceleration loading, DIANA retrieves the mass density for specific materials from the material properties in table 'MATERI' [Vol. *Material Library*]. Equivalent accelerations can also be specified for specific elements only by using element equivalent accelerations loads [Vol. *Element Library*].

In static analyses, the distributed mass, which can be modeled with line and surface distributed translational mass elements [Vol. Element Library], acts not as mass for equivalent acceleration loads.

syntax

```
'LOADS'
EQUIAC
```

1	80
---	----

```
dirn eqaccer
```

dir is the direction number referring to table 'DIRECT' [§ 1.5 p. 11].

eqacce is the equivalent acceleration.

Only one equivalent acceleration can be defined per load case. It is possible to combine multiple equivalent acceleration load cases in a load set [§ 2.3.8].

Note that the transformation of pressure heads into pore pressures in mixture or coupled groundwater flow-stress analysis is always based on the gravity acceleration specified in table 'MODEL' [§ 1.2 p. 6] and not on equivalent accelerations.

file.dat

```
'LOADS'
CASE 6
EQUIAC
  1  20.0
CASE 7
EQUIAC
  2  10.0
```

In this example, case 6 contains an equivalent acceleration of 20.0 in direction 1 and case 7 contains an equivalent acceleration of 10.0 in direction 2.

2.3.4 Centrifugal Load

Centrifugal load, due to rotation of the finite element model, is specified in subtable **CENTRI** with turning speed and axis of rotation. To evaluate centrifugal loading, DIANA retrieves the mass density for specific materials from the material properties in table **'MATERI'** [Vol. *Material Library*].

syntax

'LOADS'

CENTRI

 $\omega_{\tau} \quad \text{axipt1}_{\tau 3} \quad \text{axipt2}_{\tau 3}$

omega is the rotational velocity ω in rad/sec.

axipt1, *axipt2* are the model XYZ coordinates of two points on the rotation axis.

Only one single centrifugal load may be specified: one case, one speed, one rotation axis.

file.dat

'LOADS'

CASE 3

CENTRI

0.1 0.1 1.3 2.6 1.0 2.0 3.0

2.3.5 Fixed Displacements or Deformation

Fixed or prescribed displacements, so-called deformation load, are input in subtable **DEFORM**. There are three forms of input syntax: (1) one nodal displacement per line, (2) nodes in a series of numbers or groups or both, with one displacement value valid for all the nodes in the series, (3) nodes in a series of numbers and/or groups, with a series of displacement values: one value for each node. A spatial function can be attached to this property [§ 1.8].

Fixed displacements in subtable *DEFORM* must be specified as supports in table 'SUPPORT' [§ 2.1 p. 15]. Consequently they will have a zero displacement for all cases, unless specified otherwise in subtable *DEFORM*.

syntax

'LOADS'

DEFORM

A horizontal number line is shown, starting at 1 and ending at 80. There is a tick mark at 56.

$node_n$	$type_w$	$dirnr_n$	$value_r$
----------	----------	-----------	-----------

1 5 6 80

/ nodes_{nq...} /

$$type_w \quad dirnr_n \quad value_r$$

/ nodes_{nq...} /

$$type_w \text{ dirnr}_n / values_{r...} /$$

node is a single node number, **nodes** is a series of nodes, it must be specified between slashes and may comprise numbers or groups or both.

type is the displacement type: TR for translation or RO for rotation. The direction number *dirnr* refers to table 'DIRECT' [§ 1.5 p. 11]: a translation in, or a rotation around the specified direction.

value is the displacement value, *values* is a series of displacement values, one for each node in *nodes*.

You can get output of fixed displacements via the `PRESCR` type specifier of the `DISPLA` output item [§ 4.2.2 p. 77].

file.dat

```

'LOADS'
CASE 3
DEFORM
  3    TR  2    1.E-2
  5    RO  3    0.25
/ 4-12 EDGE /   TR 5 .001
CASE 5
DEFORM
  3    TR  2    2.E-2

```

In this example the fixed translation of node 3 in direction 2 differs in case 3 and case 5!

2.3.6 Mobile Loads

In bridge design, mobile (or moving) loads are often applied to get the extreme values of analysis results like strains, stresses, or support reactions. In the next sections we will present the main principles for mobile load analysis with DIANA [§ 2.3.6.1], followed by a description of the input syntax for mobile loads.²

2.3.6.1 Principles

Restrictions. The application of mobile loads is subject to the following restrictions.

- *Mobile loads can only act on beam elements.*
It is allowed to connect other elements to the beam elements on which a mobile load acts, for instance in a plate model of a bridge. When the stiffness of the beam elements is negligible in comparison to the plate elements, the only function of the beam elements is to define the mobile load pathway.
- *Mobile loads can only be applied in regular linear static analysis [Ch. 4].*
- *Only one mobile load per load case may be defined.*
- *If a load case contains a mobile load, no other loads are allowed in this load case.*

Load cases. Usually, a specified mobile load will consist of a set of moving forces, which can represent a truck for example, and a static distributed load. Additionally, you must define a set of unbroken beam elements which form the ‘pathway’ of the load [Fig. 2.16]. DIANA will automatically generate one load case for each position of the truck placed on

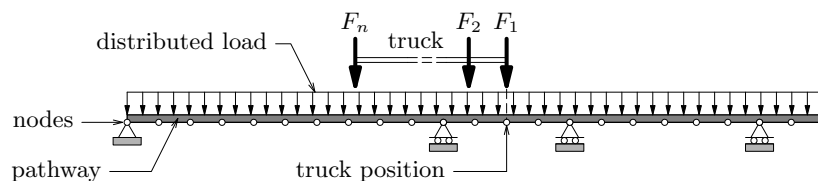


Figure 2.16: Mobile load definition

a subsequent node of the pathway. Therefore, the number of load cases will be equal to the number of nodes along the pathway. In addition to the mobile load, every load case will also have the same static distributed load along the entire pathway. So the generated load cases only differ in the position of the truck.

In the Results environment, `iDIANA` can calculate envelopes of results for all these load cases [Vol. *iDIANA*]. Other load cases, like weight or prestress, can be taken into account as well, when calculating envelopes of results.

²See also Example `bridge` in Volume *Analysis Examples*.

Truck positions. Typically, a truck load comprises multiple force loads where each force represents the loading caused by one of the truck's axles, see the forces F_1 to F_n in Figure 2.16 on the facing page. The position of the truck is defined by the position of its first axle. DIANA will spread the forces related to the other axles over the two adjacent nodes.

When the truck is at the beginning of the pathway, i.e., with its first axle on the first node [Fig. 2.17a], only the first axle will act on the model. The second axle load comes into action when the truck has moved over the distance between the two axles [Fig. 2.17b], and so on.

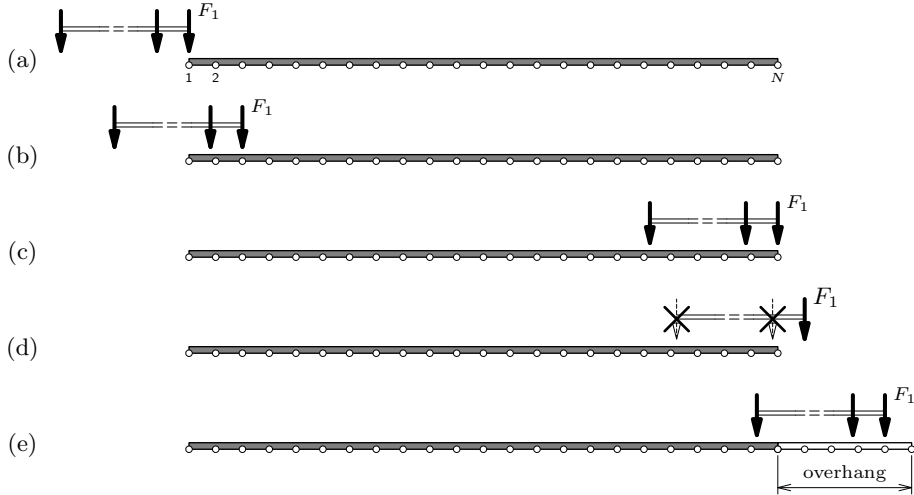


Figure 2.17: Ultimate truck positions

When the truck is positioned at the end of the pathway, i.e., with its first axle on the last node N , the other axles are still on the pathway and consequently will generate a load on the model [Fig. 2.17c]. However, when the truck leaves the pathway, i.e., when the first axle arrives beyond the last node, DIANA will not generate any truck load on the model [Fig. 2.17d]. Therefore, to account for the loading of the other axles, which are still on the model, you must define an 'overhang', i.e., a part of the pathway beyond the end of the actual model [Fig. 2.17e]. The length of this overhang must be equal to, or greater than, the length of the truck. The number of nodes in the overhang defines the number of truck positions that DIANA will take into account when the truck leaves the model.

2.3.6.2 Mobile Load Input

Input of a mobile load consists of the actual load specification in subtable **MOBILE** of input table 'LOADS' and some geometrical data for the elements in table 'GEOMET'. To apply more than one mobile load at the same time, you must specify all the **MOBILE** subtables in separate load case.

syntax

'LOADS'									
1	5	6	12	13	80				
CASE	<i>case1_n</i>								
MOBILE									
	...	<i>input parameters first mobile load</i>							
CASE	<i>case2_n</i>								
MOBILE									
	...	<i>input parameters second mobile load</i>							

You may superpose the results of mobile loads in the iDIANA Results environment [Vol. iDIANA]. You may define the mobile loads explicitly, or let them set up by DIANA according to code regulations for bridges.

syntax

'LOADS'									
1	5	6	12	13	80				
CASE	<i>case_n</i>								
MOBILE									
	ELEMEN	<i>elems_{ng...}</i>							
	[DIRECT	<i>dirnr_n</i>]							
	CODE								
		<u>NONE</u>							
		VOSB							
		VBB							
		ENV							
	<i>loading specs</i>						

CASE *case* is the number of the load case. Mobile loads may *not* be used in combination with load sets via subtable COMBIN.

ELEMEN *elems* are the elements, specified by numbers and/or groups, which form the part of the model where the traffic load must be applied. These elements (at least two) must form a contiguous open chain of connected beam elements without branches. In this sense, two elements are considered to be 'connected' if they share an end-node. DIANA automatically determines the sequence of the elements in the chain, i.e., you may specify the elements or groups in arbitrary order.

DIRECT *dirnr* specifies the direction of the mobile load, usually the gravity acceleration, with a direction number referring to table 'DIRECT' [§ 1.5 p.11]. If you do not specify a direction, then DIANA assumes $-Y$ for two-dimensional models or $-Z$ for three-dimensional models.

CODE indicates the code regulation to be applied:

NONE for 'no code', i.e., explicit specification [§ 2.3.6.3].

VOSB for the Dutch VOSB code [§ 2.3.6.4].

VBB for the Dutch VBB code [§ 2.3.6.4].

ENV for the European ENV 1991-3 code [§ 2.3.6.5].

Loading specs is a code dependent specification of the mobile load.

2.3.6.3 Explicit Input

For explicit input of mobile loads you must specify at least the truck load and the distributed load in table 'LOADS' and the effective width of the carriage way in table 'GEOMET'. To simulate the influence of the transverse position of these loads you may specify load factors in table 'GEOMET'.

Load										syntax
'LOADS'										
1	5	6	12	13	80					
CODE			NONE							
LORRY			$f1_r \ [d2_r \ f2_r \ \dots \ dn_r \ fn_r]$							
DISTR			q_r							

CODE NONE indicates that no code regulation must be applied, i.e., the mobile load will be specified explicitly.

LORRY indicates the schematic truck load [Fig. 2.18]. Values *f1* to *fn* are the axle forces $F_{i=1,n}$. Values *d2* to *dn* are the distances $d_{i=2,n}$ to the preceding force.

DISTRI *q* is the value *q* of the distributed load [*force/unitarea*].

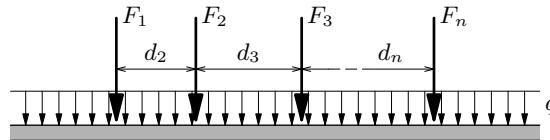


Figure 2.18: Mobile truck and distributed load

Width of carriage way*syntax*

'GEOMET'

1	5	6	12	13	80
		BWIDTH		w_r	

BWIDTH w is the effective width w of the carriage way. DIANA needs this width to determine the area at which the distributed load q is acting.

Transverse position of load*syntax*

'GEOMET'

1	5	6	12	13	80
		[LODFAC		ff_r	
		[DISFAC		fq_r	

By default DIANA neglects the influence of the transverse position of an explicitly specified mobile truck load. To let DIANA apply this transverse position you must specify multiplication factors in table 'GEOMET'.

LODFAC ff is a multiplication factor f_F for the truck forces F . This factor simulates the influence of the transverse position of the truck. [$f_F=1$]

DISFAC fq is a multiplication factor f_q for the distributed load q . This factor simulates the influence of the transverse position of the distributed load. By default this factor is equal to the effective width. [$f_q = w$]

Example of explicit input*file.dat*

```
'ELEMEN'
...
GEOMET
/ 5 8 13 24 37 40 / 2
'DIRECT'
4      0. -1. 0.
...
'GEOMET'
2  BWIDTH 4.75
   LODFAC 0.83
   DISFAC 3.72
...
'LOADS'
CASE 2
MOBILE
  ELEMEN 5 8 13 24 37 40
  DIRECT 4
  CODE   NONE
  LORRY  100. 1.50 100. 3.50 130.
  DISTRI 25.
CASE 3
WEIGHT
...
```

2.3.6.4 Dutch VOSB/VBB Code

There are two Dutch codes which define mobile loads on bridges: the VOSB [64] for steel bridges and the VBB [63] for concrete bridges. According to these codes, the magnitude of the loads depend on the class as shown in Table 2.1.

Table 2.1: MOBILE LOADS ACCORDING TO THE VOSB/VBB CODE

Class		Distributed load	Truck forces
VBB	VOSB		
60	600	4 kN/m ² (max. 12 kN/m per lane)	200 kN
45	450	3 kN/m ² (max. 9 kN/m per lane)	150 kN
30	300	2 kN/m ² (max. 6 kN/m per lane)	100 kN

To apply mobile loads according to the Dutch codes you must specify at least the loading class in table 'LOADS', and the width of the carriage way in table 'GEOMET'. Additionally in table 'GEOMET', you may specify the transverse position of the truck, and the bump and load reduction factors.

To apply the VOSB/VBB code, DIANA assumes by default that you specified the model data in standard SI units. If you used other units, then you must specify them in table 'UNITS' [§ 1.1 p. 3].

Code and class syntax

'LOADS'									
1	5	6	12	13					80
CODE									
		VOSB							
		VBB							
[CLASS		class _r]							

CODE indicates the code regulations to be applied: VOSB for the Dutch VOSB code, or VBB for the Dutch VBB code.

(class > 0) CLASS class indicates the VOSB/VBB class which determines the size of the mobile load [Table 2.1 p. 40]. For class values other than those mentioned in the table, DIANA applies a linear interpolation of the values with respect to the actual class number.

Width of carriage way and lanes syntax

'GEOMET'									
1	5	6	12	13					80
BWIDTH		w _r							
[WLANE		l _r]							

BWIDTH w is the width w of the carriage way [Fig. 2.19].

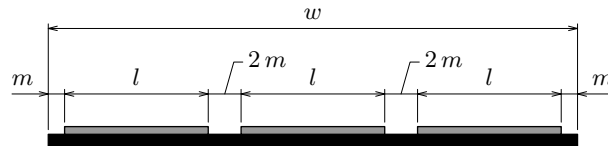


Figure 2.19: Lanes on a carriage way applied for VOSB code

($l \leq w$) WLANE l is the width l of the lanes, applied for the VOSB code.
 [$l = 3$ m] According to both Dutch codes, DIANA applies at most two lorries in transverse direction in a position which yields a maximum load contribution.

Transverse position of truck. For the Dutch codes, DIANA by default neglects the influence of the transverse position of the truck load. To let DIANA apply this transverse position you must specify either an influence line or multiplication factors in table 'GEOMET'.

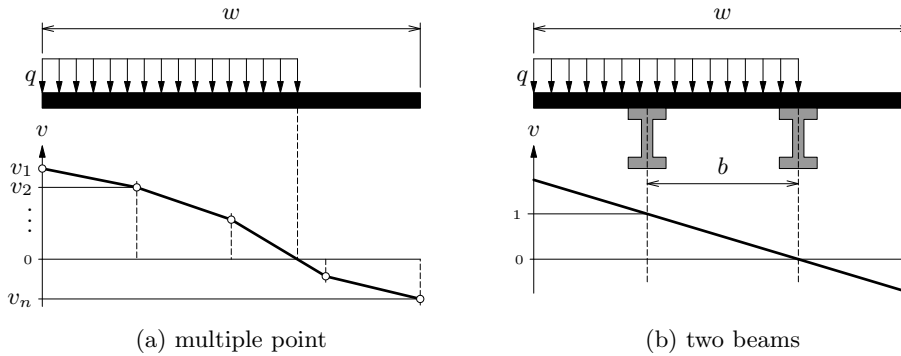


Figure 2.20: Influence line in transverse direction

Influence line

syntax

'GEOMET'

1	5	6	12	13	80
		[]	
		TRVINF	$v1_r$	$[v2_r \dots vn_r]$	
		TWOBEA	b_r		

TRVINF specifies the influence line for the truck position in transverse direction [Fig. 2.20-a]. Values $v1$ to vn are the values of the influence line $v_{i=1,n}$. You may specify at most ten points. DIANA assumes an equidistant distribution of the points. If you only specify a single point, then the influence line is a constant. ($n \leq 10$)

TWOBEA is a short-cut input for the influence line if the finite element model simulates a bridge with two beams in transverse direction [Fig. 2.20b]. The value b is the distance b between the beams. In this case, DIANA applies a straight influence line.

DIANA applies the distributed load at the part where the influence line is positive.

file.dat

```
'GEOMET'
1  BWIDTH  4.
   TRVINF  1.5 -0.5
```

In this example the influence line is a straight line. For a two-beam model the following input defines the same influence line as the previous example.

file.dat

```
'GEOMET'
1  BWIDTH  4.
   TWOBEA  2.
```

Multiplication factors

syntax

'GEOMET'

1	5	6	12	13	80
		[LODFAC	ff_r	
			[DISFAC	fqr

LODFAC ff is a multiplication factor f_F for the truck forces. This factor simulates the influence of the transverse position of the truck. If you do not specify any influence of the transverse position of truck forces DIANA applies no multiplication factor. [$f_F=1$]

DISFAC f_q is a multiplication factor f_q for the distributed load q . This factor simulates the influence of the transverse position of the distributed load. By default this factor is equal to the effective width. For the VOSB code DIANA will correct it for the sum of the width of the lanes.

[$f_q = w$]

Bumping and load reduction. Additional to the reduction factor for two lorries and the influence in transverse direction, the mobile load has to be multiplied by a bump factor and a load reduction factor. These factors depend on a certain length l . For the VOSB code [64, § A.2.10-11], DIANA uses a bump factor S and a load reduction factor B defined as

$$S = 1 + \frac{40}{100 + l} \quad \text{and} \quad B = 0.6 + \frac{40}{100 + l} \quad (2.1)$$

For the VBB [63, § 4.3.1.2-3] code the bump factor S is much more complicated because it depends strongly on geometrical properties, therefore in DIANA it must be specified explicitly, or you may indicate that bumping and load reduction must be applied according to the VOSB code. By default, DIANA will apply neither a bump factor nor load reduction.

VBB code syntax

'GEOMET'									
1	5	6	12	13					
	[BUMP		s_r						
	[LOARED		b_r						

[$S = 1$] BUMP s is the bump factor S which DIANA will apply to the mobile load.

[$B = 1$] LOARED b is the load reduction factor B which DIANA will apply to the mobile load.

VOSB code syntax

'GEOMET'									
1	5	6	12	13					
	[LOALEN		l_r						

LOALEN l is the length l which DIANA will use to compute the bump factor and the load reduction factor according to the VOSB code *even if the mobile load was input via the VBB code*.

2.3.6.5 European ENV 1991-3 Code

The European ENV 1991-3 [17] code defines traffic loads on bridges. According to this code, the magnitude of the loads depends on the width of the carriage way and the number of lanes [Fig. 2.21]. The total width w of the carriage way depends on the number of lanes

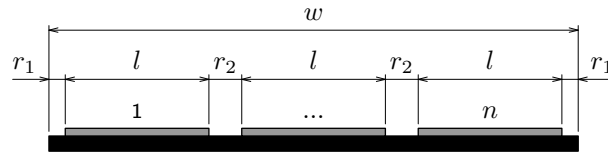


Figure 2.21: Lanes on a carriage way applied for ENV 1991-3 code

n , their width l , and the remaining width r :

$$w = n \times l + r \quad \text{with} \quad r = 2 \times r_1 + (n - 1) \times r_2 \quad (2.2)$$

Table 2.2: LANES AND CARRIAGE WAY ACCORDING TO THE ENV 1991-3 CODE

Carriage way width w [m]	Lanes		Remaining width r [m]
	number n	width l [m]	
$w < 5.4$	1	3	$w - 3$
$5.4 \leq w < 6$	2	$w/2$	0
$6 \leq w$	$\text{int}(w/3)$	3	$w - n \times 3$

To apply mobile loads according to the ENV code you must at least specify the width of the carriage way. DIANA will determine the number of lanes, their width, and the remaining width according to the ENV code [Table 2.2].

The ENV 1991-3 code indicates various classes of loading by a number [Table 2.3]. Each class gives the distributed traffic load q_{ik} , the distributed load on the remaining

Table 2.3: LOAD CLASSES ACCORDING TO THE ENV 1991-3 CODE

Class	Distributed load [kN/m ²]		Axle force [kN]
	q_{ik}	q_{rk}	Q_{ik}
1	9.0	2.5	300
2	2.5	2.5	200
3	2.5	2.5	100
> 3	2.5	2.5	0

width q_{rk} and the axle force Q_{ik} of a truck [Table 2.3]. A mobile load is applied on each lane, consisting of a truck with two axle loads Q_{ik} at a fixed distance of 1.2 m, plus the distributed force load q_{ik} over the full area of the lane [Fig. 2.22]. Additionally the distributed load q_{rk} is applied as resting on the remaining area.

By default, as prescribed by the ENV 1991-3 code, DIANA will take the values for the loads Q_{ik} and q_{ik} from the load class of the lane number: class 1 for the first lane, class 2 for the second and so on.

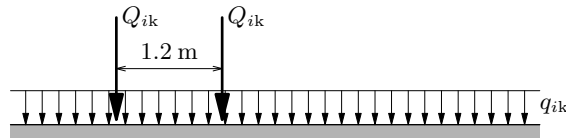


Figure 2.22: Mobile truck and distributed load according to the ENV 1991-3 code

Input data. In batch input data format you must specify the loading according to the ENV 1991-3 code in tables 'LOADS' and 'GEOMET' as indicated in the following.

To determine loading according to the ENV 1991-3 code, DIANA assumes by default that you specified the model data in standard SI units. If you used other units, then you must specify these in table 'UNITS' [§ 1.1 p. 3].

syntax

'LOADS'

1	5	6	12	13	80
CODE		ENV			
[CLASSE		class _n]			

CODE ENV indicates a load according to the ENV 1991-3 code.

CLASSE will load all lanes according to the indicated *class* [Table 2.3]. *This overrules the ENV code!* By default DIANA will apply the loads according to the ENV code: class 1 on the first lane, class 2 on the second and so on. (*class* ≤ 4)

<i>syntax</i>									
'GEOMET'									
1	5	6	12	13	80				
		BWIDTH		w_r					
		[LODFAC		ff_r					
		[DISFAC		$f q_r$					

BWIDTH w is the total width w of the carriage way.

[$ff = 1.0$] LODFAC ff is a multiplication factor for the truck axle loads. If you specify this factor then the axle loads become $ff \times Q_k$ instead of the calculated Q_k [Eq. (2.4)]. *This overrules the ENV code!*

[$fq = 1.0$] DISFAC fq is a multiplication factor for the total distributed load. If you specify this factor then the distributed load becomes $fq \times q_{tot}$ instead of the calculated q_{tot} [Eq. (2.3)]. *This overrules the ENV code!*

<i>file.dat</i>									
'ELEMEN'									
...									
GEOMET									
/ 1-5 / 1									
...									
'LOADS'									
CASE 1									
MOBILE									
ELEMEN 1-5									
CODE ENV									
'GEOMET'									
1 BWIDTH 8.0									
...									

For this example DIANA determines the total distributed load and the total axle load according to the ENV 1991-3 code as follows:

- According to Table 2.2 the number of lanes n is $\text{int}(8/3) = 2$; the width of the lanes l is equal to 3 m. According to Table 2.3 the remaining width r is equal to $8 - 2 \times 3 = 2$ m. As no class for the lanes has been specified in this example, class 1 will apply for the first lane and class 2 for the second lane. This gives the following total distributed load per unit length:

$$\begin{aligned}
 q_{tot} &= l \times (9 + 2.5) + r \times 2.5 \\
 &= 3 \times 11.5 + 2 \times 2.5 \\
 &= 39.5 \text{ kN/m}^1
 \end{aligned} \tag{2.3}$$

- According to Table 2.3 the total axle load of the truck is calculated as:

$$\begin{aligned}
 Q_k &= 300 + 200 \\
 &= 500 \text{ kN}
 \end{aligned} \tag{2.4}$$

2.3.7 Wind and Water Load

Wind and water load cause an average distributed load which can be used in static analysis.

- In a static analysis, wind and water load may be specified both, even in the same load case.
- DIANA evaluates wind or water load for a subset of the element types as indicated in a special section in Volume *Element Library*, which describes the input of necessary geometry parameters.

2.3.8 Load Sets

In subtable **COMBIN** of table 'LOADS' previously specified load cases are combined to load sets, which form the load vectors of the system of equations (right-hand-side). Solution of the system yields the displacements for these load sets.

- Specification of load sets is optional. If you do not specify load sets, DIANA assumes a one-to-one relation between load cases and load sets.
- Null-vectors will be generated for omitted load set numbers.

syntax

```
'LOADS'
COMBIN
1      5 6      80
losetn casen factorr [casen factorr ...]
1      5 6      80
losetn / casesn... / factorr [/ casesn... / factorr ...]
```

loset is the load set number.

case is a case number, referring to previously specified cases [§ 2.3 p. 31], *cases* is a series of case numbers in between slashes.

factor is a multiplication factor for *case* or *cases*.

file.dat

```
'LOADS'
CASE 1
...
CASE 2
...
CASE 3
...
CASE 4
...
COMBIN
1  1  1.0  2  1.56  3  0.7
2           2 -1.0  3  0.08  4 -0.25
4  1 -0.5           3 -2.66
5  / 1 2 / -1.0  / 3 4 / -1.5
```

In this example, DIANA will generate a null-vector for load set 3!

2.3.9 Influence Fields

An influence value is the value of a stress or strain in a specific point of the model as a function of the location of a nodal load. In other words: the influence value for node N represents the value of the stress or strain in point i of element E if a unit load acts in node N . The set of influence values obtained if the load respectively acts in each node of the model is called the *influence field*.

If you specify subtable INFLUE, with the element number and the type and direction of the nodal load, then DIANA calculates the influence fields for all the points in the element.

Influence field determination cannot be combined with other loads.

No other subtables may be present in table 'LOADS' if subtable INFLUE is specified.

syntax

'LOADS'			
INFLUE			
1	5	6	80
$elem_n$	$type_w$	$dirnr_n$	

$elem$ is an element number.

$type$ is the load type: FORCE or MOMENT. The direction number $dirnr$ refers to table 'DIRECT' [§ 1.5 p. 11]: a force in, or a moment around the specified direction.

2.3.9.1 Determination and Output

DIANA determines the influence fields with the standard commands for linear static analysis [Ch. 4] and for calculation of postanalysis results [§ 4.2 p. 73]. Influence fields may be output in tabular form, graphically or on a file for external postprocessing. The output of the influence field for a specific point may be selected with appropriate ELEMENT selection commands in the SELECT block of the OUTPUT commands. The usual STRAIN or STRESS commands specify the strain or stress for which the influence field must be determined. A node selection in the output commands indicates that the load only acts on parts of the model.

file.dcf

```
analysis commands
BEGIN OUTPUT FEMVIEW
  BEGIN SELECT
    BEGIN ELEMEN
      INTPNT 2
      NODES 4
    END ELEMEN
    NODES 35-80
  END SELECT
  STRESS MOMENT XX
END OUTPUT
```

The ELEMEN block gives the selection of points for which influence fields must be plotted. Note that the element number is not specified because it was input in subtable INFLUE. Influence fields are plotted for integration point 2 and for relative node number 4 of the element. The second NODES command indicates that the unit load acts on nodes 35 to 80. See also Volume *iDIANA* and examples bridge and skewpl [Vol. *Analysis Examples*].

Chapter 3

General Commands

See Volume *Getting Started* for a general description of DIANA batch analysis commands.

3.1 Control Commands

Control commands control general aspects of a run. They are optionally specified in the command file before the first module command.

syntax

```
[ _____ ]  
LOG  
NOLOG  
[ NUMTHR numthrn ]  
[ ERRORS{.CPF} { _____ } ]  
                MF=maxfatn  
                MW=maxwarn  
...      module commands  
*END
```

LOG to enable log lines. See Volume *Getting Started* for general description of DIANA job logging. [LOG]

NOLOG to disable log lines.

NUMTHR *numthr* is the number of threads to be used for parallel processing of element loops. The default number of threads depends on the available system resources (number of cores), and is maximized by the number of active elements in the model divided by 100. Note that the actual number of threads being used for parallel processing of element loops will never exceed the number of elements.

ERRORS to fine-tune the error handling. See Volume *Getting Started* for general description of DIANA's error message facility. [ERRORS.CPF]

ERRORS.C to print error codes, ERRORS.P to print parameters like severity and reference, ERRORS.F to print full error text.

Parameter MF=*maxfat* is the maximum number of fatal errors allowed. If the fatal error count exceeds this number, the run will be aborted. [MF=10]

Parameter MW=*maxwar* is the maximum number of warnings. If the warning count exceeds this number, additional warnings will be suppressed. [MW=50]

Example*file.dcf*

```
NOLOG
ERRORS.F MF=5 MW=20
*INPUT
*END
```

3.2 FILOS File Maintenance

The central database for each analysis project with DIANA is handled by FILOS, the FILE Organization System and hence called ‘the FILOS file’. The user must maintain a FILOS file during the whole lifetime of the analysis project for a particular finite element model. Basically there are two types of maintenance: external and internal.

External maintenance by means of operating system commands. These commands perform such things as backup on magnetic tape, transfer in a computer network, copy, delete etc.

Internal maintenance by means of DIANA commands for Module FILOS. Most notably is initialization, furthermore there are commands to expand, compress or erase the FILOS file etc.

The FILOS file is a binary file and cannot be ported between different brand computers, nor can it be printed, displayed on a terminal screen or edited by means of a text editor.

3.2.1 Tutorial Introduction

The possibilities of Module FILOS are manifold. Fortunately the commonly used commands are very simple as will be shown in the following examples.

Initialize Filos file*file.dcf*

```
*FILOS
INITIALIZE
```

This command initializes a FILOS file, with a virtually unlimited size.

Compress Filos file*file.dcf*

```
*FILOS
COMPRESS
```

Sometimes, after many runs, it may be useful to compress the FILOS file with the command as shown above. This causes unused space, due to deletion of items, to be collected and made available for reuse.

Show disk space of Filos file*file.dcf*

```
*FILOS
INFORM SPACE
```

This command tells you how many disk space the FILOS file currently uses, and how many free space it still contains.

Command sequence. It must be emphasized that the commands for Module FILOS may be placed in between other module commands as shown in the following example:

Backup the Filos file

file.dcf

```

*FILOS
INITIA
*INPUT
*LINSTA
*FILOS
COPY FILE="Backup1"
*NONLIN
...
commands for nonlinear analysis
*FILOS
COPY FILE="Backup2"
*END

```

These commands are for a complete linear and nonlinear analysis with two backup versions made: one just after the linear analysis and another one just after the nonlinear analysis.

3.2.2 Command Syntax

This section describes the commands for Module FILOS to perform internal maintenance of the FILOS file. The name of the FILOS file must be specified beforehand in an environment symbol or in the system command to start up the DIANA run [Vol. *Getting Started*].

syntax

```

*FILOS
[ INITIA ]
[ UNLOCK ]
[ CHECK ]
[ BEGIN SAVE
  [ FILE=file_s ]
  [ ITEMS=items_w... ]
  END SAVE ]
[ COPY FILE=file_s ]
[ BEGIN RESTOR
  [ FILE=file_s ]
  [ ITEMS=items_w... ]
  END RESTOR ]
[ COMPRE ]
[ BEGIN INFORM
  [ SPACE {          } ]
    DIRECT
    FILE
  END INFORM ]

```

***FILOS** starts the module to maintain the FILOS file. There are no default commands for Module FILOS, if you only specify the ***FILOS** command nothing will happen.

INITIA to initialize the FILOS file. Any file must once be initialized to become a legal FILOS file. The ***FILOS INITIA** command may only appear at the beginning of a command file, it will be ignored on other places!

The INITIA command causes all the data on the FILOS file to be lost; consequently the analysis of the finite element model must start from the beginning, with reading the input data.

UNLOCK to unlock the FILOS file. A locking mechanism prevents that two DIANA jobs simultaneously access the same FILOS file. The 'lock' is removed when a job is

legally finished. If by any chance the job is not legally finished, the FILOS file remains locked and must be unlocked at the beginning of the next job.

CHECK to check the consistency of the FILOS file. After system problems (disk errors, break down) the consistency of the FILOS file may be damaged. This command checks if the FILOS file is still OK and usable for further analysis. If not, you must delete it and either initialize a new one and start all over again or restore a backup version that you kept somewhere.

SAVE to save a selection of data *items* on another FILOS-like file. If you do not specify *items* then all data will be saved. By default DIANA saves to the file `diana_save.ff`.

COPY to copy the FILOS file. There are two reasons to use this command instead of the operating system ‘copy’ command: (1) it yields a compressed FILOS file or (2) it gives the opportunity to backup the FILOS file during the analysis process. DIANA copies the FILOS file onto the file specified by `FILE=file`.¹

RESTOR to restore a selection of data *items* from another FILOS-like file. If you do not specify *items* then all data will be restored. DIANA copies the data items from the file specified by `FILE=file`.¹ By default DIANA restores from the file `diana_save.ff`.

*Note that the FILOS **SAVE/RESTOR** commands are different from the **SAVE/RESTOR** commands to save or restore executed steps in a nonlinear structural analysis [§ 13.3.9 p. 245].*

COMPRES to compress the FILOS file. DIANA collects wasted disk space, for instance due to deletion of data items, and puts it at the end of the FILOS file to make it available for reuse.

INFORM to get information about the FILOS file:

SPACE the amount of memory space used for the FILOS file: **FILE** the number of blocks currently used, **DIRECT** the number of directory entries. By default, if you specify **SPACE** without any option, DIANA informs you about both blocks and directories.

3.3 Reading or Remaking Input Data

then

Module INPUT is primarily used to read the input file containing tables. The input files can be data files (*.dat) or FX⁺ files (*.fxd). Moreover INPUT can printout (remake) tables that were previously read and it can build tables out of data on the FILOS file. All the other modules take their information from the FILOS file and need no input data, but only user commands.

syntax

***INPUT**

[READ { *option*_w } [FILE=*infil*_s] [TABLE *tabnam*_{w...}]]

ECHO

APPEND

[DELETE TABLE *tabnam*_{w...}]

[REMAKE [FILE=*outfil*_s] [TABLE *tabnam*_{w...}]]

***INPUT** starts Module INPUT to read or write input data. If you do not specify any of the following commands then DIANA reads all tables on the input file.

[READ]

¹If there is no extension in *file* (a period and some characters) DIANA adds the extension `.ff` to it.

READ to read input tables from data files (*.dat) or FX⁺ files (*.fxd). All data from previously read tables will be overwritten! The **ECHO** option asks DIANA to echo the input on the standard output file, while being read. Due to the **APPEND** option, the input data will be appended to previously read tables.

By default, DIANA takes the file name from the run command [Vol. *Getting Started*]. With parameter **FILE=** you may overrule this name by *infil*. If there is no extension in this file name (a period and some characters) then DIANA assumes the extension *.dat*.

By default DIANA will read all tables on the input data file. With the **TABLE** option you may select one or more specific tables *tabnam* to be read.

DELETE TABLE to delete one or more specified tables *tabnam* from the FILOS file. These tables must have been read previously.

REMAKE to write tables to a file. DIANA will remake the tables, i.e., compose them out of evaluated and stored data from the FILOS file. Remade tables will contain the same data as those previously read, but possibly in an other lay out.

By default, DIANA writes the tables to the standard output file. If you specify parameter **FILE=** the tables go to a file named *outfil*. If there is no extension in this file name (a period and some characters) then DIANA adds the extension *.dat* to it.

By default DIANA will write all tables for which data is available on the FILOS file. With the **TABLE** option you may select one or more specific tables *tabnam* to be remade.

Default

file.dcf

*INPUT

By default all tables on the input file are read and not echoed if only the *INPUT command is specified, it has the same effect as:

file.dcf

*INPUT
READ

Read specified tables and echo

file.dcf

*INPUT
READ ECHO TABLE LOADS SUPPOR

These commands cause the specified tables to be read and the input lines simultaneously written to the standard output file.

Remake specified tables

file.dcf

*INPUT
REMAKE FILE="myfile" TABLE LOADS SUPPOR

These commands cause the specified tables to be written to the file *myfile.dat* even if they were never read with Module INPUT.

3.4 Element Evaluation

Due to the **EVALUA** command(s) DIANA will check and evaluate the geometric and material properties of elements. You may customize the evaluation process to check the quality of the mesh. The **OFF** option switches OFF the element evaluation which may save computing time if this task was performed previously.

syntax

```

BEGIN EVALUA
[ OFF ]
[ CHECK { _____ } ]
      SHAPE=eshaper
      RATIO=eratior
[ BEGIN AVERAG
[ OFF ]
[ ANGLE=nangler ]
END AVERAG ]
[ BEGIN REINFO
[ OFF ]
[ INTERF [ _____ ] ]
      ON
      OFF
[ GRID FULLEL ]
END REINFO ]
[ COMPOS [ OFF ] ]
END EVALUA

```

CHECK performs an *extended test* on element shape and aspect ratio. By default DIANA does not perform this test.

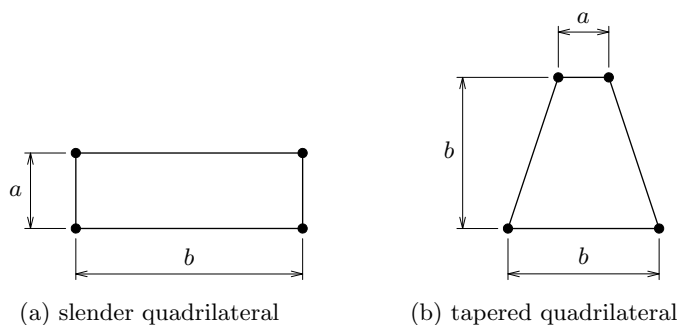


Figure 3.1: Element shapes

The extended test checks whether the element shape deviates too much from the ideal shape and whether the aspect ratio is extremely high. Two parameters are appropriate for this test [Fig. 3.1].

Table 3.1: SHAPE PARAMETERS

b/a	Slender quadrilateral		Tapered quadrilateral	
	<i>eshape</i>	<i>eratio</i>	<i>eshape</i>	<i>eratio</i>
1.0	0.0	1.00	0.0	1.00
1.5	0.0	1.25	2.5×10^{-4}	1.25
2.0	0.0	1.50	6.7×10^{-4}	1.50
5.0	0.0	3.00	$24. \times 10^{-4}$	3.00

Table 3.1 shows some typical values of the shape parameters for the slender and tapered quadrilaterals of Figure 3.1. See Roddeman [74] for background theory. You may overrule the default values by specifying parameters:

[*eshape*=0.001]

SHAPE=*eshape* the tolerance for the Shape test.

[*eratio*=3.0]

RATIO=eratio the tolerance for the Aspect Ratio test.

If one of the shape parameters of an element is greater than the specified tolerance, DIANA will produce a warning message. The extended check option checks linear and quadratic elements of the following element types: plane stress, plane strain, axisymmetric, plate bending, flat shell, solid, and regular flow elements. Excluded are interfaces, beams trusses, boundary elements etc. Especially for solid elements, the extended test consumes a considerable amount of processing time.

AVERAG performs an averaging procedure on the nodal normals of all curved shell elements and plane, 2-D line, and line-solid interface elements attached to a node. If you specify the **OFF** option, then DIANA will not perform the averaging procedure of the nodal normals. [AVERAG]

ANGLE=nangle the tolerance angle in the used units (radians or degrees) for the averaging procedure. [*nangle*=20°]

For these elements for which neither a predefined shape nor explicit thickness vectors are defined in table 'GEOMET', DIANA will calculate by default the nodal normals using the procedure given below.

1. The element normal directions at the nodes are initially determined by assuming a parabolic shape of the element surface, see Volume *Element Library*.
2. The angle between the normals for all elements connected to a node are computed.
3. When a nodal normal is less than $\frac{1}{2}$ *nangle* within line from a nodal normal of the elements with a predefined shape, the nodal normal is adjusted to the nodal normal of that element with a predefined shape having the smallest angle.
4. For all nodal normals which have not been adjusted, element sets are created whose normals are mutually within line of the defined tolerance angle *nangle*.
5. For each created set, the element normals in the set are averaged and the corresponding nodal normal for all elements in the set are replaced by this averaged normal. No averaging occurs for elements which are not put in any created set.

REINFO indicates specific settings for evaluation of reinforcements. The **OFF** option switches OFF the reinforcement evaluation which may save computing time if this task was performed previously. [REINFO]

INTERF activates the generation of reinforcement parts in interface elements during the preprocessing of section input [Vol. *Element Library*]. Diana does not generate embedded reinforcement parts in interface elements if you specify the **OFF** option, or if you do not specify the **REINFO INTERF** command at all. For bond-slip reinforcements and pile foundations, always a connection is being made when the interface does not have a physical distance between both sides, i.e. the opposite nodes of the interface have the same coordinates. In case the interface is modelled with a physical distance, however, no truss or beam element will be generated in the interface if you specify the **OFF** option, or if you do not specify the **REINFO INTERF** command at all.

GRID FULLEL indicates that grid reinforcement particles will only be defined in those elements which are fully covered by a reinforcement section and not in elements that are partly covered by reinforcement grid sections. By default DIANA generates reinforcement particles in elements that are either fully or partly covered by a reinforcement grid section.

COMPOS indicates evaluation of composed elements. The **OFF** option switches OFF the composed element evaluation which may save computing time if this task was performed previously. [COMPOS]

file.dcf

```

BEGIN EVALUA
  CHECK SHAPE=1.E-4 RATIO=2.5
  REINFO INTERF
END EVALUA

```

file.dcf

```

EVALUA CHECK SHAPE=1.E-4 RATIO=2.5

```

3.5 Elements Assembly

Due to the **ASSEMB** command(s) DIANA will assemble the elements of the model and create the system degrees of freedom. The **OFF** option switches OFF the elements assembly which may save computing time if this task was performed previously.

syntax

```

BEGIN ASSEMB
[ OFF ]
[ TOLERA=eps r ]
[ BEGIN AUTOTY
  [ OFF ]
  [ BEGIN SELECT
    [ ELEMEN elems ng... ]
    [ NODES nodes ng... ]
  END SELECT ]
END AUTOTY ]
END ASSEMB

```

[$\epsilon = 10^{-6}$] **TOLERA=eps** is the criterion ϵ for coincidence of a potential system degree of freedom and an existing one.

AUTOTY turns on the *automatic tying* option. This option generates tyings for coupling of elements with rotational degrees of freedom to elements with translational degrees of freedom only, like beam-plane stress, beam-solid and shell-solid. See § 2.2.8 on page 28 for conditions. If you do not specify **AUTOTY** commands, then DIANA does not perform automatic tying.

[OFF]

OFF explicitly turns off the auto tying option.

SELECT selects parts of the finite element model for which the automatic tying option is active.

ELEMEN elems is a set of elements to be used by the automatic tying option.

If you do not select elements, then all elements will be used.

NODES nodes is a set of nodes to be used by the automatic tying option. If you do not select nodes, then all nodes will be used.

file.dcf

```

BEGIN ASSEMB
  TOLERA=1.E-5
  BEGIN AUTOTY
    SELECT ELEMEN 1-5
  END AUTOTY
END ASSEMB

```


file.dcf

ASSEMB TOLERA=1.E-7

3.6 Output

After analysis of the finite element model, DIANA can output analysis results like displacements, strains and stresses. The actual data that can be output depends on the type of analysis and on the element type, as described in the other chapters in this part. The commands to get output of analysis results are grouped in the following command blocks.

syntax

BEGIN OUTPUT

[OFF]
[SELECT ...]
[LAYOUT ...]
...

END OUTPUT

OUTPUT selects analysis results for output [§ 3.6.1 p. 56].

SELECT selects parts of the model [§ 3.6.2]. This selection is optional, if you do not specify it then DIANA gives output for the entire model, i.e., all nodes, elements, and reinforcements.

LAYOUT customizes the layout of tabular output [§ 3.6.4.1 p. 65].

Scope of command blocks. There may be multiple OUTPUT blocks. In this case, the selections and layout specifications are only valid for that particular OUTPUT block.

file.dcf

*LINSTA
BEGIN OUTPUT
 BEGIN SELECT
 LOADS 4 6
 NODES 10-80(5)
 ELEMENTS 25 38
 END SELECT
 DISPLA GLOBAL
 STRESS VONMIS
END OUTPUT
BEGIN OUTPUT
 BEGIN SELECT
 LOADS 8
 END SELECT
 STRESS PRINCI
END OUTPUT

These commands select load sets 4 and 6 for output of the global displacements and Von Mises stresses in linear static analysis results. You will get the displacements in the global coordinate system for nodes 10, 15, ..., 80, and the Von Mises stresses for elements 25 and 38. For load set 8 you will get the principal stresses for all elements. The same output would have resulted from the following commands. In short format the commands for the last OUTPUT block could have been as follows.

file.dcf

```
*LINSTA
BEGIN OUTPUT
  SELECT LOADS 8
  STRESS PRINCI
END OUTPUT
```

3.6.1 Output Selection

The OUTPUT block comprises commands to specify for which analysis results DIANA must produce output. The output ‘device’ (tabular, postprocessing etc.), and some options and parameters may be specified in the starting line of this command block.

syntax

```
BEGIN OUTPUT [ devicew ] { outoptw } { paramss }
               NDIANA      BINARY      FILE=files
               TABULA      ASCII       ...
               FEMVIE      APPEND
               FXPLUS

[ OFF ]

itemw [ typew ] [ formw ] [ operw ] { compw } { locaw } { optiw } ...
TOTAL    TRANSL  LOCAL    ...      NODES    AXES
INCREM   ROTATI  GLOBAL                    INTPNT  NOAXES
INITIA   FORCE   PRINCI                    CENTER  COOR
PHASE    DISFOR  VONMIS                    AVJSCE   NOCOOR
REACTI   MOMENT  INVARI                               SMOOTH
RESIDU   DISMOM  REINFO                               ERROR
PRESCR   CAUCHY  VOLUME                               NOBOND
ELEMEN   PIOLAK  REAXES                               CENTAX
REINFO   GREEN   ...                               RELATI
...      TRACTI
          PRESSU
          TEMPER

END OUTPUT
```

OUTPUT command block to specify desired output of analysis results. You may put the following additional keywords in the starting line:

- [NDIANA] *device* specifies the way of output:
- NDIANA for interactive postprocessing of analysis results with DianaIE [§ 3.6.3 p. 63]. This is the default output device when the analysis is neither started from iDIANA nor FX⁺.
 - TABULA for tabular output [§ 3.6.4 p. 64].
 - FEMVIE for interactive postprocessing of analysis results with iDIANA [§ 3.6.5 p. 66]. This is the default output device when the analysis is started from iDIANA.
 - FXPLUS for interactive postprocessing of analysis results with FX⁺ [§ 3.6.6 p. 67]. This is the default output device when the analysis is started from FX⁺.
- outopt* Are additional output options, availability depend on the output device.
- BINARY to get output in binary format.
 - ASCII to get output in ASCII text format.
 - APPEND to append the output to an existing output file.

params are additional parameters, depending on device and analysis type. *FILE=file* is a general parameter which specifies the name of the output file. This name overrules the file name specified in the *diana* run command [Vol. *Getting Started*]. DIANA adds an appropriate extension to the specified file name for instance *.tb* for tabular output [§ 3.6.4 p. 64]. See also Volume *Getting Started* for files in a DIANA job.

OFF suppresses all output of analysis results.

item is the name of the result item to be output. Names depend on analysis type for instance *DISPLA* for displacements, *STRESS* for stresses. Most result items have additional specifiers for type, location etc. Item names and specifiers depend on the item itself, the analysis type and the element type, they are described in the appropriate volumes of the User's Manual.

type is the type of the result item.

TOTAL for total.

INCREM for incremental.

INITIA for initial.

PHASE for the increment of a phase in phased analysis.

REACTI for reaction.

RESIDU for residual.

PRESCR for prescribed.

... more *types* for specific analysis types.

form is the theoretical formulation of the result item.

TRANSL for translation.

ROTATI for rotation.

FORCE for force.

DISFOR for distributed force.

MOMENT for moment.

DISMOM for distributed moment.

CAUCHY for Cauchy stress.

PIOLAK for Piola–Kirchhoff stress.

GREEN for Green–Lagrange strain.

TRACTI for tractions of interface elements.

PRESSU for pressure.

TEMPER for temperature.

See also the description of particular result items.

oper is an operation on the result item to be performed prior to its output. General operations are listed below. For analysis and/or result specific operations see the descriptions of the result items.

LOCAL for transformation to local *xyz* directions,

GLOBAL for transformation to global *XYZ* directions,

PRINCI for transformation to principal directions for strains [§ 47.1.2 p. 574] or stresses [§ 47.2.2 p. 575].

VONMIS for transformation to equivalent Von Mises strains [§ 47.1.1 p. 573], or Von Mises stresses [§ 47.2.1 p. 575].

INVARI for stress invariants [§ 47.2.5 p. 576].

REINFO for reinforcement moments and forces [§ 47.2.6 p. 576].

VOLUME for volumetric strain [§ 47.1.3 p. 574].

REAXES for strain vectors. You may give special commands in the *SELECT* block if you ask output of strain vectors [§ 3.6.2.3 p. 60].

... more *operations* for specific analysis types.

comp are the components of the result item to be output, for instance **XX** for σ_{xx} or **XY** for σ_{xy} . Components depend on the particular result item, see the descriptions. If you do not select components then DIANA gives output for all components of the result item. Components only apply for tabular output; for output to the iDIANA Results environment all components will be written.

loca specifies the location, see Volume *Element Library* for definitions.

NODES for nodes. For class-I beams elements the default nodal stress and strain output results are made at the extreme fiber locations of the beam cross-section instead of the specified or default stress points. The same style is followed also for nodal stress and strain output results of class-II and class-III beam elements instead of providing the output results at all fiber locations at the integration points of the beam cross-section.

INTPNT for element integration points.

CENTER for averaged results in the element center point.

AVJSCE for averaged results over a region with radius r according to the Japan Society of Civil Engineers (JSCE) [§ 3.6.2.4 p. 61]. The radius r is defined in the element selection block by the **RAJSCE** parameter.

opti are additional options for tabular output. If appropriate, options are extensively explained in the chapters for a particular type of analysis or in Volume *Element Library*.

[NOAXES]

AXES gives output of the relevant axes directions, for example:

- directions of the degree of freedom for nodal local results
- directions of the element axes for element local results
- directions of the principal stresses or strains

NOAXES suppresses this output.

[NOCOOR]

COOR gives output of the coordinates of the location of the result item, for instance the integration points, **NOCOOR** suppresses this output.

SMOOTH to ‘smooth’ the element data values like strains and stresses, after extrapolation to the nodes. Smoothing yields the average value of the element contributions at the node. This option can only be applied for results in global orientation. Smoothing is particularly useful for graphic output, it avoids overlapped plotting of strain and stress rosettes.

ERROR gives the maximum value of the ‘error’ (deviation) of the smoothed value and the extrapolated value.

NOBOND includes the contribution of prestress in posttensioned reinforcements to element forces and moments [Vol. *Element Library*].

CENTAX will automatically shift the specified composed line or composed surface to the centroidal axis of the structure and output the moment and/or distributed moment with respect to the shifted composed line and/or surface.

RELATI gives the displacement, velocity, or acceleration results relative to an explicitly defined base node indicated by the **BASNOD** command [§ 3.6.2 p. 59] in a linear static, linear transient, hybrid frequency time domain (HFTD), or a structural nonlinear analysis, or to the defined base excitation [§ 6.3 p. 104] in a transient analysis when no base node is defined explicitly.

... other options may be special for a particular type of analysis, for instance **AMPLIT** and **COMPLE** for a frequency response analysis [§ 8.3 p. 153].

file.dcf

```
BEGIN OUTPUT TABULA NONLIN FILE="myfile"
DISPLA TOTAL GLOBAL Y COOR
STRESS TOTAL CAUCHY LOCAL XX YY XY INTPNT AXES
END OUTPUT
```

These commands produce tabulated output of nonlinear analysis on a file named `myfile.tb`. The output comprises displacements and stresses: the total displacements in global Y direction u_Y including the original coordinates of the nodes and the total Cauchy stresses in local element direction, σ_{xx} σ_{yy} σ_{xy} , located in the integration points. The directions of the local xy axes are also output.

3.6.2 Model Selection

The **SELECT** block comprises commands to select parts of the model for which DIANA must produce output of analysis results.

syntax

```

BEGIN SELECT
[ NODES ... ] ...
[ ELEMEN ... ] ...
[ REINFO ... ] ...
[ BASNOD bnn ]
[ fieldw [ _____ ] [ special ]
      fldnrsn...
      ALL
END SELECT

```

NODES is a subcommand block for node selection [§ 3.6.2.1].

ELEMEN is a subcommand block for regular element selection [§ 3.6.2.2] and composed element selection [§ 3.6.2.6].

REINFO is a subcommand block for reinforcement selection [§ 3.6.2.5].

BASNOD *bn* defines the base node for relative displacements, relative velocities, and relative accelerations indicated by the **RELATI** option for the respective result item [§ 3.6.1]. When the base node *bn* is defined, the results will be output relative to this base node. When no base node is defined, result items with the **RELATI** option will lead to results relative to the defined base excitation [§ 6.3 p. 104].

field select fields: *fldnrs* are field numbers, **ALL** indicates all fields. The field name [ALL] *field* depends on the analysis type, for instance **LOADS** for load sets in linear static analysis, **STEPS** for nonlinear or dynamic analysis, **MODES** for eigenvalue or stability analysis. See the appropriate chapters in this part.

Special selections may be available for a particular application, for instance in linear static analysis you may select the minimum and maximum values for (selected) load sets [§ 4.2.1 p. 75].

file.dcf

```

BEGIN SELECT
  NODES 1 3 10-80(5) 123 125 \
        200-300
  LOADS 4 6
END SELECT

```

3.6.2.1 Node Selection

The **NODES** command block is a sub-block in the **SELECT** block to select nodes for output.

syntax

```

BEGIN NODES [ _____ ]
              nodesng...
              ALL
              NONE
END NODES

```

- [ALL] **NODES** node selection: *nodes* is a series of nodes specified by numbers or groups or both. ALL indicates all nodes. NONE indicates no nodes at all which is useful to suppress output for nodes, for instance to get element results only.

3.6.2.2 Element Selection

The ELEMEN command block is a sub-block in the SELECT block to select elements for output.

syntax

```

BEGIN ELEMEN [ _____ ]
              elemsng...
              ALL
              NONE
[ NODES _____ ]
              rnonrsn...
              ALL
[ INTPNT _____ ]
              intnrsn...
              ALL
[ REAXES parameters for stress and strain transformation ]
[ AVJSCE parameters for result averaging according to JSCE ]
END ELEMEN

```

- [ALL] **ELEMEN** element selection: *elems* is a series of elements specified by numbers or groups or both. ALL indicates all elements. NONE indicates no elements at all which is useful to suppress output for elements, for instance to get reinforcement results only.

- [ALL] **NODES** selects element nodes: *rnonrs* are relative node numbers, 1 for the first node, 2 for the second etc., ALL indicates all element nodes.

- [ALL] **INTPNT** selects element integration points: *intnrs* are integration point numbers, ALL indicates all element integration points.

REAXES specifies some parameters that DIANA applies to transform stresses and strains [§ 3.6.2.3].

AVJSCE specifies the parameters for averaged results over a region with radius *r* according to the Japan Society of Civil Engineers (JSCE) [§ 3.6.2.4].

file.dcf

```

BEGIN SELECT
  BEGIN ELEMENTS 5-100(5)
    NODES 1 2
  END ELEMENTS
END SELECT

```

This selection comprises the first and second node of elements 5, 10, 15, ..., 100.

3.6.2.3 Stress and Strain Transformation

*syntax***BEGIN REAXES**[R1= $x1_r$ $y1_r$ [$z1_r$]][R2= $x2_r$ $y2_r$ [$z2_r$]][ZR= $zr1_r$ [$zr2_r$]][CO= $co1_r$ [$co2_r$]]**END REAXES**

REAXES in the ELEMEN command block specifies, for the selected elements, the axes for transformation of strains or stresses. In combination with the REAXES operation, these axes specify the normal axes of the planes on which the stress or strain vectors apply. In combination with the REINFO operation, these axes specify the two reinforcement axes, see § 47.2.6 on page 576 for background theory.

R1= specifies the first axis. Two values $x1$ and $y1$ specify a vector in the element xy axes. Three values $x1$, $y1$ and $z1$ specify a vector in the model XYZ axes. Default is the element x axis.

[R1=1. 0.]

R2= specifies the second axis. Two or three values are interpreted analogous to parameter R1. Default is the element y axis.

[R2=0. 1.]

For stress and strain vectors, there are no restrictions for the two axes. For reinforcement forces and moments, the model axes will be projected into the elements xy plane and therefore the specified axes should not be perpendicular to the xy plane of the element. Moreover, the two reinforcement axes should not coincide. If required, DIANA will take the second axis opposite to the specified direction, in order to ensure that the third axis coincides with the element z axis.

ZR= specifies the relative internal beam arms z_r : $zr1$ for the first reinforcement axis, $zr2$ for the second. If you only specify $zr1$, then DIANA applies the same internal beam arm for both reinforcement axes. ($z_r > 0$)

CO= specifies the coverages co , i.e., the distance from the center of the reinforcement to the outer surface: $co1$ for the first reinforcement axis, $co2$ for the second. If you only specify $co1$, then DIANA applies the same coverage for both reinforcement axes. ($co > 0$)

Parameters ZR and CO only apply for reinforcement forces and moments but not for plane stress elements.

3.6.2.4 Result Averaging over a Region According to JSCE

For the damage index output (STATUS DAMIND [§ 13.4.4.3 p. 262]) and the normalized cumulative energy (STATUS ENERGY NRMCM [§ 13.4.4.4 p. 262]) the results can be averaged over a region with radius r around the integration points as illustrated in Figure 3.2. The radius r of the region is defined by the RAJSCE parameter that should be specified in the element selection block according to the following syntax:

*syntax***BEGIN AVJSCE**RAJSCE r_r **END AVJSCE**

AVJSCE in the ELEMEN command block specifies, for the selected elements, the parameters needed for the result averaging over a region according to the Japan Society of Civil Engineers (JSCE):

RAJSCE r is the radius r of the region over which the results will be averaged.

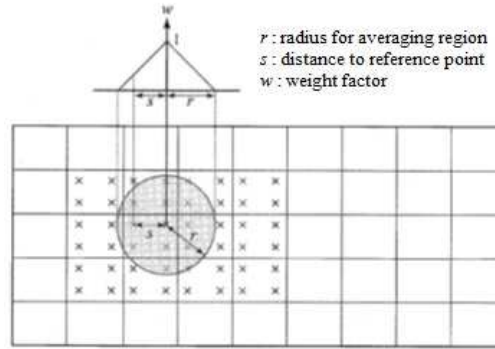


Figure 3.2: Result averaging over a region with radius r according to JSCE

The contributions of the integration points are weighted by the distance s to the reference integration point: when the distance s is larger than the radius r , the value of that integration point does not contribute to the averaged results; for integration points with a distance s smaller than radius r the weight factor w linearly decreases with the distance to the reference point [Fig. 3.2]. In different model selection blocks different values for the radius r can be defined.

3.6.2.5 Reinforcement Selection

The REINFO command block is a sub-block in the SELECT block to select reinforcements for output.

syntax

```

BEGIN REINFO [ _____ ]
               reinfsng...
               ALL
               NONE
[ INTPNT _____ ]
               intnrsn...
               ALL
[ ELEMEN _____ ]
               elemsng...
               ALL
END REINFO

```

[ALL] REINFO reinforcement selection: *reinfs* is a series of reinforcements specified by numbers or groups or both. ALL indicates all reinforcements. NONE indicates no reinforcements at all which is useful to suppress output for reinforcements, for instance to get element results only.

[ALL] INTPNT selects reinforcement integration points: *intnrs* are integration point numbers, ALL indicates all reinforcement integration points.

[ALL] ELEMEN selects elements for reinforcement output: *elems* is a series of elements specified by numbers or groups or both. Results will only be output if the reinforcement is embedded in one of the selected elements. ALL indicates all elements.

file.dcf

```

BEGIN SELECT
  BEGIN REINFO 1 4 5
    ELEMEN 30-40
  END REINFO
END SELECT

```


This selection comprises those parts of reinforcements 1, 4 and 5 which are embedded in one of the elements 30 to 40.

file.dcf

```
BEGIN SELECT
  ELEMEN NONE
  REINFO 1 4 5
END SELECT
```

This selection comprises output for reinforcements 1, 4 and 5 only. No output will be given for elements!

3.6.2.6 Composed Element Selection

For composed elements the regular structural elements in the element selection (`SELECT ELEMEN`, see § 3.6.2.2 p. 60]) and reinforcement selection (`SELECT REINFO`, see § 3.6.2.5 p. 62]) are used to define that composition results will be based on a selection of structural elements and/or reinforcements in the model. By default all structural elements and all reinforcements will be considered. When only reinforcements are selected, and no regular structural elements are selected, only the stresses in the reinforcements will contribute to the composed elements results. When only elements are selected and no reinforcements are selected, only the stresses in the elements will contribute to the composed element results.

file.dcf

```
BEGIN SELECT
  ELEMEN 3 5 11-20 /
END SELECT
BEGIN OUTPUT
  STRESS MOMENT
END OUTPUT
```

These commands ask to output the bending moments of composed elements 3 and 5, which take the contributions of elements 11 to 20 into account. See also the example cbeam in Volume *Analysis Examples*.

file.dcf

```
BEGIN SELECT
  ELEMEN "Left Composition" /
  REINFO ALL /
END SELECT
BEGIN OUTPUT
  STRESS MOMENT
END OUTPUT
```

These commands ask to output the bending moments of the composed elements in the group "Left Composition" and take the contribution of only the reinforcements into account. In this case stresses in elements do not contribute to the composed element results.

3.6.3 Output for Postprocessing with DianaIE

To get output for interactive postprocessing with DianaIE you must specify the `NDIANA` device option. Volume *Getting Started* describes the postprocessing procedures with DianaIE. See also § 3.6.1 on page 56 for general syntax description of `OUTPUT` commands.

syntax

```

BEGIN OUTPUT NDIANA { outoptw } { params }
                    BINARY
                    itemw ...

```

```
END OUTPUT
```

[BINARY] BINARY writes output to native DIANA files for DianaIE (the default). These neutral files are portable between different brands of computers.

item specifies the output item(s) [§ 3.6.1 p. 56]. Volume *Getting Started* describes an example with output to DianaIE.

3.6.4 Tabular Output

DIANA produces tabular output if you specify the result items (displacement, strain, stress etc.) in an OUTPUT command block with the device name TABULA. By default DIANA chooses a page and table layout which suits most cases. To customize the layout of the tabular output you may specify a LAYOUT command block.

syntax

```

BEGIN OUTPUT TABULA { outoptw } { params }
                    APPEND

```

```
[ TEXT texts ]
```

```
[ BEGIN LAYOUT
```

```
...
```

```
END LAYOUT ]
```

```
itemw ...
```

```
END OUTPUT
```

APPEND causes the output to be appended to an existing file.

TEXT *text* is a string of text which DIANA will put left justified in the header line. By default, DIANA applies the base name of the tabular file as header text.

LAYOUT customizes the layout of the tabular output [§ 3.6.4.1 p. 65].

item specifies the output item(s) [§ 3.6.1 p. 56].

file.dcf

```

BEGIN OUTPUT TABULA
  STRESS CAUCHY PRINC1
  STRESS CAUCHY VONMIS
END OUTPUT

```

As shown below, these commands produce two tables: one for the principal stresses and another one for the equivalent Von Mises stress.

file.tb

```

Analysis type      LINSTA
Load case nr.      1
Result             STRESS TOTAL  CAUCHY
Tensor invariant    PRINC1
Location of results NODES

Elmnr Nodnr      S1      S2      S3
1      1      3.814E+05 -3.332E+05 0.000E+00
      2      5.041E+05 -2.856E+05 0.000E+00
      6      1.503E+06  1.995E+05 0.000E+00
      5      1.174E+06  2.651E+05 0.000E+00
2      2      2.651E+06  5.335E+05 0.000E+00
      3      2.691E+06  4.167E+05 0.000E+00
      7      2.166E+06 -1.657E+05 0.000E+00
      6      2.172E+06  4.495E+04 0.000E+00

Analysis type      LINSTA

```

```

Load case nr.      1
Result            STRESS TOTAL  CAUCHY
Tensor invariant  VONMIS
Location of results  NODES

```

```

Elmnr  Nodnr      Seq
  1      1    6.193E+05
        2    6.925E+05
        6    1.414E+06
        5    1.066E+06
  2      2    2.428E+06
        3    2.509E+06
        7    2.254E+06
        6    2.150E+06

```

This output is from a linear static analysis of a model with plane stress elements. See Volume *Element Library* for description of the available stresses for these elements. See also § 4.2.4 on page 81 for description of output labels like s1, s2, s3 and Seq.

file.dcf

```

BEGIN OUTPUT TABULA LINSTA APPEND FILE="myfile"
  TEXT "Example of displacement output"
  DISPLA COOR
END OUTPUT

```

These commands append tabular output of the displacements of linear static analysis, including the coordinates of the nodes, to the file `myfile.tb`. The header text will be "Example of displacement output".

3.6.4.1 Layout Specification

The LAYOUT commands customize the layout of the tabular output.

syntax

```

BEGIN LAYOUT
[ LINPAG nlinesn ]
[ COLLIN ncolsn ]
[ DIGITS { _____ } ]
      RESULT ndgresn
      COORD ndgcoon
      AXES ndgaxen
[ COMBIN ]
END LAYOUT

```

LINPAG *nlines* specifies the number of lines per page. The default is 66 lines per page, including the header and the blank line. If *nlines* is equal to zero or negative [*nlines* = 66] no number of lines per page will be set, and therefore no additional lines will be generated in the tabulated output file.

COLLIN *ncols* specifies the number of columns per line. The default is 130 columns. [*ncols* = 130]

DIGITS specifies the amount of digits (number symbols) in the format of floating point values. DIANA always puts a single digit in front of the decimal point. For instance, if you specify three digits then the format looks like 2.38E+01.

DIANA never formats more than ten digits. As DIANA runs on machines with double precision accuracy, yielding 15 decimals, the maximum number of digits to be formatted is within the accuracy. ($1 \leq ndg... \leq 10$)

RESULT *ndgres* specifies the number of digits for the analysis results. The default format for result values is four digits which looks like 2.385E+01. [*ndgres* = 4]

COORD *ndgcoo* specifies the number of digits for point (node) coordinates. The default format for coordinates is four digits which looks like 2.385E+01. [*ndgcoo* = 4]

AXES *ndgaxe* specifies the number of digits for the direction of local axes. The default format for axes directions is three digits which looks like 0.38. DIANA [ndgaxe = 3] normalizes the axes directions which yields values between -1 and 1 therefore an exponent is omitted.

COMBIN causes the output items to be assembled in one table. In this case, the location specifier *loca* [§ 3.6.1 p. 56] must be the same for all items to be combined in one table. Therefore DIANA assumes the location of the first item to be valid for all items in the set. If you specify different locations for the items in the set, then DIANA gives a warning message.

This option may only be applied with explicitly specified output items, i.e., it does not work with default output. This option is not available for composed elements. DIANA will combine at most fifteen items in one table.

file.dcf

```
BEGIN OUTPUT TABULA
  LAYOUT COMBIN
  STRESS CAUCHY PRINCI
  STRESS CAUCHY VONMIS
END OUTPUT
```

As shown below, these commands produce one table containing the principal stresses as well as the equivalent Von Mises stress.

file.tb

```
Analysis type      LINSTA
Load case nr.      1
Result             Combination
Location of results NODES

Elmnr Nodnr      S1      S2      S3      Seq
1      1      3.814E+05 -3.332E+05 0.000E+00 6.193E+05
      2      5.041E+05 -2.856E+05 0.000E+00 6.925E+05
      6      1.503E+06 1.995E+05 0.000E+00 1.414E+06
      5      1.174E+06 2.651E+05 0.000E+00 1.066E+06
2      2      2.651E+06 5.335E+05 0.000E+00 2.428E+06
      3      2.691E+06 4.167E+05 0.000E+00 2.509E+06
      7      2.166E+06 -1.657E+05 0.000E+00 2.254E+06
      6      2.172E+06 4.495E+04 0.000E+00 2.150E+06
```

This output is from a linear static analysis of a model with plane stress elements. See Volume *Element Library* for description of the available stresses for these elements. See also § 4.2.4 on page 81 for description of output labels s1, s2, s3 and Seq.

3.6.5 Output for Postprocessing with iDIANA

To get output for interactive postprocessing with iDIANA you must specify the FEMVIE device option. Volume *iDIANA* describes the postprocessing procedures with iDIANA. See also § 3.6.1 on page 56 for general syntax description of OUTPUT commands.

syntax

```
BEGIN OUTPUT FEMVIE { outoptw } { params }
  BINARY
  ASCII
  APPEND

[ MODEL OFF ]
  itemw ...
END OUTPUT
```

[BINARY] BINARY writes output to an iDIANA database (the default).

ASCII writes an iDIANA neutral file in ASCII text format.

When choosing between database or neutral file, you should realize that neutral files are portable between different brands of computers. Databases (binary) are usually smaller than neutral files but generally they are *not* portable.

APPEND appends the output to an existing iDIANA neutral file or database.

MODEL OFF suppresses the output of the data for the finite element model, like node coordinates and element connection. In this case DIANA only outputs the analysis results. For an iDIANA database, which must contain the model data, you must also specify the **APPEND** option.

3.6.6 Output for Postprocessing with FX+

To get output for interactive postprocessing with FX⁺ you must specify the **FXPLUS** device option. Volume *FX+ for DIANA* describes the postprocessing procedures with FX⁺. See also § 3.6.1 on page 56 for general syntax description of **OUTPUT** commands.

syntax

```
BEGIN OUTPUT FXPLUS { outoptw } { params }
                        BINARY
                        itemw ...
```

```
END OUTPUT
```

BINARY writes output to Post-Neutral files for FX⁺ (the default). These neutral files are [BINARY] portable between different brands of computers.

item specifies the output item(s) [§ 3.6.1 p. 56]. Volume *Getting Started* describes an example with output to FX⁺.

Part II

Linear Static Analysis

Chapter 4

Regular Linear Static Analysis

To perform an analysis of a finite element model, you must supply DIANA with two types of information: input data and commands.

Input data describes the geometry, the topology, the boundary condition, the loading and the physical properties of the finite element model. Chapter 1 presents the input of general, analysis independent data like coordinates and units. Chapter 2 describes the additional input for structural analysis. See also Volume *Element Library* and Volume *Material Library*.

Commands to tell DIANA how to analyse the model: linear, static, dynamic, nonlinear etc., and what analysis results must be output. Commands are supplied on a command file according to the rules of grammar and syntax for the DIANA Command Language [Vol. *Getting Started*]. This chapter describes the analysis commands for Module LINSTA which performs *linear*, *static*, structural analysis.

Main tasks

syntax

*LINSTA

[MODEL ...]

[SOLVE [OFF] ...]

[OUTPUT ...] ...

MODEL evaluates and assembles the finite element model [§ 4.1].

SOLVE solves the system of equations. See Chapter 30 for customization of the solution procedure.

OUTPUT selects analysis results for output [§ 4.2 p. 73].

An example of the minimum set of commands to initialize a new FILOS file, read the input file, perform a linear static analysis, and get output of displacements and stresses is:

file.dcf

```
*FILOS
INITIA
*INPUT
*LINSTA
BEGIN OUTPUT
  DISPLA
  STRESS
END OUTPUT
*END
```

For syntax description of the *FILOS commands see § 3.2 on page 48 and for the *INPUT commands § 3.3 on page 50.

Default*file.dcf*

```
*LINSTA
*END
```

If you only give a single ***LINSTA** command, like in the above example, then DIANA evaluates the model and solves the system of equations. You will not get output of analysis results in this case. The default behaviour is as if you had given the following commands.

file.dcf

```
*LINSTA
MODEL
SOLVE
*END
```

4.1 Model Evaluation

The **MODEL** commands customize the evaluation of the finite element model prior to the actual analysis.

syntax

```
BEGIN MODEL
[ OFF ]
[ EVALUA [ OFF ] ... ]
[ ASSEMB [ OFF ] ]
[ MATRIX [ OFF ] ]
[ LOADS [ OFF ] ]
END MODEL
```

EVALUA to check and evaluate geometric and material properties for elements [§ 3.4 p. 51].

ASSEMB to assemble the elements, i.e., creating appropriate system degrees of freedom [§ 3.5 p. 54].

MATRIX to setup the element stiffness matrices.

LOADS to setup the load vectors. You can get output of load vectors via the **INTERN** and **EXTERN** type specifiers of the **FORCE** output item [§ 4.2.6 p. 88].

Provided that the **FILOS** file of a previous analysis is still available, some tasks may be skipped which may save a considerable amount of computing time, especially for large finite element models. We show a few examples.

Change loading*file.dcf*

```
*INPUT
READ TABLE LOADS
*LINSTA
BEGIN MODEL
  EVALUA OFF
  ASSEMB OFF
  MATRIX OFF
  LOADS
END MODEL
SOLVE
BEGIN OUTPUT
...
END OUTPUT
*END
```

A typical example of customized commands is the change of loading and the calculation of new analysis results. Module LINSTA only has to setup the load vectors and solve the system of equations to get the analysis results. We could also have left out the LOADS and SOLVE commands because LINSTA performs a main task unless it is explicitly switched off. The minimum set of commands is as follows.

file.dcf

```
*INPUT
READ TABLE LOADS
*LINSTA
MODEL OFF
BEGIN OUTPUT
...
END OUTPUT
*END
```

Change supports

file.dcf

```
*INPUT
READ TABLE SUPPOR
*LINSTA
BEGIN MODEL
EVALUA OFF
END MODEL
BEGIN OUTPUT
...
END OUTPUT
*END
```

If only some supports change in the finite element model, the evaluation of the material and geometric properties of elements may be skipped.

4.2 Output of Analysis Results

You can obtain output of regular linear static analysis results via an obligatory OUTPUT command block which selects the analysis results to be output. For the general, analysis type independent options for output of analysis results see § 3.6 on page 55.

syntax

```
BEGIN OUTPUT [ devicew ] [ outoptw... ] [ params ]
[ OFF ]
[ SELECT ... ]
[ LAYOUT ... ]
itemw ...
DISPLA
STRAIN
STRESS
PRESSU
FORCE
NODFOR
ELMFOR
PARAME
FRACTU
END OUTPUT
```

SELECT optional commands to customize the batch output: model selection [§ 3.6.2 p. 59], selection of load sets and extreme values [§ 4.2.1 p. 75], stress- and strain transformation [§ 3.6.2.3 p. 60].

LAYOUT optional commands to customize the layout and style of tabular output [§ 3.6.4.1 p. 65].

item is the name of the analysis result to be output. See § 3.6.1 on page 56 for complete syntax of this command.

DISPLA for displacements [§ 4.2.2].

STRAIN for strains [§ 4.2.3].

STRESS for stresses [§ 4.2.4].

PRESSU for hydrostatic pressure capacity [§ 4.2.5].

FORCE for nodal forces [§ 4.2.6].

NODFOR for element nodal forces [§ 4.2.7].

ELMFOR for internal element forces [§ 4.2.8].

PARAME for model parameters [§ 4.2.9 p. 91].

FRACTU for Linear Elastic Fracture Mechanics (LEFM) analysis parameters of crack tip elements [§ 4.2.10].

Default

file.dcf

```
*LINSTA
[ commands ]
OUTPUT
*END
```

If you only give a single **OUTPUT** command, like in the above example, or if you do not give the **OUTPUT** command at all, then DIANA gives a default output as if you had given the following commands.

file.dcf

```
*LINSTA
[ commands ]
BEGIN OUTPUT
  DISPLA TOTAL TRANSL GLOBAL
  FORCE REACTI GLOBAL
  FORCE EXTERN GLOBAL
  STRESS TOTAL CAUCHY GLOBAL NODES
  STRESS TOTAL FORCE LOCAL NODES
  STRESS TOTAL MOMENT LOCAL NODES
  STRESS TOTAL DISFOR LOCAL NODES
  STRESS TOTAL DISMOM LOCAL NODES
  STRAIN TOTAL GREEN GLOBAL NODES
END OUTPUT
*END
```

Example

file.dcf

```
BEGIN OUTPUT
BEGIN SELECT
  BEGIN ELEMEN 4-30
    REAXES R1=1.0 1.0 / R2=1.0 0.3 / ZR=0.28 0.10 / C0=0.02
  END ELEMEN
END SELECT
STRESS TOTAL DISFOR REINFO
STRESS TOTAL DISMOM REINFO
END OUTPUT
```

```

BEGIN OUTPUT
BEGIN SELECT
  BEGIN ELEMEN 31-40
    REAXES R1=1.0 0.5 / R2=1.0 0.2
  END ELEMEN
END SELECT
STRESS TOTAL CAUCHY REAXES
STRESS TOTAL DISFOR REAXES
END OUTPUT

```

Output only

file.dcf

```

*LINSTA
MODEL OFF
BEGIN OUTPUT
  STRESS TOTAL DISFOR REINFO
  [ more output ]
END OUTPUT

```

4.2.1 Loads and Extreme Values

For linear static analysis, you may select load sets for which DIANA must output analysis results. If you do not select load sets, then DIANA will output result values for all load sets. Moreover, you may ask DIANA to output only the extreme result values for the (selected) load sets.

syntax

```

BEGIN SELECT
LOADS [            {    } ]
        losetsn... MIN
        ALL           MAX
END SELECT

```

LOADS specifies a load selection: *losets* are load set numbers referring to input table 'LOADS' [§ 2.3.8 p. 45]. ALL indicates all load sets, which also is the default. [ALL]

MIN selects the minimum value of the (selected) load sets to be output, MAX selects the maximum value.

file.dcf

```

BEGIN OUTPUT TABULA
BEGIN SELECT
  LOADS 2 4 6 MAX
  BEGIN ELEMEN 415 533
    NODES 1
  END ELEMEN
END SELECT
STRESS CAUCHY LOCAL XX NODES
END OUTPUT

```

These commands could yield the following tabular output.

file.tb

```

Analysis type      LINSTA
Extreme results    MAXIMUM
Result            STRESS TOTAL  CAUCHY
Axes              LOCAL
Location of results NODES

Element type L2TRU
Extreme values Sxx 3.198E+02  Loadcasenr 2  Elmnr 415  Nodnr 315

Elmnr Nodnr      Sxx
 415  315      3.198E+02 ( 2)
 533  387      1.763E+02 ( 6)

```

This tabular output shows a maximum value for stress component σ_{xx} at node 315 in element 415 to be 319.8 for load set 2. The maximum value at node 387 in element 533 is 176.3 for load set 6. Notice that DIANA writes the maximum value for all selected output points just below the header paragraph.

Combined items. If you ask for a combination of various output items in tabular output [§ 3.6.4.1 p. 66], then DIANA determines the extreme value(s) for the output item that you specified first. The values for the other items for the same load set will be printed on the same line of the output table.

Mobile loads. If you ask for tabular output for an analysis in which a mobile load is applied, you get a tabular file like:

file.tb

```

Analysis type      LINSTA
Extreme results    MINIMUM
Result            STRESS TOTAL  DISFOR
Axes              LOCAL
Location of results NODES

Element type CT30S
Extreme values Qxz -2.337E+01  Loadcasenr 1  Truckpos 3  Elmnr 11  Nodnr 23

Elmnr Nodnr      Qxz
 11    1      -3.742E+00 (12 )
      11      -1.709E+00 (12 )
      21       3.247E-01 (12 )
      22      -7.695E+00 ( 1-3)
      23      -2.337E+01 ( 1-3)
      12      -9.492E+00 ( 1-3)
 15    23      -2.170E+01 ( 1-2)
      13       6.646E-01 ( 5-2)
      3       1.003E+01 (12 )
      2       1.095E+01 (12 )
      1       1.186E+01 (12 )
      12       1.353E+00 ( 5-2)

```

In this example in the load cases 1 and 5 a mobile load is defined whereas load case 12 is a regular loading. The minimum total distributed shear force **Qxz** has a value **-2.337E+01** for the truck position number 3 in load case number 1 at the location of the node with number 23 in the element with number 11. The numbers between brackets indicate the load case number, and in case of a mobile load, also the truck position after the hyphen.

4.2.2 Displacements

syntax

```

DISPLA [ typew ] [ formw ] [ operw ] { compw } { optiw }
      TOTAL   TRANSL  LOCAL          RELATI
      PHASE   ROTATI  GLOBAL
      PRESCR          NORM

```

DISPLA specifies displacements of the nodes as output item.

[TOTAL] *type* specifies the displacement type.

TOTAL for the total displacements of a structure, i.e., the deformed geometry after a linear static analysis.

PHASE for the phased displacements, i.e., the change in the deformed geometry between two consecutive phases in a phased structural analysis [§ 28 p. 401].

PRESCR for the prescribed displacements, i.e., the deformation load specified in sub-table DEFORM of table 'LOADS' [§ 2.3.5 p. 35]. Prescribed displacements can be output as soon as the elements have been assembled, it is not necessary that the system of equations has been solved.

form specifies the displacement formulation [§ 3.6.1 p. 57]. [TRANSL]

oper specifies an operation (transformation) to be performed on the displacements [§ 3.6.1 p. 57]. One specific operation is available: [GLOBAL]

NORM for the length of the displacement vector. Only the translational terms will be used to calculate the norm. This scalar result can be used for result scans over load cases by some output devices.

comp selects displacement components for output. Default is all available components. [all]

opti are additional options. The RELATI option gives the displacement results relative to an explicitly defined base node indicated by the BASNOD command [§ 3.6.2 p. 59]. See § 3.6.1 on page 58 for other options.

Total displacements					<i>comp</i> ...		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	<i>opti</i>	X	Y	Z
DISPLA	TOTAL	TRANSL	LOCAL		Dtx	Dty	Dtz
					u_x	u_y	u_z
DISPLA	TOTAL	TRANSL	LOCAL	RELATI	Drtx	DrtY	Drtz
					u_{x_r}	u_{y_r}	u_{z_r}
DISPLA	TOTAL	TRANSL	GLOBAL		DtX	DtY	DtZ
					u_X	u_Y	u_Z
DISPLA	TOTAL	TRANSL	GLOBAL	RELATI	DrtX	DrtY	DrtZ
					u_{X_r}	u_{Y_r}	u_{Z_r}
DISPLA	TOTAL	ROTATI	LOCAL		Drx	Dry	Drz
					ϕ_x	ϕ_y	ϕ_z
DISPLA	TOTAL	ROTATI	LOCAL	RELATI	Drrx	DrrY	Drrz
					ϕ_{x_r}	ϕ_{y_r}	ϕ_{z_r}
DISPLA	TOTAL	ROTATI	GLOBAL		DrX	DrY	DrZ
					ϕ_X	ϕ_Y	ϕ_Z
DISPLA	TOTAL	ROTATI	GLOBAL	RELATI	DrrX	DrrY	DrrZ
					ϕ_{X_r}	ϕ_{Y_r}	ϕ_{Z_r}

Total displacements				
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	
DISPLA	TOTAL	TRANSL	NORM	DXYZ
				$\ u\ $

Phased displacements					<i>comp</i> ...		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	<i>opti</i>	X	Y	Z
DISPLA	PHASE	TRANSL	LOCAL		PDtx	PDty	PDtz
					u_x	u_y	u_z
DISPLA	PHASE	TRANSL	LOCAL	RELATI	PrDtx	PrDty	PrDtz
					u_{x_r}	u_{y_r}	u_{z_r}
DISPLA	PHASE	TRANSL	GLOBAL		PDtX	PDtY	PDtZ
					u_X	u_Y	u_Z
DISPLA	PHASE	TRANSL	GLOBAL	RELATI	PrDtX	PrDtY	PrDtZ
					u_{X_r}	u_{Y_r}	u_{Z_r}
DISPLA	PHASE	ROTATI	LOCAL		PDrx	PDry	PDrz
					ϕ_x	ϕ_y	ϕ_z
DISPLA	PHASE	ROTATI	LOCAL	RELATI	PrDrx	PrDry	PrDrz
					ϕ_{x_r}	ϕ_{y_r}	ϕ_{z_r}
DISPLA	PHASE	ROTATI	GLOBAL		PDrX	PDrY	PDrZ
					ϕ_X	ϕ_Y	ϕ_Z
DISPLA	PHASE	ROTATI	GLOBAL	RELATI	PrDrX	PrDrY	PrDrZ
					ϕ_{X_r}	ϕ_{Y_r}	ϕ_{Z_r}

Phased displacements				
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	
DISPLA	PHASE	TRANSL	NORM	PXYZ
				$\ u\ $

Prescribed displacements					<i>comp</i> ...		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	<i>opti</i>	X	Y	Z
DISPLA	PRESKR	TRANSL	LOCAL		pDtx	pDty	pDtz
					u_x	u_y	u_z
DISPLA	PRESKR	TRANSL	LOCAL	RELATI	prDtx	prDty	prDtz
					u_{x_r}	u_{y_r}	u_{z_r}
DISPLA	PRESKR	TRANSL	GLOBAL		pDtX	pDtY	pDtZ
					u_X	u_Y	u_Z
DISPLA	PRESKR	TRANSL	GLOBAL	RELATI	prDtX	prDtY	prDtZ
					u_{X_r}	u_{Y_r}	u_{Z_r}
DISPLA	PRESKR	ROTATI	LOCAL		pDry	pDrz	
					ϕ_x	ϕ_y	ϕ_z
DISPLA	PRESKR	ROTATI	LOCAL	RELATI	prDry	prDrz	
					ϕ_{x_r}	ϕ_{y_r}	ϕ_{z_r}
DISPLA	PRESKR	ROTATI	GLOBAL		pDrX	pDrY	pDrZ
					ϕ_X	ϕ_Y	ϕ_Z
DISPLA	PRESKR	ROTATI	GLOBAL	RELATI	prDrX	prDrY	prDrZ
					ϕ_{X_r}	ϕ_{Y_r}	ϕ_{Z_r}

Prescribed displacements				
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	
DISPLA	PRESKR	TRANSL	NORM	pXYZ
				$\ u\ $

4.2.3 Strains

syntax

STRAIN	[<i>type_w</i>]	[<i>form_w</i>]	[<i>oper_w</i>]	{ <i>comp_w</i> }	{ <i>loca_w</i> }	{ <i>opti_w</i> }
	TOTAL	GREEN	LOCAL		INTPNT	SMOOTH
		FORCE	GLOBAL		NODES	ERROR
		MOMENT	PRINCI		CENTER	...
		DISFOR	VONMIS			
		DISMOM	REAXES			
		TRACTI	VOLUME			

DISSEI JANFOR
 JANMOM

STRAIN specifies strains as output item. Table 4.1 outlines the availability and applicability of the various strain output options for each of the element families.

Table 4.1: AVAILABILITY OF STRAIN OUTPUT FOR REGULAR LINEAR STATIC ANALYSIS

<i>item</i>	STRAIN	truss	beam	pl. stress	pl. strain	axisymm.	plate bend.	fl. shell	cu. shell	solid	str. interface	spring	p. mass	e. reinfo.
<i>type</i>	TOTAL	a	a	a	a	a	a	a	a	a	a	-	a	
<i>form</i>	GREEN	a	a	a	a	a	-	a	a	a	-	-	-	a
	FORCE	-	c	-	-	-	-	-	-	-	-	a	-	e
	MOMENT	-	c	-	-	-	-	-	-	-	-	a	-	f
	DISFOR	-	-	-	-	-	a	a	-	-	-	-	-	-
	DISMOM	-	-	-	-	-	a	a	-	-	-	-	-	-
	TRACTI	-	-	-	-	-	-	-	-	-	a	-	-	e
	DISSEI	-	-	-	-	-	-	-	-	-	d	-	-	-
<i>oper</i>	LOCAL	a	a	a	a	a	a	a	a	a	a	a	-	a
	GLOBAL	a	a	a	a	a	a	a	a	a	a	a	-	e
	PRINCI	a	a	a	a	a	a	a	a	a	-	-	-	-
	VONMIS	a	a	a	a	a	a	a	a	a	-	-	-	-
	REAXES	-	-	a	-	-	a	a	a	a	-	-	-	-
	VOLUME	a	a	a	a	a	a	a	a	a	-	-	-	-
	JANFOR	-	-	-	-	-	-	-	-	-	h	-	-	-
	JANMOM	-	-	-	-	-	-	-	-	-	h	-	-	-
<i>loca</i>	INTPNT	a	i	a	a	a	a	a	a	a	a	-	-	g
	NODES	a	a	a	a	a	a	a	a	a	-	a	-	e
	CENTER	a	a	a	a	a	a	a	a	a	a	-	-	-
<i>opti</i>	SMOOTH	a	a	a	a	a	-	-	a	a	-	-	-	-
	ERROR	a	a	a	a	a	-	-	a	a	-	-	-	-

(a) All elements. (b) Not for elements with orthotropic geometry. (c) For all beam elements, class-II and class-III only in combination with local coordinate system. (d) Only for plane structural interface elements. (e) Only for bond-slip reinforcements. (f) Only for bond-slip reinforcements modeled by beam elements. (g) Not for bond-slip reinforcements. (h) Only for line interface to shell elements. (i) Only for class-II and class-III beam elements. (-) Not available or not suitable.

type specifies the strain type [§ 3.6.1 p. 57].

[TOTAL]

form specifies the strain formulation.

[GREEN]

GREEN for Green-Lagrange strains [§ 4.2.3.1].

FORCE for deformations due to normal and shear forces [§ 4.2.3.2].

DISFOR for generalized strains [§ 4.2.3.2].

MOMENT for curvatures due to a concentrated bending moment [§ 4.2.3.3].

DISMOM for curvatures due to a distributed bending moment [§ 4.2.3.3].

TRACTI for tractions in structural interface elements [§ 4.2.3.4].

DISSEI for distributed seismic moments in plane structural interface elements [§ 4.2.3.5].

oper specifies an operation (transformation) to be performed on the primary strains [§ 3.6.1 p. 57]. Two specific operations are available for line interface to shell elements with the Janssen material model:

[GLOBAL]

JANFOR for averaged relative displacements in local *xyz* directions [§ 4.2.3.4].

JANMOM for relative rotations around the tangential direction. [§ 4.2.3.4].

comp selects strain components for output. Default is all available components.

[all]

loca specifies the location for the strains to be output [§ 3.6.1 p. 58].

opti are additional options [§ 3.6.1 p. 58].

4.2.3.1 Green–Lagrange Strains

Primary strains [§ 47.1]				comp ...						
item	type	form	oper	XX	YY	ZZ	XY	YZ	ZX	1 2 3
STRAIN TOTAL	GREEN	LOCAL		Exx	Eyy	Ezz	Gxy	Gyz	Gzx	
				ε_{xx}	ε_{yy}	ε_{zz}	γ_{xy}	γ_{yz}	γ_{zx}	
STRAIN TOTAL	GREEN	GLOBAL		EXX	EYY	EZZ	GXY	GYZ	GZX	
				ε_{XX}	ε_{YY}	ε_{ZZ}	γ_{XY}	γ_{YZ}	γ_{ZX}	
STRAIN TOTAL	GREEN	PRINCI								E1 E2 E3
										$\varepsilon_1 \varepsilon_2 \varepsilon_3$
STRAIN TOTAL	GREEN	REAXES								E1RA E2RA
										$\varepsilon_1^a \varepsilon_2^a$

Von Mises strain [§ 47.1.1]				
item	type	form	oper	
STRAIN TOTAL	GREEN	VONMIS		Eeq
				ε_{eq}

Volumetric strain [§ 47.1.3]				
item	type	form	oper	
STRAIN TOTAL	GREEN	VOLUME		Evol
				ε_{vol}

4.2.3.2 Deformations

Force deformations				comp ...		
item	type	form	oper	X	Y	Z
STRAIN TOTAL	FORCE	LOCAL		Px		
				Δu_x		
STRAIN TOTAL	FORCE	GLOBAL		PX	PY	PZ
				Δu_X	Δu_Y	Δu_Z

Generalized strains				comp ...						
itm	type	form	oper	XX	YY	ZZ	XY	YZ	ZX	1 2
STRAIN TOTAL	DISFOR	LOCAL		Pxx	Pyy	Pzz	Pxy	Pyz	Pzx	
				Ψ_{xx}	Ψ_{yy}	Ψ_{zz}	Ψ_{xy}	Ψ_{yz}	Ψ_{zx}	
STRAIN TOTAL	DISFOR	REAXES								P1RA P2RA
										$\Psi_1^a \Psi_2^a$

4.2.3.3 Curvatures

Concentrated curvatures				comp ...		
item	type	form	oper	X	Y	Z
STRAIN TOTAL	MOMENT	LOCAL		Kx	Ky	Kz
				κ_x	κ_y	κ_z

Distributed curvatures				comp ...					
item	type	form	oper	XX	YY	ZZ	XY	1 2	
STRAIN TOTAL	DISMOM	LOCAL		Kxx	Kyy	Kzz	Kxy		
				κ_{xx}	κ_{yy}	κ_{zz}	κ_{xy}		
STRAIN TOTAL	DISMOM	REAXES						K1RA K2RA	
								$\kappa_1^a \kappa_2^a$	

4.2.3.4 Relative Displacements of Interface Elements

Note that the labels for the results in the local xyz directions are dependent on the interface element type [Vol. Element Library]. Output items related to the normal direction(s) are indicated with ‘N’ and items related to the shear direction(s) are indicated with ‘S’ in their name followed by the letter indicating the corresponding element direction.

Relative displacements				<i>comp</i> ...		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	X	Y	Z
STRAIN	TOTAL	TRACTI	LOCAL	DUNx	DUNy	DUNz
				DUSx	DUSy	DUSz
				Δu_x	Δu_y	Δu_z
STRAIN	TOTAL	TRACTI	GLOBAL	DUX	DUY	DUZ
				Δu_X	Δu_Y	Δu_Z
STRAIN	TOTAL	TRACTI	JANFOR	DUJx	DUJy	DUJz
				Δu_x	Δu_y	Δu_z
STRAIN	TOTAL	TRACTI	JANMOM	DRJy		
				$\Delta \phi_x$		

4.2.3.5 Distributed Seismic Moment

Distributed seismic moment [§ 47.1.4]						
<i>item</i>	<i>type</i>	<i>form</i>				
STRAIN	TOTAL	DISSEI		PSTOT	PSNEG	PSPOS
				\mathcal{P}_S	\mathcal{P}_{S-}	\mathcal{P}_{S+}

4.2.4 Stresses

syntax

STRESS	$[type_w]$	$[form_w]$	$[oper_w]$	$\{comp_w\}$	$\{loca_w\}$	$\{opti_w\}$
	TOTAL	CAUCHY	LOCAL		INTPNT	NOBOND
	INITIA	FORCE	GLOBAL		NODES	SMOOTH
		MOMENT	PRINCI		CENTER	ERROR
		DISFOR	VONMIS			CENTAX
		DISMOM	INVARI			...
		TRACTI	REINFO			
		GRADIE	REAXES			
		SHEAR	MAXSHR			
		USRRBE	CRKIND			
		USRRSH	BIAXFE			
			MOHRCO			
			HOEKBR			
			FRICTI			
			JANFOR			
			JANMOM			
			TSAIWU			

STRESS specifies stresses as output item. Table 4.2 on the next page outlines the availability and applicability of the various stress output options for each of the element families.

type specifies the stress type [§ 3.6.1 p. 57].

[TOTAL]

form specifies the stress formulation.

[CAUCHY]

CAUCHY for Cauchy stresses [§ 4.2.4.1].

FORCE for concentrated forces and tractions [§ 4.2.4.2].

MOMENT for concentrated bending moments [§ 4.2.4.3].

DISFOR for distributed forces [§ 4.2.4.2].

DISMOM for distributed bending moments [§ 4.2.4.3].

TRACTI for tractions in structural interface elements [§ 4.2.4.4].

GRADIE for gradients of stresses in reinforcement bars [§ 4.2.4.5].

SHEAR for shear stress in the reinforcement mother element connection [§ 4.2.4.6].

Table 4.2: AVAILABILITY OF STRESS OUTPUT FOR REGULAR LINEAR STATIC ANALYSIS

<i>item</i>	STRESS	truss	beam	pl. stress	pl. strain	axisymm.	plate bend.	fl. shell	cu. shell	solid	str. interface	spring	p. mass	e. reinfo.	comp. line	comp. surf.
<i>type</i>	TOTAL	a	a	a	a	a	a	a	a	a	a	-	a	a	a	
	INITIA	a	a	a	a	a	a	a	a	a	a	-	a	a	a	
<i>form</i>	CAUCHY	a	a	a	a	b	b	a	a	-	-	-	a	-	-	
	FORCE	a	a	-	-	-	-	-	-	-	-	-	l	a	-	
	MOMENT	-	a	-	-	-	-	-	-	-	a	-	h	a	-	
	DISFOR	-	-	a	c	d	a	a	a	-	-	-	-	-	a	
	DISMOM	-	-	-	c	d	a	a	a	-	-	-	-	-	a	
	TRACTI	-	-	-	-	-	-	-	-	a	-	-	g	-	-	
	GRADIE	-	-	-	-	-	-	-	-	-	-	-	e	-	-	
	SHEAR	-	-	-	-	-	-	-	-	-	-	-	e	-	-	
	USRRBE	-	a	-	-	-	-	-	-	-	-	-	-	-	-	
	USRRSH	-	-	-	-	a	a	a	a	-	-	-	-	-	-	
<i>oper</i>	LOCAL	a	a	a	a	a	a	a	a	a	a	-	a	a	a	
	GLOBAL	a	a	a	a	a	a	a	a	a	f	-	g	-	-	
	PRINCI	a	a	a	a	a	a	a	a	a	-	-	-	-	-	
	VONMIS	a	a	a	a	a	a	a	a	a	-	-	-	-	-	
	INVARI	-	-	a	a	a	-	-	a	-	-	-	-	-	-	
	REINFO	-	-	a	-	-	a	a	a	a	-	-	-	-	a	
	REAXES	-	-	a	-	-	a	a	a	a	-	-	-	-	-	
	MAXSHR	-	-	a	a	a	-	-	a	a	-	-	-	-	-	
	CRKIND	a	i	a	a	a	-	-	a	a	-	-	-	-	-	
	BIAXFE	a	a	a	a	a	-	-	a	a	-	-	-	-	-	
	MOHRCO	-	-	a	a	-	-	-	a	-	-	-	-	-	-	
	HOEKBR	-	-	a	a	-	-	-	a	-	-	-	-	-	-	
	FRICTI	-	-	-	-	-	-	-	-	a	-	-	-	-	-	
	JANFOR	-	-	-	-	-	-	-	-	j	-	-	-	-	-	
	JANMOM	-	-	-	-	-	-	-	-	j	-	-	-	-	-	
	TSAIWU	-	-	a	-	-	a	a	a	-	-	-	-	-	-	
<i>loca</i>	INTPNT	a	k	a	a	a	a	a	a	a	-	-	i	-	-	
	NODES	a	a	a	a	a	a	a	a	a	f	-	g	a	a	
	CENTER	a	a	a	a	a	a	a	a	a	-	-	-	-	-	
<i>opti</i>	NOBOND	-	a	-	-	-	-	-	a	-	-	-	-	-	-	
	SMOOTH	a	a	a	a	a	-	-	a	a	-	-	-	-	-	
	ERROR	a	a	a	a	a	-	-	a	a	-	-	-	-	-	
	CENTAX	-	-	-	-	-	-	-	-	-	-	-	-	a	a	

(a) All elements. (b) Not for elements with orthotropic geometry. (c) For infinite shells only. (d) For shells of revolution only. (e) Only for bar reinforcements. (f) Only for base springs. (g) Only for bond-slip reinforcements. (h) Only for bond-slip reinforcements modeled by beam elements. (i) Not for bond-slip reinforcements. (j) Only for line interface to shell elements. (k) Only for Class-II and Class-III beam elements. (l) Only for bar reinforcements and bond-slip reinforcements. (-) Not available or not suitable.

USRRBE for user-defined derived results for beam elements [§ 4.2.4.7]. To determine the user-defined derived results DIANA needs additional material properties and the user-supplied subroutine USRBEA, see Volume *Material Library*.

USRRSH for user-defined derived results for plate and shell elements [§ 4.2.4.8]. To determine the user-defined derived results DIANA needs additional material properties and the user-supplied subroutine USRSHL, see Volume *Material Library*.

[GLOBAL] *oper* specifies an operation (transformation) to be performed on the primary stresses [§ 3.6.1 p. 57]. A specific operation is available for the maximum shear stress:

MAXSHR gives the maximum shear stress $\tau_{\max} = \frac{\sigma_1 - \sigma_3}{2}$ with σ_1 and σ_3 the highest and lowest principal stress respectively. For plane stress elements the maximum shear stress τ_{\max} is defined as $\tau_{\max} = \frac{\sigma_1 - \sigma_2}{2}$ because the third principal stress σ_3 is zero by definition.

A specific operation is available for crack indices:

CRKIND gives the crack indices I_{cr} (the crack indicator) and F_{tu} (the tensile strength utilization) which correspond to indications whether a crack arises or not:

$$I_{cr} \geq \gamma_{cr} \quad \text{with} \quad I_{cr} = \frac{f_t}{\sigma_I} \quad (4.1)$$

$$F_{tu} \leq \beta_{cr} \quad \text{with} \quad F_{tu} = \frac{\sigma_I}{f_t} \quad (4.2)$$

where f_t is the tensile strength of concrete, σ_I is the maximal principal stress and γ_{cr} and β_{cr} are safety factors. In order to avoid an infinite value of the crack indicator, DIANA will automatically set I_{cr} to 100 if $\sigma_I \leq 0.01f_t$. And in order to avoid negative values of the tensile strength utilization, DIANA will automatically set F_{tu} to 0 if $\sigma_I \leq 0$.

To determine the crack indices DIANA needs the values of the tensile strength f_t . You must specify these as a material property [Vol. *Material Library*].

A specific operation is available for Safety Factors for concrete under static and dynamic loading conditions with reference to a biaxial failure envelope:

BIAXFE gives the following Safety Factors for concrete under static and dynamic loading conditions with reference to a biaxial failure envelope:

- $FS_{static\ usual} = R_{static\ usual}/r$
- $FS_{static\ unusual} = R_{static\ unusual}/r$
- $FS_{dynamic\ unusual} = R_{dynamic\ unusual}/r$
- $FS_{dynamic\ extreme} = R_{dynamic\ extreme}/r$

Where r is the distance from the origin to the actual stress point (σ_1, σ_2) :

$$r = \sqrt{(\sigma_1^2 + \sigma_2^2)} \quad (4.3)$$

To determine the Safety Factors DIANA needs additional material properties. For the required parameters and additional information on the Safety Factors, see material parameters FCRATE, FCSPEC, FSC, and FST in Volume *Material Library*.

Note that all Safety Factors are limited to 100.

A specific operation is available for shear capacity of stress against Mohr–Coulomb failure criterion:

MOHRCO gives the shear capacity of stress against Mohr–Coulomb failure criterion [§ 47.2.9.1 p. 582]. To determine the shear capacity DIANA needs the initial cohesion c (COHESI) and initial friction angle ϕ (PHI) of the Mohr–Coulomb material model [Vol. *Material Library*].

A specific operation is available for shear capacity of stress against Hoek–Brown failure criterion:

HOEKBR gives the shear capacity of stress against Hoek–Brown failure criterion [§ 47.2.9.2 p. 582]. To determine the shear capacity DIANA needs the unconfined compressive strength σ_{ci} of the (intact) rock sample (COMSTR), Hoek–Brown constant m_b (HOEKMB), and the Hoek–Brown constant s (HOEKS) material properties of the Hoek–Brown rock plasticity material model [Vol. *Material Library*].

A specific operation is available for shear capacity of stress against Coulomb friction failure criterion for structural interface elements:

FRICTI gives the shear capacity of stress against Coulomb friction criterion for structural interface elements [§ 47.2.9.3 p. 583]. To determine the shear capacity DIANA needs the initial cohesion c (COHESI) and initial friction angle ϕ (PHI) of the Coulomb friction interface material model [Vol. *Material Library*].

Two specific operations are available for line interface to shell elements with the Janssen material model:

JANFOR for distributed forces in local *xyz* directions [§ 4.2.4.4].

JANMOM for the distributed moment around the tangential direction. [§ 4.2.4.4].

A specific operation is available for the failure indices of composites, i.e. plane stress, plate bending, and shell elements with orthotropic material specified:

TSAIWU for the failure indices of composites [§ 4.2.4.1].

The following failure criteria can be calculated:

Maximum criterion which compares the allowable load with actual strength for each component.

Tsai-Hill criterion which describes an early attempt to account for the widely different strengths in the various principal directions [2].

Tsai-Wu criterion which is a phenomenological material failure theory which is widely used for anisotropic composite materials which have different strengths in tension and compression [86].

To determine the failure indices DIANA needs additional material properties. For the required parameters and additional information on the failure criteria, see material parameters **COMPFA** and **TSWIXY** in Volume *Material Library*.

[all] **comp** selects stress components for output. Default is all available components.

loca specifies the location for the stresses to be output [§ 3.6.1 p. 58].

opti are additional options

NOBOND includes the contribution of prestress in posttensioned reinforcements to element forces and moments [Vol. *Element Library*].

SMOOTH to ‘smooth’ the element data values like strains and stresses, after extrapolation to the nodes. Smoothing yields the average value of the element contributions at the node. This option can only be applied for results in global orientation. Smoothing is particularly useful for graphic output, it avoids overlapped plotting of strain and stress rosettes.

ERROR gives the maximum value of the ‘error’ (deviation) of the smoothed value and the extrapolated value.

CENTAX will automatically shift the specified composed line or composed surface to the centroidal axis of the structure and output the moment and/or distributed moment with respect to the shifted composed line and/or surface.

... see § 3.6.1 on page 58 for other options.

4.2.4.1 Cauchy Stresses

Primary stresses [§ 47.2]				comp ...								
item	type	form	oper	XX	YY	ZZ	XY	YZ	ZX	1	2	3
STRESS	TOTAL	CAUCHY	LOCAL	Sxx	Syy	Szz	Sxy	Syz	Szx			
				σ_{xx}	σ_{yy}	σ_{zz}	σ_{xy}	σ_{yz}	σ_{zx}			
STRESS	TOTAL	CAUCHY	GLOBAL	SXX	SYY	SZZ	SXY	SYZ	SZX			
				σ_{XX}	σ_{YY}	σ_{ZZ}	σ_{XY}	σ_{YZ}	σ_{ZX}			
STRESS	TOTAL	CAUCHY	PRINCI							S1	S2	S3
										σ_1	σ_2	σ_3
STRESS	TOTAL	CAUCHY	REAXES							S1RA	S2RA	
										σ_1^a	σ_2^a	
STRESS	INITIA	CAUCHY	LOCAL	S0xx	S0yy	S0zz	S0xy	S0yz	S0zx			
				σ_{xx}^0	σ_{yy}^0	σ_{zz}^0	σ_{xy}^0	σ_{yz}^0	σ_{zx}^0			
STRESS	INITIA	CAUCHY	GLOBAL	S0XX	S0YY	S0ZZ	S0XY	S0YZ	S0ZX			
				σ_{XX}^0	σ_{YY}^0	σ_{ZZ}^0	σ_{XY}^0	σ_{YZ}^0	σ_{ZX}^0			
STRESS	INITIA	CAUCHY	PRINCI							S01	S02	S03
										σ_1^0	σ_2^0	σ_3^0
STRESS	INITIA	CAUCHY	REAXES							S01RA	S02RA	
										σ_1^{0a}	σ_2^{0a}	

Von Mises stress [§ 47.2.1]				
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	
STRESS	TOTAL	CAUCHY	VONMIS	Seq
				σ_{eq}
STRESS	INITIA	CAUCHY	VONMIS	S0eq
				σ_{eq}^0

Stress invariants [§ 47.2.5]				<i>comp</i> ...		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	P	Q	LODE
STRESS	TOTAL	CAUCHY	INVARI	P	Q	Lode
				p'	q	θ

Maximum shear stress				
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	
STRESS	TOTAL	CAUCHY	MAXSHR	Tmax
				τ_{\max}
STRESS	INITIA	CAUCHY	MAXSHR	T0max
				τ_{\max}^0

Biaxial failure envelope				<i>comp</i> ...			
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	STUS	STUN	DYUN	DYEX
STRESS	TOTAL	CAUCHY	BIAXFE	FSstus	FSstun	FSdyun	FSdyex
				FS_{stus}	FS_{stun}	FS_{dyun}	FS_{dyex}

Shear capacity [§ 47.2.9]				<i>comp</i> ...	
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	SHRCAP	
STRESS	TOTAL	CAUCHY	MOHRCO	SHRCAP	
				ψ	
STRESS	TOTAL	CAUCHY	HOEKBR	SHRCAP	
				ψ	
STRESS	TOTAL	TRACTI	FRICTI	SHRCAP	
				ψ	

Composite failure criteria				<i>comp</i> ...		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	MAX	HILL	TSAIWU
STRESS	TOTAL	CAUCHY	TSAIWU	MAX	HILL	TSAIWU
				FI_{max}	FI_{TH}	FI_{TW}

4.2.4.2 Forces

Concentrated forces				<i>comp</i> ...		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	X	Y	Z
STRESS	TOTAL	FORCE	LOCAL	N _x	Q _y	Q _z
				N_x	Q_y	Q_z
STRESS	TOTAL	FORCE	GLOBAL	N _X	N _Y	N _Z
				N_X	N_Y	N_Z
STRESS	INITIA	FORCE	LOCAL	N0 _x	Q0 _y	Q0 _z
				N_x^0	Q_y^0	Q_z^0
STRESS	INITIA	FORCE	GLOBAL	NO _X	NO _Y	NO _Z
				N_X^0	N_Y^0	N_Z^0

Distributed forces				comp ...							
item	type	form	oper	XX	YY	ZZ	XY	YZ	ZX	1	2
STRESS	TOTAL	DISFOR	LOCAL	Nxx	Nyy	Nzz	Nxy	Qyz	Qxz		
				n_{xx}	n_{yy}	n_{zz}	n_{xy}	q_{yz}	q_{xz}		
STRESS	TOTAL	DISFOR	REAXES							N1RA	N2RA
										n_1^a	n_2^a
STRESS	INITIA	DISFOR	LOCAL	N0xx	N0yy	N0zz	N0xy	Q0yz	Q0xz		
				n_{xx}^0	n_{yy}^0	n_{zz}^0	n_{xy}^0	q_{yz}^0	q_{xz}^0		

Reinforcement forces [§ 47.2.6]				<i>comp</i> ...				
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	1R	2R	1RC	2RC	QT
STRESS	TOTAL	DISFOR	REINFO	N1R	N2R	N1RC	N2RC	QT
				n'_1	n'_2	n_{1c}'	n_{2c}'	q'

4.2.4.3 Bending moments

Concentrated moments				<i>comp</i> ...		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	X	Y	Z
STRESS	TOTAL	MOMENT	LOCAL	M _x	M _y	M _z
				M_x	M_y	M_z
STRESS	TOTAL	MOMENT	GLOBAL	M _X	M _Y	M _Z
				M_X	M_Y	M_Z
STRESS	INITIA	MOMENT	LOCAL	M0 _x	M0 _y	M0 _z
				M_x^0	M_y^0	M_z^0
STRESS	INITIA	MOMENT	GLOBAL	M0 _X	M0 _Y	M0 _Z
				M_X^0	M_Y^0	M_Z^0

Distributed moments				<i>comp</i> ...					
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	XX	YY	ZZ	XY	1	2
STRESS	TOTAL	DISMOM	LOCAL	Mxx	Myy	Mzz	Mxy		
				m_{xx}	m_{yy}	m_{zz}	m_{xy}		
STRESS	TOTAL	DISMOM	REAXES					M1RA	M2RA
								m_1^a	m_2^a
STRESS	INITIA	DISMOM	LOCAL	M0xx	M0yy	M0zz	M0xy		
				m_{xx}^0	m_{yy}^0	m_{zz}^0	m_{xy}^0		

Reinforcement moments [§ 47.2.6]				<i>comp</i> ...	
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	1R	2R
STRESS	TOTAL	DISMOM	REINFO	M1R	M2R
				m'_1	m'_2

4.2.4.4 Tractions

Note that the labels for the results in the local xyz directions are dependent on the interface element type [Vol. Element Library]. Output items related to the normal direction(s) are indicated with ‘N’ and items related to the shear direction(s) are indicated with ‘S’ in their name followed by the letter indicating the corresponding element direction.

Traction				comp ...		
item	type	form	oper	X	Y	Z
STRESS TOTAL	TRACTI	LOCAL		STNx	STNy	STNz
				STSx	STSy	STSz
				t_x	t_y	t_z
STRESS TOTAL	TRACTI	GLOBAL		STX	STY	STZ
				t_X	t_Y	t_Z
STRESS TOTAL	TRACTI	JANFOR		NJxx	QJxy	QJxz
				t_x	t_y	t_z
STRESS TOTAL	TRACTI	JANMOM		MJxx		
				m_z		
STRESS INITIA	TRACTI	LOCAL		STONx	STONy	STONz
				STOSx	STOSy	STOSz
				t_x^0	t_y^0	t_z^0
STRESS INITIA	TRACTI	GLOBAL		STOX	STOY	STOZ
				t_X^0	t_Y^0	t_Z^0

4.2.4.5 Stress Gradients in Reinforcement Bars

Stress gradients [§ 47.2.7]			
item	type	form	oper
STRESS TOTAL	GRADIE	LOCAL	SG
			σ'_{xx}

4.2.4.6 Shear Stress in Reinforcement Mother Element Connection

Shear stress [§ 47.2.8]			
item	type	form	oper
STRESS TOTAL	SHEAR	LOCAL	SSHR
			τ

4.2.4.7 User-defined Derived Results for Beam Elements

User-defined derived results			
item	type	form	oper
STRESS TOTAL	USRRBE	LOCAL	User-defined components and labels
STRESS INITIA	USRRBE	LOCAL	User-defined components and labels

4.2.4.8 User-defined Derived Results for Plates and Shells

User-defined derived results			
item	type	form	oper
STRESS TOTAL	USRRSH	LOCAL	User-defined components and labels
STRESS INITIA	USRRSH	LOCAL	User-defined components and labels

4.2.5 Hydrostatic Pressure Capacity

You may select the hydrostatic pressure capacity of stress against the Mohr–Coulomb failure criterion or the hydrostatic pressure capacity of stress against the Coulomb friction failure criterion for output according to the following syntax:

syntax

```

PRESSU [ typew ] { locaw } { optiw }
      MOHRCO   INTPNT
      HOEKBR   NODES
      FRICTI   CENTER

```

PRESSU specifies the calculated hydrostatic pressure capacity result as output item.

type specifies the type [§ 3.6.1 p. 57]:

MOHRCO gives the hydrostatic pressure capacity of stress against the Mohr–Coulomb failure criterion [§ 47.2.9.1 p. 582]. To determine the hydrostatic pressure capacity DIANA needs the the initial cohesion c (COHESI) and initial friction angle ϕ (PHI) of the Coulomb friction interface material model [Vol. *Material Library*].

HOEKBR gives the hydrostatic pressure capacity of stress against the Hoek–Brown failure criterion [§ 47.2.9.2 p. 582]. To determine the hydrostatic pressure capacity DIANA needs the unconfined compressive strength σ_{ci} of the (intact) rock sample (COMSTR), Hoek–Brown constant m_b (HOEKMB), and the Hoek–Brown constant s (HOEKS) material properties of the Hoek–Brown rock plasticity material model [Vol. *Material Library*].

FRICTI gives the hydrostatic pressure capacity of stress against the Coulomb friction failure criterion [§ 47.2.9.3 p. 583]. To determine the hydrostatic pressure capacity DIANA needs the the initial cohesion c (COHESI) and initial friction angle ϕ (PHI) of the Coulomb friction interface material model [Vol. *Material Library*].

[NODES] *loca* specifies the location for the hydrostatic pressure capacity to be output [§ 3.6.1 p. 58].

opti are additional options [§ 3.6.1 p. 58].

Hydrostatic pressure capacity [§ 47.2.9]	
<i>item</i>	<i>type</i>
PRESSU MOHRCO	PPCAP
	χ
PRESSU FRICTI	PPCAP
	χ

4.2.6 Nodal Forces

syntax

```

FORCE [ typew ] [ formw ] [ operw ] { compw } { optiw }
      REACTI   TRANSL   LOCAL
      RESIDU   ROTATI   GLOBAL
      EXTERN
      INTERN

```

FORCE specifies forces and moments in the nodes as output item [§ 47.3 p. 583].

[REACTI] *type* specifies the type of the nodal forces.

REACTI for the reaction forces in all supported nodes.

RESIDU for the residual forces.

EXTERN for the external load vectors.

INTERN for the right-hand-side vectors (internal load).

Load vectors can be output as soon as these have been setup [§ 4.1 p. 72], it is not necessary that the system of equations has been solved.

form specifies the formulation [§ 3.6.1 p. 57] [TRANSL]

oper specifies an operation (transformation) to be performed on the forces and moments [§ 3.6.1 p. 57]. [GLOBAL]

comp selects force or moment components for output. Default is all available components. [all]

opti are additional options [§ 3.6.1 p. 58].

Support reactions [§ 47.3.3]				<i>comp</i> ...		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	X	Y	Z
FORCE REACTI	TRANSL	LOCAL		FBx F_x^b	FBY F_y^b	FBZ F_z^b
FORCE REACTI	TRANSL	GLOBAL		FBX F_X^b	FBY F_Y^b	FBZ F_Z^b
FORCE REACTI	ROTATI	LOCAL		MBx M_x^b	MBY M_y^b	MBZ M_z^b
FORCE REACTI	ROTATI	GLOBAL		MBX M_X^b	MBY M_Y^b	MBZ M_Z^b

Residuals [§ 47.3.3]				<i>comp</i> ...		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	X	Y	Z
FORCE RESIDU	TRANSL	LOCAL		FRx F_x^r	FRY F_y^r	FRZ F_z^r
FORCE RESIDU	TRANSL	GLOBAL		FRX F_X^r	FRY F_Y^r	FRZ F_Z^r
FORCE RESIDU	ROTATI	LOCAL		MRx M_x^r	MRy M_y^r	MRz M_z^r
FORCE RESIDU	ROTATI	GLOBAL		MRX M_X^r	MRY M_Y^r	MRZ M_Z^r

Load vectors [§ 47.3.2]				<i>comp</i> ...		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	X	Y	Z
FORCE EXTERN	TRANSL	LOCAL		FEx F_x^{ext}	FEY F_y^{ext}	FEZ F_z^{ext}
FORCE EXTERN	TRANSL	GLOBAL		FEX F_X^{ext}	FEY F_Y^{ext}	FEZ F_Z^{ext}
FORCE EXTERN	ROTATI	LOCAL		MExt M_x^{ext}	MEY M_y^{ext}	MEZ M_z^{ext}
FORCE EXTERN	ROTATI	GLOBAL		MEX M_X^{ext}	MEY M_Y^{ext}	MEZ M_Z^{ext}
FORCE INTERN	TRANSL	LOCAL		FIx F_x^{int}	FIY F_y^{int}	FIZ F_z^{int}
FORCE INTERN	TRANSL	GLOBAL		FIX F_X^{int}	FIY F_Y^{int}	FIZ F_Z^{int}
FORCE INTERN	ROTATI	LOCAL		MIxt M_x^{int}	MIY M_y^{int}	MIz M_z^{int}
FORCE INTERN	ROTATI	GLOBAL		MIX M_X^{int}	MIY M_Y^{int}	MIZ M_Z^{int}

4.2.7 Internal Nodal Element Forces

syntax

NODFOR $[type_w]$ $[form_w]$ $[oper_w]$ $\{comp_w\}$ $\{opti_w\}$
 ELEMEN TRANSL LOCAL
 REINFO ROTATI GLOBAL
 TOTAL

NODFOR specifies internal nodal element forces and moments in the nodes as output item. This command gives the contributions of the element or reinforcement internal nodal forces (or both), to a certain node. A selection of elements which form a ‘section’ of the model gives the total internal forces that act on that ‘section’.

[TOTAL] *type* specifies the forces type.

ELEMEN for contribution of elements only.

REINFO for contribution of embedded reinforcements only.

TOTAL for contribution of both elements and embedded reinforcements.

[TRANSL] *form* specifies the formulation [§ 3.6.1 p. 57].

[GLOBAL] *oper* specifies an operation (transformation) to be performed on the forces and moments [§ 3.6.1 p. 57].

[all] *comp* selects force or moment components for output. Default is all available components.

opti are additional options [§ 3.6.1 p. 58].

Internal nodal total forces [§ 47.3.1]				<i>comp</i> ...		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	X	Y	Z
NODFOR	TOTAL	TRANSL	LOCAL	FNT _x F_x^{nt}	FNT _y F_y^{nt}	FNT _z F_z^{nt}
NODFOR	TOTAL	TRANSL	GLOBAL	FNT _X F_X^{nt}	FNT _Y F_Y^{nt}	FNT _Z F_Z^{nt}
NODFOR	TOTAL	ROTATI	LOCAL	MNT _x M_x^{nt}	MNT _y M_y^{nt}	MNT _z M_z^{nt}
NODFOR	TOTAL	ROTATI	GLOBAL	MNT _X M_X^{nt}	MNT _Y M_Y^{nt}	MNT _Z M_Z^{nt}

Internal nodal element forces [§ 47.3.1]				<i>comp</i> ...		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	X	Y	Z
NODFOR	ELEMEN	TRANSL	LOCAL	FNE _x F_x^{ne}	FNE _y F_y^{ne}	FNE _z F_z^{ne}
NODFOR	ELEMEN	TRANSL	GLOBAL	FNE _X F_X^{ne}	FNE _Y F_Y^{ne}	FNE _Z F_Z^{ne}
NODFOR	ELEMEN	ROTATI	LOCAL	MNE _x M_x^{ne}	MNE _y M_y^{ne}	MNE _z M_z^{ne}
NODFOR	ELEMEN	ROTATI	GLOBAL	MNE _X M_X^{ne}	MNE _Y M_Y^{ne}	MNE _Z M_Z^{ne}

Internal nodal reinforcement forces [§ 47.3.1]				<i>comp</i> ...		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	X	Y	Z
NODFOR	REINFO	TRANSL	LOCAL	FNR _x F_x^{nr}	FNR _y F_y^{nr}	FNR _z F_z^{nr}
NODFOR	REINFO	TRANSL	GLOBAL	FNR _X F_X^{nr}	FNR _Y F_Y^{nr}	FNR _Z F_Z^{nr}
NODFOR	REINFO	ROTATI	LOCAL	MNR _x M_x^{nr}	MNR _y M_y^{nr}	MNR _z M_z^{nr}
NODFOR	REINFO	ROTATI	GLOBAL	MNR _X M_X^{nr}	MNR _Y M_Y^{nr}	MNR _Z M_Z^{nr}

4.2.8 Internal Element Forces

syntax

```
ELMFOR [ typew ] [ formw ] [ operw ] { compw } { optiw }
      ELEMEN  TRANSL  GLOBAL
      REINFO  ROTATI
      TOTAL
```

ELMFOR specifies internal element forces and moments in the nodes of an element as output item. This command gives the contributions of the element or reinforcement internal forces (or both), to a certain node of an element.

type specifies the forces type. [TOTAL]

ELEMEN for contribution of elements only.

REINFO for contribution of embedded reinforcements only.

TOTAL for contribution of both elements and embedded reinforcements.

form specifies the formulation [§ 3.6.1 p. 57]. [TRANSL]

oper specifies an operation (transformation) to be performed on the forces and moments [§ 3.6.1 p. 57]. [GLOBAL]

comp selects force or moment components for output. Default is all available components. [all]

opti are additional options [§ 3.6.1 p. 58].

Internal element forces				<i>comp</i> ...		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	X	Y	Z
ELMFOR TOTAL	TRANSL	GLOBAL		FETX F_X^{et}	FETY F_Y^{et}	FETZ F_Z^{et}
ELMFOR TOTAL	ROTATI	GLOBAL		METX M_X^{et}	METY M_Y^{et}	METZ M_Z^{et}
ELMFOR ELEMEN	TRANSL	GLOBAL		FEEX F_X^{ee}	FEEY F_Y^{ee}	FEEZ F_Z^{ee}
ELMFOR ELEMEN	ROTATI	GLOBAL		MEEX M_X^{ee}	MEEY M_Y^{ee}	MEEZ M_Z^{ee}
ELMFOR REINFO	TRANSL	GLOBAL		FERX F_X^{er}	FERY F_Y^{er}	FERZ F_Z^{er}
ELMFOR REINFO	ROTATI	GLOBAL		MERX M_X^{er}	MERY M_Y^{er}	MERZ M_Z^{er}

4.2.9 Model Parameters

For some models DIANA will calculate, or assume, (default) values for appropriate material or geometry parameters. Typical examples are spatial functions attached to geometry or material parameters, e.g. depth dependent Young's moduli, Poisson's ratios, mass densities [Vol. *Material Library*], or element thickness [Vol. *Element Library*]. Sometimes, these parameters may be useful in postprocessing. Via the OUTPUT block you can ask DIANA to output certain parameters, being the values that you have explicitly specified, the assumed, or else the calculated (default) values.

syntax

```
PARAM [ typew ] [ locaw ] [ operw ] { compw } { optiw }
      YOUNG  INTPNT  LOCAL
      POISON  NODES  GLOBAL
      DENSIT
```

YLDSTR
COHESI
PHI
PSI
LAMBDA
TENSTR
GF1
COMSTR
DSNX
DSSX
DSNY
DSSY
DSNZ
DSSZ
MAXSHR
GAMMAR
TAUFAC
PMLX
PMLY
PMLZ
THICK
ECCENT

PARAME for output of a model parameter. Table 4.3 outlines the availability and applicability of the various model parameters output for each of the element families.

Table 4.3: AVAILABILITY OF MODEL PARAMETER OUTPUT

<i>item</i>	PARAME	truss	beam	pl. stress	pl. strain	axisymm.	plate bend.	fl. shell	cu. shell	solid	str. interface	contact	spring	p. mass	e. reinfo.	comp. line	comp. surf.
<i>type</i>	YOUNG	a	a	a	a	a	a	a	a	a	-	-	-	-	a	-	-
	POISON	a	a	a	a	a	a	a	a	a	-	-	-	-	b	-	-
	DENSIT	a	a	a	a	a	a	a	a	a	-	-	-	-	b	-	-
	YLDSTR	a	a	a	a	a	-	-	a	a	-	-	-	-	a	-	-
	COHESI	-	-	-	a	a	-	-	-	a	a	-	-	-	c	-	-
	PHI	-	-	-	a	a	-	-	-	a	a	-	-	-	c	-	-
	PSI	-	-	-	a	a	-	-	-	a	a	-	-	-	-	-	-
	LAMBDA	-	-	-	a	a	-	-	-	a	-	-	-	-	-	-	-
	TENSTR	a	d	a	a	a	-	-	a	a	-	-	-	-	-	-	-
	GF1	a	d	a	a	a	-	-	a	a	-	-	-	-	-	-	-
	COMSTR	a	d	a	a	a	-	-	a	a	-	-	-	-	-	-	-
	DSNX	-	-	-	-	-	-	-	-	-	a	-	-	-	-	-	-
	DSSX	-	-	-	-	-	-	-	-	-	a	-	-	-	b	-	-
	DSNY	-	-	-	-	-	-	-	-	-	a	-	-	-	b	-	-
	DSSY	-	-	-	-	-	-	-	-	-	a	-	-	-	-	-	-
	DSNZ	-	-	-	-	-	-	-	-	-	a	-	-	-	b	-	-
	DSSZ	-	-	-	-	-	-	-	-	-	a	-	-	-	-	-	-
	MAXSHR	-	-	-	-	-	-	-	-	-	-	-	-	-	c	-	-
	GAMMAR	-	-	-	a	a	-	-	-	a	-	-	-	-	-	-	-
	TAUFAC	-	-	-	a	a	-	-	-	a	-	-	-	-	-	-	-
	PMLX	-	-	-	a	-	-	-	-	a	-	-	-	-	-	-	-
	PMLY	-	-	-	a	-	-	-	-	a	-	-	-	-	-	-	-
	PMLZ	-	-	-	a	-	-	-	-	a	-	-	-	-	-	-	-
	THICK	-	-	a	e	f	a	a	a	-	g	-	-	-	-	-	-
	ECCENT	-	a	-	-	-	-	-	a	-	-	-	-	-	-	-	-

(a) All elements. (b) Only for bond-slip reinforcements and pile foundations. (c) Only for pile foundations. (d) Not for Class-I beams. (e) Only for infinite shells. (f) Only for axisymmetric shells. (g) Only for line interfaces to shells and two-dimensional line interfaces between membranes. (-) Not available or not suitable.

type specifies which parameter must be output.

YOUNG gives the Young's modulus E .

POISSON gives the Poisson's ratio ν .

DENSIT gives the mass density ρ . For all regular structural elements ρ is the mass density per *unit volume*. For regular structural interface elements ρ is the mass density *per unit area*. For fluid–structure interface elements ρ is the mass density of the fluid ρ_f .

YLDSTR gives the yield stress σ_y for Von Mises and Tresca plasticity.

COHESI gives the cohesion c for Mohr–Coulomb and Drucker–Prager plasticity, and for Coulomb friction and nonlinear elastic friction interface materials.

PHI gives the friction angle ϕ in the used units (radians or degrees) for Mohr–Coulomb, Drucker–Prager, and Egg Cam-clay plasticity, and for Coulomb friction and nonlinear elastic friction interface materials.

PSI gives the dilatancy angle ψ in the used units (radians or degrees) for Mohr–Coulomb and Drucker–Prager plasticity, and for Coulomb friction interface materials.

LAMBDA gives the hardening parameter λ for Egg Cam-clay plasticity.

TENSTR gives the tensile strength f_t for the Total Strain crack model and the Maekawa–Fukuura concrete model.

GF1 gives the Mode-I fracture energy G_f^I for the Total Strain crack model and the Maekawa–Fukuura concrete model.

COMSTR gives the compressive strength f_c for the Total Strain crack model and the Maekawa–Fukuura concrete model.

DSNX gives the linear stiffness modulus D_{11} , which sets the relation between the normal traction t_{nx} in the element x direction and the normal relative displacement Δu_{nx} in the element x direction, for interface materials.

DSSX gives the linear stiffness modulus D_{11} , which sets the relation between the shear traction t_{sx} in the element x direction and the shear relative displacement Δu_{sx} in the element x direction, for interface models.

DSNY gives the linear stiffness modulus D_{22} , which sets the relation between the normal traction t_{ny} in the element y direction and the normal relative displacement Δu_{ny} in the element y direction, for interface materials.

DSSY gives the linear stiffness modulus D_{22} , which sets the relation between the shear traction t_{sy} in the element y direction and the shear relative displacement Δu_{sy} in the element y direction, for interface models.

DSNZ gives the linear stiffness modulus D_{33} , which sets the relation between the normal traction t_{nz} in the element z direction and the normal relative displacement Δu_{nz} in the element z direction, for interface materials.

DSSZ gives the linear stiffness modulus D_{33} , which sets the relation between the shear traction t_{sz} in the element z direction and the shear relative displacement Δu_{sz} in the element z direction, for interface models.

MAXSHR gives the absolute maximum value of the force per length that the pile shaft–soil interface can transfer f_{\max} for pile foundations.

GAMMAR *gammar* is the characteristic shear strain γ_r for the Hardin–Drnevich model or for the Ramberg–Osgood model.

TAUFAC *tf* is an optional factor to the shear–stress values in the multilinear diagram of shear strains γ and the corresponding shear stresses τ (GAMTAU) or to the ratio of secant and initial shear stiffness values in the multilinear diagram of shear strains γ and the corresponding ratios of secant shear stiffness and initial stiffness G/G_0 (GAMBET).

PMLX gives the perfectly matched layer (PML) parameter in X direction.

PMLY gives the perfectly matched layer (PML) parameter in Y direction.

PMLZ gives the perfectly matched layer (PML) parameter in Z direction.

THICK gives the thickness t in the respective element nodes.

ECCENT gives the eccentricity e in the respective element nodes.

loca specifies the location of the parameter to be output. For material parameters the location of the output can be either nodes or integration points. By default, material parameters are output in the element integration points. For thickness and eccentricity, the output is always in the nodes of the elements.

oper is an operation on the result item to be performed prior to its output:

LOCAL for transformation to local xyz directions,

GLOBAL for transformation to global XYZ directions,

[all] *comp* selects model parameter components for output. Default is all available components.

opti are additional options [[§ 3.6.1 p. 58](#)].

Model parameters			<i>comp</i> ...		
<i>item</i>	<i>type</i>	<i>oper</i>	X	Y	Z
PARAME YOUNG			YOUNG		
			E		
PARAME POISON			POISON		
			ν		
PARAME DENSIT			DENSIT		
			ρ		
PARAME YLDSTR			YLDSTR		
			σ_y		
PARAME COHESI			COHESI		
			c		
PARAME PHI			PHI		
			ϕ		
PARAME PSI			PSI		
			ψ		
PARAME LAMBDA			LAMBDA		
			λ		
PARAME TENSTR			TENSTR		
			f_t		
PARAME GF1			GF1		
			G_f^I		
PARAME COMSTR			COMSTR		
			f_c		
PARAME DSNX			DSNX		
			D_{11}		
PARAME DSSX			DSSX		
			D_{11}		
PARAME DSNY			DSNY		
			D_{22}		
PARAME DSSY			DSSY		
			D_{22}		
PARAME DSNZ			DSNZ		
			D_{33}		
PARAME DSSZ			DSSZ		
			D_{33}		
PARAME MAXSHR			MAXSHR		
			f_{\max}		
PARAME THICK			THICK		
			t		
PARAME ECCENT LOCAL			ECCENx	ECCENy	ECCENZ
			e_x	e_y	e_z
PARAME ECCENT GLOBAL			ECCENX	ECCENY	ECCENZ
			e_X	e_Y	e_Z

4.2.10 Fracture Mechanics Parameters

For crack tip elements [Vol. *Element Library*] DIANA can calculate and output the parameters for Linear Elastic Fracture Mechanics analysis (LEFM). See Chapter 57 for background theory. See also example crackpl in Volume *Analysis Examples*.

syntax

FRACTU

FRACTU specifies the LEFM parameters as output item. No further options apply for this output. The LEFM parameters are always output as two ‘components’.

LEFM parameters	
<i>item</i>	
FRACTU	K G
	K_I G_I

K_I is the Mode-I stress intensity factor [Eq. (57.2) p. 639]. The stress intensity factor can only be calculated if the material parameters and the thickness (for plane stress elements) of all the elements in the crack front are identical.

G_I is the Mode-I energy release rate [Eq. (57.8) p. 640].

Chapter 5

Fatigue Failure Analysis

You may perform fatigue failure analysis with DIANA's Module LINSTA. For fatigue failure analysis you must input the so-called *Wöhler diagram* [Vol. *Material Library*]. If you do so, then DIANA can output the number of load cycles to failure for various stress amplitudes [§ 5.1]. See also Example “Center Cracked Plate” (*crackpl*) in Volume *Analysis Examples*.

5.1 Number of Load Cycles to Failure

If the stress amplitude is calculated in a specific material point of the model, then the number of load cycles to fatigue failure can be easily determined for this point using the Wöhler diagram. For each selected point, DIANA automatically derives the stress amplitude. Therefore, assuming that in one load cycle all selected load sets are passed through, the difference of the extreme stress values of these load sets yield the stress amplitude of this point [Fig. 5.1].

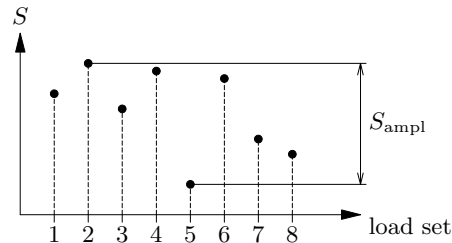


Figure 5.1: Stress Amplitude

5.1.1 Stress Amplitudes

DIANA can calculate the stress amplitude from the Von Mises stresses or from the principal stresses. For the Von Mises stresses, the amplitude is the difference between the smallest and largest stress. For the principal stresses, the amplitude is the difference between the largest absolute value of the maximum and minimum stresses.

Examples of principal stress amplitude. Consider the following stress situation of extreme principal stresses

$$\begin{aligned} S_{1.\max} &= 1.0 & S_{2.\max} &= 0.0 \\ S_{1.\min} &= 0.5 & S_{2.\min} &= 0.0 \end{aligned} \quad (5.1)$$

The stress amplitude is calculated by

$$S_{\text{ampl}} = S_{1.\max} - S_{1.\min} = 1.0 - 0.5 = 0.5 \quad (5.2)$$

Another example is the following stress situation of extreme principal stresses

$$\begin{aligned} S_{1.\max} &= 1.0 & S_{2.\max} &= 0.5 \\ S_{1.\min} &= 0.0 & S_{2.\min} &= -0.5 \end{aligned} \quad (5.3)$$

the stress amplitude yields

$$S_{\text{ampl}} = S_{1.\max} - S_{2.\min} = 1.0 - (-0.5) = 1.5 \quad (5.4)$$

5.1.2 Output Selection

syntax

```

BEGIN OUTPUT [devicew] [outoptw...] [params]
[ OFF ]
[ BEGIN SELECT
[ ... ]
LOADS [ _____ ]
      losetsn...
      ALL
END SELECT ]
NFATIG { operw } { locaw } { optiw }
        VONMIS   NODES   SMOOTH
        PRINCI   INTPNT  ERROR
END OUTPUT

```

SELECT command block to customize the batch output.

... for model selection see § 3.6.2 on page 59.

A load cycle is defined by a series of two or more load sets which subsequently occur in one cycle. By default DIANA takes all load sets specified in table 'LOADS' [§ 2.3.8 p. 45], you may select load sets with the following command.

[ALL]

LOADS *losets* is a series of load sets, referring to input table 'LOADS', which define the load cycle for fatigue failure analysis.

OUTPUT command block to select analysis results for output. See § 3.6.1 on page 56 for options.

You must specify the output of the number of load cycles to fatigue failure in separate OUTPUT blocks, because the interpretation of selected loads is not like for other output items. It is not allowed to put the item NFATIG together with other items in one OUTPUT block.

NFATIG specifies the number of load cycles to fatigue failure as output item.

For each output point, DIANA calculates the stress amplitude from the extreme stresses. Using the specified Wöhler diagram [Vol. *Material Library*], DIANA translates the stress amplitude into the number of load cycles to fatigue failure.

[VONMIS]

oper specifies the formulation of the stress amplitude used. VONMIS for difference between extreme Von Mises stresses.¹ PRINCI for difference between extreme principal stresses.

loca specifies the location for the strains to be output [§ 3.6.1 p. 58].

opti are additional options [§ 3.6.1 p. 58].

¹Von Mises stresses are not suitable for pressure-to-tension load cycles.

<i>item</i>	<i>oper</i>	
NFATIG VONMIS	NF	N_f
NFATIG PRINCI	NF	N_f

Part III

Dynamic Analysis

Chapter 6

Input for Dynamic Analysis

This chapter describes the additional input data for dynamic analysis. Input of mass and damping is appropriate for all types of dynamic analysis. Transient dynamic analysis may require additional input of initial displacements or velocities, and a time-load diagram [§ 6.5 p. 106]. Modal pushover analysis requires additional input of a pushover load [§ 6.4 p. 104]. Frequency domain dynamic analysis requires additional input of a frequency-load diagram [§ 6.6 p. 110]. Base excitation is a typical loading for all types of dynamic analysis [§ 6.3 p. 104]. In transient dynamic analysis prescribed accelerations can be used to model non-uniform nodal accelerations [§ 6.5.2 p. 107].

6.1 Mass

In dynamic analysis DIANA requires the input of mass parameters. See Volume *Material Library* for description of input syntax. There are three forms of mass input:

Mass density for structural elements. In dynamic analysis you must specify the mass density for all structural elements. In static analysis, the mass density is required to determine dead weight load.

Concentrated mass for point elements. Concentrated mass in dynamic analysis can be modelled with point mass/damping elements. Generally speaking, these elements do not influence the static behaviour of the model, i.e., they do not have stiffness, strain or stress. For rotational point mass/damping elements, concentrated mass moments of rotational inertia can be defined. In static analysis, the concentrated mass acts as concentrated loading for dead weight.

Distributed mass for line and surface mass elements. Distributed mass in dynamic analysis can be modelled with line and surface distributed translational mass elements. Generally speaking, these elements do not influence the static behaviour of the model, i.e. they do not have stiffness, strain or stress. In static analysis, the distributed mass acts not as loading for dead weight. For dam engineering, the Westergaard mass distribution can be used for the dynamic mass of the dam reservoir.

6.2 Damping

There are various forms of damping input: viscous damping and structural damping for structural elements, continuous damping (dashpots) via discrete spring/dashpot elements or point mass/damping elements. See Volume *Material Library* for description of input syntax. See also § 48.1.2 on page 586 for background theory of damping.

Viscous damping. Viscous damping is a form of damping which is proportional to the velocity. For structural elements you may specify Raleigh damping coefficients to simulate viscous damping.

Structural damping. Structural damping, also called hysteretic damping, is a form of damping which is independent of the frequency and proportional to the displacement. It may be specified for structural elements in the model and is only valid in a direct frequency response analysis.

Continuous damping. Continuous damping may be specified via discrete spring/dashpot elements or one-node point mass/damping elements.

6.3 Base Excitation

A rigid base motion may be specified with accelerations in various (orthogonal) directions. A typical application for base excitation is a well known earthquake spectrum.

Note that at maximum three orthogonal base excitations may be specified.

See § 48.1.4 on page 586 for background theory.

syntax

```
'LOADS'
CASE casen
BASE
1
80
dirnrn accelr ...
```

CASE starts a new subtable, *case* is the load case number, used for reference from the analysis commands.

BASE indicates that the load case consists of a base motion: number *dirnr* refers to a direction in table 'DIRECT', value *accel* is the uniform translational acceleration applied to the rigid base. For all translational supports, DIANA will resolve the acceleration in the supported direction.

file.dat

```
'DIRECT'
1 1. 0. 0.
2 0. 1. 0.
3 0. 0. 1.
4 1. 1. 0.
'LOADS'
CASE 1
BASE
3 6.
4 10.
```

In this example the rigid base involves a uniform translational acceleration of 6 *length/time*² in the model *Z* direction and of 10 *length/time*² in a direction of 45° with the model *X* and *Y* axes.

6.4 Modal Pushover Analysis

This section describes the load input for modal pushover analysis.

To perform a pushover analysis for assessment of seismic safety, a load pattern that approximates the distribution of inertia forces should be defined. FEMA 273 [13] proposed different load patterns, amongst them an equivalent lateral force (ELF) modal distribution. This ELF distribution is approximately proportional to the first mode shape. Entering such a schematized load distribution on nodes, edges or faces of a construction

may prove to be a lot of work, therefore DIANA offers a generalized form of mode shape dependent load.

$$\mathbf{f}_{\text{push}} = \sum_j a * \mathbf{M} * \phi_i * \mathbf{x}_j \quad (6.1)$$

Where a is the specified acceleration, \mathbf{M} is the mass matrix, ϕ_i is the i^{th} eigenvector, and x_j is the earthquake direction j . The eigenvector ϕ_i is scaled such that the maximum magnitude in the selected load direction is set to 1. This provides a load distribution over all elements that have a mass matrix. A prerequisite for use of this load is that a free vibration type eigenvalue analysis has to be performed prior to the setup of loads in linear or nonlinear static analysis. In this preliminary eigenvalue analysis, geometric stress stiffness may be included. Nonlinear material behaviour may be concentrated in predefined hinges of critical points (e.g. by means of nonlinear springs). It is equally possible to use general nonlinear material models throughout the whole finite element model and in that manner compute the regions with nonlinear material behaviour. Guidelines for performing pushover analysis can be found in Chopra and Goel [19].

The load parameters are defined in subtable **PUSHOV** of table '**LOADS**'. It is possible to define multiple pushover loads in one load case. There are two forms of input syntax for pushover loading on specific elements:

1. one element load per line,
2. elements in a series of numbers or groups or both, with one pushover load specification valid for all the elements in the series.

Single element

syntax

'LOADS'									
CASE $case_n$									
PUSHOV									
1	5	6	12	13	80				
$elem_n$	$\underline{dirtyp_w}$								
	SPECIF								
	ALLDIR								
	NOGRAV								
	DIRECT		$dirnr_n$						
	ACCELE		acc_r						
	MODE		$mode_n$						

Series of elements

syntax

'LOADS'									
CASE $case_n$									
PUSHOV									
1	5	6	12	13	80				
/ $elems_{ng...}$ /									
$\underline{dirty_p}_w$									
SPECIF									
ALLDIR									
NOGRAV									
DIRECT $dirnr_n$									
ACCELE acc_r									
MODE $mode_n$									

CASE starts a new subtable, *case* is the load case number, used for reference from the analysis commands.

PUSHOV indicates that the load case consists of a pushover load.

elem is a single element number, *elems* is a series of elements, it must be specified between slashes and may comprise numbers or groups or both.

dirtyp specifies the direction type for the pushover loading:

SPECIF indicates that the pushover load is only applied in a user defined direction, which is specified by the DIRECT parameter.

ALLDIR indicates that the pushover load is applied in all direction components of the mode shape.

NOGRAV indicates that the pushover load is applied in all direction components of the mode shape, but leaving out the gravity direction [§ 1.2 p. 6].

DIRECT *dirnr* specifies the direction of the modal pushover load using a number referring to a direction in table 'DIRECT'. This input is only applicable for the user specified direction (SPECIF).

ACCELE *acc* specifies the acceleration.

(*mode* > 0) MODE *mode* defines the eigenmode to be considered.

6.5 Transient Analysis

This section describes the input for transient structural and fluid–structure interaction analysis which is additional to the general input for the finite element model [Ch. 1] and for structural analysis [Ch. 2]. See also Chapter 7 for commands for transient analysis.

6.5.1 Initial Conditions

You may apply various types of initial conditions for a transient analysis.

- *Specified initial displacements or velocities.* You may specify initial displacements or velocities in input table 'INIVAR' [§ 6.5.1.1]. To let DIANA actually use these displacements or velocities as an initial condition for a transient analysis, you must supply the START command with special options [§ 7.3 p. 124] [§ 13.3.1 p. 220].
- *Calculated initial displacements.* You may ask DIANA to apply previously calculated displacements as an initial condition for a transient dynamic analysis. Calculated initial displacements require no further input data, you only must supply the START command with special options depending on the origin of the displacement field.
- *Specified initial stresses.* The initial stresses as specified via an element prestress load in input table 'LOADS' [Vol. *Element Library*] may be applied as initial stresses for the transient analysis. To let DIANA actually apply these stresses as an initial condition you must supply the START command with special options.
- *Calculated initial stresses.* The stresses as calculated for a load set of a linear static analysis may be applied as initial stresses for a transient analysis. Calculated initial stresses require no further input data, you only must supply the START command with special options.

6.5.1.1 Initial Displacements or Velocities

Initial displacement or velocity fields for transient analysis are input via table 'INIVAR'. These fields may be used as initial (strain free) condition for a transient analysis via special start option [§ 13.3.1.1 p. 222].

syntax

'INIVAR'			
DISPLA <i>field_n</i>			
1	5	6	80
<i>node_n</i>	<i>type_w</i>	<i>dirnr_n</i>	<i>displa_r</i>
1	5	6	80
/ <i>nodes_{ng...}</i> /			
	<i>type_w</i>	<i>dirnr_n</i>	<i>displa_r</i>
1	5	6	80
/ <i>nodes_{n...}</i> /			
	<i>type_w</i>	<i>dirnr_n</i>	/ <i>displs_{r...}</i> /
VELOCI <i>field_n</i>			
1	5	6	80
<i>node_n</i>	<i>type_w</i>	<i>dirnr_n</i>	<i>veloci_r</i>
1	5	6	80
/ <i>nodes_{ng...}</i> /			
	<i>type_w</i>	<i>dirnr_n</i>	<i>veloci_r</i>
1	5	6	80
/ <i>nodes_{n...}</i> /			
	<i>type_w</i>	<i>dirnr_n</i>	/ <i>velocs_{r...}</i> /

DISPLA is the subtable heading for initial displacements and VELOCI for initial velocities.

Number *field* is the field number for reference in the start option command.

node is a single node number. *nodes* is a series of nodes, which must be specified between slashes. Depending on the format, *nodes* may comprise numbers or groups or both.

type is the type of variable: TR for translation or RO for rotation.

dirnr specifies the direction, referring to table 'DIRECT' [§ 1.5 p. 11].

displa is the initial displacement. *displs* is a series of initial displacements, one for each node in *nodes*, it must be specified between slashes.

veloci is the initial velocity. *velocs* is a series of initial velocities, one for each node in *nodes*, it must be specified between slashes.

file.dat

'INIVAR'		
DISPLA 3		
1	TR 3	0.001
/ 10-12 / TR 2 / 0.002(3) /		
4	RO 5	0.00005

This example specifies initial displacement field number 3. The initial translation of node 1 is 0.001 in direction 3. The initial translation of nodes 10 to 12 is 0.002 in direction 2. The initial rotation of node 4 is 0.00005 around direction 5. All other translations and rotations are initially equal to zero.

6.5.2 Prescribed Accelerations

Prescribed accelerations can be used in transient dynamic analysis to model non-uniform nodal accelerations. Prescribed accelerations are input in subtable ACCELE of table LOADS. There are three forms of input syntax: (1) one nodal acceleration per line, (2) nodes in a series of numbers or groups or both, with one acceleration value valid for all the nodes in the series, (3) nodes in a series of numbers and/or groups, with a series of acceleration values: one value for each node. A spatial function can be attached to this property [§ 1.8].

Prescribed accelerations in subtable *ACCELE* must be specified as supports in table 'SUPPOR' [§ 2.1 p. 15]. Consequently they will have a zero displacement for all cases, unless acceleration values are specified in subtable *ACCELE*.

Prescribed accelerations cannot be combined with base acceleration loads and fixed displacement or deformation loads in the same load set.

syntax

```

'LOADS'
ACCELE
1      5 6                                80
noden   typew dirnrn valuer
1      5 6                                80
/ nodesng... /
      typew dirnrn valuer
1      5 6                                80
/ nodesng... /
      typew dirnrn / valuesr... /

```

node is a single node number, *nodes* is a series of nodes, it must be specified between slashes and may comprise numbers or groups or both.

type is the acceleration type: TR for translation or RO for rotation. The direction number *dirnr* refers to table 'DIRECT' [§ 1.5 p. 11]: a translation in, or a rotation around the specified direction.

value is the acceleration value, *values* is a series of acceleration values, one for each node in *nodes*.

file.dat

```

'LOADS'
CASE 3
ACCELE
  3   TR   2   1.E-2
/ 4-12 EDGE / TR 5 .001
CASE 5
ACCELE
  3   TR   2   2.E-2
CASE 7
ACCELE
  2   RO   3   0.1

```

In this example the prescribed acceleration of node 3 in direction 2 differs in case 3 and case 5!

6.5.3 Time–Load Diagram

You may specify a time–load diagram completely, i.e., times and load factors [§ 6.5.3.1], in table 'TIMELO'. Alternatively, you may specify only the times in table 'TIMELO' and import the load factors from an external file [§ 6.5.3.2].

6.5.3.1 Direct Input

Direct input is used to specify a time–load diagram by times and load factors explicitly in table 'TIMELO'.

Complete table*syntax***'TIMELO'**

1	80
---	----

LOAD tlo_n TIMES $times_{r...}$ /FACTOR $lf_{r...}$ /**'TIMELO'** is the table heading for the time-load diagram input.LOAD tlo specifies the transient load which is active during the following times. The load number tlo refers to a load set in Table 'LOADS' [§ 2.3.8 p. 45].TIMES $times$ are the times t for the corresponding load factors. Times must be specified in increasing order. ($t_{i+1} \geq t_i$)FACTOR lf are the load factors for the corresponding times.

You may specify multiple sets of times and load factors, optionally with a different load number for each set. Within one set, the number of factors lf must always match the number of times in $times$ as shown in the following example.

*file.dat***'TIMELO'**

LOAD 2

TIMES 0.00 0.10 /

FACTOR 2.5 3.6 /

LOAD 1

TIMES 0.13 0.16 0.16 0.45 /

FACTOR 6.4 4.2 2.5 0.0 /

LOAD 3

TIMES 0.45:0.501(0.01) /

FACTOR 4.8 5.3 7.9 7.8 6.2 1.4 /

In this example no load is active from $t = 0.10$ to $t = 0.13$.**6.5.3.2 External File**

An external file can be used to import the load factors of a time-load diagram. The times are specified explicitly in table 'TIMELO'.

Factors imported*syntax***'TIMELO'**

1	5	6	80
---	---	---	----

LOAD tlo_n TIMES $times_{r...}$ /FACTOR IMPORT $file_s$ [SKIP $nlin_n$]

[_____]

SCALE $sfac_r$ PEAK $pval_r$ **'TIMELO'** is the table heading for the time-load diagram input.LOAD tlo specifies the transient load which is active during the following times. The load number tlo refers to a load set in Table 'LOADS' [§ 2.3.8 p. 45].TIMES $times$ are the times t for the corresponding load factors. Times must be specified in increasing order. ($t_{i+1} \geq t_i$)

- FACTOR IMPORT** asks DIANA to read the load factors from an external file named *file*. Factors on this file must be separated by spaces, commas, tabs or newlines. There are two additional options.
- [*nlin* = 0] **SKIP** causes DIANA to skip the first *nlin* lines before starting to read load factors.
- [*sfac* = 1] **SCALE** specifies a scale factor *sfac*. DIANA will multiply each load factor from the external file by this factor before applying the load.
- (*pval* > 0) **PEAK** asks DIANA to scale the values from the external file such that the greatest absolute value of the load factors is equal to *pval*.

You may specify multiple sets of times and file, optionally with a different load number for each set. Within one set, DIANA reads as many factors from *file* as there are times in *times*. For repeated import from *file* reading continues from the last read factor of the previous import.

```
file.dat
```

```
'TIMELO'
LOAD 4
TIMES 0.00 0.10 0.13 0.16 0.16 0.45 /
FACTOR IMPORT "load.log"
      SKIP 2
      SCALE 3.
```

In this example DIANA will read six load factors from external file `load.log`. This file could be like shown below.

```
load.log
```

```
Load logging for tower footing
Thu Mar 19 16:33:55 MET 1998
 1 3 4 5.2 3.2 1.6
```

The first two lines will be skipped. The actual load factors for transient load 4 will be: $lf_1 = 3 \times 1 = 3$, $lf_2 = 3 \times 3 = 9$, etc. Note that in this example the load factor suddenly drops at time $t = 0.16$ from $3 \times 5.2 = 15.6$ to $3 \times 3.2 = 9.6$.

All regular Fortran formats are allowed for the factors on the external file. Integers (without decimal point) are interpreted as reals. Exponents of ten may be specified via E or D format. For instance, 45, 45.0, 45.0E0, and 45.0D0 all represent the real value of 45.0 for the scale factor.

6.5.4 Ambient Time Dependency

Time dependent temperature, concentration, and pressure is taken into account as a loading in a transient analysis. See input tables 'TEMPER', 'CONCEN', and 'PRESSU' in Volume *Material Library*. This transient loading is analogous to specification via element loads TEMPER, CONCEN, and PRESSU in subtable ELEMEN of table 'LOADS' [Vol. *Element Library*], in combination with a time-load diagram in table 'TIMELO' [§ 6.5.3].

6.6 Frequency Domain Analysis

This section describes the input for frequency domain structural and fluid-structure interaction analysis which is additional to the general input for the finite element model [Ch. 1] and for structural analysis [Ch. 2]. See also Chapter 8 for commands for frequency response analysis and Chapter 9 for commands for Response Spectrum analysis.

6.6.1 Frequency–Load Diagram

You may specify a frequency–load diagram completely [§ 6.6.1.1], i.e., frequencies and load factors, or periods and load factors, in table 'FREQLO'. Alternatively, you may specify only the frequencies or periods in table 'FREQLO' and import the load factors from an external file [§ 6.6.1.2]. For complex loads, i.e. loads with non-zero phase angles, you may specify either frequency–load diagrams for both the real and imaginary part of the load, or a frequency–amplitude diagram and a frequency–phase diagram [§ 6.6.1.3]. Input as function of periods is also possible for complex loads. Furthermore, elastic response spectra [§ 6.6.1.4] and design spectra [§ 6.6.1.5] according to Eurocode 8 EN 1998-1 [65] for seismic analys, or elastic response spectra [§ 6.6.1.6] and design spectra [§ 6.6.1.7] as described in the Dutch NEN NPR 9998:2015 [68] guidelines for induced earthquakes, can also be specified in table 'FREQLO'.

6.6.1.1 Direct Input

Direct input is used to specify a frequency–load diagram by frequencies and load factors or by periods and load factors explicitly in table 'FREQLO'.

Complete table

syntax

'FREQLO'

1 80

LOAD *flo_n*

FREQUE *frequencies_{r...}* /

PERIOD *periods_{r...}* /

FACTOR *lf_{r...}* /

'FREQLO' is the table heading for the frequency–load diagram input.

LOAD *flo* specifies the load which is active during the following frequencies or periods. The load number *flo* refers to a load set in Table 'LOADS' [§ 2.3.8 p. 45].

FREQUE *frequencies* are the frequencies *f* for the corresponding load factors. Frequencies must be specified in increasing order. $(f_{i+1} \geq f_i)$

PERIOD *periods* are the periods *T* for the corresponding load factors. Periods must be specified in increasing order. $(T_{i+1} \geq T_i)$

FACTOR *lf* are the load factors for the corresponding frequencies or periods.

You may specify multiple sets of frequencies ior periods and load factors, optionally with a different load number for each set. Within one set, the number of factors *lf* must always match the number of frequencies in *frequencies* as shown in the following example.

file.dat

'FREQLO'

LOAD 2

FREQUE 0.00 0.10 /

FACTOR 2.5 3.6 /

LOAD 1

FREQUE 0.13 0.16 0.16 0.45 /

FACTOR 6.4 4.2 2.5 0.0 /

LOAD 3

FREQUE 0.45:0.501(0.01) /

FACTOR 4.8 5.3 7.9 7.8 6.2 1.4 /

In this example no load is active from $f = 0.10$ to $f = 0.13$.

6.6.1.2 External File

An external file can be used to import the load factors of a frequency-load diagram. The frequencies or periods are specified explicitly in table 'FREQLO'.

Factors imported

syntax

```
'FREQLO'
```

```
LOAD flon
```

```
FREQUE frequenciesr... /
```

```
PERIOD periodsr... /
```

```
FACTOR IMPORT files
```

```
    [SKIP nlinn]
```

```
    [_____]
```

```
    SCALE sfacr
```

```
    PEAK pvalr
```

'FREQLO' is the table heading for the frequency-load diagram input.

LOAD *flo* specifies the load which is active during the following frequencies or periods. The load number *flo* refers to a load set in Table 'LOADS' [§ 2.3.8 p. 45].

FREQUE *frequencies* are the frequencies *f* for the corresponding load factors. Frequencies must be specified in increasing order.

PERIOD *periods* are the periods *T* for the corresponding load factors. Periods must be specified in increasing order.

FACTOR IMPORT asks DIANA to read the load factors from an external file named *file*. Factors on this file must be separated by spaces, commas, tabs or newlines. There are two additional options.

[*nlin* = 0] SKIP causes DIANA to skip the first *nlin* lines before starting to read load factors.

[*sfac* = 1] SCALE specifies a scale factor *sfac*. DIANA will multiply each load factor from the external file by this factor before applying the load.

(*pval* > 0) PEAK asks DIANA to scale the values from the external file such that the greatest absolute value of the load factors is equal to *pval*.

You may specify multiple sets of frequencies or periods and file, optionally with a different load number for each set. Within one set, DIANA reads as many factors from *file* as there are frequencies or periods. For repeated import from *file* reading continues from the last read factor of the previous import.

file.dat

```
'FREQLO'
```

```
LOAD 4
```

```
FREQUE 0.00 0.10 0.13 0.16 0.16 0.45 /
```

```
FACTOR IMPORT "load.log"
```

```
    SKIP 2
```

```
    SCALE 3.
```

In this example DIANA will read six load factors from external file *load.log*. This file could be like shown below.

load.log

```
Load logging for tower footing
```

```
Thu Mar 19 16:33:55 MET 1998
```

```
1 3 4 5.2 3.2 1.6
```

The first two lines will be skipped. The actual load factors for transient load 4 will be: $lf_1 = 3 \times 1 = 3$, $lf_2 = 3 \times 3 = 9$, etc. Note that in this example the load factor suddenly drops at frequency $f = 0.16$ from $3 \times 5.2 = 15.6$ to $3 \times 3.2 = 9.6$.

All regular Fortran formats are allowed for the factors on the external file. Integers (without decimal point) are interpreted as reals. Exponents of ten may be specified via E or D format. For instance, 45, 45.0, 45.0E0, and 45.0D0 all represent the real value of 45.0 for the scale factor.

6.6.1.3 Complex Loads

Load multiplication factors for complex loads, i.e. loads with non-zero phase angles, can be specified by either frequency-load diagrams for both the real and complex part of the load, or a frequency-amplitude diagram and a frequency-phase diagram. Input as function of periods is also possible for complex loads.

Real and imaginary multiplication factors

syntax

'FREQLO'

1 80

LOAD flo_n

FREQUE $frequencies_{r...}$ /

PERIOD $periods_{r...}$ /

REAL $lfre_{r...}$ /

IMAGIN $lfim_{r...}$ /

'FREQLO' is the table heading for the frequency-load diagram input.

LOAD flo specifies the load which is active during the following frequencies or periods. The load number flo refers to a load set in Table 'LOADS' [§ 2.3.8 p. 45].

FREQUE $frequencies$ are the frequencies f for the corresponding load factors. Frequencies must be specified in increasing order. $(f_{i+1} \geq f_i)$

PERIOD $periods$ are the periods T for the corresponding load factors. Periods must be specified in increasing order. $(T_{i+1} \geq T_i)$

REAL $lfre$ are the load multiplication factors of the real part of the complex load for the corresponding frequencies or periods.

IMAGIN $lfim$ are the load multiplication factors of the imaginary part of the complex load for the corresponding frequencies or periods.

You may specify multiple sets of frequencies or periods and real and imaginary load factors, optionally with a different load number for each set. Within one set, the number of real factors $lfre$ and imaginary factors $lfim$ must always match the number of frequencies or periods as shown in the following example.

file.dat

'FREQLO'

LOAD 2

FREQUE 0.00 0.10 /

REAL 0.0 0.0 /

IMAGIN 2.5 3.6 /

LOAD 1

FREQUE 0.13 0.16 0.16 0.45 /

REAL 6.4 4.2 2.5 0.0 /

IMAGIN 6.4 4.2 2.5 0.0 /

LOAD 3

FREQUE 0.45:0.501(0.01) /

FACTOR 4.8 5.3 7.9 7.8 6.2 1.4 /

In this example no load is active from $f = 0.10$ to $f = 0.13$.

Amplitude multiplication factors and phase angles

syntax

'FREQLO'

1

80

LOAD flo_n

FREQUE $frequencies_{r...}$ /

PERIOD $periods_{r...}$ /

AMPLIT $ampl_{r...}$ /

PHASE $ph_{r...}$ /

'FREQLO' is the table heading for the frequency-load diagram input.

LOAD flo specifies the load which is active during the following frequencies or periods. The load number flo refers to a load set in Table 'LOADS' [§ 2.3.8 p. 45].

($f_{i+1} \geq f_i$) FREQUE $frequencies$ are the frequencies f for the corresponding load factors. Frequencies must be specified in increasing order.

($T_{i+1} \geq T_i$) PERIOD $periods$ are the periods T for the corresponding load factors. Periods must be specified in increasing order.

AMPLIT $ampl$ are the load amplitude multiplication factors of the complex load for the corresponding frequencies or periods.

PHASE ph are the phase angles of the complex load for the corresponding frequencies or periods.

You may specify multiple sets of frequencies or periods and load amplitude multiplication factors and phase angles, optionally with a different load number for each set. Within one set, the number of real factors $lfre$ and imaginary factors $lfim$ must always match the number of frequencies or periods as shown in the following example.

file.dat

'UNITS'

ANGLE DEGREE

'FREQLO'

LOAD 2

FREQUE 0.00 0.10 /

AMPLIT 2.5 3.6 /

PHASE 90.0 90.0 /

LOAD 1

FREQUE 0.13 0.16 0.16 0.45 /

AMPLIT 40.96 17.64 6.25 0.0 /

PHASE 45.0 45.0 45.0 45.0 /

LOAD 3

FREQUE 0.45:0.501(0.01) /

FACTOR 4.8 5.3 7.9 7.8 6.2 1.4 /

Note that in this example no load is active from $f = 0.10$ Hz to $f = 0.13$ Hz.

6.6.1.4 Eurocode 8 EN 1998-1 Elastic Response Spectrum

Elastic response spectrum

syntax

```

'FREQL0'
1 5 6 80
LOAD blon
CODE      EURO8
SPECTR    ELASTI
[ ORIENT { _____ } ]
          HORIZO
          VERTIC
[ REGION { _____ } ]
          TYPE1
          TYPE2
[ GROUND { _____ } ]
          A
          B
          C
          D
          E
          CUSTOM sr tbr tcr tdr
[ IMPORT   gammar ]
[ DAMPIN   cr ]

```

'FREQLQ' is the table heading for the frequency-load diagram input.

LOAD *blo* specifies the base excitation load [§ 6.3 p. 104] with an acceleration equal to the reference peak ground acceleration on type A ground a_{gR} which is active during the seismic analysis. The load number *blo* refers to a load set in Table 'LOADS' [§ 2.3.8 p. 45].

CODE EURO8 specifies that the spectrum will be set up according to Eurocode 8 [65].

SPECTR ELASTI specifies that an elastic response spectrum will be set up.

ORIENT indicates the orientation of the spectrum:

[HORIZO]

HORIZO for a horizontal spectrum.

VERTIC for a vertical spectrum.

REGION indicates the type of seismicity region:

[TYPE1]

TYPE1 indicates a high or moderate seismicity region ($M_s > 5.5$).

TYPE2 indicates a low seismicity region ($M_s \leq 5.5$).

GROUND indicates the ground types as described in Eurocode 8 EN 1998-1 to define the recommended values for the soil factor S , the lower limit of the period of the constant spectral acceleration branch T_B , the upper limit of the period of the constant spectral acceleration branch T_C and beginning of the constant displacement response range of the spectrum T_D , user-defined values for these parameters, or the elastic response spectrum according to the Dutch NEN NPR 9998 [68] guidelines for induced earthquakes:

A for rock or other rock-like geological formations, including at most 5 m of weaker material at the surface.

B for deposits of very dense sand, gravel, or very stiff clay, at least several tens of meters in thickness, characterized by a gradual increase of mechanical properties with depth.

C for deep deposits of dense or medium-dense sand, gravel or stiff clay with thickness from several tens to many hundreds of meters.

D for deposits of loose-to-medium cohesionless soil (with or without some soft cohesive layers), or of predominantly soft-to-firm cohesive soil.

E for a soil profile consisting of a surface alluvium layer with average shear wave velocity values of type C or D and thickness varying between 5 m and 20 m, underlaid by stiffer material with an average shear wave velocity ν_s larger than 800m/s.

CUSTOM for user-defined values in case the recommended ground types cannot be used: s for the soil factor S ; tb for the lower limit of the period of the constant spectral acceleration branch T_B ; tc for the upper limit of the period of the constant spectral acceleration branch T_C ; and td for the beginning of the constant displacement response range of the spectrum T_D

[$\gamma_I = 1.0$] IMPORT *gamma* is the importance factor γ_I .

[$\xi = 0.05$] DAMPIN *c* indicates the viscous damping ratio of the structure ξ .

6.6.1.5 Eurocode 8 EN 1998-1 Design Spectrum

Design spectrum	<i>syntax</i>
'FREQLO'	
1	5 6 80
LOAD <i>blo_n</i>	
CODE EURO8	
SPECTR DESIGN	
[BEHAVI <i>q_r</i>]	
[ORIENT { _____ }]	
HORIZO	
VERTIC	
[REGION { _____ }]	
TYPE1	
TYPE2	
[GROUND { _____ }]	
A	
B	
C	
D	
E	
CUSTOM <i>s_r tb_r tc_r td_r</i>	
[IMPORT <i>gamma_r</i>]	
[LBOUND <i>beta_r</i>]	

'FREQLO' is the table heading for the frequency-load diagram input.

LOAD *blo* specifies the base excitation load [§ 6.3 p. 104] with an acceleration equal to the reference peak ground acceleration on type A ground a_{gR} which is active during the seismic analysis. The load number *blo* refers to a load set in Table 'LOADS' [§ 2.3.8 p. 45].

CODE EURO8 specifies that the spectrum will be set up according to Eurocode 8 EN 1998-1 [65].

SPECTR DESIGN specifies that a design spectrum will be set up.

BEHAVI *q* specifies the behaviour factor *q*, which is an approximation of the ratio of the seismic forces that the structure would experience if its response was completely elastic with 5 % viscous damping, to the seismic forces that may be used in the design, with a conventional elastic analysis model, still ensuring a satisfactory response of the structure. The value of behaviour factor *q* may be different in different horizontal directions of the structure. For the vertical component of the seismic action a behaviour factor *q* up to 1.5 should generally be adopted for all materials and structural systems. [*q* = 1.5]

ORIENT indicates the orientation of the spectrum: [HORIZO]

HORIZO for a horizontal spectrum.

VERTIC for a vertical spectrum.

REGION indicates the type of seismicity region: [TYPE1]

TYPE1 indicates a high or moderate seismicity region ($M_s > 5.5$).

TYPE2 indicates a low seismicity region ($M_s \leq 5.5$).

GROUND indicates the ground types as described in Eurocode 8 EN 1998-1 to define the recommended values for the soil factor *S*, the lower limit of the period of the constant spectral acceleration branch T_B , the upper limit of the period of the constant spectral acceleration branch T_C and beginning of the constant displacement response range of the spectrum T_D , user-defined values for these parameters, or the design spectrum according to the Dutch NEN NPR 9998 [68] guidelines for induced earthquakes:

A for rock or other rock-like geological formations, including at most 5 m of weaker material at the surface.

B for deposits of very dense sand, gravel, or very stiff clay, at least several tens of meters in thickness, characterized by a gradual increase of mechanical properties with depth.

C for deep deposits of dense or medium-dense sand, gravel or stiff clay with thickness from several tens to many hundreds of meters.

D for deposits of loose-to-medium cohesionless soil (with or without some soft cohesive layers), or of predominantly soft-to-firm cohesive soil.

E for a soil profile consisting of a surface alluvium layer with average shear wave velocity values of type C or D and thickness varying between 5 m and 20 m, underlayed by stiffer material with an average shear wave velocity ν_s larger than 800m/s.

CUSTOM for user-defined values in case the recommended ground types cannot be used: *s* for the soil factor *S*; *tb* for the lower limit of the period of the constant spectral acceleration branch T_B ; *tc* for the upper limit of the period of the constant spectral acceleration branch T_C ; and *td* for the beginning of the constant displacement response range of the spectrum T_D

IMPORT *gamma* is the importance factor γ_I . [$\gamma_I = 1.0$]

LBOUND *beta* is the lower bound factor β for the horizontal design spectrum. [$\beta = 0.2$]

6.6.1.6 NPR 9998:2015 Elastic Response Spectrum

Elastic response spectrum*syntax*

'FREQLO'		
1	5	680
LOAD <i>blo_n</i>		
CODE	NPR	
SPECTR	ELASTI	
[ORIENT {	<u> </u>	}]
	HORIZO	
	VERTIC	
[KAG	<i>kag_r</i>]
[IMPORT	<i>gamma_r</i>]
[DAMPIN	<i>c_r</i>]

'FREQLO' is the table heading for the frequency-load diagram input.

LOAD *blo* specifies the base excitation load [§ 6.3 p. 104] with an acceleration equal to the reference peak ground acceleration a_{gR} which is active during the seismic analysis. The load number *blo* refers to a load set in Table 'LOADS' [§ 2.3.8 p. 45].

CODE NPR specifies that the spectrum will be set up according to the Dutch NEN NPR 9998:2015 [68] guidelines for induced earthquakes.

SPECTR ELASTI specifies that an elastic response spectrum will be set up.

[HORIZO] ORIENT indicates the orientation of the spectrum:

HORIZO for a horizontal spectrum.

VERTIC for a vertical spectrum.

KAG *kag* is the dimensionless factor k_{ag} which is depending on the Consequence Class according to Tables 2.1 and 2.2 of the Dutch NEN NPR 9998:2015 [68] guidelines for induced earthquakes.

[$k_{ag} = 1.9$]

[$\gamma_I = 1.0$] IMPORT *gamma* is the importance factor γ_I .

[$\xi = 0.05$] DAMPIN *c* indicates the viscous damping ratio of the structure ξ .

6.6.1.7 NPR 9998:2015 Design Spectrum**Design spectrum***syntax*

'FREQLO'		
1	5	6
LOAD blo_n		
CODE	NPR	
SPECTR	DESIGN	
[BEHAVI	q_r]	
[ORIENT {	<u> </u> }]	
	HORIZO	
	VERTIC	
[DESSPC {	<u> </u> }]	
	SD	
	SA	
[KAG	kag_r]	
[IMPORT	$gamma_r$]	

'FREQLO' is the table heading for the frequency-load diagram input.

LOAD *blo* specifies the base excitation load [§ 6.3 p. 104] with an acceleration equal to the reference peak ground acceleration a_{gR} which is active during the seismic analysis. The load number *blo* refers to a load set in Table 'LOADS' [§ 2.3.8 p. 45].

CODE NPR specifies that the spectrum will be set up according to the Dutch NEN NPR 9998:2015 [68] guidelines for induced earthquakes.

SPECTR DESIGN specifies that a design spectrum will be set up.

BEHAVI *q* specifies the behaviour factor q , which is an approximation of the ratio of the seismic forces that the structure would experience if its response was completely elastic with 5 % viscous damping, to the seismic forces that may be used in the design, with a conventional elastic analysis model, still ensuring a satisfactory response of the structure. The value of behaviour factor q may be different in different horizontal directions of the structure. For the vertical component of the seismic action a behaviour factor q up to 1.5 should generally be adopted for all materials and structural systems.

[$q = 1.5$]

ORIENT indicates the orientation of the spectrum:

[HORIZO]

HORIZO for a horizontal spectrum.

VERTIC for a vertical spectrum.

DESSPC indicates which horizontal design spectrum to use:

[SD]

SD for a horizontal design spectrum according to Eqs. (3.21)-(3.23) of the Dutch NEN NPR 9998:2015 [68] guidelines for induced earthquakes.

SA for a horizontal design spectrum according to Eqs. (3.24)-(3.26) of the Dutch NEN NPR 9998:2015 [68] guidelines for induced earthquakes. This design spectrum is only used in case $V_{s;30} > 250$ m/s, without individual layers with $V_s < 200$ m/s, and $a_{gR} < 0.2$ g.

KAG *kag* is the dimensionless factor k_{ag} which is depending on the Consequence Class according to Tables 2.1 and 2.2 of the Dutch NEN NPR 9998:2015 [68] guidelines for induced earthquakes.

[$k_{ag} = 1.9$]

IMPORT *gamma* is the importance factor γ_I .

[$\gamma_I = 1.0$]

Chapter 7

Linear Transient Structural Analysis

DIANA's Module `NONLIN` may be used to perform a linear transient structural analysis if the forcing function varies in time and a linear behaviour of the system is assumed. Note that dynamic frequency response analysis must be performed with Module `MODAL` or `FREQUE` [Ch. 8]. In order to perform a linear transient analysis with DIANA you must take the following actions:

1. Invoke Module `FILOS` to initialize an analysis database [§ 3.2 p. 48].
2. Invoke Module `INPUT` to read the finite element model into the database [§ 3.3 p. 50].
3. Invoke Module `NONLIN` to perform a linear transient analysis.
4. You may get plots of the analysis results in the Results working environment of iDIANA [Vol. *iDIANA*].

See also § 6.5 on page 106 for the additional input for transient analysis and Chapter 48 on page 585 for background theory. The command sequence for linear transient analysis is as follows.

syntax

```
*NONLIN
[ MODEL ... ]
[ TYPE ]
[ SOLVE ... ]
[ EXECUT ... ] ...
[ OUTPUT ... ] ...
*END
```

`MODEL` evaluates the finite element model [§ 7.1 p. 122].

`TYPE` specifies the type of the analysis [§ 7.2].

`SOLVE` customizes the settings for the solution method [Ch. 30 p. 421].

`EXECUT` executes steps [§ 7.3].

`OUTPUT` selects analysis results for output [§ 7.4 p. 125]. This output selection, at the `*NONLIN` command level, is valid for all `EXECUT` blocks of the current analysis. However, an output selection within any `EXECUTE` block [§ 7.3 p. 124], overrules the output selection at the `*NONLIN` level.

Example. The commands for linear transient analysis are a sub-set of those for non-linear transient analysis as described in Chapter 13. The following is an example of commands for linear transient analysis.

file.dcf

```

*FILOS
INITIA
*INPUT
*NONLIN
BEGIN TYPE
  BEGIN TRANSI
    METHOD NEWMAR GAMMA=0.5 BETA=0.25
    DYNAMI
  END TRANSI
END TYPE
BEGIN EXECUTE
  BEGIN TIME
    BEGIN STEPS
      step sizes
    END STEPS
  END TIME
  BEGIN ITERAT
    MAXITE=0 no iteration!
  END ITERAT
END EXECUTE
*END

```

The ***FILOS** command with **INITIA** invokes Module FILOS to initialize the FILOS file which is the central database for each analysis with DIANA [§ 3.2 p. 48]. The ***INPUT** command invokes Module INPUT to read the complete input data file. This file must contain the general input data for the finite element mesh [Ch. 1], additional data for structural analysis [Ch. 2], the data for mass and damping [Vol. *Material Library*], and additional data for transient analysis [§ 6.5 p. 106] like the time-load diagram as shown below.

file.dat

```

'TIMELO'
LOAD 2
TIMES 0.0 0.00001 1.0 /
FACTOR 0.0 2.85 2.85 /

```

Preliminary static analysis. If you define an initial state with calculated displacements or stresses, then the displacements must be determined via an **EXECUT START** command block [§ 13.3.1 p. 220]. In all other cases the commands described previously are formally sufficient for a linear transient analysis. However, you are urged to perform a linear static analysis and some postprocessing before continuing with the transient dynamic analysis [Ch. 4].

Fluid-structure interaction analysis. In case of a model containing fluid, structural elements and fluid-structure interface elements, automatically a fluid-structure linear transient analysis will be performed. DIANA recognizes this element combination and will perform the required preparation to perform a linear transient fluid-structure analysis. The added mass effect of the fluid [Eq. (48.100) p. 599] is taken into account to calculate the linear transient response. Results will be available for the structural part of the model.

Lumped element matrices may not be used in a fluid-structure response analysis.

7.1 Model Evaluation

The **MODEL** commands customize the evaluation of the finite element model prior to the actual analysis.

syntax

```

BEGIN MODEL
[ OFF ]
[ EVALUA [ OFF ] ... ]
[ ASSEMB [ OFF ] ... ]
[ MATRIX [ OFF ] ]
[ LOADS [ OFF ] ]
END MODEL

```

EVALUA to check and evaluate geometric and material properties for elements and reinforcements [§ 3.4 p. 51].

ASSEMB to assemble the elements, i.e., creating appropriate system degrees of freedom [§ 3.5 p. 54].

MATRIX to setup the element stiffness matrices.

LOADS to setup the load vectors.

7.2 Type of Transient Analysis

The TYPE commands indicate the type of the transient analysis.

syntax

```

BEGIN TYPE
[ OFF ]
PHYSIC OFF
TRANSI ...
END TYPE

```

PHYSIC OFF may switch off physical nonlinear analysis, which by default is on for Module NONLIN.

TRANSI sets options and parameters for transient analysis. Commands are analogous to those for nonlinear transient analysis [§ 13.2.3].

7.3 Step Execution

With the EXECUT commands you ask DIANA to execute load or time steps. A command file may contain one or more EXECUT blocks. Details of various commands are given in the referred sections.

syntax

```

BEGIN EXECUT
[ OFF ]
[ typew ... ]
START
LOAD
TIME
[ ITERAT ... ]
[ STOP ... ]
[ OUTPUT ... ] ...
END EXECUT

```

type indicates the type of step(s) to be executed.

[LOAD]

START to evaluate the initial state [§ 13.3.1 p. 220].

LOAD to execute load steps [§ 7.3.1].

TIME to execute time steps [§ 7.3.2].

ITERAT switches off the process of equilibrium iteration [§ 7.3.3 p. 125].

STOP specifies a stop criterion for step execution [§ 13.3.6 p. 240].

OUTPUT selects output for steps [§ 7.4 p. 125]. This selection overrules any output selection specified at the *NONLIN command level [Ch. 7 p. 121].

7.3.1 Load Steps

The LOAD commands specify the execution of load steps.

syntax

```

BEGIN LOAD
[ OFF ]
[ LOADNR=losetn ]
[ RESTOR stepn ]
[ BEGIN STEPS
  [ methodw ... ]
  EXPLIC
  ENERGY
  AUTOMA
  END STEPS ]
[ SAVE ... ]
END LOAD

```

LOADNR *loset* refers to a load set from table 'LOADS' [§ 2.3.8 p. 45]. Default is the lowest available load set number.

RESTOR restores data for step *step* which must have been saved previously [§ 13.3.9 p. 245].

[EXPLIC] STEPS specifies how to apply step sizes. Via *method* you may specify the step sizes explicitly, or you may let DIANA determine them automatically.

EXPLIC explicitly specified step sizes [§ 13.3.2.1 p. 225].

ENERGY energy based automatic step size control [§ 13.3.2.3 p. 226].

AUTOMA automatic step size control [§ 13.3.2.4 p. 227].

Note that Arc-length methods are not profitable in linear analysis.

SAVE saves data of specified steps for future restart [§ 13.3.9 p. 245].

7.3.2 Time Steps

The TIME commands specify the execution of time steps.

syntax

```

BEGIN TIME
[ OFF ]
[ RESTOR stepn ]
[ BEGIN STEPS
  [ methodw ... ]
  EXPLIC
  EXPONE
  AUTOMA
END STEPS ]
[ SAVE ... ]
END TIME

```

RESTOR restores data for step *step* which must have been saved previously.

STEPS specifies how to apply step sizes. Via *method* you may specify the step sizes explicitly, or you may let DIANA determine them automatically. [EXPLIC]

EXPLIC explicitly specified step sizes [§ 13.3.3.1 p. 232].

EXPONE applies exponential time increments [§ 13.3.3.3 p. 233].

AUTOMA automatic step size control [§ 13.3.2.4 p. 227].

SAVE saves data of specified steps for future restart [§ 13.3.9 p. 245].

7.3.3 No Equilibrium Iteration

Linear transient analysis does not require equilibrium iteration. Therefore you may switch off the iteration process via the ITERAT commands.

syntax

```

BEGIN ITERAT
  MAXITE=0
END ITERAT

```

MAXITE=0 sets the maximum number of iterations for each time or load step to zero.

7.4 Output of Analysis results

You may indicate the analysis results to be output via the OUTPUT commands. For the general, analysis type independent options for output of analysis results see § 3.6 on page 55.

syntax

```

BEGIN OUTPUT [devicew] [outoptw...] [params]
[ OFF ]
[ BEGIN SELECT
  [ ... ]
  [ STEPS ... ]
END SELECT ]
[ LAYOUT ... ]
  itemw ...
DISPLA
VELOCI
ACCELE

```

STRAIN
STRESS
STATUS
FORCE
NODFOR
ELMFOR
FSPRES
END OUTPUT

SELECT command block to customize the batch output.

... for model selection see § 3.6.2 on page 59, for stress- and strain transformation § 3.6.2.3 on page 60.

STEPS selects steps for output [§ 13.4.1 p. 248].

LAYOUT optional commands to specify the layout and style of tabular output [§ 3.6.4 p. 64].

item is the name of the analysis result to be output. See § 3.6.1 on page 56 for complete syntax of this command.

DISPLA for displacements [§ 7.4.1].

VELOCI for velocities [§ 7.4.2].

ACCELE for accelerations [§ 7.4.3].

STRAIN for strains [§ 7.4.4].

STRESS for stresses [§ 7.4.5].

STATUS for status [§ 7.4.6].

FORCE for nodal forces and moments [§ 7.4.7].

NODFOR for element nodal forces and moments [§ 7.4.8].

ELMFOR for internal element forces and moments [§ 7.4.9].

FSPRES for dynamic pressures of fluid–structure interface elements [§ 7.4.10].

7.4.1 Displacements

<i>syntax</i>				
DISPLA	[<i>type_w</i>]	[<i>form_w</i>]	[<i>oper_w</i>]	{ <i>comp_w</i> } { <i>opti_w</i> }
	TOTAL	TRANSL	LOCAL	RELATI
	INCREM	ROTATI	GLOBAL	...
	PHASE		NORM	

DISPLA specifies displacements of the nodes as output item.

[TOTAL] *type* specifies the displacement type.

TOTAL for the total displacements of a structure, i.e., the deformed geometry at a certain stage.

INCREM for the incremental displacements, i.e., the change in the deformed geometry between two consecutive steps.

PHASE for the phased displacements, i.e., the change in the deformed geometry between two consecutive phases in a phased analysis.

[TRANSL] *form* specifies the displacement formulation [§ 3.6.1 p. 57].

[GLOBAL] *oper* specifies an operation (transformation) to be performed on the displacements [§ 3.6.1 p. 57]. One specific operation is available:

NORM for the length of the displacement vector. Only the translational terms will be used to calculate the norm. This scalar result can be used for result scans over time steps by some output devices.

[all] *comp* selects displacement components for output. Default is all available components.

opti are additional options. The RELATI option gives the displacement results relative to an explicitly defined base node indicated by the BASNOD command [§ 3.6.2 p. 59] or to the defined base excitation [§ 6.3 p. 104] when no base node is defined explicitly. See § 3.6.1 on page 58 for other options.

Total displacements					<i>comp</i> ...		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	<i>opti</i>	X	Y	Z
DISPLA	TOTAL	TRANSL	LOCAL		TDtx	TDty	TDtz
					u_x	u_y	u_z
DISPLA	TOTAL	TRANSL	LOCAL	RELATI	TrDtx	TrDty	TrDtz
					u_{x_r}	u_{y_r}	u_{z_r}
DISPLA	TOTAL	TRANSL	GLOBAL		TDtX	TDtY	TDtZ
					u_X	u_Y	u_Z
DISPLA	TOTAL	TRANSL	GLOBAL	RELATI	TrDtX	TrDtY	TrDtZ
					u_{X_r}	u_{Y_r}	u_{Z_r}
DISPLA	TOTAL	ROTATI	LOCAL		TDrx	TDry	TDrz
					ϕ_x	ϕ_y	ϕ_z
DISPLA	TOTAL	ROTATI	LOCAL	RELATI	TrDrx	TrDry	TrDrz
					ϕ_{x_r}	ϕ_{y_r}	ϕ_{z_r}
DISPLA	TOTAL	ROTATI	GLOBAL		TDrX	TDrY	TDrZ
					ϕ_X	ϕ_Y	ϕ_Z
DISPLA	TOTAL	ROTATI	GLOBAL	RELATI	TrDrX	TrDrY	TrDrZ
					ϕ_{X_r}	ϕ_{Y_r}	ϕ_{Z_r}

Total displacements				
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	
DISPLA	TOTAL	TRANSL	NORM	TDXYZ
				$\ u\ $

Incremental displacements					<i>comp</i> ...		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	<i>opti</i>	X	Y	Z
DISPLA	INCREM	TRANSL	LOCAL		IDtx	IDty	IDtz
					Δu_x	Δu_y	Δu_z
DISPLA	INCREM	TRANSL	LOCAL	RELATI	IrDtx	IrDty	IrDtz
					Δu_{x_r}	Δu_{y_r}	Δu_{z_r}
DISPLA	INCREM	TRANSL	GLOBAL		IDtX	IDtY	IDtZ
					Δu_X	Δu_Y	Δu_Z
DISPLA	INCREM	TRANSL	GLOBAL	RELATI	IrDtX	IrDtY	IrDtZ
					Δu_{X_r}	Δu_{Y_r}	Δu_{Z_r}
DISPLA	INCREM	ROTATI	LOCAL		IDrx	IDry	IDrz
					$\Delta \phi_x$	$\Delta \phi_y$	$\Delta \phi_z$
DISPLA	INCREM	ROTATI	LOCAL	RELATI	IrDrx	IrDry	IrDrz
					$\Delta \phi_{x_r}$	$\Delta \phi_{y_r}$	$\Delta \phi_{z_r}$
DISPLA	INCREM	ROTATI	GLOBAL		IDrX	IDrY	IDrZ
					$\Delta \phi_X$	$\Delta \phi_Y$	$\Delta \phi_Z$
DISPLA	INCREM	ROTATI	GLOBAL	RELATI	IrDrX	IrDrY	IrDrZ
					$\Delta \phi_{X_r}$	$\Delta \phi_{Y_r}$	$\Delta \phi_{Z_r}$

Incremental displacements				
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	
DISPLA	INCREM	TRANSL	NORM	IDXYZ
				$\ u\ $

Phased displacements					<i>comp</i> . . .		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	<i>opti</i>	X	Y	Z
DISPLA	PHASE	TRANSL	LOCAL		PDtx	PDty	PDtz
					u_x	u_y	u_z
DISPLA	PHASE	TRANSL	LOCAL	RELATI	PrDtx	PrDty	PrDtz
					u_{x_r}	u_{y_r}	u_{z_r}
DISPLA	PHASE	TRANSL	GLOBAL		PDtX	PDtY	PDtZ
					u_X	u_Y	u_Z
DISPLA	PHASE	TRANSL	GLOBAL	RELATI	PrDtX	PrDtY	PrDtZ
					u_{X_r}	u_{Y_r}	u_{Z_r}
DISPLA	PHASE	ROTATI	LOCAL		PDrx	PDry	PDrz
					ϕ_x	ϕ_y	ϕ_z
DISPLA	PHASE	ROTATI	LOCAL	RELATI	PrDrx	PrDry	PrDrz
					ϕ_{x_r}	ϕ_{y_r}	ϕ_{z_r}
DISPLA	PHASE	ROTATI	GLOBAL		PDrX	PDrY	PDrZ
					ϕ_X	ϕ_Y	ϕ_Z
DISPLA	PHASE	ROTATI	GLOBAL	RELATI	PrDrX	PrDrY	PrDrZ
					ϕ_{X_r}	ϕ_{Y_r}	ϕ_{Z_r}

Phased displacements			
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>
DISPLA	PHASE	TRANSL	NORM
			PXYZ
			$\ u\ $

7.4.2 Velocities

syntax

VELOCI	[<i>type_w</i>]	[<i>form_w</i>]	[<i>oper_w</i>]	{ <i>comp_w</i> }	{ <i>opti_w</i> }
	TOTAL	TRANSL	LOCAL		RELATI
	PHASE	ROTATI	GLOBAL		. . .
			NORM		

VELOCI specifies the velocity of the nodes as output item.

[TOTAL] *type* specifies the velocity type [§ 3.6.1 p. 57].

TOTAL for the total velocities of a structure, i.e., the velocity at a certain time.

PHASE for the phased velocities, i.e., the change in the velocity of the nodes between two consecutive phases in a phased analysis.

[TRANSL] *form* specifies the velocity formulation [§ 3.6.1 p. 57].

[GLOBAL] *oper* specifies an operation (transformation) to be performed on the velocities [§ 3.6.1 p. 57]. One specific operation is available:

NORM for the length of the velocity vector. Only the translational terms will be used to calculate the norm. This scalar result can be used for result scans over time steps by some output devices.

[all] *comp* selects velocity components for output. Default is all available components.

opti are additional options. The RELATI option gives the velocity results relative to an explicitly defined base node indicated by the BASNOD command [§ 3.6.2 p. 59] or to the defined base excitation [§ 6.3 p. 104] when no base node is defined explicitly. See § 3.6.1 on page 58 for other options.

Total velocities					comp	...		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	<i>opti</i>	X	Y	Z	
VELOCI	TOTAL	TRANSL	LOCAL		TVtx	TVty	TVtz	
					\dot{u}_x	\dot{u}_y	\dot{u}_z	
VELOCI	TOTAL	TRANSL	LOCAL	RELATI	TrVtx	TrVty	TrVtz	
					\dot{u}_{x_r}	\dot{u}_{y_r}	\dot{u}_{z_r}	
VELOCI	TOTAL	TRANSL	GLOBAL		TVtX	TVtY	TVtZ	
					\dot{u}_X	\dot{u}_Y	\dot{u}_Z	
VELOCI	TOTAL	TRANSL	GLOBAL	RELATI	TrVtX	TrVtY	TrVtZ	
					\dot{u}_{X_r}	\dot{u}_{Y_r}	\dot{u}_{Z_r}	
VELOCI	TOTAL	ROTATI	LOCAL		TVrx	TVry	TVrz	
					$\dot{\phi}_x$	$\dot{\phi}_y$	$\dot{\phi}_z$	
VELOCI	TOTAL	ROTATI	LOCAL	RELATI	TrVrx	TrVry	TrVrz	
					$\dot{\phi}_{x_r}$	$\dot{\phi}_{y_r}$	$\dot{\phi}_{z_r}$	
VELOCI	TOTAL	ROTATI	GLOBAL		TVrX	TVrY	TVrZ	
					$\dot{\phi}_X$	$\dot{\phi}_Y$	$\dot{\phi}_Z$	
VELOCI	TOTAL	ROTATI	GLOBAL	RELATI	TrVrX	TrVrY	TrVrZ	
					$\dot{\phi}_{X_r}$	$\dot{\phi}_{Y_r}$	$\dot{\phi}_{Z_r}$	

Total velocities					
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>		
VELOCI	TOTAL	TRANSL	NORM	TVXYZ	
					$\ \dot{u}\ $

Phased velocities					comp	...		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	<i>opti</i>	X	Y	Z	
VELOCI	PHASE	TRANSL	LOCAL		PVtx	PVty	PVtz	
					\dot{u}_x	\dot{u}_y	\dot{u}_z	
VELOCI	PHASE	TRANSL	LOCAL	RELATI	PrVtx	PrVty	PrVtz	
					\dot{u}_{x_r}	\dot{u}_{y_r}	\dot{u}_{z_r}	
VELOCI	PHASE	TRANSL	GLOBAL		PVtX	PVtY	PVtZ	
					\dot{u}_X	\dot{u}_Y	\dot{u}_Z	
VELOCI	PHASE	TRANSL	GLOBAL	RELATI	PrVtX	PrVtY	PrVtZ	
					\dot{u}_{X_r}	\dot{u}_{Y_r}	\dot{u}_{Z_r}	
VELOCI	PHASE	ROTATI	LOCAL		PVrx	PVry	PVrz	
					$\dot{\phi}_x$	$\dot{\phi}_y$	$\dot{\phi}_z$	
VELOCI	PHASE	ROTATI	LOCAL	RELATI	PrVrx	PrVry	PrVrz	
					$\dot{\phi}_{x_r}$	$\dot{\phi}_{y_r}$	$\dot{\phi}_{z_r}$	
VELOCI	PHASE	ROTATI	GLOBAL		PVrX	PVrY	PVrZ	
					$\dot{\phi}_X$	$\dot{\phi}_Y$	$\dot{\phi}_Z$	
VELOCI	PHASE	ROTATI	GLOBAL	RELATI	PrVrX	PrVrY	PrVrZ	
					$\dot{\phi}_{X_r}$	$\dot{\phi}_{Y_r}$	$\dot{\phi}_{Z_r}$	

Phased velocities					
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>		
VELOCI	PHASE	TRANSL	NORM	PVXYZ	
					$\ \dot{u}\ $

7.4.3 Accelerations

syntax

ACCELE	[<i>type_w</i>]	[<i>form_w</i>]	[<i>oper_w</i>]	{ <i>comp_w</i> }	{ <i>opti_w</i> }
	TOTAL	TRANSL	LOCAL		RELATI
	PHASE	ROTATI	GLOBAL		
			NORM		

ACCELE specifies the acceleration to be output in the nodes.

type specifies the acceleration type [§ 3.6.1 p. 57].

[TOTAL]

TOTAL for the total accelerations of a structure, i.e., the acceleration at a certain time.

PHASE for the phased accelerations, i.e., the change in the acceleration of the nodes between two consecutive phases in a phased analysis.

[TRANSL] *form* specifies the acceleration formulation [§ 3.6.1 p. 57].

[GLOBAL] *oper* specifies an operation (transformation) to be performed on the accelerations [§ 3.6.1 p. 57]. One specific operation is available:

NORM for the length of the acceleration vector. Only the translational terms will be used to calculate the norm. This scalar result can be used for result scans over time steps by some output devices.

[all] *comp* selects acceleration components for output. Default is all available components.

opti are additional options. The RELATI option gives the acceleration results relative to an explicitly defined base node indicated by the BASNOD command [§ 3.6.2 p. 59] or to the defined base excitation [§ 6.3 p. 104] when no base node is defined explicitly. See § 3.6.1 on page 58 for other options.

opti are additional options [§ 3.6.1 p. 58].

Total accelerations					<i>comp</i> ...		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	<i>opti</i>	X	Y	Z
ACCELE	TOTAL	TRANSL	LOCAL		TAtx \ddot{u}_x	TAty \ddot{u}_y	TAtz \ddot{u}_z
ACCELE	TOTAL	TRANSL	LOCAL	RELATI	TrAtx \ddot{u}_{x_r}	TrAty \ddot{u}_{y_r}	TrAtz \ddot{u}_{z_r}
ACCELE	TOTAL	TRANSL	GLOBAL		TAtX \ddot{u}_X	TAtY \ddot{u}_Y	TAtZ \ddot{u}_Z
ACCELE	TOTAL	TRANSL	GLOBAL	RELATI	TrAtX \ddot{u}_{X_r}	TrAtY \ddot{u}_{Y_r}	TrAtZ \ddot{u}_{Z_r}
ACCELE	TOTAL	ROTATI	LOCAL		TArx $\ddot{\phi}_x$	TArY $\ddot{\phi}_y$	TArz $\ddot{\phi}_z$
ACCELE	TOTAL	ROTATI	LOCAL	RELATI	TrArx $\ddot{\phi}_{x_r}$	TrArY $\ddot{\phi}_{y_r}$	TrArz $\ddot{\phi}_{z_r}$
ACCELE	TOTAL	ROTATI	GLOBAL		TArX $\ddot{\phi}_X$	TArY $\ddot{\phi}_Y$	TArZ $\ddot{\phi}_Z$
ACCELE	TOTAL	ROTATI	GLOBAL	RELATI	TrArX $\ddot{\phi}_{X_r}$	TrArY $\ddot{\phi}_{Y_r}$	TrArZ $\ddot{\phi}_{Z_r}$

Total accelerations				
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	
ACCELE	TOTAL	TRANSL	NORM	TAXYZ $\ \ddot{u}\ $

Phased accelerations					<i>comp</i> ...		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	<i>opti</i>	X	Y	Z
ACCELE	PHASE	TRANSL	LOCAL		PAtx \ddot{u}_x	PAty \ddot{u}_y	PAtz \ddot{u}_z
ACCELE	PHASE	TRANSL	LOCAL	RELATI	PrAtx \ddot{u}_{x_r}	PrAty \ddot{u}_{y_r}	PrAtz \ddot{u}_{z_r}
ACCELE	PHASE	TRANSL	GLOBAL		PAtX \ddot{u}_X	PAtY \ddot{u}_Y	PAtZ \ddot{u}_Z
ACCELE	PHASE	TRANSL	GLOBAL	RELATI	PrAtX \ddot{u}_{X_r}	PrAtY \ddot{u}_{Y_r}	PrAtZ \ddot{u}_{Z_r}
ACCELE	PHASE	ROTATI	LOCAL		PArx $\ddot{\phi}_x$	PArY $\ddot{\phi}_y$	PArz $\ddot{\phi}_z$
ACCELE	PHASE	ROTATI	LOCAL	RELATI	PrArx $\ddot{\phi}_{x_r}$	PrArY $\ddot{\phi}_{y_r}$	PrArz $\ddot{\phi}_{z_r}$
ACCELE	PHASE	ROTATI	GLOBAL		PArX $\ddot{\phi}_X$	PArY $\ddot{\phi}_Y$	PArZ $\ddot{\phi}_Z$
ACCELE	PHASE	ROTATI	GLOBAL	RELATI	PrArX $\ddot{\phi}_{X_r}$	PrArY $\ddot{\phi}_{Y_r}$	PrArZ $\ddot{\phi}_{Z_r}$

Phased accelerations			
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>
ACCELE	PHASE	TRANSL	NORM
			PXYZ $\ \ddot{u}\ $

7.4.4 Strains

syntax

STRAIN	[<i>type_w</i>]	[<i>form_w</i>]	[<i>oper_w</i>]	{ <i>comp_w</i> }	{ <i>loca_w</i> }	{ <i>opti_w</i> }
	TOTAL	GREEN	LOCAL		INTPNT	
		FORCE	GLOBAL		NODES	
		MOMENT	PRINCI		CENTER	
		DISFOR	VONMIS			
		DISMOM	REAXES			
		TRACTI	VOLUME			

STRAIN specifies strains as output item. Table 7.1 on the next page outlines the availability and applicability of the various strain output options for each of the element families.

type specifies the strain type [§ 3.6.1 p. 57].

[TOTAL]

form specifies the strain formulation.

[GREEN]

GREEN for Green–Lagrange strains [§ 7.4.4.1].

FORCE for deformations due to normal and shear forces [§ 7.4.4.2].

DISFOR for generalized strains [§ 7.4.4.2].

MOMENT for curvatures due to concentrated bending moments [§ 7.4.4.3].

DISMOM for curvatures due to distributed bending moments [§ 7.4.4.3].

TRACTI for tractions in structural interface elements [§ 7.4.4.4].

oper specifies an operation (transformation) to be performed on the primary strains [§ 3.6.1 p. 57].

[LOCAL]

comp selects strain components for output. Default is all available components.

[all]

loca specifies the location for the strains to be output [§ 3.6.1 p. 58].

opti are additional options [§ 3.6.1 p. 58].

Table 7.1: AVAILABILITY OF STRAIN OUTPUT FOR LINEAR TRANSIENT ANALYSIS

<i>item</i>	STRAIN	truss	beam	pl. stress	pl. strain	axisymm.	plate bend.	fl. shell	cu. shell	solid	str. interface	spring	p. mass	e. reinfo.
<i>type</i>	TOTAL	a	a	a	a	a	a	a	a	a	a	-	a	
<i>form</i>	GREEN	a	a	a	a	-	a	a	a	-	-	-	a	
	FORCE	-	b	-	-	-	-	-	-	-	-	a	-	c
	MOMENT	-	b	-	-	-	-	-	-	-	-	a	-	d
	DISFOR	-	-	-	-	-	a	a	-	-	-	-	-	-
	DISMOM	-	-	-	-	-	a	a	-	-	-	-	-	-
	TRACTI	-	-	-	-	-	-	-	-	a	-	-	-	c
<i>oper</i>	LOCAL	a	a	a	a	a	a	a	a	a	a	-	a	
	GLOBAL	a	a	a	a	a	a	a	a	a	a	-	c	
	PRINCI	a	a	a	a	a	a	a	a	-	-	-	-	
	VONMIS	a	a	a	a	a	a	a	a	-	-	-	-	
	REAXES	-	-	a	-	-	a	a	a	-	-	-	-	
	VOLUME	a	a	a	a	a	a	a	a	-	-	-	-	
<i>loca</i>	INTPNT	a	f	a	a	a	a	a	a	a	-	-	e	
	NODES	a	a	a	a	a	a	a	a	-	a	-	c	
	CENTER	a	a	a	a	a	a	a	a	a	-	-	-	

(a) All elements. (b) For all beam elements, class-II and class-III only in combination with local coordinate system. (c) Only for bond-slip reinforcements. (d) Only for bond-slip reinforcements modeled by beam elements. (e) Not for bond-slip reinforcements. (f) Only for class-II and class-III beam elements. (-) Not available or not suitable.

7.4.4.1 Green-Lagrange Strains

Primary strains [§ 47.1]				<i>comp</i> ...									
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	XX	YY	ZZ	XY	YZ	ZX	1	2	3	
STRAIN	TOTAL	GREEN	LOCAL	E _{xx}	E _{yy}	E _{zz}	G _{xy}	G _{yz}	G _{zx}				
				ε_{xx}	ε_{yy}	ε_{zz}	γ_{xy}	γ_{yz}	γ_{zx}				
STRAIN	TOTAL	GREEN	GLOBAL	EXX	EYY	EZZ	GXY	GYZ	GZX				
				ε_{XX}	ε_{YY}	ε_{ZZ}	γ_{XY}	γ_{YZ}	γ_{ZX}				
STRAIN	TOTAL	GREEN	PRINCI							E1	E2	E3	
										ε_1	ε_2	ε_3	
STRAIN	TOTAL	GREEN	REAXES							E1RA	E2RA		
										ε_1^a	ε_2^a		

Von Mises strain [§ 47.1.1]				
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	
STRAIN	TOTAL	GREEN	VONMIS	Eeq
				ε_{eq}

Volumetric strain [§ 47.1.3]				
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	
STRAIN	TOTAL	GREEN	VOLUME	Evol
				ε_{vol}

7.4.4.2 Deformations

Force deformations				<i>comp</i> ...		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	X	Y	Z
STRAIN	TOTAL	FORCE	LOCAL	P _x		
				Δu_x		
STRAIN	TOTAL	FORCE	GLOBAL	PX	PY	PZ
				Δu_X	Δu_Y	Δu_Z

Generalized strains				<i>comp</i> ...						
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	XX	YY	ZZ	XY	YZ	ZX	1 2
STRAIN TOTAL	DISFOR	LOCAL		Pxx	Pyx	Pzz	Pxy	Pyz	Pzx	
				Ψ_{xx}	Ψ_{yy}	Ψ_{zz}	Ψ_{xy}	Ψ_{yz}	Ψ_{zx}	
STRAIN TOTAL	DISFOR	REAXES								P1RA P2RA
										Ψ_1^a Ψ_2^a

7.4.4.3 Curvatures

Concentrated curvatures				<i>comp</i> ...		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	X	Y	Z
STRAIN TOTAL	MOMENT	LOCAL		Kx	Ky	Kz
				κ_x	κ_y	κ_z

Distributed curvatures				<i>comp</i> ...						
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	XX	YY	ZZ	XY	1	2	
STRAIN TOTAL	DISMOM	LOCAL		Kxx	Kyy	Kzz	Kxy			
				κ_{xx}	κ_{yy}	κ_{zz}	κ_{xy}			
STRAIN TOTAL	DISMOM	REAXES						K1RA	K2RA	
								κ_1^a	κ_2^a	

7.4.4.4 Relative Displacements of Interface Elements

Note that the labels for the results in the local *xyz* directions are dependent on the interface element type [Vol. Element Library]. Output items related to the normal direction(s) are indicated with ‘N’ and items related to the shear direction(s) are indicated with ‘S’ in their name followed by the letter indicating the corresponding element direction.

Relative displacements				<i>comp</i> ...		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	X	Y	Z
STRAIN TOTAL	TRACTI	LOCAL		DUNx	DUNy	DUNz
				DUSx	DUSy	DUSz
				Δu_x	Δu_y	Δu_z
STRAIN TOTAL	TRACTI	GLOBAL		DUX	DUY	DUZ
				Δu_X	Δu_Y	Δu_Z

7.4.5 Stresses

syntax

STRESS	[<i>type_w</i>]	[<i>form_w</i>]	[<i>oper_w</i>]	{ <i>comp_w</i> }	{ <i>loca_w</i> }	{ <i>opti_w</i> }
	TOTAL	CAUCHY	LOCAL		INTPNT	
		FORCE	GLOBAL		NODES	
		MOMENT	PRINCI		CENTER	
		DISFOR	VONMIS			
		DISMOM	INVARI			
		TRACTI	REINFO			
		GRADIE	REAXES			
		SHEAR	MAXSHR			
			BIAXFE			
			MOHRCO			
			HOEKBR			

STRESS specifies stresses as output item. Table 7.2 on the following page outlines the availability and applicability of the various stress output options for each of the element families.

Table 7.2: AVAILABILITY OF STRESS OUTPUT FOR LINEAR TRANSIENT ANALYSIS

<i>item</i>	STRESS	truss	beam	pl. stress	pl. strain	axisymm.	plate bend.	fl. shell	cu. shell	solid	str. interface	spring	p. mass	e. reinfo.	comp. line	comp. surf.
<i>type</i>	TOTAL	a	a	a	a	a	a	a	a	a	a	-	a	a	a	
<i>form</i>	CAUCHY	a	a	a	a	a	a	a	a	-	-	-	a	-	-	
	FORCE	a	a	-	-	-	-	-	-	-	a	-	a	a	-	
	MOMENT	-	a	-	-	-	-	-	-	-	a	-	c	a	-	
	DISFOR	-	-	a	g	h	a	a	a	-	-	-	-	-	a	
	DISMOM	-	-	-	g	h	a	a	a	-	-	-	-	-	a	
	TRACTI	-	-	-	-	-	-	-	-	a	-	-	b	-	-	
	GRADIE	-	-	-	-	-	-	-	-	-	-	-	e	-	-	
	SHEAR	-	-	-	-	-	-	-	-	-	-	-	e	-	-	
<i>oper</i>	LOCAL	a	a	a	a	a	a	a	a	a	a	-	a	a	a	
	GLOBAL	a	a	a	a	a	a	a	a	a	-	-	b	-	-	
	PRINCI	a	a	a	a	a	a	a	a	-	-	-	-	-	-	
	VONMIS	a	a	a	a	a	a	a	a	-	-	-	-	-	-	
	INVARI	-	-	a	a	a	-	-	a	-	-	-	-	-	-	
	REINFO	-	-	a	-	-	a	a	a	-	-	-	-	-	-	
	REAXES	-	-	a	-	-	a	a	a	-	-	-	-	-	-	
	MAXSHR	-	-	a	a	a	-	-	a	-	-	-	-	-	-	
	BIAXFE	a	a	a	a	a	-	-	a	-	-	-	-	-	-	
	MOHRCO	-	-	a	a	-	-	-	a	-	-	-	-	-	-	
<i>loca</i>	HOEKER	-	-	a	a	-	-	-	a	-	-	-	-	-	-	
	INTPNT	a	f	a	a	a	a	a	a	a	-	-	d	-	-	
	NODES	a	a	a	a	a	a	a	a	-	-	-	b	a	a	
	CENTER	a	a	a	a	a	a	a	a	a	-	-	-	-	-	

(a) All elements. (b) Only for bond-slip reinforcements. (c) Only for bond-slip reinforcements modeled by beam elements. (d) Not for bond-slip reinforcements. (e) Only for bar reinforcements. (f) Only for class-II and class-III beam elements. (g) For infinite shells only. (h) For shells of revolution only. (i) Only for bar reinforcements and bond-slip reinforcements. (-) Not available or not suitable.

type specifies the stress type [§ 3.6.1 p. 57]. [TOTAL]

[CAUCHY] *form* specifies the stress formulation.

CAUCHY for Cauchy stresses [§ 7.4.5.1].

FORCE for concentrated forces [§ 7.4.5.2].

MOMENT for concentrated bending moments [§ 7.4.5.3].

DISFOR for distributed forces [§ 7.4.5.2].

DISMOM for distributed moments [§ 7.4.5.3].

TRACTI for tractions in structural interface elements [§ 7.4.5.4].

GRADIE for gradients of stresses in reinforcement bars [§ 4.2.4.5 p. 87].

SHEAR for shear stress in the reinforcement mother element connection [§ 4.2.4.6 p. 87].

[LOCAL] *oper* specifies an operation (transformation) to be performed on the primary stresses [§ 3.6.1 p. 57]. A specific operation is available for the maximum shear stress:

MAXSHR gives the maximum shear stress $\tau_{\max} = \frac{\sigma_1 - \sigma_3}{2}$ with σ_1 and σ_3 the highest and lowest principal stress respectively. For plane stress elements the maximum shear stress τ_{\max} is defined as $\tau_{\max} = \frac{\sigma_1 - \sigma_2}{2}$ because the third principal stress σ_3 is zero by definition.

A specific operation is available for Safety Factors for concrete under static and dynamic loading conditions with reference to a biaxial failure envelope:

BIAXFE gives the following Safety Factors for concrete under static and dynamic loading conditions with reference to a biaxial failure envelope:

- $FS_{static\ usual} = R_{static\ usual}/r$
- $FS_{static\ unusual} = R_{static\ unusual}/r$
- $FS_{dynamic\ unusual} = R_{dynamic\ unusual}/r$
- $FS_{dynamic\ extreme} = R_{dynamic\ extreme}/r$

Where r is the distance from the origin to the actual stress point (σ_1, σ_2) :

$$r = \sqrt{(\sigma_1^2 + \sigma_2^2)} \quad (7.1)$$

To determine the Safety Factors DIANA needs additional material properties. For the required parameters and additional information on the Safety Factors, see *Volume Material Library*.

Note that all Safety Factors are limited to 100.

A specific operation is available for shear capacity of stress against Mohr–Coulomb failure criterion:

MOHRCO gives the shear capacity of stress against Mohr–Coulomb failure criterion [§ 47.2.9.1 p. 582]. To determine the shear capacity DIANA needs additional material properties, see *Volume Material Library*.

A specific operation is available for shear capacity of stress against Hoek–Brown failure criterion:

HOEKBR gives the shear capacity of stress against Hoek–Brown failure criterion [§ 47.2.9.2 p. 582]. To determine the shear capacity DIANA needs the unconfined compressive strength σ_{ci} of the (intact) rock sample (COMSTR), Hoek–Brown constant m_b (HOEKMB), and the Hoek–Brown constant s (HOEKS) material properties of the Hoek–Brown rock plasticity material model [Vol. *Material Library*].

comp selects strain components for output. Default is all available components. [all]

loca specifies the location for the strains to be output [§ 3.6.1 p. 58].

opti are additional options [§ 3.6.1 p. 58].

7.4.5.1 Cauchy Stresses

Primary stresses [§ 47.2]				comp ...								
item	type	form	oper	XX	YY	ZZ	XY	YZ	ZX	1	2	3
STRESS TOTAL	CAUCHY	LOCAL		Sxx	Syy	Szz	Sxy	Syz	Szx			
				σ_{xx}	σ_{yy}	σ_{zz}	σ_{xy}	σ_{yz}	σ_{zx}			
STRESS TOTAL	CAUCHY	GLOBAL		SXX	SYX	SZZ	SXY	SYZ	SZX			
				σ_{XX}	σ_{YY}	σ_{ZZ}	σ_{XY}	σ_{YZ}	σ_{ZX}			
STRESS TOTAL	CAUCHY	PRINCI								S1	S2	S3
										σ_1	σ_2	σ_3
STRESS TOTAL	CAUCHY	REAXES								S1RA	S2RA	
										σ_1^a	σ_2^a	

Von Mises stress [§ 47.2.1]			
item	type	form	oper
STRESS TOTAL	CAUCHY	VONMIS	Seq
			σ_{eq}

Stress invariants [§ 47.2.5]				comp ...		
item	type	form	oper	P	Q	LODE
STRESS TOTAL	CAUCHY	INVARI		P	Q	Lode
				p'	q	θ

Maximum shear stress			
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>
STRESS TOTAL	CAUCHY	MAXSHR	Tmax τ_{\max}
STRESS INITIA	CAUCHY	MAXSHR	T0max τ_{\max}^0

Biaxial failure envelope				<i>comp</i> ...			
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	STUS	STUN	DYUN	DYEX
STRESS TOTAL	CAUCHY	BIAXFE	FSstus	FSstun	FSdyun	FSdyex	
			FS_{stus}	FS_{stun}	FS_{dyun}	FS_{dyex}	

Shear capacity [§ 47.2.9]				<i>comp</i> ...			
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	SHRCAP			
STRESS TOTAL	CAUCHY	MOHRCO	SHRCAP				
			ψ				
STRESS TOTAL	CAUCHY	HOEKBR	SHRCAP				
			ψ				

7.4.5.2 Forces

Concentrated forces				<i>comp</i> ...		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	X	Y	Z
STRESS TOTAL	FORCE	LOCAL	Nx	Qy	Qz	
			N_x	Q_y	Q_z	
STRESS TOTAL	FORCE	GLOBAL	NX	NY	NZ	
			N_X	N_Y	N_Z	

Distributed forces				<i>comp</i> ...							
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	XX	YY	ZZ	XY	YZ	ZX	1	2
STRESS TOTAL	DISFOR	LOCAL	Nxx	Nyy	Nzz	Nxy	Qyz	Qxz			
			n_{xx}	n_{yy}	n_{zz}	n_{xy}	q_{yz}	q_{xz}			
STRESS TOTAL	DISFOR	REAXES								N1RA	N2RA
										n_1^a	n_2^a

Reinforcement forces [§ 47.2.6]				<i>comp</i> ...			
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	1R	2R	1RC	2RC
STRESS TOTAL	DISFOR	REINFO	N1R	N2R	N1RC	N2RC	
			n_1'	n_2'	$n_1'^c$	$n_2'^c$	

7.4.5.3 Bending Moments

Concentrated moments				<i>comp</i> ...		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	X	Y	Z
STRESS TOTAL	MOMENT	LOCAL	Mx	My	Mz	
			M_x	M_y	M_z	
STRESS TOTAL	MOMENT	GLOBAL	MX	MY	MZ	
			M_X	M_Y	M_Z	

Distributed moments				comp ...						
item	type	form	oper	XX	YY	ZZ	XY	1	2	
STRESS	TOTAL	DISMOM	LOCAL	Mxx	Myy	Mzz	Mxy			
				m_{xx}	m_{yy}	m_{zz}	m_{xy}			
STRESS	TOTAL	DISMOM	REAXES					M1RA	M2RA	
								m_1^a	m_2^a	

Reinforcement moments [§ 47.2.6]				<i>comp</i> ...	
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	1R	2R
STRESS TOTAL	DISMOM	REINFO	M1R	M2R	
			m_1'	m_2'	

7.4.5.4 Traction

Note that the labels for the results in the local xyz directions are dependent on the interface element type [Vol. Element Library]. Output items related to the normal direction(s) are indicated with ‘N’ and items related to the shear direction(s) are indicated with ‘S’ in their name followed by the letter indicating the corresponding element direction.

Traction				comp ...		
item	type	form	oper	X	Y	Z
STRESS TOTAL	TRACTI	LOCAL		STNx	STNy	STNz
				STSx	STSy	STSz
				t_x	t_y	t_z
STRESS TOTAL	TRACTI	GLOBAL		STX	STY	STZ
				t_X	t_Y	t_Z

7.4.6 Status

The post-analysis result **STATUS** is used in DIANA to output the internal state parameters of the material model at the integration points of an element, for instance the specific kinetic or gravitational energy. For each internal state parameter, a syntax will be presented in the following.

7.4.6.1 Specific Kinetic Energy

For transient dynamic analyses the actual kinetic energy per volume can be calculated and output. The kinetic energy per volume is defined by the following equation:

$$W_k = \rho \cdot v \cdot v \quad (7.2)$$

where ρ is the density and v is the norm of the total velocity, respectively.

				syntax
STATUS	[\underline{type}_w]	{ \underline{form}_w }	{ \underline{loca}_w }	{ \underline{opti}_w }
	ENERGY	KINETI	INTPNT	

ENERGY gives output of energy.

form specifies the energy formulation. **KINETI** gives specific kinetic energy.

loca specifies the location for the status to be output [§ 3.6.1 p. 58]. For the specific kinetic energy only output at the integration points is possible.

opti are additional options [§ 3.6.1 p. 58].

Specific kinetic energy			
item	type	form	
STATUS	ENERGY	KINETI	WK
			W_k

7.4.6.2 Specific Gravitational Energy

For transient dynamic analyses the field energy per volume related to a particle moving in a gravitational force field can be calculated and output. The gravitational energy per volume is defined by the following equation:

$$W_g = \rho g u \quad (7.3)$$

where ρ is the density, g is the gravity vector, and u is the total displacement, respectively.

syntax

```
STATUS [ typew ] { formw } { locaw } { optiw }
      ENERGY  GRAVIT  INTPNT
```

ENERGY gives output of energy.

form specifies the energy formulation. GRAVIT gives specific gravitational energy.

[INTPNT] *loca* specifies the location for the status to be output [§ 3.6.1 p. 58]. For the specific gravitational energy only output at the integration points is possible.

opti are additional options [§ 3.6.1 p. 58].

Specific gravitational energy			
<i>item</i>	<i>type</i>	<i>form</i>	
STATUS	ENERGY	GRAVIT	WG W _g

7.4.7 Nodal Forces and Moments

syntax

```
FORCE [ typew ] [ formw ] [ operw ] { compw } { optiw }
      TOTAL  TRANSL  LOCAL
      RESIDU  ROTATI  GLOBAL
      REACTI
```

FORCE specifies forces (and moments) in the nodes as output item [§ 47.3 p. 583].

[TOTAL] *type* specifies the type [§ 3.6.1 p. 57].

TOTAL for the total forces and moments.

RESIDU for the residual forces.

REACTI for the reaction forces in all supported nodes.

[TRANSL] *form* specifies the formulation [§ 3.6.1 p. 57]

[GLOBAL] *oper* specifies an operation (transformation) to be performed on the forces and moments [§ 3.6.1 p. 57].

[all] *comp* selects force or moment components for output. Default is all available components.

opti are additional options [§ 3.6.1 p. 58].

Total forces and moments				<i>comp</i> ...		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	X	Y	Z
FORCE	TOTAL	TRANSL	LOCAL	FTx F _x	FTy F _y	FTz F _z
FORCE	TOTAL	TRANSL	GLOBAL	FTX F _X	FTY F _Y	FTZ F _Z
FORCE	TOTAL	ROTATI	LOCAL	MTx M _x	MTy M _y	MTz M _z
FORCE	TOTAL	ROTATI	GLOBAL	MTX M _X	MTY M _Y	MTZ M _Z

Residuals [§ 47.3.3]				<i>comp</i> ...		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	X	Y	Z
FORCE	RESIDU	TRANSL	LOCAL	FRx F_x^R	FRx F_y^R	FRx F_z^R
FORCE	RESIDU	TRANSL	GLOBAL	FRX F_X^R	FRX F_Y^R	FRX F_Z^R
FORCE	RESIDU	ROTATI	LOCAL	MRx M_x^R	MRy M_y^R	MRz M_z^R
FORCE	RESIDU	ROTATI	GLOBAL	MRX M_X^R	MRY M_Y^R	MRZ M_Z^R

Support reactions [§ 47.3.3]				<i>comp</i> ...		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	X	Y	Z
FORCE	REACTI	TRANSL	LOCAL	FBx F_x^B	FBY F_y^B	FBz F_z^B
FORCE	REACTI	TRANSL	GLOBAL	FBX F_X^B	FBY F_Y^B	FBZ F_Z^B
FORCE	REACTI	ROTATI	LOCAL	MBx M_x^B	MBY M_y^B	MBz M_z^B
FORCE	REACTI	ROTATI	GLOBAL	MBX M_X^B	MBY M_Y^B	MBZ M_Z^B

7.4.8 Nodal Element Forces

syntax

NODFOR [*type_w*] [*form_w*] [*oper_w*] { *comp_w* } { *opti_w* }

ELEMEN TRANSL LOCAL

REINFO ROTATI GLOBAL

TOTAL

DAMPIN

INERTI

NODFOR specifies internal nodal element forces and moments, nodal element damping forces and moments, and nodal element inertia forces and moments in the nodes as output item. This command gives the contributions of the element or reinforcement internal nodal forces and moments (or both), element damping forces and moments, or element inertia forces and moments to a certain node. A selection of elements which form a ‘section’ of the model gives the total internal forces and moments that act on that ‘section’.

type specifies the forces type.

[TOTAL]

ELEMEN for the internal nodal force and moment contribution of elements only.

REINFO for the internal nodal force and moment contribution of embedded reinforcements only.

TOTAL for the internal nodal force and moment contribution of both elements and embedded reinforcements.

DAMPIN for nodal element damping forces and moments.

INERTI for nodal element inertia forces and moments.

form specifies the formulation [§ 3.6.1 p. 57].

[TRANSL]

oper specifies an operation (transformation) to be performed on the forces and moments [§ 3.6.1 p. 57].

[GLOBAL]

comp selects force or moment components for output. Default is all available components.

[all]

opti are additional options [§ 3.6.1 p. 58].

Internal nodal total forces [§ 47.3.1]				<i>comp</i> ...		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	X	Y	Z
NODFOR	TOTAL	TRANSL	LOCAL	FNTx F_x^{nt}	FNTy F_y^{nt}	FNTz F_z^{nt}
NODFOR	TOTAL	TRANSL	GLOBAL	FNTX F_X^{nt}	FNTY F_Y^{nt}	FNTZ F_Z^{nt}
NODFOR	TOTAL	ROTATI	LOCAL	MNTx M_x^{nt}	MNTy M_y^{nt}	MNTz M_z^{nt}
NODFOR	TOTAL	ROTATI	GLOBAL	MNTX M_X^{nt}	MNTY M_Y^{nt}	MNTZ M_Z^{nt}

Internal nodal element forces [§ 47.3.1]				<i>comp</i> ...		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	X	Y	Z
NODFOR	ELEMEN	TRANSL	LOCAL	FNEx F_x^{ne}	FNEy F_y^{ne}	FNEz F_z^{ne}
NODFOR	ELEMEN	TRANSL	GLOBAL	FNEX F_X^{ne}	FNEY F_Y^{ne}	FNEZ F_Z^{ne}
NODFOR	ELEMEN	ROTATI	LOCAL	MNEx M_x^{ne}	MNEy M_y^{ne}	MNEz M_z^{ne}
NODFOR	ELEMEN	ROTATI	GLOBAL	MNEX M_X^{ne}	MNEY M_Y^{ne}	MNEZ M_Z^{ne}

Internal nodal reinforcement forces [§ 47.3.1]				<i>comp</i> ...		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	X	Y	Z
NODFOR	REINFO	TRANSL	LOCAL	FNRx F_x^{nr}	FNRy F_y^{nr}	FNRz F_z^{nr}
NODFOR	REINFO	TRANSL	GLOBAL	FNRX F_X^{nr}	FNRY F_Y^{nr}	FNRZ F_Z^{nr}
NODFOR	REINFO	ROTATI	LOCAL	MNRx M_x^{nr}	MNRy M_y^{nr}	MNRz M_z^{nr}
NODFOR	REINFO	ROTATI	GLOBAL	MNRX M_X^{nr}	MNRY M_Y^{nr}	MNRZ M_Z^{nr}

Nodal element damping forces				<i>comp</i> ...		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	X	Y	Z
NODFOR	DAMPIN	TRANSL	LOCAL	FDx F_x^d	FDy F_y^d	FDz F_z^d
NODFOR	DAMPIN	TRANSL	GLOBAL	FDX F_X^d	FDY F_Y^d	FDZ F_Z^d
NODFOR	DAMPIN	ROTATI	LOCAL	MDx M_x^d	MDy M_y^d	MDz M_z^d
NODFOR	DAMPIN	ROTATI	GLOBAL	MDX M_X^d	MDY M_Y^d	MDZ M_Z^d

Nodal element inertia forces				<i>comp</i> ...		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	X	Y	Z
NODFOR	INERTI	TRANSL	LOCAL	FMx F_x^m	FMY F_y^m	FMZ F_z^m
NODFOR	INERTI	TRANSL	GLOBAL	FMX F_X^m	FMY F_Y^m	FMZ F_Z^m
NODFOR	INERTI	ROTATI	LOCAL	MMx M_x^m	MMY M_y^m	MMZ M_z^m
NODFOR	INERTI	ROTATI	GLOBAL	MMX M_X^m	MMY M_Y^m	MMZ M_Z^m

7.4.9 Element Forces

syntax

```
ELMFOR [ typew ] [ formw ] [ operw ] { compw } { optiw }
      ELEMEN  TRANSL  GLOBAL
      REINFO  ROTATI
      TOTAL
      DAMPIN
      INERTI
```

ELMFOR specifies internal element forces and moments, element damping forces and moments, or element inertia forces and moments in the nodes of an element as output item. This command gives the contributions of the element or reinforcement internal forces and moments (or both), element damping forces and moments, or element inertia forces and moments to a certain node of an element.

type specifies the forces type.

[TOTAL]

ELEMEN for internal force and moment contribution of elements only.

REINFO for internal force and moment contribution of embedded reinforcements only.

TOTAL for internal force and moment contribution of both elements and embedded reinforcements.

DAMPIN for element damping forces and moments.

INERTI for element inertia forces and moments.

form specifies the formulation [§ 3.6.1 p. 57].

[TRANSL]

oper specifies an operation (transformation) to be performed on the forces and moments [§ 3.6.1 p. 57].

[GLOBAL]

comp selects force or moment components for output. Default is all available components.

[all]

opti are additional options [§ 3.6.1 p. 58].

Internal element forces				<i>comp</i> ...		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	X	Y	Z
ELMFOR TOTAL	TRANSL	GLOBAL		FETX F_X^{et}	FETY F_Y^{et}	FETZ F_Z^{et}
ELMFOR TOTAL	ROTATI	GLOBAL		METX M_X^{et}	METY M_Y^{et}	METZ M_Z^{et}
ELMFOR ELEMEN	TRANSL	GLOBAL		FEEX F_X^{ee}	FEEY F_Y^{ee}	FEEZ F_Z^{ee}
ELMFOR ELEMEN	ROTATI	GLOBAL		MEEX M_X^{ee}	MEEY M_Y^{ee}	MEEZ M_Z^{ee}
ELMFOR REINFO	TRANSL	GLOBAL		FERX F_X^{er}	FERY F_Y^{er}	FERZ F_Z^{er}
ELMFOR REINFO	ROTATI	GLOBAL		MERX M_X^{er}	MERY M_Y^{er}	MERZ M_Z^{er}

Element damping forces				<i>comp</i> ...		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	X	Y	Z
ELMFOR DAMPIN	TRANSL	GLOBAL		FEDX F_X^{ed}	FEDY F_Y^{ed}	FEDZ F_Z^{ed}
ELMFOR DAMPIN	ROTATI	GLOBAL		MEDX M_X^{ed}	MEDY M_Y^{ed}	MEDZ M_Z^{ed}

Element inertia forces				<i>comp</i> ...		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	X	Y	Z
ELMFOR INERTI	TRANSL	GLOBAL		FEMX F_X^{em}	FEMY F_Y^{em}	FEMZ F_Z^{em}
ELMFOR INERTI	ROTATI	GLOBAL		MEMX M_X^{em}	MEMY M_Y^{em}	MEMZ M_Z^{em}

7.4.10 Dynamic Pressures

For fluid–structure interface elements [Vol. *Element Library*] DIANA can calculate and output the dynamic pressures. Positive dynamic pressures indicate additional pressure on the structure, while negative dynamic pressures, i.e. suction pressures, reduce the pressure on the structure. See [§ 48.5 p. 595] for background theory.

syntax

```
FSPRES [ typew ] [ locaw ] { optiw }
        TOTAL    NODES
```

FSPRES specifies the dynamic pressures of the fluid–structure interface elements as output item.

[TOTAL] *type* specifies the dynamic pressure type.

TOTAL for the total dynamic pressures of the fluid–structure interface elements.

[NODES] *loca* specifies the location for the dynamic pressures to be output [§ 3.6.1 p. 58].

opti are additional options [§ 3.6.1 p. 58].

Dynamic pressures			
<i>item</i>	<i>type</i>	<i>loca</i>	
FSPRES	TOTAL	NODES	PRfs
<i>p_{fs}</i>			

Chapter 8

Frequency Response Analysis

This chapter describes the procedures to perform a linear dynamic analysis in the frequency domain, i.e., a *steady-state harmonic* response analysis. To perform a frequency response analysis you must supply input data for mass and damping [Vol. *Material Library*], and an optional base excitation [§ 6.3 p.104]. Then you may invoke one of the following modules to perform a frequency response analysis.

- *Modal response analysis* with Module MODAL [§ 8.1].
- *Direct response analysis* with Module FREQUE [§ 8.2].

The steady-state harmonic analysis may be applied when the loading is a deterministic frequency content and the system is subjected to continuous harmonic excitations.

The response may be determined via a mode-superposition technique or via a direct steady-state method (direct dynamic stiffness method). The mode-superposition method requires a preliminary eigenfrequency analysis. By means of a selected number of eigenpairs DIANA performs a transformation to modal coordinates. See § 8.1 on the next page for analysis commands.

Damping. In case of damping, only modal damping can be used in a mode superposition technique. For local damping or for heavily damped systems, a mode superposition technique is no longer appropriate. In these cases a direct method must be used.

Loading. The system can be subjected to two types of loading excitation. The first is a mechanical load in terms of externally applied concentrated forces or distributed forces like pressures. The second type of excitation is a base motion which must be prescribed via a support motion.

A mechanical load can be applied via nodal or element loads. For a base or support motion, DIANA assumes that all supports have the same prescribed motion. You must specify the motion in terms of accelerations, translational accelerations only. DIANA converts these accelerations to equivalent inertia loads. For a base motion, the response is relative to the base.

For a direct response analysis the system can also be subjected to a prescribed deformation load excitation. All other types of loading will be ignored.

Multiple excitation loads may be used in a modal or direct frequency response analysis. For each excitation frequency the responses of the excitation loads are superposed. Different loading types may be combined in the same frequency response analysis. In case a deformation excitation load is applied in combination with a base acceleration loading the deformation is relative to the base.

Multiplication coefficient. For each load set the frequency dependent multiplication coefficient is piecewise linearly interpolated from the corresponding diagram specified in table 'FREQLO' [§ 6.6.1 p.111]. If table 'FREQLO' is not specified a default multiplication coefficient equal to 1.0 will be applied for each excitation frequency on the first existing load set.

Example*file.dat*

```

'FREQL0'
LOAD 1
FREQUE 3.0 5.0 9.0 12.0 /
FACTOR 1.0 6.0 4.0 9.0 /
LOAD 3
FREQUE 0.0 20.0 /
FACTOR 2.0 2.0 /

```

In this example the diagram is specified by four points $(f_i, \lambda_i) = (3, 1), (5, 6), (9, 4), (12, 9)$ [Fig. 8.1]. The multiplication factor for load set 3 is equal to 2.0 for the frequency range 0.0 to 20.0.

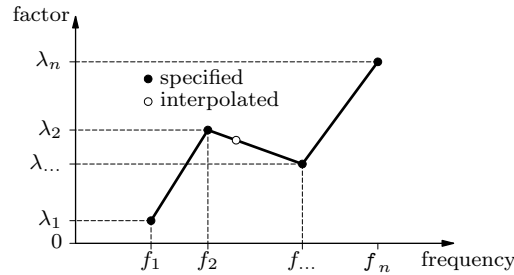


Figure 8.1: Loading multiplication specification

Fluid–structure interaction analysis. In case of a model containing fluid, structural elements and fluid–structure interface elements, automatically a fluid–structure frequency response analysis will be performed. DIANA recognizes this element combination and will perform the required preparation to perform a frequency response fluid–structure analysis. In a direct response analysis compression of the fluid as well as free surface waves are admitted and a radiation boundary may be incorporated. To include these effects specific material data is required, see ‘Fluid–structure interaction analysis’ in Volume *Material Library*. In a modal response analysis the added mass effect of the fluid [Eq. (48.100) p. 599] is taken into account. Results will be available for the structural part of the model.

Lumped element matrices may not be used in a fluid–structure response analysis.

8.1 Modal Response Analysis

The primary tasks for Module MODAL are invoked via the following command sequence. The OFF options suppress the execution of the specified task, which may save computing time if it was performed previously.

syntax

```

*MODAL
[ MODEL ... ]
[ EIGEN ... ]
[ RESPON ... ]
*END

```

MODEL evaluates and assembles the finite element model [§ 8.1.1].

EIGEN to solve the eigenvalue problem for a modal response analysis [§ 8.1.2 p. 145].

RESPON to perform the actual modal response analysis [§ 8.1.3 p. 146].

8.1.1 Model Evaluation

The `MODEL` commands customize the evaluation of the finite element model prior to the actual modal response analysis.

syntax

```
BEGIN MODEL
[ OFF ]
[ EVALUA ... ]
[ ASSEMB ]
END MODEL
```

`EVALUA` to check and evaluate geometric and material properties for elements [§ 3.4 p. 51].

`ASSEMB` to assemble the elements, i.e., creating appropriate system degrees of freedom [§ 3.5 p. 54].

8.1.2 Eigenmodes and Natural Frequencies

To solve the free vibration eigenvalue problem in a modal response analysis, i.e., to determine the eigenmodes and natural frequencies, you must give commands in the `EIGEN` block which are analogous to those for Module `EIGEN` [§ 31.2 p. 428]. The `OFF` options suppress the execution of the specified task. This may be useful to save computing time if previously determined results still reside on the `FILOS` file.

syntax

```
BEGIN EIGEN
[ OFF ]
[ FREEVI [ OFF ] ... ]
[ EXECUT [ OFF ] ... ]
[ OUTPUT ... ] ...
END EIGEN
```

`FREEVI` sets up the free vibration eigenmode analysis [§ 31.2.1 p. 429].

`EXECUT` specifies how to solve the eigenvalue problem. In particular this involves the type of the solution procedure [§ 31.3 p. 432].

`OUTPUT` to specify the eigenmodes to be output [§ 31.5.10 p. 438].

Default output

file.dcf

```
*MODAL
BEGIN EIGEN
  OUTPUT
END EIGEN
[ commands ]
*END
```

If you only give a single `OUTPUT` command, like in the above example, or if you omit the `OUTPUT` command, then `DIANA` gives a default output as if you had given the following commands.

file.dcf

```

*MODAL
BEGIN EIGEN
  BEGIN OUTPUT
    DISPLA TOTAL TRANSL GLOBAL
  END OUTPUT
END EIGEN
[ commands ]
*END

```

Example*file.dcf*

```

*MODAL
BEGIN EIGEN
  FREEVI MASS
  BEGIN EXECUT
    NMODES=20
  END EXECUT
  OUTPUT DISPLA
END EIGEN
*END

```

This command sequence is a typical example for a preliminary eigenvalue analysis. In the free vibration problem, indicated by **FREEVIB**, the system mass matrix has to be used which is invoked by the **MASS** command option. By default DIANA assembles the consistent element mass matrices. The **EXECUT** command block calculates the natural frequencies and corresponding mode shapes needed for modal analysis. Parameter **NMODES=20** asks for an arbitrarily chosen number of twenty frequencies. Finally, printed output of results like natural frequencies and mode shapes is obtained through the **OUTPUT** command block.

8.1.3 Response Analysis

To perform the actual modal response analysis, i.e., to determine the response of a construction at a certain load frequency, you must give commands in the **RESPON** block. The **OFF** options suppress the specified tasks. This may be useful to save computing time if previously determined results still reside on the **FILOS** file.

syntax

```

BEGIN RESPON [ OFF ]
[ REDUCE [ OFF ] ... ]
[ EXECUT [ OFF ] ... ]
[ OUTPUT ... ] ...
END RESPON

```

REDUCE reduces the system of equations [§ 8.1.3.1].

EXECUT executes the actual modal response analysis [§ 8.1.3.2].

OUTPUT specifies the modal response analysis results to be output [§ 8.3].

Default output*file.dcf*

```

*MODAL
[ commands ]
BEGIN RESPON
  OUTPUT
END RESPON
*END

```

If you only give a single **OUTPUT** command, like in the above example, or if you omit the **OUTPUT** command, then DIANA gives a default output as if you had given the following commands.

file.dcf

```

*MODAL
[ commands ]
BEGIN RESPON
  BEGIN OUTPUT
    DISPLA TOTAL TRANSL GLOBAL COMPLE
    VELOCI TOTAL TRANSL GLOBAL COMPLE
    ACCELE TOTAL TRANSL GLOBAL COMPLE
  END OUTPUT
END RESPON
*END

```

8.1.3.1 Reduction of System of Equations

To reduce the full set of equations, you must give commands in the REDUCE block within the RESPON block.

syntax

```

BEGIN REDUCE
[ OFF ]
[ BEGIN SELECT
  MODES _____
      modes n...
      ALL
  END SELECT ]
END REDUCE

```

OFF suppresses the reduction of the system of equations. This may save computing time when the reduced system of equations is still available from a previous modal response analysis.

SELECT MODES selects eigenmodes for the modal response analysis. The modes must have been determined previously [§ 8.1.2 p. 145]. You may explicitly specify a set of *modes*, or you may require a modal response analysis for ALL determined modes. All modes is also the default if you do not select modes. [ALL]

8.1.3.2 Analysis Execution

The EXECUT command block actually performs the modal response analysis by applying the excitation frequencies.

syntax

```

BEGIN EXECUT
[ OFF ]
[ ACCELE ]
[ MODES [ _____ ] ]
  OFF
  NUMBER=i n...
[ EXPLIC [ _____ ] ]
  OFF
  FREQUE=om r...
[ DAMPIN c1 r [ ... cn r ] ]
END EXECUT

```

OFF suppresses the execution of the modal response analysis.

ACCELE employs a static correction to the displacement response according to a *mode acceleration* method.

[OFF] **MODES** indicates an excitation with natural frequencies. Parameter **NUMBER** specifies a series of natural frequency numbers i indicating the natural frequencies f_i . If you do not specify any frequency numbers then DIANA assumes excitation with the first natural frequency f_1 . No frequency number may exceed the highest natural frequency number calculated [§ 8.1.2].

*Note that when a reduced system of equations is used [§ 8.1.3.1], the natural frequency numbers refer to the reduced set of eigenmodes and no longer to the full set of calculated eigenmodes, i.e. **MODES NUMBER 3** refers to the third eigenmode of the reduced set of eigenmodes, which may be different from the third eigenmode that has been calculated.*

[EXPLICIT] **EXPLICIT** indicates an excitation with explicitly specified frequencies Ω_i . Parameter **FREQUE** specifies a series of frequencies ω . If you do not specify any frequencies then DIANA assumes excitation with a discrete frequency of 1 by default.

[c1=0.01] **DAMPIN** specifies modal damping. Values **c1** to **cn** are a series of damping ratios in parts of the critical damping factor c_{crit} [Eq. (48.5) p. 586]; a value of 0.01 indicates 1 %. If only one factor **c1** is specified, then the damping is the same for all natural frequencies of the system. Alternatively, the damping ratios must be specified for all natural frequencies of the system, which may be reduced [§ 8.1.3.1].

Modal damping must be applied if excitations coincide with the natural frequencies.

[$\Omega_1 = 1.0$] If you specify neither natural nor discrete frequencies, then DIANA assumes an excitation with a discrete frequency of 1.

Default *file.dcf*

```
*MODAL
[ commands ]
BEGIN RESPON
EXECUT
END RESPON
[ commands ]
```

If you only give a single **EXECUT** command, then DIANA will execute a frequency response analysis according to the following command sequence.

file.dcf

```
*MODAL
BEGIN RESPON
BEGIN EXECUT
ACCELE
EXPLICIT FREQUE=1.0
END EXECUT
END RESPON
[ commands ]
```

8.1.4 Example Commands

8.1.4.1 Response to Forced Periodic Excitation

The following commands are an example for steady-state response to forced periodic excitations. In this case, Module MODAL gives amplitudes of displacement and velocity. The finite element model is subjected to an harmonically varying forcing function.

file.dcf

```

*MODAL
BEGIN RESPON
  BEGIN EXECUT
    ACCELE
    EXPLIC FREQUE=1. 7. 12.
  END EXECUT
END RESPON
[ output commands ]

```

A modal analysis is performed by the EXECUT command block and a static correction is employed to the displacement response by the ACCELE command. The EXPLIC command requests the analysis of response for three explicit excitation frequencies of 1, 7, and 12 Hz, if the unit of time is second [s]. See § 8.3 on page 152 for output commands.

8.1.4.2 Frequency Response

This example shows how to setup a frequency response analysis. In this analysis a combination of response amplitude, like displacement and phase angle versus frequency ratio, are determined for excitations covering a specified range with some resonance¹ frequencies when the system is subjected to a certain input. In order to avoid an infinite response at resonance, effects of damping must be included.

file.dcf

```

*MODAL
BEGIN RESPON
  REDUCE OFF
  BEGIN EXECUTE
    MODES NUMBER=1-5(2)
    DAMPIN 0.02
  END EXECUTE
END RESPON
[ commands ]

```

REDUCE OFF specifies that no reduction will be used. Modal damping is specified in the DAMPIN command, where 0.02 is a ratio in parts of the critical damping and which is, in this example, the same for all natural frequencies. The excitation frequencies at which the response has to be calculated are specified in the EXECUT command block. The MODES command indicates a sequence of excitations in the first, third, and fifth natural frequency (f_1 , f_3 , f_5).

8.2 Direct Response Analysis

The primary tasks for Module FREQUE are invoked via the following command sequence.

syntax

```

*FREQUE
[ MODEL ... ]
[ EXECUT ... ]
[ OUTPUT ... ] ...
*END

```

MODEL evaluates and assembles the finite element model [§ 8.2.1].

EXECUT executes the actual direct response analysis [§ 8.2.2].

OUTPUT specifies the direct response analysis results to be output [§ 8.3].

¹resonance = the condition when the excitation frequency equals the natural frequency.

Default output*file.dcf*

```
*FREQUE
[ commands ]
OUTPUT
*END
```

If you only give a single **OUTPUT** command, like in the above example, or if you omit the **OUTPUT** command, then **DIANA** gives a default output as if you had given the following commands.

file.dcf

```
*FREQUE
[ commands ]
BEGIN OUTPUT
  DISPLA TOTAL TRANSL GLOBAL COMPLE
  VELOCI TOTAL TRANSL GLOBAL COMPLE
  ACCELE TOTAL TRANSL GLOBAL COMPLE
END OUTPUT
*END
```

Example*file.dcf*

```
*FREQUE
BEGIN EXECUT
  EXPLIC FREQUE=1. 7. 12.
END EXECUT
[ output commands ]
```

This command sequence is an example for direct response to forced periodic excitations. In this case, Module **FREQUE** gives amplitudes of displacement and velocity. The finite element model is subjected to an harmonically varying forcing function. A direct response analysis is performed by the **EXECUT** command block. The **EXPLIC** command requests the analysis of response for three explicit excitation frequencies of 1, 7, and 12 Hz, if the unit of time is second [s]. See § 8.3 on page 152 for output commands.

8.2.1 Model Evaluation

The **MODEL** commands customize the evaluation of the finite element model prior to the actual direct response analysis.

syntax

```
BEGIN MODEL
[ OFF ]
[ EVALUA ... ]
[ ASSEMB ]
[ MATRIX ]
END MODEL
```

EVALUA to check and evaluate geometric and material properties for elements [§ 3.4 p. 51].

ASSEMB to assemble the elements, i.e., creating appropriate system degrees of freedom [§ 3.5 p. 54].

MATRIX to setup the element matrices [§ 8.2.1.1].

8.2.1.1 Element Matrices

The **MATRIX** commands specify the type of element matrices to be applied in the direct response analysis.

syntax

```

BEGIN MATRIX
[ OFF ]
[ MASS [ OFF ] [ _____ ] [ ROTATI [ OFF ] ] ]
      CONSIS
      LUMPED
[ DAMPIN [ OFF ] [ _____ ] ]
      CONSIS
      LUMPED
[ STRESS [ OFF ] [ _____ ] ]
      INPUT { _____ }
              LOAD=losetn
              FACTOR=facr
      CALCUL { _____ }
              LOAD=losetn
              FACTOR=facr
      PHASE
END MATRIX

```

OFF suppresses the setup of element matrices.

MASS specifies the kind of mass matrices to be applied:

[CONSIS]

CONSIS for consistent matrices (the default).

LUMPED for lumped matrices.

ROTATI indicates that rotational terms must be included in the mass matrices.

The OFF option suppresses the inclusion of rotational terms. If you do not specify the ROTATI option explicitly, then DIANA will include rotational terms by default.

OFF suppresses the setup of mass matrices.

DAMPIN specifies the kind of damping matrices to be applied:

[CONSIS]

CONSIS for consistent matrices (the default).

LUMPED for lumped matrices.

OFF suppresses the setup of damping matrices.

STRESS adds the geometric stress-stiffness matrix \mathbf{K}_G to the linear elastic stiffness matrix \mathbf{K}_{L0} . [CALCUL]

INPUT indicates a geometric stress-stiffness matrix, with stresses specified via a prestress load for load set *loset*, in subtable **ELEMEN** and/or **REINFO** of table 'LOADS' [Vol. *Element Library*].

CALCUL indicates a geometric stress-stiffness matrix, with stresses which DIANA will calculate automatically in a linear static analysis for load set *loset* (the default).

PHASE indicates a geometric stress-stiffness matrix, with stresses present from a previous phase.

To setup the geometric stress-stiffness matrix \mathbf{K}_G from a stress field you may specify a load set number *loset* via the **LOAD** parameter. This load set number corresponds to a load set in input table 'LOADS' [§ 2.3.8 p. 45]. Default is the lowest available load set number. The optional parameter **FACTOR** specifies a multiplication factor *fac*. [fac=1.0]

8.2.2 Analysis Execution

The EXECUT command block actually performs the direct response analysis by applying the excitation frequencies.

syntax

```
BEGIN EXECUT
[ OFF ]
[ EXPLIC [FREQUE=omr...] ]
END EXECUT
```

OFF suppresses the execution of the direct response analysis.

EXPLIC indicates an excitation with explicitly specified frequencies Ω_i . Parameter FREQUE specifies a series of frequencies *om*. If you do not specify any frequencies then DIANA assumes excitation with a discrete frequency of 1 by default.

[*om*=1.0]

Default

file.dcf

```
*FREQUE
[ commands ]
EXECUT
[ commands ]
```

If you only give a single EXECUT command, then DIANA will execute a direct response analysis according to the following command sequence.

file.dcf

```
*FREQUE
BEGIN EXECUT
  EXPLIC FREQUE=1.0
END EXECUT
[ commands ]
```

8.3 Output of Analysis Results

With command in the OUTPUT block you can get output of results from a frequency response analysis. For the general, analysis type independent options for output of analysis results see § 3.6 on page 55.

syntax

```
BEGIN OUTPUT [devicew] [outoptw...] [params]
[ OFF ]
[ BEGIN SELECT
  [ ... ]
  [ FREQUE [ _____ ] ]
    frqnrsn...
    ALL
END SELECT ]
[ LAYOUT ... ]
itemw { typew } { formw } { operw } { compw } { locaw } { optiw }
DISPLA                                     AMPLIT
VELOCI                                    COMPLE
ACCELE                                   RELATI
FORCE                                     ...
```

STRAIN
STRESS
FSPRES
END OUTPUT

SELECT command block to customize the batch output:

... for model selection see § 3.6.2 on page 59.

FREQUE selects specific frequencies for output: *frqnrs* is a series of frequency numbers, ALL selects all frequencies. [ALL]

LAYOUT commands to customize the layout and style of tabular output [§ 3.6.4 p. 64].

item is the name of the analysis result to be output. See § 3.6.1 on page 56 for complete syntax of this command.

DISPLA for displacements [§ 8.3.1].

VELOCI for velocities [§ 8.3.2].

ACCELE for accelerations [§ 8.3.3].

FORCE for nodal forces and moments [§ 8.3.6].

STRAIN for strains [§ 8.3.4].

STRESS for stresses [§ 8.3.5].

FSPRES for dynamic pressures of fluid–structure interface elements [§ 8.3.7].

opti are options applied to the output *item* [§ 3.6.1 p. 58]. Special options for a frequency response analysis indicate how the result must be expressed. [COMPLE]

AMPLIT to get the result of *item* expressed in amplitude and phase angle. This option is not available for iDIANA, because the amplitude and phase angle can be calculated in the iDIANA Results environment from the complex representation of the result *item*.

COMPLE to get the complex representation of the result *item*, i.e., the real and imaginary part.

RELATI gives the displacement, velocity, or acceleration results relative to an explicitly defined base node indicated by the BASNOD command [§ 3.6.2 p. 59].

8.3.1 Displacements

syntax

DISPLA	[<i>type_w</i>]	[<i>form_w</i>]	[<i>oper_w</i>]	{ <i>comp_w</i> }	{ <i>opti_w</i> }
	TOTAL	TRANSL	LOCAL		AMPLIT
		ROTATI	GLOBAL		COMPLE
					RELATI
					...

DISPLA specifies displacements of the nodes as output item.

type specifies the displacement type. [TOTAL]

TOTAL for the total displacements of a structure, i.e., the deformed geometry at a certain stage.

form specifies the displacement formulation [§ 3.6.1 p. 57]. [TRANSL]

oper specifies an operation (transformation) to be performed on the displacements [§ 3.6.1 p. 57]. [GLOBAL]

comp selects displacement components for output. Default is all available components. [all]

opti are additional options [§ 8.3].

Translation displacements					<i>comp</i> . . .		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	<i>opti</i>	X	Y	Z
DISPLA	TOTAL	TRANSL	LOCAL	AMPLIT	DCtxA	DCtyA	DCtzA
					\hat{u}_x	\hat{u}_y	\hat{u}_z
					DCtxP	DCtyP	DCtzP
					φu_x	φu_y	φu_z
DISPLA	TOTAL	TRANSL	LOCAL	AMPLIT RELATI	DrCtxA	DrCtyA	DrCtzA
					\hat{u}_{x_r}	\hat{u}_{y_r}	\hat{u}_{z_r}
					DrCtxP	DrCtyP	DrCtzP
					φu_{x_r}	φu_{y_r}	φu_{z_r}
DISPLA	TOTAL	TRANSL	LOCAL	COMPLE	DCtxR	DCtyR	DCtzR
					$\Re u_x$	$\Re u_y$	$\Re u_z$
					DCtxI	DCtyI	DCtzI
					$\Im u_x$	$\Im u_y$	$\Im u_z$
DISPLA	TOTAL	TRANSL	LOCAL	COMPLE RELATI	DrCtxR	DrCtyR	DrCtzR
					$\Re u_{x_r}$	$\Re u_{y_r}$	$\Re u_{z_r}$
					DrCtxI	DrCtyI	DrCtzI
					$\Im u_{x_r}$	$\Im u_{y_r}$	$\Im u_{z_r}$
DISPLA	TOTAL	TRANSL	GLOBAL	AMPLIT	DCtXA	DCtYA	DCtZA
					\hat{u}_X	\hat{u}_Y	\hat{u}_Z
					DCtXP	DCtYP	DCtZP
					φu_X	φu_Y	φu_Z
DISPLA	TOTAL	TRANSL	GLOBAL	AMPLIT RELATI	DrCtXA	DrCtYA	DrCtZA
					\hat{u}_{X_r}	\hat{u}_{Y_r}	\hat{u}_{Z_r}
					DrCtXP	DrCtYP	DrCtZP
					φu_{X_r}	φu_{Y_r}	φu_{Z_r}
DISPLA	TOTAL	TRANSL	GLOBAL	COMPLE	DCtXA	DCtYA	DCtZA
					$\Re u_X$	$\Re u_Y$	$\Re u_Z$
					DCtXP	DCtYP	DCtZP
					$\Im u_X$	$\Im u_Y$	$\Im u_Z$
DISPLA	TOTAL	TRANSL	GLOBAL	COMPLE RELATI	DrCtXA	DrCtYA	DrCtZA
					$\Re u_{X_r}$	$\Re u_{Y_r}$	$\Re u_{Z_r}$
					DrCtXP	DrCtYP	DrCtZP
					$\Im u_{X_r}$	$\Im u_{Y_r}$	$\Im u_{Z_r}$

Rotation displacements					<i>comp</i> ...		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	<i>opti</i>	X	Y	Z
DISPLA	TOTAL	ROTATI	LOCAL	AMPLIT	DCrxA	DCryA	DCrzA
					$\hat{\phi}_x$	$\hat{\phi}_y$	$\hat{\phi}_z$
					DCrxP	DCryP	DCrzP
					$\varphi\phi_x$	$\varphi\phi_y$	$\varphi\phi_z$
DISPLA	TOTAL	ROTATI	LOCAL	AMPLIT RELATI	DrCrxA	DrCryA	DrCrzA
					$\hat{\phi}_{x_r}$	$\hat{\phi}_{y_r}$	$\hat{\phi}_{z_r}$
					DrCrXP	DrCryP	DrCrzP
					$\varphi\phi_{x_r}$	$\varphi\phi_{y_r}$	$\varphi\phi_{z_r}$
DISPLA	TOTAL	ROTATI	LOCAL	COMPLE	DCrxR	DCryR	DCrzR
					$\Re\phi_x$	$\Re\phi_y$	$\Re\phi_z$
					DCrxI	DCryI	DCrzI
					$\Im\phi_x$	$\Im\phi_y$	$\Im\phi_z$
DISPLA	TOTAL	ROTATI	LOCAL	COMPLE RELATI	DrCrXR	DrCryR	DrCrzR
					$\Re\phi_{x_r}$	$\Re\phi_{y_r}$	$\Re\phi_{z_r}$
					DrCrXI	DrCryI	DrCrzI
					$\Im\phi_{x_r}$	$\Im\phi_{y_r}$	$\Im\phi_{z_r}$
DISPLA	TOTAL	ROTATI	GLOBAL	AMPLIT	DCrXA	DCrYA	DCrZA
					$\hat{\phi}_X$	$\hat{\phi}_Y$	$\hat{\phi}_Z$
					DCrXP	DCrYP	DCrZP
					$\varphi\phi_X$	$\varphi\phi_Y$	$\varphi\phi_Z$
DISPLA	TOTAL	ROTATI	GLOBAL	AMPLIT RELATI	DrCrXA	DrCrYA	DrCrZA
					$\hat{\phi}_{X_r}$	$\hat{\phi}_{Y_r}$	$\hat{\phi}_{Z_r}$
					DrCrXP	DrCrYP	DrCrZP
					$\varphi\phi_{X_r}$	$\varphi\phi_{Y_r}$	$\varphi\phi_{Z_r}$
DISPLA	TOTAL	ROTATI	GLOBAL	COMPLE	DCrXR	DCrYR	DCrZR
					$\Re\phi_X$	$\Re\phi_Y$	$\Re\phi_Z$
					DCrXP	DCrYP	DCrZP
					$\Im\phi_X$	$\Im\phi_Y$	$\Im\phi_Z$
DISPLA	TOTAL	ROTATI	GLOBAL	COMPLE RELATI	DrCrXR	DrCrYR	DrCrZR
					$\Re\phi_{X_r}$	$\Re\phi_{Y_r}$	$\Re\phi_{Z_r}$
					DrCrXP	DrCrYP	DrCrZP
					$\Im\phi_{X_r}$	$\Im\phi_{Y_r}$	$\Im\phi_{Z_r}$

8.3.2 Velocities

syntax

```

VELOCI [ typew ] [ formw ] [ operw ] { compw } { optiw }
      TOTAL      TRANSL  LOCAL      AMPLIT
              ROTATI  GLOBAL      COMPLE
                      RELATI
                      ...

```

VELOCI specifies velocities of the nodes as output item.

type specifies the velocity type. [TOTAL]

TOTAL for the total velocities of a structure.

form specifies the velocity formulation [§ 3.6.1 p. 57]. [TRANSL]

oper specifies an operation (transformation) to be performed on the velocities [§ 3.6.1 p. 57]. [GLOBAL]

comp selects velocity components for output. Default is all available components. [all]

opti are additional options [§ 8.3].

Translation velocities					<i>comp</i> . . .		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	<i>opti</i>	X	Y	Z
VELOC	TOTAL	TRANSL	LOCAL	AMPLIT	VtxA	VtyA	VtzA
					\hat{u}_x	\hat{u}_y	\hat{u}_z
					VtxP	VtyP	VtzP
					$\varphi \dot{u}_x$	$\varphi \dot{u}_y$	$\varphi \dot{u}_z$
VELOC	TOTAL	TRANSL	LOCAL	AMPLIT RELATI	VrtxA	VrtyA	VrtzA
					\hat{u}_{x_r}	\hat{u}_{y_r}	\hat{u}_{z_r}
					VrtxP	VrtyP	VrtzP
					$\varphi \dot{u}_{x_r}$	$\varphi \dot{u}_{y_r}$	$\varphi \dot{u}_{z_r}$
VELOC	TOTAL	TRANSL	LOCAL	COMPLE	VtxR	VtyR	VtzR
					$\Re \dot{u}_x$	$\Re \dot{u}_y$	$\Re \dot{u}_z$
					VtxI	VtyI	VtzI
					$\Im \dot{u}_x$	$\Im \dot{u}_y$	$\Im \dot{u}_z$
VELOC	TOTAL	TRANSL	LOCAL	COMPLE RELATI	VrtxR	VrtyR	VrtzR
					$\Re \dot{u}_{x_r}$	$\Re \dot{u}_{y_r}$	$\Re \dot{u}_{z_r}$
					VrtxI	VrtyI	VrtzI
					$\Im \dot{u}_{x_r}$	$\Im \dot{u}_{y_r}$	$\Im \dot{u}_{z_r}$
VELOC	TOTAL	TRANSL	GLOBAL	AMPLIT	VtXA	VtYA	VtZA
					\hat{u}_X	\hat{u}_Y	\hat{u}_Z
					VtXP	VtYP	VtZP
					$\varphi \dot{u}_X$	$\varphi \dot{u}_Y$	$\varphi \dot{u}_Z$
VELOC	TOTAL	TRANSL	GLOBAL	AMPLIT RELATI	VrtXA	VrtYA	VrtZA
					\hat{u}_{X_r}	\hat{u}_{Y_r}	\hat{u}_{Z_r}
					VrtXP	VrtYP	VrtZP
					$\varphi \dot{u}_{X_r}$	$\varphi \dot{u}_{Y_r}$	$\varphi \dot{u}_{Z_r}$
VELOC	TOTAL	TRANSL	GLOBAL	COMPLE	VtXR	VtYR	VtZR
					$\Re \dot{u}_X$	$\Re \dot{u}_Y$	$\Re \dot{u}_Z$
					VtXI	VtYI	VtZI
					$\Im \dot{u}_X$	$\Im \dot{u}_Y$	$\Im \dot{u}_Z$
VELOC	TOTAL	TRANSL	GLOBAL	COMPLE RELATI	VrtXR	VrtYR	VrtZR
					$\Re \dot{u}_{X_r}$	$\Re \dot{u}_{Y_r}$	$\Re \dot{u}_{Z_r}$
					VrtXI	VrtYI	VrtZI
					$\Im \dot{u}_{X_r}$	$\Im \dot{u}_{Y_r}$	$\Im \dot{u}_{Z_r}$

Rotation velocities					<i>comp</i> ...		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	<i>opti</i>	X	Y	Z
VELOCI	TOTAL	ROTATI	LOCAL	AMPLIT	VrxA	VryA	VrzA
					$\hat{\phi}_x$	$\hat{\phi}_y$	$\hat{\phi}_z$
					VrxP	VryP	VrzP
					$\varphi \dot{\phi}_x$	$\varphi \dot{\phi}_y$	$\varphi \dot{\phi}_z$
VELOCI	TOTAL	ROTATI	LOCAL	AMPLIT RELATI	VrrxA	VrryA	VrrzA
					$\hat{\phi}_{x_r}$	$\hat{\phi}_{y_r}$	$\hat{\phi}_{z_r}$
					VrrxP	VrryP	VrrzP
					$\varphi \dot{\phi}_{x_r}$	$\varphi \dot{\phi}_{y_r}$	$\varphi \dot{\phi}_{z_r}$
VELOCI	TOTAL	ROTATI	LOCAL	COMPLE	VrxR	VryR	VrzR
					$\Re \dot{\phi}_x$	$\Re \dot{\phi}_y$	$\Re \dot{\phi}_z$
					VrxI	VryI	VrzI
					$\Im \dot{\phi}_x$	$\Im \dot{\phi}_y$	$\Im \dot{\phi}_z$
VELOCI	TOTAL	ROTATI	LOCAL	COMPLE RELATI	VrrxR	VrryR	VrrzR
					$\Re \dot{\phi}_{x_r}$	$\Re \dot{\phi}_{y_r}$	$\Re \dot{\phi}_{z_r}$
					VrrxI	VrryI	VrrzI
					$\Im \dot{\phi}_{x_r}$	$\Im \dot{\phi}_{y_r}$	$\Im \dot{\phi}_{z_r}$
VELOCI	TOTAL	ROTATI	GLOBAL	AMPLIT	VrXA	VrYA	VrZA
					$\hat{\phi}_X$	$\hat{\phi}_Y$	$\hat{\phi}_Z$
					VrXP	VrYP	VrZP
					$\varphi \dot{\phi}_X$	$\varphi \dot{\phi}_Y$	$\varphi \dot{\phi}_Z$
VELOCI	TOTAL	ROTATI	GLOBAL	AMPLIT RELATI	VrrXA	VrrYA	VrrZA
					$\hat{\phi}_{X_r}$	$\hat{\phi}_{Y_r}$	$\hat{\phi}_{Z_r}$
					VrrXP	VrrYP	VrrZP
					$\varphi \dot{\phi}_{X_r}$	$\varphi \dot{\phi}_{Y_r}$	$\varphi \dot{\phi}_{Z_r}$
VELOCI	TOTAL	ROTATI	GLOBAL	COMPLE	VrXR	VrYR	VrZR
					$\Re \dot{\phi}_X$	$\Re \dot{\phi}_Y$	$\Re \dot{\phi}_Z$
					VrXI	VrYI	VrZI
					$\Im \dot{\phi}_X$	$\Im \dot{\phi}_Y$	$\Im \dot{\phi}_Z$
VELOCI	TOTAL	ROTATI	GLOBAL	COMPLE RELATI	VrrXR	VrrYR	VrrZR
					$\Re \dot{\phi}_{X_r}$	$\Re \dot{\phi}_{Y_r}$	$\Re \dot{\phi}_{Z_r}$
					VrrXI	VrrYI	VrrZI
					$\Im \dot{\phi}_{X_r}$	$\Im \dot{\phi}_{Y_r}$	$\Im \dot{\phi}_{Z_r}$

8.3.3 Accelerations

syntax

ACCELE	[<i>type</i> _w]	[<i>form</i> _w]	[<i>oper</i> _w]	{ <i>comp</i> _w }	{ <i>opti</i> _w }
	TOTAL	TRANSL	LOCAL		AMPLIT
		ROTATI	GLOBAL		COMPLE
					RELATI
					...

ACCELE specifies accelerations of the nodes as output item.

type specifies the acceleration type.

[TOTAL]

TOTAL for the total accelerations of a structure, i.e., the acceleration at a certain stage.

form specifies the acceleration formulation [§ 3.6.1 p. 57].

[TRANSL]

oper specifies an operation (transformation) to be performed on the accelerations [§ 3.6.1 p. 57].

[GLOBAL]

comp selects acceleration components for output. Default is all available components.

[all]

opti are additional options [§ 8.3].

Translation accelerations					<i>comp</i> . . .		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	<i>opti</i>	X	Y	Z
ACCELE	TOTAL	TRANSL	LOCAL	AMPLIT	AtxA	AtyA	AtzA
					\hat{u}_x	\hat{u}_y	\hat{u}_z
					AtxP	AtyP	AtzP
					$\varphi \ddot{u}_x$	$\varphi \ddot{u}_y$	$\varphi \ddot{u}_z$
ACCELE	TOTAL	TRANSL	LOCAL	AMPLIT RELATI	ArtxA	ArtyA	ArtzA
					\hat{u}_{x_r}	\hat{u}_{y_r}	\hat{u}_{z_r}
					ArtxP	ArtyP	ArtzP
					$\varphi \ddot{u}_{x_r}$	$\varphi \ddot{u}_{y_r}$	$\varphi \ddot{u}_{z_r}$
ACCELE	TOTAL	TRANSL	LOCAL	COMPLE	AtxR	AtyR	AtzR
					$\Re \ddot{u}_x$	$\Re \ddot{u}_y$	$\Re \ddot{u}_z$
					AtxI	AtyI	AtzI
					$\Im \ddot{u}_x$	$\Im \ddot{u}_y$	$\Im \ddot{u}_z$
ACCELE	TOTAL	TRANSL	LOCAL	COMPLE RELATI	ArtxR	ArtyR	ArtzR
					$\Re \ddot{u}_{x_r}$	$\Re \ddot{u}_{y_r}$	$\Re \ddot{u}_{z_r}$
					ArtxI	ArtyI	ArtzI
					$\Im \ddot{u}_{x_r}$	$\Im \ddot{u}_{y_r}$	$\Im \ddot{u}_{z_r}$
ACCELE	TOTAL	TRANSL	GLOBAL	AMPLIT	AtXA	AtYA	AtZA
					\hat{u}_X	\hat{u}_Y	\hat{u}_Z
					AtXP	AtYP	AtZP
					$\varphi \ddot{u}_X$	$\varphi \ddot{u}_Y$	$\varphi \ddot{u}_Z$
ACCELE	TOTAL	TRANSL	GLOBAL	AMPLIT RELATI	ArtXA	ArtYA	ArtZA
					\hat{u}_{X_r}	\hat{u}_{Y_r}	\hat{u}_{Z_r}
					ArtXP	ArtYP	ArtZP
					$\varphi \ddot{u}_{X_r}$	$\varphi \ddot{u}_{Y_r}$	$\varphi \ddot{u}_{Z_r}$
ACCELE	TOTAL	TRANSL	GLOBAL	COMPLE	AtXR	AtYR	AtZR
					$\Re \ddot{u}_X$	$\Re \ddot{u}_Y$	$\Re \ddot{u}_Z$
					AtXI	AtYI	AtZI
					$\Im \ddot{u}_X$	$\Im \ddot{u}_Y$	$\Im \ddot{u}_Z$
ACCELE	TOTAL	TRANSL	GLOBAL	COMPLE RELATI	ArtXR	ArtYR	ArtZR
					$\Re \ddot{u}_{X_r}$	$\Re \ddot{u}_{Y_r}$	$\Re \ddot{u}_{Z_r}$
					ArtXI	ArtYI	ArtZI
					$\Im \ddot{u}_{X_r}$	$\Im \ddot{u}_{Y_r}$	$\Im \ddot{u}_{Z_r}$

Rotation accelerations					<i>comp</i> ...		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	<i>opti</i>	X	Y	Z
ACCELE	TOTAL	ROTATI	LOCAL	AMPLIT	ArxA	ArYA	ArZA
					$\hat{\phi}_x$	$\hat{\phi}_y$	$\hat{\phi}_z$
					ArxP	ArYP	ArzP
					$\varphi \ddot{\phi}_x$	$\varphi \ddot{\phi}_y$	$\varphi \ddot{\phi}_z$
ACCELE	TOTAL	ROTATI	LOCAL	AMPLIT RELATI	ArrxA	ArrYA	ArrZA
					$\hat{\phi}_{x_r}$	$\hat{\phi}_{y_r}$	$\hat{\phi}_{z_r}$
					ArrxP	ArrYP	ArrzP
					$\varphi \ddot{\phi}_{x_r}$	$\varphi \ddot{\phi}_{y_r}$	$\varphi \ddot{\phi}_{z_r}$
ACCELE	TOTAL	ROTATI	LOCAL	COMPLE	ArxR	ArYR	ArZR
					$\Re \ddot{\phi}_x$	$\Re \ddot{\phi}_y$	$\Re \ddot{\phi}_z$
					ArxI	ArYI	ArZI
					$\Im \ddot{\phi}_x$	$\Im \ddot{\phi}_y$	$\Im \ddot{\phi}_z$
ACCELE	TOTAL	ROTATI	LOCAL	COMPLE RELATI	ArrxR	ArrYR	ArrZR
					$\Re \ddot{\phi}_{x_r}$	$\Re \ddot{\phi}_{y_r}$	$\Re \ddot{\phi}_{z_r}$
					ArrxI	ArrYI	ArrZI
					$\Im \ddot{\phi}_{x_r}$	$\Im \ddot{\phi}_{y_r}$	$\Im \ddot{\phi}_{z_r}$
ACCELE	TOTAL	ROTATI	GLOBAL	AMPLIT	ArXA	ArYA	ArZA
					$\hat{\phi}_X$	$\hat{\phi}_Y$	$\hat{\phi}_Z$
					ArXP	ArYP	ArZP
					$\varphi \ddot{\phi}_X$	$\varphi \ddot{\phi}_Y$	$\varphi \ddot{\phi}_Z$
ACCELE	TOTAL	ROTATI	GLOBAL	AMPLIT RELATI	ArrXA	ArrYA	ArrZA
					$\hat{\phi}_{X_r}$	$\hat{\phi}_{Y_r}$	$\hat{\phi}_{Z_r}$
					ArrXP	ArrYP	ArrZP
					$\varphi \ddot{\phi}_{X_r}$	$\varphi \ddot{\phi}_{Y_r}$	$\varphi \ddot{\phi}_{Z_r}$
ACCELE	TOTAL	ROTATI	GLOBAL	COMPLE	ArXR	ArYR	ArZR
					$\Re \ddot{\phi}_X$	$\Re \ddot{\phi}_Y$	$\Re \ddot{\phi}_Z$
					ArXI	ArYI	ArZI
					$\Im \ddot{\phi}_X$	$\Im \ddot{\phi}_Y$	$\Im \ddot{\phi}_Z$
ACCELE	TOTAL	ROTATI	GLOBAL	COMPLE RELATI	ArrXR	ArrYR	ArrZR
					$\Re \ddot{\phi}_{X_r}$	$\Re \ddot{\phi}_{Y_r}$	$\Re \ddot{\phi}_{Z_r}$
					ArrXI	ArrYI	ArrZI
					$\Im \ddot{\phi}_{X_r}$	$\Im \ddot{\phi}_{Y_r}$	$\Im \ddot{\phi}_{Z_r}$

8.3.4 Strains

syntax

STRAIN	[<i>type_w</i>]	[<i>form_w</i>]	[<i>oper_w</i>]	{ <i>comp_w</i> }	{ <i>loca_w</i> }	{ <i>opti_w</i> }
	TOTAL	GREEN	LOCAL		INTPNT	AMPLIT
		FORCE	GLOBAL		NODES	COMPLE
		MOMENT	VONMIS		CENTER	...
		DISFOR				
		DISMOM				
		TRACTI				

STRAIN specifies strains as output item. Table 8.1 on the next page outlines the availability and applicability of the various strain output options for each of the element families.

type specifies the strain type [§ 3.6.1 p. 57].

[TOTAL]

form specifies the strain formulation.

[GREEN]

GREEN for Green–Lagrange strains [§ 8.3.4.1].

FORCE for deformations due to normal and shear forces [§ 8.3.4.2].

DISFOR for generalized strains [§ 8.3.4.2].

Table 8.1: AVAILABILITY OF STRAIN OUTPUT FOR FREQUENCY RESPONSE ANALYSIS

<i>item</i>	STRAIN	truss	beam	pl. stress	pl. strain	axisymm.	plate bend.	fl. shell	cu. shell	solid	str. interface	spring	p. mass	e. reinfo.
<i>type</i>	TOTAL	a	a	a	a	a	a	a	a	a	a	-	-	a
<i>form</i>	GREEN	a	a	a	a	a	a	a	a	a	-	-	-	a
	FORCE	-	b	-	-	-	-	-	-	-	-	a	-	c
	MOMENT	-	b	-	-	-	-	-	-	-	-	a	-	d
	DISFOR	-	-	-	-	-	a	a	-	-	-	-	-	-
	DISMOM	-	-	-	-	-	a	a	-	-	-	-	-	-
	TRACTI	-	-	-	-	-	-	-	-	-	a	-	-	c
<i>oper</i>	LOCAL	a	a	a	a	a	a	a	a	a	a	-	-	a
	GLOBAL	a	a	a	a	a	a	a	a	a	-	a	-	c
	VONMIS	-	-	a	a	a	a	a	a	a	-	-	-	-
<i>loca</i>	INTPNT	a	f	a	a	a	a	a	a	a	a	-	-	e
	NODES	a	a	a	a	a	a	a	a	a	-	a	-	c
	CENTER	a	a	a	a	a	a	a	a	a	a	-	-	-
<i>opti</i>	AMPLIT	a	a	a	a	a	a	a	a	a	a	a	-	a
	COMPLE	a	a	a	a	a	a	a	a	a	a	a	-	a

(a) All elements. (b) For all beam elements, class-II and class-III only in combination with local coordinate system. (c) Only for bond-slip reinforcements. (d) Only for bond-slip reinforcements modeled by beam elements. (e) Not for bond-slip reinforcements. (f) Only for class-II and class-III beam elements. (-) Not available or not suitable.

MOMENT for curvatures due to concentrated bending moments [§ 8.3.4.3].

DISMOM for curvatures due to distributed bending moments [§ 8.3.4.3].

TRACTI for tractions in structural interface elements [§ 8.3.4.4].

[GLOBAL] *oper* specifies an operation (transformation) to be performed on the primary strains [§ 3.6.1 p. 57].

opti are additional options [§ 8.3 p. 153].

8.3.4.1 Green-Lagrange Strains

Primary strains [§ 47.1]					<i>comp</i> ...						
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	<i>opti</i>	XX	YY	ZZ	XY	YZ	ZX	
STRAIN	TOTAL	GREEN	LOCAL	AMPLIT	ExxA	EyyA	EzzA	GxyA	GyzA	GzxA	
					$\hat{\epsilon}_{xx}$	$\hat{\epsilon}_{yy}$	$\hat{\epsilon}_{zz}$	$\hat{\gamma}_{xy}$	$\hat{\gamma}_{yz}$	$\hat{\gamma}_{zx}$	
					ExxP	EyyP	EzzP	GxyP	GyzP	GzxP	
					$\varphi\epsilon_{xx}$	$\varphi\epsilon_{yy}$	$\varphi\epsilon_{zz}$	$\varphi\gamma_{xy}$	$\varphi\gamma_{yz}$	$\varphi\gamma_{zx}$	
					ExxR	EyyR	EzzR	GxyR	GyzR	GzxR	
					$\Re\epsilon_{xx}$	$\Re\epsilon_{yy}$	$\Re\epsilon_{zz}$	$\Re\gamma_{xy}$	$\Re\gamma_{yz}$	$\Re\gamma_{zx}$	
STRAIN	TOTAL	GREEN	LOCAL	COMPLE	ExxI	EyyI	EzzI	GxyI	GyzI	GzxI	
					$\Im\epsilon_{xx}$	$\Im\epsilon_{yy}$	$\Im\epsilon_{zz}$	$\Im\gamma_{xy}$	$\Im\gamma_{yz}$	$\Im\gamma_{zx}$	
					EXXA	EYYA	EZZA	GXYA	GYZA	GZXA	
					$\hat{\epsilon}_{XX}$	$\hat{\epsilon}_{YY}$	$\hat{\epsilon}_{ZZ}$	$\hat{\gamma}_{XY}$	$\hat{\gamma}_{YZ}$	$\hat{\gamma}_{ZX}$	
					EXXP	EYYP	EZZP	GXYP	GYZP	GZXP	
					$\varphi\epsilon_{XX}$	$\varphi\epsilon_{YY}$	$\varphi\epsilon_{ZZ}$	$\varphi\gamma_{XY}$	$\varphi\gamma_{YZ}$	$\varphi\gamma_{ZX}$	
STRAIN	TOTAL	GREEN	GLOBAL	COMPLE	EXXR	EYXR	EZZR	GXYR	GYZR	GZXR	
					$\Re\epsilon_{XX}$	$\Re\epsilon_{YY}$	$\Re\epsilon_{ZZ}$	$\Re\gamma_{XY}$	$\Re\gamma_{YZ}$	$\Re\gamma_{ZX}$	
					EXXI	EYXI	EZZI	GXYI	GYZI	GZXI	
					$\Im\epsilon_{XX}$	$\Im\epsilon_{YY}$	$\Im\epsilon_{ZZ}$	$\Im\gamma_{XY}$	$\Im\gamma_{YZ}$	$\Im\gamma_{ZX}$	
					EXXR	EYXR	EZZR	GXYR	GYZR	GZXR	
					$\Re\epsilon_{XX}$	$\Re\epsilon_{YY}$	$\Re\epsilon_{ZZ}$	$\Re\gamma_{XY}$	$\Re\gamma_{YZ}$	$\Re\gamma_{ZX}$	

Von Mises strains [§ 47.1.1]					
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	<i>opti</i>	
STRAIN TOTAL GREEN VONMIS	AMPLIT	EeqA			
		$\hat{\epsilon}_{eq}$			
		EeqP			
		$\varphi \epsilon_{eq}$			
STRAIN TOTAL GREEN VONMIS	COMPLE	EeqR			
		$\Re \epsilon_{eq}$			
		EeqI			
		$\Im \epsilon_{eq}$			

8.3.4.2 Deformations

Force deformations					<i>comp</i> ...		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	<i>opti</i>	X	Y	Z
STRAIN TOTAL FORCE	LOCAL	AMPLIT			PxA		
					$\Delta \hat{u}_x$		
					PxP		
					$\Delta_{\varphi} u_x$		
STRAIN TOTAL FORCE	LOCAL	COMPLE			PxR		
					$\Delta_{\Re} u_x$		
					PxI		
					$\Delta_{\Im} u_x$		
STRAIN TOTAL FORCE	GLOBAL	AMPLIT			PXA	PYA	PZA
					$\Delta \hat{u}_X$	$\Delta \hat{u}_Y$	$\Delta \hat{u}_Z$
					PXP	PYP	PZP
					$\Delta_{\varphi} u_X$	$\Delta_{\varphi} u_Y$	$\Delta_{\varphi} u_Z$
STRAIN TOTAL FORCE	GLOBAL	COMPLE			PXR	PYR	PZR
					$\Delta_{\Re} u_X$	$\Delta_{\Re} u_Y$	$\Delta_{\Re} u_Z$
					PXI	PYI	PZI
					$\Delta_{\Im} u_X$	$\Delta_{\Im} u_Y$	$\Delta_{\Im} u_Z$

Generalized strains					<i>comp</i> ...					
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	<i>opti</i>	XX	YY	ZZ	XY	YZ	ZX
STRAIN TOTAL DISFOR	LOCAL	AMPLIT			PxxA	PyyA	PzzA	PxyA	PyzA	PzxA
					$\hat{\Psi}_{xx}$	$\hat{\Psi}_{yy}$	$\hat{\Psi}_{zz}$	$\hat{\Psi}_{xy}$	$\hat{\Psi}_{yz}$	$\hat{\Psi}_{zx}$
					PxxP	PyyP	PzzP	PxyP	PyzP	PzxP
					$\varphi \Psi_{xx}$	$\varphi \Psi_{yy}$	$\varphi \Psi_{zz}$	$\varphi \Psi_{xy}$	$\varphi \Psi_{yz}$	$\varphi \Psi_{zx}$
STRAIN TOTAL DISFOR	LOCAL	COMPLE			PxxR	PyyR	PzzR	PxyR	PyzR	PzxR
					$\Re \Psi_{xx}$	$\Re \Psi_{yy}$	$\Re \Psi_{zz}$	$\Re \Psi_{xy}$	$\Re \Psi_{yz}$	$\Re \Psi_{zx}$
					PxxI	PyyI	PzzI	PxyI	PyzI	PzxI
					$\Im \Psi_{xx}$	$\Im \Psi_{yy}$	$\Im \Psi_{zz}$	$\Im \Psi_{xy}$	$\Im \Psi_{yz}$	$\Im \Psi_{zx}$

8.3.4.3 Curvatures

Concentrated curvatures					<i>comp</i> ...		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	<i>opti</i>	X	Y	Z
STRAIN TOTAL MOMENT	LOCAL	AMPLIT			KxA	KyA	KzA
					$\hat{\kappa}_x$	$\hat{\kappa}_y$	$\hat{\kappa}_z$
					KxP	KyP	KzP
					$\varphi \kappa_x$	$\varphi \kappa_y$	$\varphi \kappa_z$
STRAIN TOTAL MOMENT	LOCAL	COMPLE			KxR	KyR	KzR
					$\Re \kappa_x$	$\Re \kappa_y$	$\Re \kappa_z$
					KxI	KyI	KzI
					$\Im \kappa_x$	$\Im \kappa_y$	$\Im \kappa_z$

Distributed curvatures					<i>comp</i> ...			
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	<i>opti</i>	XX	YY	ZZ	XY
STRAIN	TOTAL	DISMOM	LOCAL	AMPLIT	KxxA	KyyA	KzzA	KxyA
					\hat{K}_{xx}	\hat{K}_{yy}	\hat{K}_{zz}	\hat{K}_{xy}
					KxxP	KyyP	KzzP	KxyP
					φK_{xx}	φK_{yy}	φK_{zz}	φK_{xy}
STRAIN	TOTAL	DISMOM	LOCAL	COMPLE	KxxR	KyyR	KzzR	KxyR
					$\Re K_{xx}$	$\Re K_{yy}$	$\Re K_{zz}$	$\Re K_{xy}$
					KxxI	KyyI	KzzI	KxyI
					$\Im K_{xx}$	$\Im K_{yy}$	$\Im K_{zz}$	$\Im K_{xy}$

8.3.4.4 Relative Displacements of Interface Elements

Note that the labels for the results in the local *xyz* directions are dependent on the interface element type [Vol. Element Library]. Output items related to the normal direction(s) are indicated with ‘N’ and items related to the shear direction(s) are indicated with ‘S’ in their name followed by the letter indicating the corresponding element direction.

Relative displacements					<i>comp</i> ...		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	<i>opti</i>	X	Y	Z
STRAIN	TOTAL	TRACTI	LOCAL	AMPLIT	DUNxA	DUNyA	DUNzA
					DUSxA	DUSyA	DUSzA
					$\Delta \hat{u}_x$	$\Delta \hat{u}_y$	$\Delta \hat{u}_z$
					DUNxP	DUNyP	DUNzP
					DUSxP	DUSyP	DUSzP
					$\varphi \Delta u_x$	$\varphi \Delta u_y$	$\varphi \Delta u_z$
STRAIN	TOTAL	TRACTI	LOCAL	COMPLE	DUNxR	DUNyR	DUNzR
					DUSxR	DUSyR	DUSzR
					$\Re \Delta u_x$	$\Re \Delta u_y$	$\Re \Delta u_z$
					DUNxI	DUNyI	DUNzI
					DUSxI	DUSyI	DUSzI
					$\Im \Delta u_x$	$\Im \Delta u_y$	$\Im \Delta u_z$
STRAIN	TOTAL	TRACTI	GLOBAL	AMPLIT	DUXA	DUYA	DUZA
					$\Delta \hat{u}_X$	$\Delta \hat{u}_Y$	$\Delta \hat{u}_Z$
					DUXP	DUYP	DUZP
					$\varphi \Delta u_X$	$\varphi \Delta u_Y$	$\varphi \Delta u_Z$
STRAIN	TOTAL	TRACTI	GLOBAL	COMPLE	DUXR	DUYR	DUZR
					$\Re \Delta u_X$	$\Re \Delta u_Y$	$\Re \Delta u_Z$
					DUXI	DUYI	DUZI
					$\Im \Delta u_X$	$\Im \Delta u_Y$	$\Im \Delta u_Z$

8.3.5 Stresses

syntax

STRESS	[<i>type_w</i>]	[<i>form_w</i>]	[<i>oper_w</i>]	{ <i>comp_w</i> }	{ <i>loca_w</i> }	{ <i>opti_w</i> }
	TOTAL	CAUCHY	LOCAL		INTPNT	AMPLIT
		FORCE	GLOBAL		NODES	COMPLE
		MOMENT	VONMIS		CENTER	...
		DISFOR				
		DISMOM				
		TRACTI				

STRESS specifies stresses as output item. Table 8.2 on the facing page outlines the availability and applicability of the various stress output options for each of the element families.

[TOTAL] *type* specifies the stress type [§ 3.6.1 p. 57]

Table 8.2: AVAILABILITY OF STRESS OUTPUT FOR FREQUENCY RESPONSE ANALYSIS

<i>item</i>	STRESS	truss	beam	pl. stress	pl. strain	axisymm.	plate bend.	fl. shell	cu. shell	solid	str. interface	spring	p. mass	e. reinfo.	comp. line	comp. surf.
<i>type</i>	TOTAL	a	a	a	a	a	a	a	a	a	a	-	a	a	a	a
<i>form</i>	CAUCHY	a	a	a	a	a	a	a	a	-	-	-	a	-	-	-
	FORCE	a	a	-	-	-	-	-	-	-	a	-	h	a	-	-
	MOMENT	-	a	-	-	-	-	-	-	-	a	-	c	a	-	-
	DISFOR	-	-	a	f	g	a	a	a	-	-	-	-	-	a	-
	DISMOM	-	-	-	f	g	a	a	a	-	-	-	-	-	a	-
	TRACTI	-	-	-	-	-	-	-	-	a	-	-	b	-	-	-
<i>oper</i>	LOCAL	a	a	a	a	a	a	a	a	a	a	-	a	a	a	a
	GLOBAL	a	a	a	a	a	a	a	a	a	-	-	b	-	-	-
	VONMIS	a	a	a	a	a	a	a	a	-	-	-	-	-	-	-
<i>loca</i>	INTPNT	a	e	a	a	a	a	a	a	a	-	-	d	-	-	-
	NODES	a	a	a	a	a	a	a	a	-	-	-	b	a	a	-
	CENTER	a	a	a	a	a	a	a	a	a	-	-	-	-	-	-
<i>opti</i>	AMPLIT	a	a	a	a	a	a	a	a	a	a	-	a	a	a	a
	COMPLE	a	a	a	a	a	a	a	a	a	a	-	a	a	a	a

(a) All elements. (b) Only for bond-slip reinforcements. (c) Only for bond-slip reinforcements modeled by beam elements. (d) Not for bond-slip reinforcements. (e) Only for class-II and class-III beam elements. (f) For infinite shells only. (g) For shells of revolution only. (h) Only for bar reinforcements and bond-slip reinforcements. (-) Not available or not suitable.

[CAUCHY] *form* specifies the stress formulation.

CAUCHY for Cauchy stresses [§ 8.3.5.1].

FORCE for concentrated forces [§ 8.3.5.2].

DISFOR for distributed forces [§ 8.3.5.2].

MOMENT for concentrated bending moments [§ 8.3.5.3].

DISMOM for distributed bending moments [§ 8.3.5.3].

TRACTI for tractions in structural interface elements [§ 8.3.5.4].

oper specifies an operation (transformation) to be performed on the primary stresses [GLOBAL] [§ 3.6.1 p. 57].

opti are additional options [§ 8.3 p. 153].

8.3.5.1 Cauchy Stresses

Primary stresses					<i>comp</i> ...					
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	<i>opti</i>	XX	YY	ZZ	XY	YZ	ZX
STRESS	TOTAL	CAUCHY	LOCAL	AMPLIT	SxxA	SyyA	SzzA	SxyA	SyzA	SzxA
					$\hat{\sigma}_{xx}$	$\hat{\sigma}_{yy}$	$\hat{\sigma}_{zz}$	$\hat{\sigma}_{xy}$	$\hat{\sigma}_{yz}$	$\hat{\sigma}_{zx}$
					SxxP	SyyP	SzzP	SxyP	SyzP	SzxP
					$\varphi\sigma_{xx}$	$\varphi\sigma_{yy}$	$\varphi\sigma_{zz}$	$\varphi\sigma_{xy}$	$\varphi\sigma_{yz}$	$\varphi\sigma_{zx}$
STRESS	TOTAL	CAUCHY	LOCAL	COMPLE	SxxR	SyyR	SzzR	SxyR	SyzR	SzxR
					$\Re\sigma_{xx}$	$\Re\sigma_{yy}$	$\Re\sigma_{zz}$	$\Re\sigma_{xy}$	$\Re\sigma_{yz}$	$\Re\sigma_{zx}$
					SxxI	SyyI	SzzI	SxyI	SyzI	SzxI
					$\Im\sigma_{xx}$	$\Im\sigma_{yy}$	$\Im\sigma_{zz}$	$\Im\sigma_{xy}$	$\Im\sigma_{yz}$	$\Im\sigma_{zx}$
STRESS	TOTAL	CAUCHY	GLOBAL	AMPLIT	SXXA	SYXA	SZZA	SXYA	SYZA	SZXA
					$\hat{\sigma}_{XX}$	$\hat{\sigma}_{YY}$	$\hat{\sigma}_{ZZ}$	$\hat{\sigma}_{XY}$	$\hat{\sigma}_{YZ}$	$\hat{\sigma}_{ZX}$
					SXXP	SYYP	SZZP	SXYP	SYZP	SZXP
					$\varphi\sigma_{XX}$	$\varphi\sigma_{YY}$	$\varphi\sigma_{ZZ}$	$\varphi\sigma_{XY}$	$\varphi\sigma_{YZ}$	$\varphi\sigma_{ZX}$
STRESS	TOTAL	CAUCHY	GLOBAL	COMPLE	SXXR	SYXR	SZZR	SXYR	SYZR	SZXR
					$\Re\sigma_{XX}$	$\Re\sigma_{YY}$	$\Re\sigma_{ZZ}$	$\Re\sigma_{XY}$	$\Re\sigma_{YZ}$	$\Re\sigma_{ZX}$
					SXXI	SYXI	SZZI	SXYI	SYZI	SZXI
					$\Im\sigma_{XX}$	$\Im\sigma_{YY}$	$\Im\sigma_{ZZ}$	$\Im\sigma_{XY}$	$\Im\sigma_{YZ}$	$\Im\sigma_{ZX}$

Von Mises stresses					
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	<i>opti</i>	
STRESS TOTAL CAUCHY VONMIS	AMPLIT	SeqA			
		$\hat{\sigma}_{eq}$			
		SeqP			
		$\varphi\sigma_{eq}$			
STRESS TOTAL CAUCHY VONMIS	COMPLE	SeqR			
		$\Re\sigma_{eq}$			
		SeqI			
		$\Im\sigma_{eq}$			

8.3.5.2 Forces

Concentrated forces					<i>comp</i>	\dots		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	<i>opti</i>	X	Y	Z	
STRESS TOTAL FORCE	LOCAL	AMPLIT			NxA	QyA	QzA	
					\hat{N}_x	\hat{Q}_y	\hat{Q}_z	
					NxP	QyP	QzP	
					φN_x	φQ_y	φQ_z	
STRESS TOTAL FORCE	LOCAL	COMPLE			NxR	NyR	NzR	
					$\Re N_x$	$\Re Q_y$	$\Re Q_z$	
					NxI	NyI	NzI	
					$\Im N_x$	$\Im Q_y$	$\Im Q_z$	
STRESS TOTAL FORCE	GLOBAL	AMPLIT			NXA	NYA	NZA	
					\hat{N}_X	\hat{N}_Y	\hat{N}_Z	
					NXP	NYP	NZP	
					φN_X	φN_Y	φN_Z	
STRESS TOTAL FORCE	GLOBAL	COMPLE			NXR	NYR	NZR	
					$\Re N_X$	$\Re N_Y$	$\Re N_Z$	
					NXI	NYI	NZI	
					$\Im N_X$	$\Im N_Y$	$\Im N_Z$	

Distributed forces					<i>comp</i>	\dots				
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	<i>opti</i>	XX	YY	ZZ	XY	YZ	ZX
STRESS TOTAL DISFOR	LOCAL	AMPLIT			NxxA	NyyA	Nzza	NxyA	Qyza	Qxza
					\hat{n}_{xx}	\hat{n}_{yy}	\hat{n}_{zz}	\hat{n}_{xy}	\hat{q}_{yz}	\hat{q}_{xz}
					NxxP	NyyP	NzzP	NxyP	QyzP	QxzP
					φn_{xx}	φn_{yy}	φn_{zz}	φn_{xy}	φq_{yz}	φq_{xz}
STRESS TOTAL DISFOR	LOCAL	COMPLE			NxxR	NyyR	NzzR	NxyR	QyzR	QxzR
					$\Re n_{xx}$	$\Re n_{yy}$	$\Re n_{zz}$	$\Re n_{xy}$	$\Re q_{yz}$	$\Re q_{xz}$
					NxxI	NyyI	NzzI	NxyI	QyzI	QxzI
					$\Im n_{xx}$	$\Im n_{yy}$	$\Im n_{zz}$	$\Im n_{xy}$	$\Im q_{yz}$	$\Im q_{xz}$

8.3.5.3 Bending moments

Concentrated moments					comp ...		
item	type	form	oper	opti	X	Y	Z
STRESS TOTAL MOMENT LOCAL	AMPLIT				MxA	MyA	MzA
					\hat{M}_x	\hat{M}_y	\hat{M}_z
					MxP	MyP	MzP
					φM_x	φM_y	φM_z
STRESS TOTAL MOMENT LOCAL	COMPLE				MxR	MyR	MzR
					$\Re M_x$	$\Re M_y$	$\Re M_z$
					MxI	MyI	MzI
					$\Im M_x$	$\Im M_y$	$\Im M_z$
STRESS TOTAL MOMENT GLOBAL	AMPLIT				MXA	MYA	MZA
					\hat{M}_X	\hat{M}_Y	\hat{M}_Z
					MXP	MYP	MZP
					φM_X	φM_Y	φM_Z
STRESS TOTAL MOMENT GLOBAL	COMPLE				MXR	MYR	MZR
					$\Re M_X$	$\Re M_Y$	$\Re M_Z$
					MXI	MYI	MZI
					$\Im M_X$	$\Im M_Y$	$\Im M_Z$

Distributed moments					comp ...			
item	type	form	oper	opti	XX	YY	ZZ	XY
STRESS TOTAL DISMOM LOCAL	AMPLIT				MxxA	MyyA	MzzA	MxyA
					\hat{m}_{xx}	\hat{m}_{yy}	\hat{m}_{zz}	\hat{m}_{xy}
					MxxP	MyyP	MzzP	MxyP
					φm_{xx}	φm_{yy}	φm_{zz}	φm_{xy}
STRESS TOTAL DISMOM LOCAL	COMPLE				MxxR	MyyR	MzzR	MxyR
					$\Re m_{xx}$	$\Re m_{yy}$	$\Re m_{zz}$	$\Re m_{xy}$
					MxxI	MyyI	MzzI	MxyI
					$\Im m_{xx}$	$\Im m_{yy}$	$\Im m_{zz}$	$\Im m_{xy}$

8.3.5.4 Traction

Note that the labels for the results in the local xyz directions are dependent on the interface element type [Vol. Element Library]. Output items related to the normal direction(s) are indicated with ‘N’ and items related to the shear direction(s) are indicated with ‘S’ in their name followed by the letter indicating the corresponding element direction.

Traction					comp ...		
item	type	form	oper	opti	X	Y	Z
STRESS TOTAL TRACTI LOCAL	AMPLIT				STNxA	STNyA	STNzA
					STSxA	STSyA	STSzA
					\hat{t}_x	\hat{t}_y	\hat{t}_z
					STNxP	STNyP	STNzP
					STSxP	STSyP	STSzP
					φt_x	φt_y	φt_z
					STNxR	STNyR	STNzR
					STSxR	STSyR	STSzR
STRESS TOTAL TRACTI LOCAL	COMPLE				$\Re t_x$	$\Re t_y$	$\Re t_z$
					STNxI	STNyI	STNzI
					STSxI	STSyI	STSzI
					$\Im t_x$	$\Im t_y$	$\Im t_z$
STRESS TOTAL TRACTI GLOBAL	AMPLIT				STXA	STYA	STZA
					\hat{t}_X	\hat{t}_Y	\hat{t}_Z
					STXP	STYP	STZP
					φt_X	φt_Y	φt_Z
STRESS TOTAL TRACTI GLOBAL	COMPLE				STXR	STYR	STZR
					$\Re t_X$	$\Re t_Y$	$\Re t_Z$
					STXI	STYI	STZI
					$\Im t_X$	$\Im t_Y$	$\Im t_Z$

8.3.6 Nodal Forces and Moments

syntax

FORCE	[<i>type_w</i>]	[<i>form_w</i>]	[<i>oper_w</i>]	{ <i>comp_w</i> }	{ <i>opti_w</i> }
	REACTI	TRANSL	LOCAL		AMPLIT
		ROTATI	GLOBAL		COMPLE
					...

FORCE specifies forces (and moments) in the nodes as output item [§ 47.3 p. 583].

[REACTI] *type* specifies the type [§ 3.6.1 p. 57].

REACTI for the reaction forces in all supported nodes.

[TRANSL] *form* specifies the formulation [§ 3.6.1 p. 57].

[GLOBAL] *oper* specifies an operation (transformation) to be performed on the forces and moments [§ 3.6.1 p. 57].

[all] *comp* selects force or moment components for output. Default is all available components.

opti are additional options [§ 8.3].

Force reactions					<i>comp</i> ...		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	<i>opti</i>	X	Y	Z
FORCE	REACTI	TRANSL	LOCAL	AMPLIT	FBxA	FByA	FBzA
					\hat{F}_x^b	\hat{F}_y^b	\hat{F}_z^b
					FBxP	FBYP	FBzP
					φF_x^b	φF_y^b	φF_z^b
FORCE	REACTI	TRANSL	LOCAL	COMPLE	FBxR	FBYR	FBzR
					$\Re F_x^b$	$\Re F_y^b$	$\Re F_z^b$
					FBxI	FBYI	FBzI
					$\Im F_x^b$	$\Im F_y^b$	$\Im F_z^b$
FORCE	REACTI	TRANSL	GLOBAL	AMPLIT	FBXA	FBYA	FBZA
					\hat{F}_X^b	\hat{F}_Y^b	\hat{F}_Z^b
					FBXP	FBYP	FBZP
					φF_X^b	φF_Y^b	φF_Z^b
FORCE	REACTI	TRANSL	GLOBAL	COMPLE	FBXR	FBYR	FBZR
					$\Re F_X^b$	$\Re F_Y^b$	$\Re F_Z^b$
					FBXI	FBYI	FBZI
					$\Im F_X^b$	$\Im F_Y^b$	$\Im F_Z^b$

Moment reactions					<i>comp</i> ...		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	<i>opti</i>	X	Y	Z
FORCE	REACTI	ROTATI	LOCAL	AMPLIT	MBxA	MByA	MBzA
					\hat{M}_x^b	\hat{M}_y^b	\hat{M}_z^b
					MBxP	MBYP	MBzP
					φM_x^b	φM_y^b	φM_z^b
FORCE	REACTI	ROTATI	LOCAL	COMPLE	MBxR	MBYR	MBzR
					$\Re M_x^b$	$\Re M_y^b$	$\Re M_z^b$
					MBxI	MBYI	MBzI
					$\Im M_x^b$	$\Im M_y^b$	$\Im M_z^b$
FORCE	REACTI	ROTATI	GLOBAL	AMPLIT	MBXA	MBYA	MBZA
					\hat{M}_X^b	\hat{M}_Y^b	\hat{M}_Z^b
					MBXP	MBYP	MBZP
					φM_X^b	φM_Y^b	φM_Z^b
FORCE	REACTI	ROTATI	GLOBAL	COMPLE	MBXR	MBYR	MBZR
					$\Re M_X^b$	$\Re M_Y^b$	$\Re M_Z^b$
					MBXI	MBYI	MBZI
					$\Im M_X^b$	$\Im M_Y^b$	$\Im M_Z^b$

8.3.7 Dynamic Pressures

For fluid–structure interface elements [Vol. *Element Library*] DIANA can calculate and output the dynamic pressures. Positive dynamic pressures indicate additional pressure on the structure, while negative dynamic pressures, i.e. suction pressures, reduce the pressure on the structure. See [§ 48.5 p. 595] for background theory.

syntax

```

FSPRES [ typew ] [ locaw ] { optiw }
      TOTAL   NODES   AMPLIT
                        COMPLE
                        ...

```

FSPRES specifies the dynamic pressures of the fluid–structure interface elements as output item.

type specifies the dynamic pressure type. [TOTAL]

TOTAL for the total dynamic pressures of the fluid–structure interface elements.

loca specifies the location for the strains to be output [§ 3.6.1 p. 58]. [NODES]

opti are additional options [§ 8.3 p. 153]. [COMPLE]

Dynamic pressures				
<i>item</i>	<i>type</i>	<i>loca</i>	<i>opti</i>	
FSPRES	TOTAL	NODES	AMPLIT	PRfsA
				\hat{p}_{fs}
				PRfsP
				φp_{fs}
FSPRES	TOTAL	NODES	COMPLE	PRfsR
				$\Re p_{fs}$
				PRfsI
				$\Im p_{fs}$

Chapter 9

Response Spectrum Analysis

This chapter describes the response spectrum analysis (RSA) with Module SPECTR. A response spectrum analysis is a linear dynamic statistical analysis method which measures the contribution from each eigenmode to indicate the likely maximum seismic response of an elastic structure.

A response spectrum analysis requires the specification of an excitation spectrum, also known as a frequency–acceleration spectrum. The excitation spectrum is specified by a base excitation, nodal, or element load, and a diagram with frequency dependent load multiplication coefficients. All other types of loading will be ignored.

The combined response spectrum of the model for all the selected modes is generated according to the absolute sum (ABS) rule, the standard Complete Quadratic Combination (CQC) rule, an alternative Complete Quadratic Combination with absolute superposition terms (CQCABS) rule, or the Square-Root-of-Sum-of-Squares (SRSS) rule per excitation spectrum.

Multiple excitation spectra may be used in a response spectrum analysis. The response of the individual excitation spectra are combined using either linear combinations or the Square-Root-of-Sum-of-Squares (SRSS) rule. See § 48.3 on page 590 for background theory.

Multiplication coefficient. The multiplication coefficient for each load set and each harmonic is piecewise linearly interpolated from a diagram specified in table 'FREQLO' [§ 6.6.1 p. 111]. If table 'FREQLO' is not specified a default multiplication coefficient equal to 1.0 will be applied for each harmonic on the first existing load set.

Example

file.dat

```
'FREQLO'  
LOAD 1  
FREQUE 3.0 5.0 9.0 12.0 /  
FACTOR 1.0 6.0 4.0 9.0 /  
LOAD 3  
FREQUE 0.0 20.0 /  
FACTOR 2.0 2.0 /
```

In this example the diagram for load set 1 is specified by four points $(f_i, \lambda_i) = (3, 1)$, $(5, 6)$, $(9, 4)$, $(12, 9)$ [Fig. 9.1]. The multiplication factor for load set 3 is equal to 2.0 for the frequency range 0.0 to 20.0.

Fluid–structure interaction analysis. In case of a model containing fluid, structural elements and fluid–structure interface elements, automatically a fluid–structure response spectrum analysis will be performed. DIANA recognizes this element combination and will perform the required preparation to perform a fluid–structure response spectrum analysis. In a response spectrum analysis the added mass effect of the fluid [Eq. (48.100) p. 599] is taken into account. Results will be available for the structural part of the model.

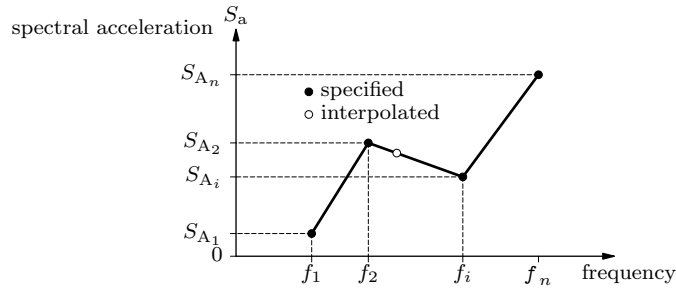


Figure 9.1: Loading multiplication specification

Lumped element matrices may not be used in a fluid–structure response spectrum analysis.

Command sequence. The command sequence for Module SPECTR is as follows.

syntax

```
*SPECTR
[ MODEL ... ]
[ EIGEN ... ]
[ RESPON ... ]
*END
```

MODEL evaluates the finite element model [§ 9.1].

EIGEN solves the eigenvalue problem for a response spectrum analysis [§ 9.2].

RESPON performs the actual response spectrum analysis [§ 9.3].

9.1 Model Evaluation

The MODEL commands customize the evaluation of the finite element model prior to the actual analysis.

syntax

```
BEGIN MODEL
[ OFF ]
[ EVALUA [OFF] ... ]
[ ASSEMB [OFF] ... ]
END MODEL
```

EVALUA checks and evaluates geometric and material properties for elements and reinforcements [§ 3.4 p. 51].

ASSEMB assembles the elements, i.e., creating system degrees of freedom [§ 3.5 p. 54].

9.2 Eigenmodes and Natural Frequencies

To solve the free vibration eigenvalue problem in a response spectrum analysis, i.e., to determine the eigenmodes and natural frequencies, you must give commands in the EIGEN block which are analogous to those for Module EIGEN [§ 31.2 p. 428]. The OFF options suppress the execution of the specified task. This may be useful to save computing time if previously determined results still reside on the FILOS file.

syntax

```

BEGIN EIGEN [ OFF ]
[ FREEVI [ OFF ] ... ]
[ EXECUT [ OFF ] ... ]
[ OUTPUT ... ] ...
END EIGEN

```

FREEVI sets up the free vibration eigenmode analysis [§ 31.2.1 p. 429].

EXECUT specifies how to solve the eigenvalue problem. In particular this involves the type of the solution procedure [§ 31.3 p. 432].

OUTPUT specifies the eigenmodes to be output [§ 31.5.10 p. 438].

9.3 Response Analysis

To perform the actual response spectrum analysis, i.e., to determine the response of a construction to a certain excitation spectrum, you must give commands in the RESPON block.

syntax

```

BEGIN RESPON
[ OUTPUT [ OFF ] ... ] ...
END RESPON

```

OUTPUT executes the actual response spectrum analysis, i.e. combining the modal results and excitation spectra, and specifies the results to be output [§ 9.4].

9.4 Output of Analysis Results

The OUTPUT command block actually performs the Response Spectrum Analysis by applying the excitation spectra specified by a base excitation loads and corresponding diagrams with frequency dependent load multiplication coefficients. See § 48.3 on page 590 for background theory.

With the commands in the OUTPUT block you also get output of results from a response spectrum analysis in the form of maximum and minimum result values representing the extreme values of the result items being calculated over all selected modes and all included load components according to specified superposition rules. The maximum result values are indicated with H and the minimum result values with L at the end of the result label.

In case a static load set is being added the maximum result will be the static solution plus the response spectrum analysis result amplitudes representing the maximum result values that are expected to occur in combination with a static load. The minimum result will be the static solution minus the Response Spectrum Analyses result amplitudes representing the minimum result values to occur in combination with a static load. Consequently, if no additional static load is being applied the minimum results are equal to minus the maximum results.

For principal strains or stresses of individual modes the amplitudes of the major compressive principal stress S3 or strain E3 may yield a larger amplitude than the major tensile principal stress S1 or strain E1. Note that during the modal superposition and load superposition signs of the results are lost due to the superposition rules. Therefore the superposed results, are rearranged such that $S1H \geq S2H \geq S3H \geq S1L \geq S2L \geq S3L$ and $E1H \geq E2H \geq E3H \geq E1L \geq E2L \geq E3L$. Furthermore, for combination with a linear static solution, only the extreme dynamic amplitudes

$S1H = -S3L$ and $E1H = -E3L$ are used to form the widest result envelopes around linear static results $S1$, $S2$ and $S3$ for principal stresses, and $E1$, $E2$ and $E3$ for principal strains.

For the general, analysis type independent options for output of analysis results see § 3.6 on page 55. For modal superposition according to the CQC rule the correlation factors among the eigenmodes are written to the standard output file *file.out* [§ 9.4.9 p. 187].

syntax

```

BEGIN OUTPUT [devicew] [outoptw...] [params]
[ OFF ]
[ BEGIN MODES [ _____ ]
               modesn...
               ALL
[ MODSUP [ _____ ]
          SRSS
          CQC [DAMPIN c1r [ ... cnr ] ]
          CQCABS [DAMPIN c1r [ ... cnr ] ]
          ABS
[ LODSUP [ _____ ]
          SRSS
          LINCOM [FACTOR f1r ... fnr ]
[STATIC [LOAD=loadnrn] [FACTOR=lodfacr ] ]
END MODES ]
[ SELECT ... ]
[ LAYOUT ... ]
itemw ...
DISPLA
VELOCI
ACCELE
STRAIN
STRESS
FORCE
NODFOR
ELMFOR
END OUTPUT

```

OFF suppresses the execution and output of the response spectrum analysis.

MODES specifies a mode selection: *modes* are the mode numbers referring to the calculated modes in the eigenvalue analysis. ALL indicates all calculated modes, which is also the default. Further, the combination rules for the modes and excitation spectra can be defined:

MODSUP specifies how the modal results should be combined for each excitation spectrum:

SRSS gives the square root of sum of squares of the (selected) modal result values, the SRSS rule [Eq. (48.39) p. 591].

CQC gives the standard Complete Quadratic Combination of the (selected) modal result values, the standard CQC rule [Eq. (48.40) p. 591]. DAMPIN specifies modal damping: Values *c1* to *cn* are a series of damping ratios in parts of the critical damping factor c_{crit} [Eq. (48.5) p. 586]; a value of 0.01 indicates 1 %. If only one factor *c1* is specified, then the damping is the same for all natural frequencies of the system. Alternatively, the damping ratios must be specified for all natural frequencies of the system.

[DAMPIN 0.05]
(*ci* ≥ 0)

Modal damping ratios will be used to superpose the results of the modes according to the CQC rule. If no damping ratios are specified zero damping will be assumed.

CQCABS gives an alternative Complete Quadratic Combination of the (selected) modal result values with absolute superposition terms [Eq. (48.41) p. 592]. **DAMPIN** specifies modal damping: Values **c1** to **cn** are a series of damping ratios in parts of the critical damping factor c_{crit} [Eq. (48.5) p. 586]; a value of 0.01 indicates 1 %. If only one factor **c1** is specified, then the damping is the same for all natural frequencies of the system. Alternatively, the damping ratios must be specified for all natural frequencies of the system.

[DAMPIN 0.05]
($c_i \geq 0$)

Modal damping ratios will be used to superpose the results of the modes according to the CQC rule. If no damping ratios are specified zero damping will be assumed.

ABS gives the absolute sum of the (selected) modal result values, the ABS rule [Eq. (48.38) p. 591].

LODSUP specifies how the responses of the individual excitation spectra are combined in case of multiple excitation spectra according to Eurocode 8 EN 1998-1 [65]: [LODSUP SRSS]

SRSS gives the square root of sum of squares of the responses of the excitation spectra, the SRSS rule [Eq. (48.39) p. 591].

LINCOM gives a linear combination of the responses of the excitation spectra. **FACTOR** gives the load contribution factors $f_1 \dots f_n$ in case of multiple excitation spectra. [FACTOR 1.0 0.3 0.3]
($f_i \geq 0$)

Note that the load contribution factors give the requested distribution between the individual excitation spectra and that DIANA will determine the combination leading to the highest response.

STATIC adds a linear static solution to the response spectrum analysis results, which means that the amplitudes of the response spectrum analysis are shifted with the linear static solution from the load set specified by **LOAD=loadnr** with an amplification factor specified by **FACTOR=loadfac**. If no explicit load set is given the first existing load set will be used. In case a static load set is being added the maximum result will be the static solution plus the response spectrum analysis result amplitudes representing the maximum result values that are expected to occur in combination with a static load. The minimum result will be the static solution minus the Response Spectrum Analyses result amplitudes representing the minimum result values to occur in combination with a static load. [loadfac=1.0]

SELECT command block to customize the batch output for model selection see § 3.6.2 on page 59.

LAYOUT commands to customize the layout and style of tabular output [§ 3.6.4 p. 64].

item is the name of the analysis result to be output. See § 3.6.1 on page 56 for complete syntax of this command.

DISPLA for displacements [§ 9.4.1].

VELOCI for displacements [§ 9.4.2].

ACCELE for displacements [§ 9.4.3].

STRAIN for strains [§ 9.4.4].

STRESS for stresses [§ 9.4.5].

FORCE for nodal forces [§ 9.4.6].

NODFOR for element nodal forces [§ 9.4.7].

ELMFOR for internal element forces [§ 9.4.8].

9.4.1 Displacements

syntax

```
DISPLA [ typew ] [ formw ] [ operw ] { compw } { optiw }
```

TOTAL

TRANSL
ROTATI

LOCAL
GLOBAL
NORM

DISPLA specifies maximum and minimum values for the nodal displacements as output item.

[TOTAL] *type* specifies the displacement type.

TOTAL for the total displacements of a structure, i.e., the deformed geometry after a response spectrum analysis.

[TRANSL] *form* specifies the displacement formulation [§ 3.6.1 p. 57].

[GLOBAL] *oper* specifies an operation (transformation) to be performed on the displacements [§ 3.6.1 p. 57]. One specific operation is available:

NORM for the length of the displacement vector. Only the translational terms will be used to calculate the norm.

[all] *comp* selects displacement components for output. Default is all available components.

opti are additional options [§ 3.6.1 p. 58].

Displacements				<i>comp</i> ...		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	X	Y	Z
DISPLA	TOTAL	TRANSL	LOCAL	DtxH	DtyH	DtzH
				u_x	u_y	u_z
				DtxL	DtyL	DtzL
				u_x	u_y	u_z
DISPLA	TOTAL	TRANSL	GLOBAL	DtXH	DtYH	DtZH
				u_X	u_Y	u_Z
				DtXL	DtYL	DtZL
				u_X	u_Y	u_Z
DISPLA	TOTAL	ROTATI	LOCAL	DrxH	DryH	DrzH
				ϕ_x	ϕ_y	ϕ_z
				DrxL	DryL	DrzL
				ϕ_x	ϕ_y	ϕ_z
DISPLA	TOTAL	ROTATI	GLOBAL	DrXH	DrYH	DrZH
				ϕ_X	ϕ_Y	ϕ_Z
				DrXL	DrYL	DrZL
				ϕ_X	ϕ_Y	ϕ_Z

Displacements				
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	
DISPLA	TOTAL	TRANSL	NORM	DtXYZH
				$\ u\ $
				DtXYZL
				$\ u\ $

9.4.2 Velocities

syntax

```

VELOCI [ typew ] [ formw ] [ operw ] { compw } { optiw }
      TOTAL      TRANSL  LOCAL
              ROTATI  GLOBAL
                  NORM

```

VELOCI specifies maximum and minimum values for the nodal velocities as output item.

type specifies the velocity type. [TOTAL]

TOTAL for the total velocities of a structure.

form specifies the velocity formulation [§ 3.6.1 p. 57]. [TRANSL]

oper specifies an operation (transformation) to be performed on the velocities [§ 3.6.1 p. 57]. One specific operation is available: [GLOBAL]

NORM for the length of the velocity vector. Only the translational terms will be used to calculate the norm.

comp selects velocity components for output. Default is all available components. [all]

opti are additional options [§ 3.6.1 p. 58].

Velocities				<i>comp</i> ...		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	X	Y	Z
VELOCI	TOTAL	TRANSL	LOCAL	VtxH VtyH VtzH		
				\dot{u}_x \dot{u}_y \dot{u}_z		
				VtxL VtyL VtzL		
				\dot{u}_x \dot{u}_y \dot{u}_z		
VELOCI	TOTAL	TRANSL	GLOBAL	VtXH VtYH VtZH		
				\dot{u}_X \dot{u}_Y \dot{u}_Z		
				VtXL VtYL VtZL		
				\dot{u}_X \dot{u}_Y \dot{u}_Z		
VELOCI	TOTAL	ROTATI	LOCAL	VrxH VryH VrzH		
				$\dot{\phi}_x$ $\dot{\phi}_y$ $\dot{\phi}_z$		
				VrxL VryL VrzL		
				$\dot{\phi}_x$ $\dot{\phi}_y$ $\dot{\phi}_z$		
VELOCI	TOTAL	ROTATI	GLOBAL	VrXH VrYH VrZH		
				$\dot{\phi}_X$ $\dot{\phi}_Y$ $\dot{\phi}_Z$		
				VrXL VrYL VrZL		
				$\dot{\phi}_X$ $\dot{\phi}_Y$ $\dot{\phi}_Z$		

Velocities				
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	
VELOCI	TOTAL	TRANSL	NORM	VtXYZH
				$\ \dot{u}\ $
				VtXYZL
				$\ \dot{u}\ $

9.4.3 Accelerations

syntax

```

ACCELE [ typew ] [ formw ] [ operw ] { compw } { optiw }
      TOTAL      TRANSL  LOCAL
              ROTATI  GLOBAL
                  NORM

```

ACCELE specifies maximum and minimum values for the nodal accelerations as output item.

[TOTAL] *type* specifies the acceleration type.

TOTAL for the total accelerations of a structure.

[TRANSL] *form* specifies the acceleration formulation [§ 3.6.1 p. 57].

[GLOBAL] *oper* specifies an operation (transformation) to be performed on the accelerations [§ 3.6.1 p. 57]. One specific operation is available:

NORM for the length of the acceleration vector. Only the translational terms will be used to calculate the norm.

[all] *comp* selects acceleration components for output. Default is all available components.

opti are additional options [§ 3.6.1 p. 58].

Accelerations				<i>comp</i> ...		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	X	Y	Z
ACCELE	TOTAL	TRANSL	LOCAL	AtxH	AtyH	AtzH
				\ddot{u}_x	\ddot{u}_y	\ddot{u}_z
				AtxL	AtyL	AtzL
				\ddot{u}_x	\ddot{u}_y	\ddot{u}_z
ACCELE	TOTAL	TRANSL	GLOBAL	AtXH	AtYH	AtZH
				\ddot{u}_X	\ddot{u}_Y	\ddot{u}_Z
				AtXL	AtYL	AtZL
				\ddot{u}_X	\ddot{u}_Y	\ddot{u}_Z
ACCELE	TOTAL	ROTATI	LOCAL	ArxH	AryH	ArzH
				$\ddot{\phi}_x$	$\ddot{\phi}_y$	$\ddot{\phi}_z$
				ArxL	AryL	ArzL
				$\ddot{\phi}_x$	$\ddot{\phi}_y$	$\ddot{\phi}_z$
ACCELE	TOTAL	ROTATI	GLOBAL	ArXH	ArYH	ArZH
				$\ddot{\phi}_X$	$\ddot{\phi}_Y$	$\ddot{\phi}_Z$
				ArXL	ArYL	ArZL
				$\ddot{\phi}_X$	$\ddot{\phi}_Y$	$\ddot{\phi}_Z$

Accelerations				
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	
ACCELE	TOTAL	TRANSL	NORM	AtXYZH
				$\ \ddot{u}\ $
				AtXYZL
				$\ \ddot{u}\ $

9.4.4 Strains

syntax

STRAIN	TOTAL	[<i>form_w</i>]	[<i>oper_w</i>]	{ <i>comp_w</i> }	{ <i>loca_w</i> }	{ <i>opti_w</i> }
		GREEN	LOCAL		INTPNT	SMOOTH
		FORCE	GLOBAL		NODES	ERROR
		MOMENT	PRINCI		CENTER	...
		DISFOR	VONMIS			
		DISMOM	REAXES			
		TRACTI	VOLUME			
		DISSEI				

STRAIN specifies strain amplitudes as output item. Table 9.1 on the facing page outlines the availability and applicability of the various strain output options for each of the element families.

Table 9.1: AVAILABILITY OF STRAIN OUTPUT FOR RESPONSE SPECTRUM ANALYSIS

<i>item</i>	STRAIN TOTAL	truss	beam	pl. stress	pl. strain	axisymm.	plate bend.	fl. shell	cu. shell	solid	str. interface	spring	p. mass	e. reinfo.
<i>type</i>	TOTAL	a	a	a	a	a	a	a	a	a	a	-	a	
<i>hline</i>	fREEN	a	a	a	a	-	a	a	a	-	-	-	a	
	FORCE	-	c	-	-	-	-	-	-	-	-	a	-	e
	MOMENT	-	c	-	-	-	-	-	-	-	-	a	-	g
<i>form</i>	DISFOR	-	-	-	-	a	a	-	-	-	-	-	-	-
	DISMOM	-	-	-	-	a	a	-	-	-	-	-	-	-
	TRACTI	-	-	-	-	-	-	-	-	a	-	-	e	
	DISSEI	-	-	-	-	-	-	-	-	d	-	-	-	-
	LOCAL	a	a	a	a	a	a	a	a	a	a	-	a	
	GLOBAL	a	a	a	a	a	a	a	a	a	a	-	e	
<i>oper</i>	PRINCI	a	a	a	a	a	a	a	a	-	-	-	-	
	VONMIS	a	a	a	a	a	a	a	a	-	-	-	-	
	REAXES	-	-	a	-	a	a	a	a	-	-	-	-	
	VOLUME	a	a	a	a	a	a	a	a	-	-	-	-	
	INTPNT	a	h	a	a	a	a	a	a	a	-	-	g	
	NODES	a	a	a	a	a	a	a	a	-	a	-	e	
<i>loca</i>	CENTER	a	a	a	a	a	a	a	a	a	-	-	-	
	SMOOTH	a	a	a	a	-	-	a	a	-	-	-	-	
<i>opti</i>	ERROR	a	a	a	a	-	-	a	a	-	-	-	-	

(a) All elements. (b) Not for elements with orthotropic geometry. (c) For all beam elements, class-II and class-III only in combination with local coordinate system. (d) Only for plane structural interface elements. (e) Only for bond-slip reinforcements. (f) Only for bond-slip reinforcements modeled by beam elements. (g) Not for bond-slip reinforcements. (h) Only for class-II and class-III beam elements. (-) Not available or not suitable.

TOTAL specifies total strains as output type [§ 3.6.1 p. 57].

form specifies the strain formulation.

[GREEN]

GREEN for Green-Lagrange strains [§ 9.4.4.1].

For principal strains of individual modes the amplitudes of the major compressive principal strain E3 may yield a larger amplitude than the major tensile principal strain E1. Note that during the modal superposition and load superposition signs of the results are lost due to the superposition rules. Therefore the superposed results, are rearranged such that $E1H \geq E2H \geq E3H \geq E1L \geq E2L \geq E3L$. Furthermore, for combination with a linear static solution, only the extreme dynamic amplitudes $E1H = -E3L$ are used to form the widest result envelopes around linear static principal strain results E1, E2 and E3.

FORCE for deformations due to normal and shear forces [§ 9.4.4.2].

DISFOR for generalized strains [§ 9.4.4.2].

MOMENT for curvatures due to a concentrated bending moment [§ 9.4.4.3].

DISMOM for curvatures due to a distributed bending moment [§ 9.4.4.3].

TRACTI for tractions in structural interface elements [§ 9.4.4.4].

DISSEI for distributed seismic moments in plane structural interface elements [§ 9.4.4.5].

oper specifies an operation (transformation) to be performed on the primary strains [GLOBAL] [§ 3.6.1 p. 57].

comp selects strain components for output. Default is all available components.

[all]

loca specifies the location for the strains to be output [§ 3.6.1 p. 58].

opti are additional options [§ 3.6.1 p. 58].

9.4.4.1 Green–Lagrange Strains

For principal strains of individual modes the amplitudes of the major compressive principal strain $E3$ may yield a larger amplitude than the major tensile principal strain $E1$. Note that during the modal superposition and load superposition signs of the results are lost due to the superposition rules. Therefore the superposed results, are rearranged such that $E1H \geq E2H \geq E3H \geq E1L \geq E2L \geq E3L$. Furthermore, for combination with a linear static solution, only the extreme dynamic amplitudes $E1H = -E3L$ are used to form the widest result envelopes around linear static principal strain results $E1$, $E2$ and $E3$.

Primary strains [§ 47.1]				comp ...								
item	type	form	oper	XX	YY	ZZ	XY	YZ	ZX	1	2	3
STRAIN TOTAL	GREEN LOCAL			ExxH	EyyH	EzzH	GxyH	GyzH	GzxH			
				ε_{xx}	ε_{yy}	ε_{zz}	γ_{xy}	γ_{yz}	γ_{zx}			
				ExxL	EyyL	EzzL	GxyL	GyzL	GzxL			
				ε_{xx}	ε_{yy}	ε_{zz}	γ_{xy}	γ_{yz}	γ_{zx}			
STRAIN TOTAL	GREEN GLOBAL			EXXH	EYYH	EZZH	GXYH	GYZH	GZXH			
				ε_{XX}	ε_{YY}	ε_{ZZ}	γ_{XY}	γ_{YZ}	γ_{ZX}			
				EXXL	EYYL	EZZL	GXYL	GYZL	GZXL			
				ε_{XX}	ε_{YY}	ε_{ZZ}	γ_{XY}	γ_{YZ}	γ_{ZX}			
STRAIN TOTAL	GREEN PRINCI									E1H	E2H	E3H
										ε_1	ε_2	ε_3
										E1L	E2L	E3L
										ε_1	ε_2	ε_3
STRAIN TOTAL	GREEN REAXES									E1RAH	E2RAH	
										ε_1^a	ε_2^a	
										E1RAL	E2RAL	
										ε_1^a	ε_2^a	

Von Mises strain [§ 47.1.1]				
item	type	form	oper	
STRAIN TOTAL	GREEN VONMIS			EeqH
				ε_{eq}
				EeqL
				ε_{eq}

Volumetric strain [§ 47.1.3]				
item	type	form	oper	
STRAIN TOTAL	GREEN VOLUME			EvolH
				ε_{vol}
				EvolL
				ε_{vol}

9.4.4.2 Deformations

Force deformations				comp ...		
item	type	form	oper	X	Y	Z
STRAIN TOTAL	FORCE LOCAL			PxH		
				Δu_x		
				PxL		
				Δu_x		
STRAIN TOTAL	FORCE GLOBAL			PXH	PYH	PZH
				Δu_X	Δu_Y	Δu_Z
				PXL	PYL	PZL
				Δu_X	Δu_Y	Δu_Z

Generalized strains				comp ...						
itm	type	form	oper	XX	YY	ZZ	XY	YZ	ZX	1 2
STRAIN TOTAL DISFOR LOCAL				PxxH	PyyH	PzzH	PxyH	PyzH	PzxH	
				Ψ_{xx}	Ψ_{yy}	Ψ_{zz}	Ψ_{xy}	Ψ_{yz}	Ψ_{zx}	
				PxxL	PyyL	PzzL	PxyL	PyzL	PzxL	
				Ψ_{xx}	Ψ_{yy}	Ψ_{zz}	Ψ_{xy}	Ψ_{yz}	Ψ_{zx}	
STRAIN TOTAL DISFOR REAXES										P1RAH P2RAH
										Ψ_1^a Ψ_2^a
										P1RAL P2RAL
										Ψ_1^a Ψ_2^a

9.4.4.3 Curvatures

Concentrated curvatures				comp ...		
item	type	form	oper	X	Y	Z
STRAIN TOTAL MOMENT LOCAL				KxH	KyH	KzH
				κ_x	κ_y	κ_z
				KxL	KyL	KzL
				κ_x	κ_y	κ_z

Distributed curvatures				comp ...					
item	type	form	oper	XX	YY	ZZ	XY	1	2
STRAIN TOTAL DISMOM LOCAL				KxxH	KyyH	KzzH	KxyH		
				κ_{xx}	κ_{yy}	κ_{zz}	κ_{xy}		
				KxxL	KyyL	KzzL	KxyL		
				κ_{xx}	κ_{yy}	κ_{zz}	κ_{xy}		
STRAIN TOTAL DISMOM REAXES								K1RAH K2RAH	
								κ_1^a	κ_2^a
								K1RAL K2RAL	
								κ_1^a	κ_2^a

9.4.4.4 Relative Displacements of Interface Elements

Note that the labels for the results in the local xyz directions are dependent on the interface element type [Vol. Element Library]. Output items related to the normal direction(s) are indicated with ‘N’ and items related to the shear direction(s) are indicated with ‘S’ in their name followed by the letter indicating the corresponding element direction.

Relative displacements				comp ...		
item	type	form	oper	X	Y	Z
STRAIN TOTAL TRACTI LOCAL				DUNxH	DUNyH	DUNzH
				DUSxH	DUSyH	DUSzH
				Δu_x	Δu_y	Δu_z
				DUNxL	DUNyL	DUNzL
				DUSxL	DUSyL	DUSzL
				Δu_x	Δu_y	Δu_z
STRAIN TOTAL TRACTI GLOBAL				DUXH	DUYH	DUZH
				Δu_X	Δu_Y	Δu_Z
				DUXL	DUYL	DUZL
				Δu_X	Δu_Y	Δu_Z

9.4.4.5 Distributed Seismic Moment

Distributed seismic moment [§ 47.1.4]						
item	type	form				
STRAIN TOTAL DISSEI				PSTOTH	PSNEGH	PSPOSH
				\mathcal{P}_S	\mathcal{P}_{S-}	\mathcal{P}_{S+}
				PSTOTL	PSNEGL	PSPOS L
				\mathcal{P}_S	\mathcal{P}_{S-}	\mathcal{P}_{S+}

9.4.5 Stresses

syntax

STRESS TOTAL	[<i>form_w</i>]	[<i>oper_w</i>]	{ <i>comp_w</i> }	{ <i>loca_w</i> }	{ <i>opti_w</i> }
	CAUCHY	LOCAL		INTPNT	NOBOND
	FORCE	GLOBAL		NODES	SMOOTH
	MOMENT	PRINCI		CENTER	ERROR
	DISFOR	VONMIS			...
	DISMOM	INVARI			
	TRACTI	REINFO			
	GRADIE	REAXES			
	SHEAR	MAXSHR			

Table 9.2: AVAILABILITY OF STRESS OUTPUT FOR RESPONSE SPECTRUM ANALYSIS

<i>item</i>	STRESS	truss	beam	pl. stress	pl. strain	axisymm.	plate bend.	fl. shell	cu. shell	solid	str. interface	spring	p. mass	e. reinfo.	comp. line	comp. surf.
<i>type</i>	TOTAL	a	a	a	a	a	a	a	a	a	a	a	-	a	a	a
<i>form</i>	CAUCHY	a	a	a	a	a	b	b	a	a	-	-	-	a	-	-
	FORCE	a	a	-	-	-	-	-	-	-	-	a	-	k	a	-
	MOMENT	-	a	-	-	-	-	-	-	-	-	a	-	h	a	-
	DISFOR	-	-	a	c	d	a	a	a	-	-	-	-	-	-	a
	DISMOM	-	-	-	c	d	a	a	a	-	-	-	-	-	-	a
	TRACTI	-	-	-	-	-	-	-	-	a	-	-	-	g	-	-
	GRADIE	-	-	-	-	-	-	-	-	-	-	-	-	e	-	-
	SHEAR	-	-	-	-	-	-	-	-	a	-	-	-	e	-	-
<i>oper</i>	LOCAL	a	a	a	a	a	a	a	a	a	a	-	-	a	a	a
	GLOBAL	a	a	a	a	a	a	a	a	a	a	f	-	g	-	-
	PRINCI	a	a	a	a	a	a	a	a	a	-	-	-	-	-	-
	VONMIS	a	a	a	a	a	a	a	a	a	-	-	-	-	-	-
	INVARI	-	-	a	a	a	-	-	a	-	-	-	-	-	-	-
	REINFO	-	-	a	-	-	a	a	a	-	-	-	-	-	-	-
	REAXES	-	-	a	-	-	a	a	a	-	-	-	-	-	-	-
	MAXSHR	-	-	a	a	a	-	-	a	-	-	-	-	-	-	-
<i>loca</i>	INTPNT	a	j	a	a	a	a	a	a	a	-	-	-	i	-	-
	NODES	a	a	a	a	a	a	a	a	-	f	-	g	a	a	
	CENTER	a	a	a	a	a	a	a	a	a	-	-	-	-	-	-
<i>opti</i>	NOBOND	-	a	-	-	-	-	-	a	-	-	-	-	-	-	-
	SMOOTH	a	a	a	a	a	-	-	a	a	-	-	-	-	-	-
	ERROR	a	a	a	a	a	-	-	a	a	-	-	-	-	-	-

(a) All elements. (b) Not for elements with orthotropic geometry. (c) For infinite shells only. (d) For shells of revolution only. (e) Only for bar reinforcements. (f) Only for base springs. (g) Only for bond-slip reinforcements. (h) Only for bond-slip reinforcements modeled by beam elements. (i) Not for bond-slip reinforcements. (j) Only for class-II and class-III beam elements. (k) Only for bar reinforcements and bond-slip reinforcements. (-) Not available or not suitable.

STRESS specifies stress amplitudes as output item. Table 9.2 outlines the availability and applicability of the various stress output options for each of the element families.

TOTAL specifies total stresses as output type [§ 3.6.1 p. 57].

[CAUCHY] specifies the stress formulation.

CAUCHY for Cauchy stresses [§ 9.4.5.1].

For principal stresses of individual modes the amplitudes of the major compressive principal stress S3 may yield a larger amplitude than the major tensile principal stress S1. Note that during the modal superposition and load superposition signs of the results are lost due to the

superposition rules. Therefore the superposed results, are rearranged such that $S1H \geq S2H \geq S3H \geq S1L \geq S2L \geq S3L$. Furthermore, for combination with a linear static solution, only the extreme dynamic amplitudes $S1H = -S3L$ are used to form the widest result envelopes around linear static principal stress results $S1$, $S2$ and $S3$.

FORCE for concentrated forces and tractions [§ 9.4.5.2].

MOMENT for concentrated bending moments [§ 9.4.5.3].

DISFOR for distributed forces [§ 9.4.5.2].

DISMOM for distributed bending moments [§ 9.4.5.3].

TRACTI for tractions in structural interface elements [§ 9.4.5.4].

GRADIE for gradients of stresses in reinforcement bars [§ 4.2.4.5 p. 87].

SHEAR for shear stress in the reinforcement mother element connection [§ 4.2.4.6 p. 87].

oper specifies an operation (transformation) to be performed on the primary stresses [GLOBAL] [§ 3.6.1 p. 57]. A specific operation is available for the maximum shear stress:

MAXSHR gives the maximum shear stress $\tau_{\max} = \frac{\sigma_1 - \sigma_3}{2}$ with σ_1 and σ_3 the highest and lowest principal stress respectively. For plane stress elements the maximum shear stress τ_{\max} is defined as $\tau_{\max} = \frac{\sigma_1 - \sigma_2}{2}$ because the third principal stress σ_3 is zero by definition.

comp selects stress components for output. Default is all available components. [all]

loca specifies the location for the stresses to be output [§ 3.6.1 p. 58].

opti are additional options [§ 3.6.1 p. 58].

9.4.5.1 Cauchy Stresses

For principal stresses of individual modes the amplitudes of the major compressive principal stress $S3$ may yield a larger amplitude than the major tensile principal stress $S1$. Note that during the modal superposition and load superposition signs of the results are lost due to the superposition rules. Therefore the superposed results, are rearranged such that $S1H \geq S2H \geq S3H \geq S1L \geq S2L \geq S3L$. Furthermore, for combination with a linear static solution, only the extreme dynamic amplitudes $S1H = -S3L$ are used to form the widest result envelopes around linear static principal stress results $S1$, $S2$ and $S3$.

Primary stresses [§ 47.2]				<i>comp</i> ...								
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	XX	YY	ZZ	XY	YZ	ZX	1	2	3
STRESS TOTAL	CAUCHY	LOCAL		SxxH	SyyH	SzzH	SxyH	SyzH	SzxH			
				σ_{xx}	σ_{yy}	σ_{zz}	σ_{xy}	σ_{yz}	σ_{zx}			
				SxxL	SyyL	SzzL	SxyL	SyzL	SzxL			
				σ_{xx}	σ_{yy}	σ_{zz}	σ_{xy}	σ_{yz}	σ_{zx}			
STRESS TOTAL	CAUCHY	GLOBAL		SXXH	SYXH	SZZH	SXYH	SYZH	SZXH			
				σ_{XX}	σ_{YY}	σ_{ZZ}	σ_{XY}	σ_{YZ}	σ_{ZX}			
				SXXL	SYXL	SZZL	SXYL	SYZL	SZXL			
				σ_{XX}	σ_{YY}	σ_{ZZ}	σ_{XY}	σ_{YZ}	σ_{ZX}			
STRESS TOTAL	CAUCHY	PRINCI								S1H	S2H	S3H
										σ_1	σ_2	σ_3
										S1L	S2L	S3L
										σ_1	σ_2	σ_3
STRESS TOTAL	CAUCHY	REAXES								S1RAH	S2RAH	
										σ_1^a	σ_2^a	
										S1RAL	S2RAL	
										σ_1^a	σ_2^a	

Von Mises stress [§ 47.2.1]			
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>
STRESS TOTAL	CAUCHY	VONMIS	SeqH
			σ_{eq}
			SeqL
			σ_{eq}

Stress invariants [§ 47.2.5]				<i>comp</i> ...		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	P	Q	LODE
STRESS TOTAL	CAUCHY	INVARI		PH	QH	LodeH
				p'	q	θ
				PL	QL	LodeL
				p'	q	θ

Maximum shear stress			
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>
STRESS TOTAL	CAUCHY	MAXSHR	TmaxH
			τ_{max}
			TmaxL
			τ_{max}

9.4.5.2 Forces

Concentrated forces				<i>comp</i> ...		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	X	Y	Z
STRESS TOTAL	FORCE	LOCAL		NxH	QyH	QzH
				N_x	Q_y	Q_z
				NxL	QyL	QzL
				N_x	Q_y	Q_z
STRESS TOTAL	FORCE	GLOBAL		NXH	NYH	NZH
				N_X	N_Y	N_Z
				NXL	NYL	NZL
				N_X	N_Y	N_Z

Distributed forces				<i>comp</i> ...							
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	XX	YY	ZZ	XY	YZ	ZX	1	2
STRESS TOTAL	DISFOR	LOCAL		NxxH	NyyH	NzzH	NxyH	QyzH	QxzH		
				n_{xx}	n_{yy}	n_{zz}	n_{xy}	q_{yz}	q_{xz}		
				NxxL	NyyL	NzzL	NxyL	QyzL	QxzL		
				n_{xx}	n_{yy}	n_{zz}	n_{xy}	q_{yz}	q_{xz}		
STRESS TOTAL	DISFOR	REAXES								N1RAH	N2RAH
										n_1^a	n_2^a
										N1RAL	N2RAL
										n_1^a	n_2^a

Reinforcement forces [§ 47.2.6]				<i>comp</i> ...				
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	1R	2R	1RC	2RC	QT
STRESS TOTAL	DISFOR	REINFO		N1RH	N2RH	N1RCH	N2RCH	QTH
				n_1'	n_2'	$n_1'^c$	$n_2'^c$	q'
				N1RL	N2RL	N1RCL	N2RCL	QTL
				n_1'	n_2'	$n_1'^c$	$n_2'^c$	q'

9.4.5.3 Bending moments

Concentrated moments				comp ...		
item	type	form	oper	X	Y	Z
STRESS	TOTAL	MOMENT	LOCAL	MxH	MyH	MzH
				M_x	M_y	M_z
				MxL	MyL	MzL
				M_x	M_y	M_z
STRESS	TOTAL	MOMENT	GLOBAL	MXH	MYH	MZH
				M_X	M_Y	M_Z
				MXL	MYL	MZL
				M_X	M_Y	M_Z

Distributed moments				comp ...					
item	type	form	oper	XX	YY	ZZ	XY	1	2
STRESS	TOTAL	DISMOM	LOCAL	MxxH	MyyH	MzzH	MxyH		
				m_{xx}	m_{yy}	m_{zz}	m_{xy}		
				MxxL	MyyL	MzzL	MxyL		
				m_{xx}	m_{yy}	m_{zz}	m_{xy}		
STRESS	TOTAL	DISMOM	REAXES					M1RAH	M2RAH
								m_1^a	m_2^a
								M1RAL	M2RAL
								m_1^a	m_2^a

Reinforcement moments [§ 47.2.6]				comp ...	
item	type	form	oper	1R	2R
STRESS	TOTAL	DISMOM	REINFO	M1RH	M2RH
				m'_1	m'_2
				M1RL	M2RL
				m'_1	m'_2

9.4.5.4 Traction

Note that the labels for the results in the local xyz directions are dependent on the interface element type [Vol. Element Library]. Output items related to the normal direction(s) are indicated with 'N' and items related to the shear direction(s) are indicated with 'S' in their name followed by the letter indicating the corresponding element direction.

Traction				comp ...		
item	type	form	oper	X	Y	Z
STRESS	TOTAL	TRACTI	LOCAL	STNxH	STNyH	STNzH
				STsH	STsH	STsH
				t_x	t_y	t_z
				STNxL	STNyL	STNzL
				STsL	STsL	STsL
				t_x	t_y	t_z
STRESS	TOTAL	TRACTI	GLOBAL	STXH	STYH	STZH
				t_X	t_Y	t_Z
				STXL	STYL	STZL
				t_X	t_Y	t_Z

9.4.6 Nodal Forces

syntax

FORCE [$type_w$] [$form_w$] [$oper_w$] { $comp_w$ } { $opti_w$ }

REACTI TRANSL LOCAL

RESIDU ROTATI GLOBAL

NORM

FORCE specifies maximum and minimum forces and moments in the nodes as output item [§ 47.3 p. 583].

[REACTI] **type** specifies the type of the nodal forces.

REACTI for the reaction forces in all supported nodes.

RESIDU for the residual forces.

[TRANSL] **form** specifies the formulation [§ 3.6.1 p. 57]

[GLOBAL] **oper** specifies an operation (transformation) to be performed on the forces and moments [§ 3.6.1 p. 57]. One specific operation is available:

NORM for the length of the force vector. Only the translational terms will be used to calculate the norm.

[all] **comp** selects force or moment components for output. Default is all available components.

opti are additional options [§ 3.6.1 p. 58].

Support reactions [§ 47.3.3]				<i>comp</i> ...		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	X	Y	Z
FORCE	REACTI	TRANSL	LOCAL	FBxH	FByH	FBzH
				F_x^b	F_y^b	F_z^b
				FBxL	FByL	FBzL
				F_x^b	F_y^b	F_z^b
FORCE	REACTI	TRANSL	GLOBAL	FBXH	FBYH	FBZH
				F_X^b	F_Y^b	F_Z^b
				FBXL	FBYL	FBZL
				F_X^b	F_Y^b	F_Z^b
FORCE	REACTI	ROTATI	LOCAL	MBxH	MByH	MBzH
				M_x^b	M_y^b	M_z^b
				MBxL	MByL	MBzL
				M_x^b	M_y^b	M_z^b
FORCE	REACTI	ROTATI	GLOBAL	MBXH	MBYH	MBZH
				M_X^b	M_Y^b	M_Z^b
				MBXL	MBYL	MBZL
				M_X^b	M_Y^b	M_Z^b

Support reactions				
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	
FORCE	REACTI	TRANSL	NORM	FBXYZH
				$\ F^b\ $
				FBXYZL
				$\ F^b\ $

Residuals [§ 47.3.3]				<i>comp</i> ...		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	X	Y	Z
FORCE	RESIDU	TRANSL	LOCAL	FRxH	FRyH	FRzH
				F_x^r	F_y^r	F_z^r
				FRxL	FRyL	FRzL
				F_x^r	F_y^r	F_z^r
FORCE	RESIDU	TRANSL	GLOBAL	FRXH	FRYH	FRZH
				F_X^r	F_Y^r	F_Z^r
				FRXL	FRYL	FRZL
				F_X^r	F_Y^r	F_Z^r
FORCE	RESIDU	ROTATI	LOCAL	MRxH	MRyH	MRzH
				M_x^r	M_y^r	M_z^r
				MRxL	MRyL	MRzL
				M_x^r	M_y^r	M_z^r
FORCE	RESIDU	ROTATI	GLOBAL	MRXH	MRYH	MRZH
				M_X^r	M_Y^r	M_Z^r
				MRXL	MRYL	MRZL
				M_X^r	M_Y^r	M_Z^r

Residuals				
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	
FORCE	RESIDU	TRANSL	NORM	FRXYZH
				$\ F^r\ $
				FRXYZL
				$\ F^r\ $

9.4.7 Internal Nodal Element Forces

syntax

NODFOR [*type_w*] [*form_w*] [*oper_w*] { *comp_w* } { *opti_w* }

ELEMEN TRANSL LOCAL

REINFO ROTATI GLOBAL

TOTAL

NODFOR specifies maximum and minimum internal nodal element forces and moments in the nodes as output item. This command gives the contributions of the element or reinforcement internal nodal forces (or both), to a certain node. A selection of elements which form a ‘section’ of the model gives the total internal forces that act on that ‘section’.

type specifies the forces type. [TOTAL]

ELEMEN for contribution of elements only.

REINFO for contribution of embedded reinforcements only.

TOTAL for contribution of both elements and embedded reinforcements.

form specifies the formulation [§ 3.6.1 p. 57]. [TRANSL]

oper specifies an operation (transformation) to be performed on the forces and moments [§ 3.6.1 p. 57]. [GLOBAL]

comp selects force or moment components for output. Default is all available components. [all]

opti are additional options [§ 3.6.1 p. 58].

Internal nodal total forces [§ 47.3.1]				comp ...		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	X	Y	Z
NODFOR TOTAL TRANSL LOCAL				FNTxH	FNTyH	FNTzH
				F_x^{nt}	F_y^{nt}	F_z^{nt}
				FNTxL	FNTyL	FNTzL
				F_x^{nt}	F_y^{nt}	F_z^{nt}
NODFOR TOTAL TRANSL GLOBAL				FNTXH	FNTYH	FNTZH
				F_X^{nt}	F_Y^{nt}	F_Z^{nt}
				FNTXL	FNTYL	FNTZL
				F_X^{nt}	F_Y^{nt}	F_Z^{nt}
NODFOR TOTAL ROTATI LOCAL				MNTxH	MNTyH	MNTzH
				M_x^{nt}	M_y^{nt}	M_z^{nt}
				MNTxL	MNTyL	MNTzL
				M_x^{nt}	M_y^{nt}	M_z^{nt}
NODFOR TOTAL ROTATI GLOBAL				MNTXH	MNTYH	MNTZH
				M_X^{nt}	M_Y^{nt}	M_Z^{nt}
				MNTXL	MNTYL	MNTZL
				M_X^{nt}	M_Y^{nt}	M_Z^{nt}

Internal nodal element forces [§ 47.3.1]				comp ...		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	X	Y	Z
NODFOR ELEMEN TRANSL LOCAL				FNExH	FNEyH	FNEzH
				F_x^{ne}	F_y^{ne}	F_z^{ne}
				FNExL	FNEyL	FNEzL
				F_x^{ne}	F_y^{ne}	F_z^{ne}
NODFOR ELEMEN TRANSL GLOBAL				FNEXH	FNEYH	FNEZH
				F_X^{ne}	F_Y^{ne}	F_Z^{ne}
				FNEXL	FNEYL	FNEZL
				F_X^{ne}	F_Y^{ne}	F_Z^{ne}
NODFOR ELEMEN ROTATI LOCAL				MNExH	MNEyH	MNEzH
				M_x^{ne}	M_y^{ne}	M_z^{ne}
				MNExL	MNEyL	MNEzL
				M_x^{ne}	M_y^{ne}	M_z^{ne}
NODFOR ELEMEN ROTATI GLOBAL				MNEXH	MNEYH	MNEZH
				M_X^{ne}	M_Y^{ne}	M_Z^{ne}
				MNEXL	MNEYL	MNEZL
				M_X^{ne}	M_Y^{ne}	M_Z^{ne}

Internal nodal reinforcement forces [§ 47.3.1]				comp ...		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	X	Y	Z
NODFOR REINFO TRANSL LOCAL				FNRxH	FNRyH	FNRzH
				F_x^{nr}	F_y^{nr}	F_z^{nr}
				FNRxL	FNRyL	FNRzL
				F_x^{nr}	F_y^{nr}	F_z^{nr}
NODFOR REINFO TRANSL GLOBAL				FNRXH	FNRyH	FNRZH
				F_X^{nr}	F_Y^{nr}	F_Z^{nr}
				FNRXL	FNRyL	FNRZL
				F_X^{nr}	F_Y^{nr}	F_Z^{nr}
NODFOR REINFO ROTATI LOCAL				MNRxH	MNRyH	MNRzH
				M_x^{nr}	M_y^{nr}	M_z^{nr}
				MNRxL	MNRyL	MNRzL
				M_x^{nr}	M_y^{nr}	M_z^{nr}
NODFOR REINFO ROTATI GLOBAL				MNRXH	MNRyH	MNRZH
				M_X^{nr}	M_Y^{nr}	M_Z^{nr}
				MNRXL	MNRyL	MNRZL
				M_X^{nr}	M_Y^{nr}	M_Z^{nr}

9.4.8 Internal Element Forces

syntax

ELMFOR [*type_w*] [*form_w*] [*oper_w*] { *comp_w* } { *opti_w* }

ELEMEN TRANSL GLOBAL

REINFO ROTATI

TOTAL

ELMFOR specifies maximum and minimum internal element forces and moments in the nodes of an element as output item. This command gives the contributions of the element or reinforcement internal forces (or both), to a certain node of an element.

type specifies the forces type. [TOTAL]

ELEMEN for contribution of elements only.

REINFO for contribution of embedded reinforcements only.

TOTAL for contribution of both elements and embedded reinforcements.

form specifies the formulation [§ 3.6.1 p. 57]. [TRANSL]

oper specifies an operation (transformation) to be performed on the forces and moments [§ 3.6.1 p. 57]. [GLOBAL]

comp selects force or moment components for output. Default is all available components. [all]

opti are additional options [§ 3.6.1 p. 58].

Internal element forces				<i>comp</i> ...		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	X	Y	Z
ELMFOR TOTAL	TRANSL	GLOBAL		FETXH	FETYH	FETZH
				F_X^{et}	F_Y^{et}	F_Z^{et}
				FETXL	FETYL	FETZL
				F_X^{et}	F_Y^{et}	F_Z^{et}
ELMFOR TOTAL	ROTATI	GLOBAL		METXH	METYH	METZH
				M_X^{et}	M_Y^{et}	M_Z^{et}
				METXL	METYL	METZL
				M_X^{et}	M_Y^{et}	M_Z^{et}
ELMFOR ELEMEN	TRANSL	GLOBAL		FEEXH	FEEYH	FEEZH
				F_X^{ee}	F_Y^{ee}	F_Z^{ee}
				FEEXL	FEEYL	FEEZL
				F_X^{ee}	F_Y^{ee}	F_Z^{ee}
ELMFOR ELEMEN	ROTATI	GLOBAL		MEEHX	MEEYH	MEEZH
				M_X^{ee}	M_Y^{ee}	M_Z^{ee}
				MEEXL	MEEYL	MEEZL
				M_X^{ee}	M_Y^{ee}	M_Z^{ee}
ELMFOR REINFO	TRANSL	GLOBAL		FERXH	FERYH	FERZH
				F_X^{er}	F_Y^{er}	F_Z^{er}
				FERXL	FERYL	FERZL
				F_X^{er}	F_Y^{er}	F_Z^{er}
ELMFOR REINFO	ROTATI	GLOBAL		MERXH	MERYH	MERZH
				M_X^{er}	M_Y^{er}	M_Z^{er}
				MERXL	MERYL	MERZL
				M_X^{er}	M_Y^{er}	M_Z^{er}

9.4.9 Correlation Factors

When the user requests superposition according to the Complete Quadratic Combination (CQC) rule the correlation factors among the eigen modes ρ_{ij} , as defined in Eq. (48.42), are written to the standard output file *file.out*. The correlation factors ρ_{ij} vary between zero and unity.

Chapter 10

Hybrid Frequency Time Domain Analysis

This chapter describes the hybrid frequency time domain (HFTD) analysis with Module HFTD. Actually, a hybrid frequency time domain response analysis is a combination of a frequency response analysis and a transient nonlinear transient analysis. This combination makes it possible to calculate the response of systems that are frequency dependent and have nonlinear behaviour. A typical field of application is seismic analysis of dam constructions, where the reservoir is considered compressible and bottom absorption effects are taken into account, and a cracking model is being used for the dam material. In DIANA modal superposition is used to reduce the system of equations. The eigenmodes will be calculated for the structural part of the model, i.e. no added fluid effects will be incorporated in case of a fluid-structure analysis.

Note that due to the usage of modal superposition techniques, the nonlinearities considered should only have local effects, i.e. they will not result in large changes of the shapes of the eigenmodes used to calculate the time history response.

A hybrid frequency time domain analysis requires the specification of maximum three base excitation loads and maximum three time-load diagrams with equidistant time intervals, also known as accelerograms. The hybrid frequency time domain analysis as implemented in DIANA is mainly based on the procedure described by Darbre [24], and by Fenves and Chaves [30] and is described in detail in the work of Sirumbal [81]. See § 49 on page 601 for background theory.

Fluid-structure interaction analysis. In case of a model containing fluid, structural elements and fluid-structure interface elements, automatically a fluid-structure hybrid frequency time domain analysis will be performed. DIANA recognizes this element combination and will perform the required preparation to perform a fluid-structure hybrid frequency time domain analysis. In a hybrid frequency time domain analysis compression of the fluid as well as free surface waves are admitted, bottom absorption effects and a Sommerfeld radiation boundary may be incorporated. To include these effects specific material data is required, see ‘Fluid-structure interaction analysis’ in Volume *Material Library*. Results will be available for the structural part of the model. Further, dynamic fluid pressures can be output for the fluid-structure interface elements.

Lumped element matrices may not be used in a fluid-structure response analysis.

Command sequence The command sequence for Module HFTD is as follows.

syntax

*HFTD

```
[ MODEL ... ]
[ EIGEN ... ]
[ RESPON ... ]
*END
```

MODEL evaluates the finite element model [§ 10.1].

EIGEN solves the eigenvalue problem for a hybrid frequency time domain analysis [§ 10.2].

RESPON performs the actual hybrid frequency time domain analysis [§ 10.3].

10.1 Model Evaluation

The MODEL commands customize the evaluation of the finite element model prior to the actual analysis.

syntax

```
BEGIN MODEL
[ OFF ]
[ EVALUA [ OFF ] ... ]
[ ASSEMB [ OFF ] ... ]
[ MATRIX [ OFF ] ... ]
[ LOADS [ OFF ] ... ]
END MODEL
```

EVALUA checks and evaluates geometric and material properties for elements and reinforcements [§ 3.4 p. 51].

ASSEMB assembles the elements, i.e., creating appropriate system degrees of freedom [§ 3.5 p. 54].

MATRIX to setup the element matrices [§ 10.1.1].

LOADS to setup the load vectors.

10.1.1 Element Matrices

The MATRIX commands specify the type of element matrices.

syntax

```
BEGIN MATRIX
[ OFF ]
[ MASS [ OFF ] [ _____ ] [ ROTATI [ OFF ] ] ]
      CONSIS
      LUMPED
[ DAMPIN [ OFF ] [ _____ ] ]
      CONSIS
      LUMPED
[ STRESS [ OFF ] [ _____ ] ]
      INPUT { _____ }
              LOAD=losetn
              FACTOR=facr
      CALCUL { _____ }
              LOAD=losetn
              FACTOR=facr
      PHASE
END MATRIX
```

OFF suppresses the setup of element matrices.

MASS specifies the kind of mass matrices to be applied: [CONSIS]

CONSIS for consistent matrices (the default).

LUMPED for lumped matrices.

ROTATI indicates that rotational terms must be included in the mass matrices.

The OFF option suppresses the inclusion of rotational terms. If you do not specify the ROTATI option explicitly, then DIANA will include rotational terms by default.

OFF suppresses the setup of mass matrices.

DAMPIN specifies the kind of damping matrices to be applied: [CONSIS]

CONSIS for consistent matrices (the default).

LUMPED for lumped matrices.

OFF suppresses the setup of damping matrices.

Note that damping is required to meet the initial conditions. Insufficient damping may result in spurious oscillations at the beginning of the time history response.

STRESS adds the geometric stress-stiffness matrix \mathbf{K}_G to the linear elastic stiffness matrix \mathbf{K}_{L0} . [CALCUL]

INPUT indicates a geometric stress-stiffness matrix, with stresses specified via a prestress load for load set *loset*, in subtable ELEMEN and/or REINFO of table 'LOADS' [Vol. *Element Library*].

CALCUL indicates a geometric stress-stiffness matrix, with stresses which DIANA will calculate automatically in a linear static analysis for load set *loset* (the default).

PHASE indicates a geometric stress-stiffness matrix, with stresses present from a previous phase.

To setup the geometric stress-stiffness matrix \mathbf{K}_G from a stress field you may specify a load set number *loset* via the LOAD parameter. This load set number corresponds to a load set in input table 'LOADS' [§ 2.3.8 p. 45]. Default is the lowest available load set number. The optional parameter FACTOR specifies a multiplication factor *fac*. [fac=1.0]

10.2 Eigenmodes and Natural Frequencies

To solve the free vibration eigenvalue problem in a hybrid frequency time domain analysis, i.e., to determine the eigenmodes and natural frequencies, you must give commands in the EIGEN block which are analogous to those for a free vibration eigenmode analysis of Module EIGEN [§ 31.2 p. 428]. The OFF options suppress the execution of the specified task. This may be useful to save computing time if previously determined results still reside on the FILOS file.

syntax

BEGIN EIGEN

[OFF]

[EXECUT [OFF] ...]

[OUTPUT ...] ...

END EIGEN

EXECUT specifies how to solve the free vibration eigenvalue problem. In particular this involves the type of the solution procedure [§ 31.3 p. 432].

OUTPUT to specify the eigenmodes to be output [§ 31.5.10 p. 438].

Default output*file.dcf*

```
*HFTD
BEGIN EIGEN
  OUTPUT
END EIGEN
[ commands ]
*END
```

If you only give a single **OUTPUT** command, like in the above example, or if you omit the **OUTPUT** command, then DIANA gives a default output as if you had given the following commands.

file.dcf

```
*HFTD
BEGIN EIGEN
  BEGIN OUTPUT
    DISPLA TOTAL TRANSL GLOBAL
  END OUTPUT
END EIGEN
[ commands ]
*END
```

Example*file.dcf*

```
*HFTD
BEGIN EIGEN
  EXECUT NMODES=30
  OUTPUT DISPLA
END EIGEN
*END
```

This command sequence is a typical example for a preliminary eigenvalue analysis. The **EXECUT** command block calculates the natural frequencies and corresponding mode shapes needed for the hybrid frequency time domain analysis. Parameter **NMODES=30** asks for an arbitrarily chosen number of thirty frequencies. Finally, printed output of results like natural frequencies and mode shapes is obtained through the **OUTPUT** command block.

10.3 Response Analysis

To perform the actual hybrid frequency time domain analysis, i.e., to determine the response of a construction due to base excitation loading at a certain time, you must give commands in the **RESPON** block. The **OFF** options suppress the specified tasks.

syntax

```
BEGIN RESPON [ OFF ]
[ REDUCE [ OFF ] ... ]
[ EXECUT [ OFF ] ... ]
[ OUTPUT [ OFF ] ... ] ...
END RESPON
```

REDUCE reduces the system of equations [§ 10.3.1].

EXECUT executes the actual hybrid frequency time domain analysis [§ 10.3.2].

OUTPUT specifies the hybrid frequency time domain analysis results to be output [§ 10.3.4].

Default output*file.dcf*

```
*HFTD
[ commands ]
BEGIN RESPON
  OUTPUT
END RESPON
*END
```

If you only give a single **OUTPUT** command, like in the above example, or if you omit the **OUTPUT** command, then DIANA gives a default output as if you had given the following commands.

file.dcf

```
*HFTD
[ commands ]
BEGIN RESPON
  BEGIN OUTPUT
    DISPLA TOTAL TRANSL GLOBAL
    FORCE REACTI
    FORCE EXTERN
    STRESS TOTAL CAUCHY GLOBAL
    STRESS TOTAL FORCE LOCAL
    STRESS TOTAL TRACTI LOCAL
    STRESS TOTAL MOMENT LOCAL
    STRESS TOTAL DISFOR LOCAL
    STRESS TOTAL DISMOM LOCAL
    STRAIN TOTAL GREEN GLOBAL
    STRAIN TOTAL TRACTI LOCAL
  END OUTPUT
END RESPON
*END
```

10.3.1 Reduction of System of Equations

To reduce the full set of equations, you must give commands in the **REDUCE** block within the **RESPON** block.

syntax

```
BEGIN REDUCE
[ OFF ]
[ BEGIN SELECT
  MODES _____
        modes n...
        ALL
  END SELECT ]
END REDUCE
```

OFF suppresses the reduction of the system of equations. This may save computing time when the reduced system of equations is still available from a previous modal response analysis.

SELECT MODES selects eigenmodes for the hybrid frequency time domain analysis. The modes must have been determined previously [§ 10.2 p. 191]. You may explicitly specify a set of *modes*, or you may require a hybrid frequency time domain analysis for **ALL** determined modes. All modes is also the default if you do not select modes. [ALL]

10.3.2 Analysis Execution

The EXECUT command block actually performs the hybrid frequency time domain analysis by calculating the response due to the base excitation loading.

syntax

```

BEGIN EXECUT
[ OFF ]
[ TIMESE ... ]
[ ITERAT ... ]
[ MINFRE minfrer ]
[ MAXFRE maxfrer ]
[ LOGGIN ... ]
END EXECUT

```

TIMESE defines the time segments [§ 10.3.2.1].

ITERAT specifies the time segment based convergence criterion to be applied [§ 10.3.2.2].

MINFRE defines the minimum excitation frequency *minfre*. The unit is depending on the chosen time unit. MINFRE can be used as a high-pass filter.

[*minfre* = 0]

MAXFRE defines the maximum excitation frequency *maxfre*. The unit is depending on the chosen time unit. MAXFRE can be used as a low-pass filter.

[*maxfre* = 50]

LOGGIN customizes the logging information that will be output for converged steps [§ 10.3.3 p. 195].

10.3.2.1 Time Segment

The TIMESE commands specify the settings for the time segments.

syntax

```

BEGIN TIMESE
[ BEGIN STEPS
[ EXPLIC NUMBER numberi... ]
END STEPS ]
END TIMESE

```

STEPS sets the number of time steps contained by the time segments.

EXPLIC explicitly specified number of time steps contained by the time segments.

NUMBER= *number* defines the number of time steps contained by the time segments.

Smaller time segments are advised for nonlinear structural behaviour. For linear structural behaviour larger time segments can be used. In case of no convergence or divergence reducing the time segments and restarting the analysis may improve the convergence.

10.3.2.2 Equilibrium Iteration

The CONVER commands specify the convergence criterion for the equilibrium iteration process. DIANA terminates the iteration process if the specified criterion is satisfied.

syntax

```

BEGIN ITERAT
[ MAXITE= $mi_n$  ]
[ BEGIN CONVER
  FORCE [  $noconv_w$  ] { _____ }
          TERMIN      TOLCON= $tc_r$ 
          CONTIN      TOLABT= $ta_r$ 
  END CONVER ]
END ITERAT

```

MAXITE= mi is the maximum number of iterations for each time segment. [$mi = 10$]

CONVER specifies the convergence criterion for the iteration process.

FORCE specifies an convergence criterion based on the pseudo-forces.

$noconv$ indicates what to do if no convergence occurs within the maximum number of iterations. With option **TERMIN** DIANA stops the analysis run, no further steps will be executed (the default). However, further steps may be executed in subsequent runs. With option **CONTIN** the analysis run will be continued, you should check the relevance of the analysis results carefully in this case. [**TERMIN**]

Two parameters specify the tolerance values. The default values depend on the used criterion as indicated in the examples below.

TOLCON= tc is the tolerance for convergence. When the norm has become less than $tc \times$ the reference norm, DIANA assumes sufficient accuracy, stops the iteration process and continues with the next step. [$tc = 10^{-2}$]

TOLABT= ta is the tolerance for divergence. When the norm exceeds a value of $ta \times$ the reference norm, DIANA assumes divergence and aborts the nonlinear analysis. [$ta = 10^4$]

10.3.3 Logging

You may ask DIANA to produce a certain amount of log information during step execution via the LOGGIN commands.

syntax

```

BEGIN LOGGIN
[ OFF ]
[ REPORT [  $verbo_w$  ] [ STEP ] ]
      BRIEF
      FULL
[ PLASTI [ OFF ] ]
[ CRACKI [ OFF ] ]
[ REACTI [ OFF ] ]
END LOGGIN

```

OFF turns logging off, i.e., DIANA will not give any log information. When logging is on (the default), you may customize the logging information via the following commands: [ON]

REPORT indicates what to be logged and when.

$verbo$ indicates the verbosity, i.e., the amount of log information to be produced: [**BRIEF**]
BRIEF gives a brief summary, **FULL** gives full log information including a summary.

STEP indicates that log information will be produced for each converged step.

[ON] PLASTI turns the logging of the plastic points in the model on or off.

[ON] CRACKI turns the logging of the cracks in the model on or off.

[ON] REACTI turns the logging of the cumulative reaction forces and moments in the model on or off.

Default

file.dcf

```
*HFTD
[ commands ]
BEGIN RESPON
[ commands ]
BEGIN EXECUT
[ commands ]
END EXECUT
END RESPON
*END
```

If, like in the above example, you do not specify LOGGIN commands then DIANA will briefly log the occurrence of plasticity and cracking, and cumulative reaction forces and moments for each converged step. This is also achieved via the following commands:

file.dcf

```
*HFTD
[ commands ]
BEGIN RESPON
[ commands ]
BEGIN EXECUT
[ commands ]
BEGIN LOGGIN
REPORT BRIEF STEP
PLASTI
CRACKI
REACTI
END LOGGIN
END EXECUT
END RESPON
*END
```

Comprehensive

file.dcf

```
*NONLIN
[ commands ]
BEGIN EXECUT
[ commands ]
BEGIN LOGGIN
REPORT FULL STEP
PLASTI
CRACKI
REACTI
END LOGGIN
END EXECUT
*END
```

These commands ask for the most comprehensive logging: full information of plasticity and cracking, including a summary, and the cumulative reaction forces and moments for each converged step.

10.3.4 Output of Response Analysis Results

You may indicate the analysis results to be output via the `OUTPUT` command block. For the general, analysis type independent options for output of analysis results see § 3.6 on page 55.

syntax

```

BEGIN OUTPUT [devicew] [outoptw...] [params]
[ OFF ]
[ BEGIN SELECT
  [ ... ]
  [ STEPS ... ]
  END SELECT ]
[ LAYOUT ... ]
itemw ...
DISPLA
VELOCI
ACCELE
STRAIN
STRESS
STATUS
PRESSU
TEMPER
CONCEN
MATURI
FORCE
NODFOR
ELMFOR
PARAME
FRACTU
FSPRES
END OUTPUT

```

`SELECT` command block to customize the batch output.

... for model selection see § 3.6.2 on page 59, for stress- and strain transformation § 3.6.2.3 on page 60.

`STEPS` selects steps for output [§ 10.3.4.1 p. 198].

`LAYOUT` optional commands to customize the layout and style of tabular output [§ 3.6.4 p. 64].

item is the name of the analysis result to be output. See § 3.6.1 on page 56 for complete syntax of this command.

`DISPLA` for displacements, syntax and output is analogous to linear transient analysis [§ 7.4.1 p. 126].

`VELOCI` for velocities, syntax and output is analogous to linear transient analysis [§ 7.4.2 p. 128].

`ACCELE` for accelerations, syntax and output is analogous to linear transient analysis [§ 7.4.3 p. 129].

`STRAIN` for strains, syntax and output is analogous to nonlinear transient analysis [§ 13.4.2 p. 248].

`STRESS` for stresses, syntax and output is analogous to nonlinear transient analysis [§ 13.4.3 p. 253].

`STATUS` for status, syntax and output is analogous to nonlinear transient analysis [§ 13.4.4 p. 260].

PRESSU for pressure, syntax and output is analogous to nonlinear transient analysis [§ 13.4.5 p. 268].

TEMPER for temperature, syntax and output is analogous to nonlinear transient analysis [§ 13.4.6 p. 269].

CONCEN for concentration, syntax and output is analogous to nonlinear transient analysis [§ 13.4.6 p. 269].

MATURI for maturity, syntax and output is analogous to nonlinear transient analysis [§ 13.4.6 p. 269].

FORCE for nodal forces and moments, syntax and output is analogous to linear transient analysis [§ 7.4.7 p. 138].

NODFOR for element nodal forces, syntax and output is analogous to linear transient analysis [§ 7.4.8 p. 139].

ELMFOR for internal element forces, syntax and output is analogous to linear transient analysis [§ 7.4.9].

PARAME for model parameters, syntax and output is analogous to nonlinear transient analysis [§ 13.4.10 p. 274].

FRACTU for Linear Elastic Fracture Mechanics analysis parameters of crack tip elements, syntax and output is analogous to nonlinear transient analysis [§ 13.4.11].

FSPRES for dynamic pressures of fluid–structure interface elements, syntax and output is analogous to linear transient analysis [§ 7.4.10].

For frequency dependent fluid properties, the dynamic pressures can only be calculated when the complete time history response is known. Therefore, these results will be written in a separate output file of which the filename is extended with `_FSPRES`. Calculation of the dynamic pressures may take considerable extra time for frequency dependent fluid properties, because it can only be done after the calculation of the time history response. For fluid properties that are not frequency dependent, i.e. incompressible fluids without first order surface waves, bottom absorption, or Sommerfeld radiation boundaries, the dynamic pressures are calculated during the calculation of the time history response and will be stored in the same result files as other output items.

10.3.4.1 Step Selection

The STEPS command selects steps for output of response analysis results.

		<i>syntax</i>
<hr/>		
STEPS [_____] { _____ }		
<i>steps</i> _{n...} MIN		
ALL MAX		
LAST		
<hr/>		

steps are numbers of selected steps.

[ALL] ALL will produce output for all steps (the default).

LAST selects the final step.

DIANA will output the extreme values that occurred up to and including the selected steps if you specify one of the options:

MIN for the minimum values,

MAX for the maximum values.

For step-wise analyses output of extreme values cannot be used in combination with the option `LAYOUT COMBIN` [§ 3.6.4.1 p. 65] to assemble various results in one table for tabular output.

Chapter 11

Nonlinear Vibration Analysis

Under large excitations, when vibrations with large amplitudes occur, it is appropriate to include geometrically nonlinear effects in the dynamic analysis of a structure. For the natural frequencies of the structure, the nonlinearity results in a softening or in a hardening effect, i.e. a specific frequency decreases or increases with increasing amplitude, respectively. In nonlinear vibration analysis, DIANA makes use of a perturbation method (analogous to the perturbation method used in the Module EULER [Ch. 18 p. 299]) that gives an approximation for the amplitude–frequency relation of the structure. Background theory is given in § 50 on page 607.

In order to perform a nonlinear vibration analysis with DIANA you must take the following actions:

1. Invoke Module FILOS to initialize an analysis database [§ 3.2 p. 48].
2. Invoke Module INPUT to read the finite element model into the database [§ 3.3 p. 50].
3. Invoke Module VIBRAT to perform a nonlinear vibration analysis.

The primary tasks for Module VIBRAT are invoked via the following command sequence.

syntax

```
*VIBRAT
[ MODEL ... ]
[ EIGEN [ OFF ] ... ]
[ REDUCE [ OFF ] ... ]
[ CONTIN [ OFF ] ... ]
*END
```

MODEL evaluates and assembles the finite element model [§ 11.1 p. 200].

EIGEN solves the eigenvalue problem in a free vibration analysis [§ 11.2 p. 200].

REDUCE performs a perturbation analysis [§ 11.3.1 p. 201].

CONTIN performs a continuation analysis [§ 11.3.2 p. 204].

Default. If you only give the *VIBRAT command, then DIANA will perform a normal linear vibration analysis, i.e., as if you specified the following.

file.dcf

```
*VIBRAT
MODEL
EIGEN
*END
```

11.1 Model Evaluation

The `MODEL` commands customize the evaluation of the finite element model prior to the actual nonlinear vibration analysis.

syntax

```
BEGIN MODEL
[ OFF ]
[ EVALUA ... ]
[ ASSEMB ... ]
END MODEL
```

`EVALUA` checks and evaluates geometric and material properties for elements [§ 3.4 p. 51].

`ASSEMB` assembles the elements, i.e., creates appropriate system degrees of freedom [§ 3.5 p. 54].

11.2 Eigenmodes and Natural Frequencies

To do the nonlinear vibration analysis, first you need to do the linear vibration analysis i.e. you need to determine the eigenmodes and natural frequencies. To do that you must give commands in the `EIGEN` block which are analogous to those for Module `EIGEN` [§ 31.2 p. 428]. The `OFF` options suppress the execution of the specified task. This may be useful to save computing time if previously determined results still reside on the `FILOS` file.

syntax

```
BEGIN EIGEN
[ OFF ]
[ FREEVI [ OFF ] ... ]
[ EXECUT [ OFF ] ... ]
[ OUTPUT ... ] ...
END EIGEN
```

`FREEVI` sets up the free vibration eigenmode analysis [§ 31.2.1 p. 429].

`EXECUT` specifies how to solve the eigenvalue problem. In particular this involves the type of the solution procedure [§ 31.3 p. 432].

`OUTPUT` specifies the eigenmodes to be output [§ 31.5.10 p. 438].

Default output

file.dcf

```
*VIBRAT
BEGIN EIGEN
  OUTPUT
END EIGEN
[ commands ]
*END
```

If you only give a single `OUTPUT` command, like in the above example, or if you omit the `OUTPUT` command, then `DIANA` gives a default output as if you had given the following commands.

file.dcf

```

*VIBRAT
BEGIN EIGEN
  BEGIN OUTPUT
    DISPLA TOTAL TRANSL GLOBAL
  END OUTPUT
END EIGEN
[ commands ]
*END

```

Example*file.dcf*

```

*VIBRAT
BEGIN EIGEN
  FREEVI MASS
  BEGIN EXECUT
    NMODES=20
  END EXECUT
  OUTPUT DISPLA
END EIGEN
*END

```

This command sequence is a typical example for a preliminary eigenvalue analysis. In the free vibration problem, indicated by **FREEVI**, the system mass matrix has to be used which is invoked by the **MASS** command option. By default **DIANA** assembles the consistent element mass matrices. The **EXECUT** command block calculates the natural frequencies and corresponding mode shapes needed for the nonlinear vibration analysis. Parameter **NMODES=20** asks for an arbitrarily chosen number of twenty frequencies. Finally, printed output of results like natural frequencies and mode shapes is obtained through the **OUTPUT** command block.

11.3 Nonlinear Vibration Analysis

To perform a nonlinear vibration analysis with Module **VIBRAT** you must give **REDUCE** and **CONTIN** commands. Note that a nonlinear vibration analysis can only be performed if the results of a previous linear vibration analysis are available on the **FILOS** file.

syntax

```

BEGIN REDUCE
[ OFF ]
...
END REDUCE
BEGIN CONTIN
[ OFF ]
...
END CONTIN

```

REDUCE specifies how to perform the perturbation analysis [§ 11.3.1]. The **OFF** option suppresses the execution of a perturbation analysis. This may save computing time when the reduced system of equations is still available from a previous perturbation analysis.

CONTIN specifies how to perform the continuation analysis [§ 11.3.2]. The **OFF** option suppresses the execution of a continuation analysis.

11.3.1 Perturbation Analysis

To specify the perturbation analysis, i.e., the reduction of the full set of equations, you must give commands in the **REDUCE** block.

syntax

```

BEGIN REDUCE
[ BEGIN EXECUT
  [ OFF ]
  [ SELECT MODES            ]
                        modesn...
                        ALL
  [ LOADFA=lambdar ]
  [ MAXAMP=ampr ]
  END EXECUT ]
[ BEGIN OUTPUT [devicew] [outoptw...] [params]
  [ OFF ]
  [ SELECT ... ]
  [ LAYOUT ... ]
  DISPLA [typew] [formw] [operw] { compw } { optiw }
          SECMD1  TRANSL  LOCAL
          SECMD2  ROTATI  GLOBAL
  END OUTPUT ]
END REDUCE

```

EXECUT defines how to execute the perturbation analysis.

- OFF suppresses the execution of a perturbation analysis. This may save computing time when you only want to get output of a previous perturbation analysis.
- [ALL]

SELECT MODES selects vibration modes for the perturbation analysis. The modes must have been determined previously [§ 11.2 p. 200]. You may explicitly specify is a set of *modes*, or you may require a perturbation analysis for ALL determined modes. All modes is also the default if you do not select modes.
- [$\lambda_p = 0.95$]

LOADFA=*lambda* specifies the relative load factor λ_p of the perturbation point.

MAXAMP=*amp* specifies that the selected vibration modes will be scaled in the perturbation analysis such that the maximum displacement amplitude of each selected vibration model will be equal to *amp*.
- OUTPUT specifies the desired output of the perturbation analysis. For the general, analysis type independent options for output of analysis results see § 3.6 on page 55.
- SELECT customizes the batch output selection. Appropriate for perturbation analysis is node selection [§ 3.6.2 p. 59].
- LAYOUT specifies optional commands to customize the layout and style of tabular output [§ 3.6.4.1 p. 65].
- [SECMD1]

DISPLA will output the second order displacements [§ 11.3.1.1].

Default

file.dcf

```

*VIBRAT
[ commands ]
BEGIN REDUCE
[ commands ]
  OUTPUT
END REDUCE
*END

```

If you only give a single OUTPUT command, like in the above example, or if you do not give the OUTPUT command at all, then DIANA gives output of the translational displacements in global *XYZ* orientation, as if you had given the following commands.

file.dcf

```

*VIBRAT
[ commands ]
BEGIN REDUCE
[ commands ]
BEGIN OUTPUT
  DISPLA SECMD1 TRANSL GLOBAL
END OUTPUT
END REDUCE
*END

```

Example.*file.dcf*

```

BEGIN REDUCE
  BEGIN EXECUT
    SELECT MODES 1 2
    LOADFAC=0.99
  END EXECUT
  BEGIN OUTPUT
    DISPLA SECMD1
    DISPLA SECMD2
  END OUTPUT
END REDUCE

```

file.dcf

```

BEGIN REDUCE
  EXECUT SELECT MODES 1 2 LOADFAC=0.99
  OUTPUT DISPLA
END REDUCE

```

11.3.1.1 Output of Second Order Displacements

The SECMD1 and SECMD2 type specifiers give output of displacements of a perturbation analysis.

<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	<i>comp ...</i>		
				X	Y	Z
DISPLA	SECMD1	TRANSL	LOCAL	D2tx	D2ty	D2tz
				u_x	u_y	u_z
DISPLA	SECMD1	TRANSL	GLOBAL	D2tX	D2tY	D2tZ
				u_X	u_Y	u_Z
DISPLA	SECMD1	ROTATI	LOCAL	D2rx	D2ry	D2rz
				ϕ_x	ϕ_y	ϕ_z
DISPLA	SECMD1	ROTATI	GLOBAL	D2rX	D2rY	D2rZ
				ϕ_X	ϕ_Y	ϕ_Z
DISPLA	SECMD2	TRANSL	LOCAL	D2tx	D2ty	D2tz
				u_x	u_y	u_z
DISPLA	SECMD2	TRANSL	GLOBAL	D2tX	D2tY	D2tZ
				u_X	u_Y	u_Z
DISPLA	SECMD2	ROTATI	LOCAL	D2rx	D2ry	D2rz
				ϕ_x	ϕ_y	ϕ_z
DISPLA	SECMD2	ROTATI	GLOBAL	D2rX	D2rY	D2rZ
				ϕ_X	ϕ_Y	ϕ_Z

11.3.2 Continuation Analysis

To specify the continuation analysis you must give commands within the CONTIN block.

syntax

```

BEGIN CONTIN
  BEGIN EXECUT
    [ OFF ]
    [ BEGIN START
      [ FREQUE=frequer ]
      [ AMPLIT amplr... ]
      END START ]
    [ STEPS {                      } ]
      NSTEPS=nstepsn
      SIZE=sizer
    [ NORM [ DISPLA ] CONVER=epsr ]
  END EXECUT
  [ BEGIN OUTPUT [devicew] [outoptw...] [params]
    [ OFF ]
    [ BEGIN SELECT
      [ ... ]
      [ STEPS                      ]
        stepsn...
        ALL
        LAST
      END SELECT ]
    [ LAYOUT ... ]
    DISPLA [ typew ] [ formw ] [ operw ] { compw } { optiw }
      NLVIBR  TRANSL  LOCAL
      ROTATI  GLOBAL

  END OUTPUT ]
END CONTIN

```

EXECUT specifies how to perform the continuation analysis.

OFF suppresses the execution of the continuation analysis.

START specifies the starting point in the load factor–mode amplitude space. Parameter FREQUE=*freque* is an estimation for the normalized frequency ω/ω_0 . AMPLIT *ampl* are estimations for the respective mode amplitudes, one value for each selected mode. AMPIMP *ampi* are the respective imperfection amplitudes for each selected mode.

STEPS specifies the execution of steps in the continuation procedure. Parameter NSTEPS=*nsteps* indicates the number of steps, SIZE=*size* is the step size.

NORM specifies the convergence criterion for the iteration process. The tolerance ε is indicated by parameter CONVER=*eps*. In nonlinear vibration analysis the only available criterion is on the displacement norm, therefore you may omit the DISPLA option.

OUTPUT specifies the desired output of the perturbation analysis. For the general, analysis type independent options for output of analysis results see § 3.6 on page 55.

SELECT customizes the batch output selection.

... is a model selection [§ 3.6.2 p. 59]. Appropriate for continuation analysis is node selection

STEPS selects steps for output of analysis results. Where *steps* are numbers of selected steps, ALL will produce output for all steps that are executed (the default), LAST selects the final step. [ALL]

LAYOUT specifies optional commands to customize the layout and style of tabular output [§ 3.6.4.1 p. 65].

DISPLA NLVIBR will output the nonlinear vibration displacements [§ 11.3.2.1]. [NLVIBR]

Default

file.dcf

```
*VIBRAT
[ commands ]
BEGIN CONTIN
[ commands ]
OUTPUT
END CONTIN
*END
```

If you only give a single OUTPUT command, like in the above example, or if you do not give the OUTPUT command at all, then DIANA gives output of the translational displacements in global XYZ orientation, as if you had given the following commands.

file.dcf

```
*VIBRAT
[ commands ]
BEGIN CONTIN
[ commands ]
BEGIN OUTPUT
DISPLA NLVIBR TRANSL GLOBAL
END OUTPUT
END CONTIN
*END
```

Example.

file.dcf

```
BEGIN CONTIN
BEGIN EXECUT
BEGIN START
FREQUE=1.0
AMPLIT 0.0 0.0
END START
STEPS NSTEPS=10 SIZE=1.0
NORM DISPLA CONVER=0.001
END EXECUT
BEGIN OUTPUT
DISPLA NLVIBR
END OUTPUT
END CONTIN
```

In the above example a nonlinear vibration continuation analysis of ten steps is executed, starting in the initial buckling point. A step size of 1.0 is applied while the convergence criterion is 0.001.

11.3.2.1 Output of Nonlinear Vibration Displacements

The NLVIBR type specifier gives output of displacements of a nonlinear vibration continuation analysis. As outlined in the background theory for nonlinear vibration analysis [§ 50 p. 607], DIANA calculates the total displacements field \mathbf{u}_{nv} for the actual nonlinear vibration configuration according to

$$\mathbf{u}_{nv} = a_i \phi_i + a_i a_j \mathbf{u}_{ij} \quad (11.1)$$

where \mathbf{u}_{ij} is called the second order displacement vector and a_i should be interpreted as amplitude of the respective mode.

<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	<i>comp ...</i>		
				X	Y	Z
DISPLA	NLVIBR	TRANSL	LOCAL	DPtx	DPty	DPtz
				u_x	u_y	u_z
DISPLA	NLVIBR	TRANSL	GLOBAL	DPtX	DPtY	DPtZ
				u_X	u_Y	u_Z
DISPLA	NLVIBR	ROTATI	LOCAL	DPrx	DPrY	DPrz
				ϕ_x	ϕ_y	ϕ_z
DISPLA	NLVIBR	ROTATI	GLOBAL	DPrX	DPrY	DPrZ
				ϕ_X	ϕ_Y	ϕ_Z

Part IV

Nonlinear Static and Transient Analysis

Chapter 12

Introduction to Nonlinear Structural Analysis

This part is a user's reference guide for Module NONLIN, which performs nonlinear static and transient *structural analysis*. See Chapter 46 for background theory on solution procedures for nonlinear systems. NAFEMS has published a very useful general introduction to nonlinear Finite Element Analysis, see Becker [6].

12.1 Input for Nonlinear Analysis

In nonlinear structural analysis you may specify data additional to the input data for linear static analysis as described in Chapter 1 and Chapter 2.

Material properties. The additional data mainly comprises the nonlinear phenomena like material behaviour [Vol. *Material Library*].

Contact analysis. For contact analysis you must specify contact zones with contact elements [Vol. *Element Library*].

Transient analysis. For nonlinear transient (dynamic) analysis you may specify additional data analogous to the data for linear transient analysis as described in Chapter 6 and §6.5. This data comprises the specification of mass and damping, the initial displacements or velocities, the time-load diagram, and the base excitation. Moreover the time and/or ambient dependency of some material properties may be specified [Vol. *Material Library*].

12.2 Element Choice for Nonlinear Analysis

In models for nonlinear analysis you are advised not to apply linearly interpolated isoparametric elements, like for instance Q8MEM, Q8OME, Q8EPS, Q8AXI, or HX24L. These elements have intrinsic shortcomings, like parasitic shear and volumetric locking, which cannot be easily dealt with in nonlinear analysis. In linear elastic analysis so-called *assumed strain* concepts can be applied to improve the element behaviour [Vol. *Element Library*].

Unfortunately, in nonlinear analysis the assumed strain concepts harm the stability of the solution procedure and may cause spurious kinematic modes. Furthermore, the assumed strain concepts are not compatible with elements for mixture analysis and for Total and Updated Lagrange geometric nonlinearity. It is therefore recommended to apply higher order elements, for instance the quadratically interpolated CQ16M, CQ16O, CQ16E, CQ16A, or CHX60.

Chapter 13

Nonlinear Structural Analysis

In order to perform a nonlinear structural analysis with DIANA you must take the following actions:

1. Invoke Module FILOS to initialize an analysis database [§ 3.2 p. 48].
2. Invoke Module INPUT to read the finite element model into the database [§ 3.3 p. 50].
3. Invoke Module NONLIN to perform a nonlinear analysis.
4. You may perform interactive postprocessing of results with DianaIE [§ 3.6.3 p. 63], which is the default output device when the analysis is neither started from iDIANA nor FX^+ , or with iDIANA [§ 3.6.5 p. 66], which is the default output device when the analysis is started from iDIANA, or with FX^+ [§ 3.6.6 p. 67], which is the default output device when the analysis is started from FX^+ . Alternatively, tabulated output is available [§ 3.6.4 p. 64].

The command sequence for Module NONLIN is as follows.

syntax

```
*NONLIN
[ MODEL ... ]
[ TYPE ... ]
[ SOLVE ... ]
[ EXECUT ... ] ...
[ OUTPUT ... ] ...
*END
```

MODEL evaluates the finite element model [§ 13.1].

TYPE specifies the type of the nonlinear analysis [§ 13.2].

SOLVE customizes the settings for the solution method [Ch. 30 p. 421].

EXECUT executes steps [§ 13.3 p. 219].

OUTPUT selects analysis results for output [§ 13.4 p. 246]. This output selection, at the *NONLIN command level, is valid for all EXECUT blocks of the current analysis. However, an output selection within any EXECUTE block overrules the output selection at the *NONLIN level [§ 13.3 p. 219].

13.1 Model Evaluation

The MODEL commands customize the evaluation of the finite element model prior to the actual analysis.

syntax

```

BEGIN MODEL
[ OFF ]
[ EVALUA [ OFF ] ... ]
[ ASSEMB [ OFF ] ... ]
[ MATRIX [ OFF ] ]
[ LOADS [ OFF ] ]
END MODEL

```

EVALUA to check and evaluate geometric and material properties for elements and reinforcements [§ 3.4 p. 51].

ASSEMB to assemble the elements, i.e., creating appropriate system degrees of freedom [§ 3.5 p. 54].

MATRIX to setup the element stiffness matrices.

LOADS to setup the load vectors.

13.2 Type of Nonlinear Analysis

The TYPE commands indicate the type of the nonlinear analysis.

It is recommended to keep the settings specified by the TYPE commands unchanged during phased analysis. It is, however, possible to switch from nonlinear to linear behaviour, but switching from linear to nonlinear behaviour may lead to unpredictable consequences. A limited number of physical nonlinear settings specified by the TYPE commands, can be modified in the physical nonlinear analysis options of the execute block [§ 13.3.4 p. 234].

syntax

```

BEGIN TYPE
[ OFF ]
[ PHYSIC ... ]
[ GEOMET ... ]
[ TRANSI ... ]
[ LINSTR [ OFF ] ]
[ RETSTR [ OFF ] ]
END TYPE

```

PHYSIC sets options and parameters for physical nonlinear analysis [§ 13.2.1].

GEOMET sets options and parameters for geometrical nonlinear analysis [§ 13.2.2].

TRANSI sets options and parameters for transient nonlinear analysis [§ 13.2.3].

[OFF] LINSTR activates linear stress/strain determination for linear elements. By default this option is not active. If you activate it then DIANA will handle *any element without physical and geometrical nonlinearities* in a special way:

- Strains and stresses will be determined only if you request these for output.
- Requested strains and stress will be determined directly from the total displacement field, like for linear static analysis. No incremental update will be performed.
- Recalculation of the tangential matrix will be skipped, because this matrix remains constant.

For models with relatively large areas of linearly behaving elements this option may yield a considerable saving of computing time and data storage, but please note the following:

*The **LINSTR** option yields incorrect results for modified elasticity [Vol. Material Library], or if you change the reinforcement in a phased nonlinear analysis [Ch. 28 p. 401], or if you suppress the superposition of displacements and strains via the **SUPPRE STRAIN** command [§ 13.3.4 p. 235], or if you suppress the superposition of stresses via the **SUPPRE STRESS** command [§ 13.3.4 p. 235].*

RETSTR triggers the recomputation of total strains and stresses based on modified elastic parameters in case of modified elasticity [Vol. Material Library]. In this case, the following stress update is done:

$$\sigma_i = E_i \cdot \epsilon_i \quad (13.1)$$

This implies that the stress state history will be lost, and may result in additional displacements even when the loading is unchanged. By default, this option is not active, and DIANA adds the incremental strains and stresses based on the modified elastic parameters to the previous stress state, i.e. the stress state history remains and no additional displacements will occur when the loading is unchanged:

$$\sigma_i = \sigma_{i-1} + E_i \Delta \epsilon_i = E_{i-1} \cdot \epsilon_{i-1} + E_i \Delta \epsilon_i \quad (13.2)$$

Default

file.dcf

```
*NONLIN
TYPE
[ commands ]
*END
```

These commands initiate a physical nonlinear analysis with application of all nonlinear phenomena as specified in the input data file. The same would occur if you had given the following commands:

file.dcf

```
*NONLIN
BEGIN TYPE
  PHYSIC
END TYPE
[ commands ]
*END
```

13.2.1 Physical Nonlinear Analysis

The use of the phenomena specified in the input data can be overruled for a limited number of options via the **PHYSIC** commands in the **TYPE** block. Phenomena and parameters are overruled for all the subsequent **EXECUT** blocks.

syntax

```
BEGIN PHYSIC
[ OFF ]
[ PLASTI [ OFF ] [ _____ ] { _____ } ]
      FIRST  MITERA=nin
      SECOND TOLERA=tyr
      SUBSTP=alpminr
```

```

[ CREEP [ OFF ] [ _____ ] { _____ } ]
      ZERO      MITERA=nin
      FIRST     TOLERA=tcr

[ CORROS [ OFF ] ]
[ TEMPER [ OFF ] ]
[ CONCEN [ OFF ] ]
[ CRACKI [ OFF ] [ _____ ] { _____ } ]
      SECANT     TOLDIR=atdr
      CONSIS     TOLTCO=tcor

[ TOTCRK [ OFF ] [ _____ ] ]
      SECANT
      CONSIS

[ ELASTI [ OFF ] ]
[ VISCOE [ OFF ] ]
[ VISCOP [ OFF ] ]
[ MATURI [ OFF ] ]
[ PRESSU [ OFF ] ]
[ HYPERE [ OFF ] ]
[ INTERF [ OFF ] ]
[ CONTAC [ OFF ] ]
[ SHRINK [ OFF ] ]
[ SOIL [ OFF ] ]
[ PML [ OFF ] ]
[ STRESS [ OFF ] ]
END PHYSIC

```

PLASTI for use of plasticity. An option indicates the tangent stiffness to be used.

[FIRST] FIRST for plasticity with classical (first order) tangent stiffness.

SECOND for plasticity with consistent (second order) tangent stiffness. Reduces the number of necessary iterations when used in combination with Regular Newton–Raphson iteration.

Via three parameters you may customize the internal plasticity iteration process: MITERA=*ni* is the maximum number of iterations and TOLERA=*ty* the tolerance on satisfaction of the yield function. Additionally, via the SUBSTP command the sub-stepping for Modified Mohr-Coulomb and anisotropic Rankine-Hill plasticity models can be controlled [Vol. *Material Library*]: *alpmi*n is the minimum relative sub-step size $\alpha_{k,\min}$ that can be set as an escape route in case the sub-steps do not converge and the plasticity algorithm keeps decreasing the sub-step size. When $\alpha_{k,\min}$ is set to 1, no sub-stepping will be applied.

CREEP for use of metal or transient creep. Note that for creep of concrete under long term load, viscoelasticity is used. An option indicates the creep order.

[ZERO] ZERO for creep with zero order approximation.

FIRST for creep with properly linearized tangent stiffness (first order creep). Requires less computing time if used in combination with Regular Newton–Raphson iteration (only for transient creep).

Via two parameters you may customize the internal creep iteration process: MITERA=*ni* is the maximum number of iterations and TOLERA=*tc* is the tolerance on stress accuracy.

[*ni*=1]
[*tc*=0.0001]

CORROS for influence of corrosion.

TEMPER takes the influence of temperature on the strains into account (thermal strains).

CONCEN takes the influence of concentration on the strains into account ('concentration' strains).

CRACKI for use of cracking. An option indicates the crack normal stiffness to be used. See also the background theory on the multi-directional fixed crack model in Volume *Material Library*.

SECANT for cracking with secant crack normal stiffness, i.e., for all cracks (either on softening branch or not) the positive secant unloading modulus is used as 'tangent' stiffness. This is the default cracking model. [SECANT]

CONSIS for cracking with consistent crack normal stiffness, i.e., for cracks on softening branch the true negative softening modulus is used as tangent stiffness. This option may require less computing time, especially when used in combination with Regular Newton–Raphson iteration. It opens up the possibility to analyse snap-back behaviour in combination with Arc-length control. Be careful when the crack pattern is diffuse (for instance with reinforced concrete), then the method easily fails because of bifurcations. In such cases, continue with the secant crack normal stiffness (**SECANT**).

Two parameters may be used to affect the internal cracking iteration process: **TOLDIR=atd** the threshold angle α_{TD} between multi-directional cracks (in degrees) and **TOLTCO=tco** the tolerance on satisfaction of the tension-cut-off criterion. [$\alpha_{TD} = 60^\circ$] [$tco = 0.001$]

TOTCRK for use of crack models based on total strain. An option indicates the crack normal stiffness to be used. See also the background theory on total strain crack models in Volume *Material Library*.

SECANT for cracking with secant crack normal stiffness, i.e., for all cracks (either on softening branch or not) the positive secant unloading modulus is used as 'tangent' stiffness. This is the default cracking model. [SECANT]

CONSIS for cracking with consistent crack normal stiffness, i.e., for cracks on the softening branch the true negative softening modulus is used as tangent stiffness.

ELASTI for use of elastic material behaviour.

VISCOE for use of viscoelastic material behaviour.

VISCOP for use of viscoplastic material behaviour.

MATURI for use of maturity dependent material properties in stress analysis of young, hardening concrete in combination with Power Law viscoelasticity.

PRESSU takes the influence of pressure on the strains into account.

HYPERE for use of hyperelastic (rubbery) material properties.

INTERF for use of interface nonlinearity. The type of nonlinearity depends on input, for instance discrete cracking, crack dilatancy, bond-slip or friction.

CONTAC for use of contact nonlinearity.

SHRINK for use of material shrinkage.

SOIL for use of simple soil material behaviour.

PML for use of perfectly matched layers (PML) [Vol. *Material Library*].

STRESS for use of simple stress dependent material behaviour.

13.2.2 Geometrical Nonlinear Analysis

The **GEOMET** commands initiate a geometrical nonlinear analysis. See § 46.2 on page 566 for background theory.

syntax

```

BEGIN GEOMET
[ OFF ]
[ formulw ]
TOTAL
UPDATE
[ BEGIN NCLOAD
[ OFF ]
LOADS losetsn...
END NCLOAD ]
[ NLPREB ]
END GEOMET

```

formul specifies the formulation to be applied.

TOTAL applies the Total Lagrange formulation [§ 46.2.2 p. 568] (the default).

UPDATE applies the Updated Lagrange formulation [§ 46.2.3 p. 568]. Updated Lagrange analysis is available for solid elements, curved shell elements, plane strain elements (except infinite shells), regular and three-dimensional plane stress elements, axisymmetric elements (except shells of revolution), three-dimensional Class-III beam elements, and three-dimensional cable elements. For other elements DIANA automatically applies Total Lagrange. Plasticity, creep, and viscoelasticity are the only physical nonlinearities that can be applied in combination with Updated Lagrange analysis. Anisotropic effects of cracking, material anisotropy and embedded reinforcements are not handled correctly in Updated Lagrange analysis, because the rotations are not accounted for in the current implementation. Therefore, cracking, material anisotropy and embedded reinforcements can not be used in combination with Updated Lagrange analysis.

[OFF] NCLOAD specifies whether or not nonconservative pressure load must be applied.

OFF switches off the application of nonconservative pressure load.

LOADS *losets* are load set numbers referring to table 'LOADS' [§ 2.3.8 p. 45]. DIANA will update the forces according to the load definition in the input file: if the direction was defined in a *global direction* then only the magnitude will be updated, if the direction was defined in an *element direction* then both the magnitude and the direction will be updated.

Only element loads can be nonconservative, other loads may not be part of the chosen load set number.

NLPREB prepares the deformed configuration for a subsequent perturbation analysis with *EULER [§ 18 p. 299].

Default

file.dcf

```

*NONLIN
TYPE GEOMET
[ commands ]
*END

```

These commands initiate a geometrical nonlinear analysis according to the Total Lagrange formulation. The same would occur if you had given the following commands:

file.dcf

```

*NONLIN
BEGIN TYPE
  GEOMET TOTAL
END TYPE
[ commands ]
*END

```

13.2.3 Transient Nonlinear Analysis

The TRANSI commands initiate a transient nonlinear analysis, i.e., applying time integration. As an option, you may specify the integration method to be used and the type of matrices to be applied in dynamic transient analysis.

syntax

```

BEGIN TRANSI
[ OFF ]
[ BEGIN METHOD
  [ methodw ]
  BACKWA
  NEWMAR { _____ }
           GAMMA=gammar
           BETA=betar
  HHT    [ ALPHA=alphar ]
  WILSON [ THETA=thetar ]
  SDIRK2
END METHOD ]
[ BEGIN DYNAMI
  [ OFF ]
  [ MASS [ OFF ] [ _____ ] [ ROTATI [ OFF ] ] ]
    CONSIS
    LUMPED
  [ DAMPIN [ OFF ] [ _____ ] [ RAYLEI [ OFF ] ] ]
    CONSIS
    LUMPED
  [ RELBAC [ OFF ] ]
END DYNAMI ]
[ TIMEDE [ OFF ] ]
END TRANSI

```

METHOD *method* specifies the time integration method.

BACKWA for Euler Backward time integration [§ 48.4.2 p. 593].

NEWMAR for time integration according to Newmark's method [§ 48.4.1 p. 593]. [NEWMAR]
 You may customize the integration algorithm with two optional parameters: [$\gamma = \frac{1}{2}$, $\beta = \frac{1}{4}$]
 GAMMA=*gamma* specifies factor γ , and BETA=*beta* specifies factor β . ($\gamma \geq \frac{1}{2}$)
 The conditions for γ and β are to guarantee the stability of the algorithm. ($\beta \geq \frac{1}{4}(\frac{1}{2} + \gamma)^2$)

HHT for the Hilber–Hughes–Taylor time integration method [§ 48.4.3 p. 593]. The
 optional parameter ALPHA=*alpha* specifies the factor α . [$\alpha = -0.1$]

WILSON for the Wilson- θ time integration method [§ 48.4.4 p. 593]. The optional
 parameter THETA=*theta* specifies the factor θ . [$\theta = 1.40$]
 ($\theta > 1.37$)

SDIRK2 for two stage Runge–Kutta time integration [§ 48.4.5 p. 594].

[OFF] DYNAMI will apply a transient dynamic analysis.¹ Please note that for transient dynamic analysis the creation of mass and damping matrices may require additional input data [Ch. 6 p. 103].

[MASS] MASS specifies the kind of mass matrices to be applied for inclusion of inertia forces.

[CON SIS] CON SIS for consistent mass matrices (the default).

LUMPED for lumped mass matrices.

[ON] ROTATI to include rotational terms in the mass matrices. The OFF options suppresses the inclusion of rotational terms. If you do not specify the ROTATI option explicitly, then DIANA will include rotational terms by default.

[CON SIS] DAMPIN specifies the kind of damping matrices to be applied for inclusion of damping forces.

CON SIS uses consistent damping matrices.

LUMPED uses lumped damping matrices.

RAYLEI uses Rayleigh damping based on the current stiffness matrices. With this option DIANA will update the damping matrix within each time increment. The OFF options suppresses the updating of the damping matrices within each time increment. If you do not specify the RAYLEI option explicitly, then DIANA will not update the Rayleigh damping matrices, which implies that the damping matrices will be based on the initial stiffness matrices in case of Rayleigh damping. This option requires the specification of the Rayleigh damping parameters as a material property, see ‘Viscous Damping’ in Volume *Material Library*, as model information [§ 1.2 p. 6], or calculated by a preceeding eigenvalue analysis [§ 31.4 p. 435].

[OFF] RELBAC specifies that the dynamic response due to an applied base acceleration is defined in a relative coordinate system with respect to the base [§ 48.1.4 p. 586]. This implies that in the equivalent force vector $\hat{\mathbf{f}}_i$ [Eq. (48.14)] no damping due to the base movement is added in case of Rayleigh damping. See also the *Effect of Rayleigh Damping in Case of Applied Base Acceleration* example in Volume *Analysis Examples* to illustrate the effect of this option in a transient dynamic analysis.

[TIMEDE] TIMEDE will take the time derivative terms $\dot{\mathbf{u}}$ and $\dot{\phi}$ in to account in soil–pore fluid (mixture) analysis [Eq. (56.28) p. 635].

Default

file.dcf

```
*NONLIN
TYPE TRANSI DYNAMI
[ commands ]
*END
```

These commands initiate a transient dynamic nonlinear analysis with Newmark time integration. Consistent mass matrices will be applied, including rotational terms. The same would occur if you had given the following commands:

file.dcf

```
*NONLIN
BEGIN TYPE
  BEGIN TRANSI
    BEGIN METHOD
      BEGIN NEWMAR
        GAMMA=0.5
```

¹Dynamic frequency response analysis must be performed with Module MODAL or FREQUE [Ch. 8].

```

      BETA=0.25
    END NEWMAR
  END METHOD
  DYNAMI MASS CONSIG ROTATI ON
END TRANSI
END TYPE
[ commands ]
*END

```

13.3 Step Execution

With the EXECUT commands you ask DIANA to execute load or time steps. A command file may contain one or more EXECUT blocks. Details of various commands are given in the referred sections.

syntax

```

BEGIN EXECUT
[ OFF ]
[ typew ... ]
START
LOAD
TIME
[ PHYSIC ... ]
[ ITERAT ... ]
[ STOP ... ]
[ LOGGIN ... ]
[ OUTPUT ... ] ...
[ REFERE ... ]
END EXECUT

```

type indicates the type of step(s) to be executed.

[LOAD]

START to evaluate the initial state [§ 13.3.1].

LOAD to execute load steps [§ 13.3.2 p. 224].

TIME to execute time steps [§ 13.3.3 p. 231].

PHYSIC sets options for physical nonlinear analysis [§ 13.3.4 p. 234].

ITERAT specifies the process of equilibrium iteration to be applied [§ 13.3.5 p. 235].

STOP specifies a stop criterion for step execution [§ 13.3.6 p. 240].

LOGGIN customizes the logging information that will be output during execution of the steps [§ 13.3.7 p. 243].

OUTPUT selects output for steps [§ 13.4 p. 246]. This selection overrules any output selection specified at the *NONLIN command level [Ch. 13 p. 211].

REFERE sets initial state references for geotechnical analysis [§ 13.3.8 p. 244].

Default

file.dcf

```

*NONLIN
[ commands ]
EXECUT
*END

```

Due to these commands DIANA will execute a load step with a default factor, applying the default integration scheme and solution procedure, but no stop criteria. DIANA will give the default logging information and output of analysis results. The same would occur if you had given the following commands:

file.dcf

```
*NONLIN
[ commands ]
BEGIN EXECUT
  LOAD
  PHYSIC OFF
  ITERAT
  LOGGIN
  OUTPUT
END EXECUT
*END
```

Note that the **PHYSIC OFF** command only switches off the setting of special physical options for step execution [§ 13.3.4 p. 234]. The physical phenomena as specified on the input file and/or overruled via the **TYPE PHYSIC** command will be applied anyhow [§ 13.2.1 p. 213].

13.3.1 Initial State Evaluation

The **START** commands evaluate the initial state of the model, i.e, prior to the execution of the first load or time step. The **START** commands balance the external forces, which are defined by the **LOAD** command, and the internal forces, which are defined by the stresses in the elements. The stresses in the elements can come from either a previous construction stage or from user input (**INITIA STRESS INPUT**) for new elements, or from a preliminary linear analysis (**INITIA STRESS CALCUL**) for new elements. Consequently, the **START** commands may only appear in the first **EXECUT** block of a nonlinear analysis.

syntax

```
BEGIN START
[ OFF ]
[ TIME=stimer ]
[ INITIA ... ]
[ LOAD[OFF]{ _____ } ]
    PREVIO[OFF]
    REAFOR[OFF]
    ADD{ _____ } ]
        LOADNR=losetn
        FACTOR=facr
[ STEPS ... ]
END START
```

[*stime*=0] **TIME** *stime* is a user specified starting time.

INITIA specifies initial conditions [§ 13.3.1.1].

[LOAD PREVIO] **LOAD** applies an initial external load $\mathbf{f}_{\text{ext}}^0$ [Eq. (13.3) p. 223]:

OFF indicates that no initial external load will be applied.

PREVIO indicates that the external loads from the end of the previous phase is retained in the current phase. The **OFF** option suppresses retaining the external loads from previous phases.

Note that any prescribed displacements included in the load of the previous phase are omitted here. Inclusion of the prescribed displacements would lead to unwanted doubling of the displacement field.

REAFFOR indicates that reaction forces at the deactivated supported nodes in the current phase will be applied. This feature is particularly important when perfectly matched layers (PML) [Vol. *Material Library*] are activated, because the applied reaction forces ensure that the PML layers are not subjected to any initial stresses or deformations. The OFF option suppresses the application of reaction forces at the deactivated nodes in the current phase.

ADD indicates that a user defined loadset is added to the initial external load $\mathbf{f}_{\text{ext}}^0$. Prescribed displacement loads specified here will also be applied. The parameter LOADNR=*loadset* indicates a load set number. Default is the lowest available load set number. The optional parameter FACTOR specifies a multiplication factor *fac*. [fac=1.0]

STEPS executes steps for incremental adaptation to the initial conditions [§ 13.3.1.2 p. 223] [STEPS]

Default

file.dcf

```
*NONLIN
[ commands ]
EXECUT START
*END
```

These commands will set the initial state with start time $t = 0$, with zero stresses in new elements, retain the state of existing elements, and retain the external load from the end of the previous phase. Incremental adaptation of initial conditions is done by executing an explicit step with size 1.0. The same would occur if you had given the following commands:

file.dcf

```
*NONLIN
[ commands ]
BEGIN EXECUT
  BEGIN START
    TIME=0.
    LOAD PREVIO
    STEPS EXPLIC SIZE 1.0
  END START
END EXECUT
*END
```

13.3.1.1 Initial Conditions

To start with static equilibrium, you must let DIANA first perform one or more steps [§ 13.3.1.2 p. 223].

Generally speaking there are three kinds of initial conditions: displacements, velocities, and stresses which may also be applied simultaneously (see also § 6.5.1 on page 106). You may either specify initial displacements, velocities and stresses explicitly, or apply displacements and stresses calculated in a preliminary analysis as an initial condition for the nonlinear analysis.

syntax

```

BEGIN INITIA
[ OFF ]
[ DISPLA [OFF] [ _____ ] ]
                        INPUT { _____ }
                              FIELD=fieldn
                              FACTOR=facr
                        CALCUL { _____ }
                              LOAD=losetn
                              FACTOR=facr
                        POSTBU
[ VELOCI [OFF] [ INPUT ] { _____ } ]
                              FIELD=fieldn
                              FACTOR=facr
[ STRESS [OFF] [ _____ ] ]
                        INPUT { _____ }
                              LOAD=losetn
                              FACTOR=facr
                        CALCUL { _____ }
                              LOAD=losetn
                              FACTOR=facr

END INITIA

```

[*fac*=1.0] For initial fields the optional parameter **FACTOR** specifies a multiplication factor *fac*. For most initial fields you may either specify a *field* number via the **FIELD** parameter, or a load set number *loset* via the **LOAD** parameter. Default is the lowest available field or load set number.

[INPUT] **DISPLA** specifies a start with initial displacements.

INPUT indicates a start with specified initial displacements. The *field* number refers to a field in subtable **DISPLA** of table 'INIVAR' [§ 6.5.1.1 p. 106].

CALCUL indicates a start with initial displacements which DIANA will calculate automatically in a linear static analysis for load set *loset*.

POSTBU indicates a start with a postbuckling displacement field from the most recent continuation stability analysis with Module **EULER** [§ 18.3.2 p. 308]. In combination with a Continuation iteration scheme for the first step [§ 13.3.2.2 p. 226] [§ 13.3.2.3 p. 227], DIANA will apply the postbuckling displacement field as a first predictor of the nonlinear analysis. The load value calculated by Module **EULER** overrules the load step size that you specify for the first step. This option is only useful in geometrically nonlinear analysis.

VELOCI specifies a start with initial velocities. This option only applies for transient dynamic analysis.

INPUT indicates a start with specified velocities, which is the only and default option. The *field* number refers to a field in subtable **VELOCI** of table 'INIVAR' [§ 6.5.1.1 p. 106],

[INPUT] **STRESS** specifies a start with initial stresses. This option also sets the initial state references. You may reset these references during the analysis by means of the **REFERE** command [§ 13.3.8 p. 244]. In a nonlinear phased analysis the **STRESS INPUT** and **STRESS CALCUL** commands initialize the stresses in elements that become active for the first time in the current phase. Stresses in all other active elements originate from a previous phase.

INPUT indicates a start with stresses, specified via a prestress load for load set *loset*, in subtable **ELEMEN** and/or **REINFO** of table 'LOADS' [Vol. *Element Library*].

CALCUL indicates a start with the stresses which DIANA will calculate automatically in a linear static analysis for load set *loset*.

DIANA does not accept the combination of explicitly specified and calculated fields of the same type. Table 13.1 indicates the possibility of various combinations of options which you may apply via multiple START commands.

Table 13.1: COMBINATIONS OF INITIAL CONDITION OPTIONS

	DISPLA INPUT	DISPLA CALCUL	DISPLA POSTBU	VELOCI	STRESS INPUT	STRESS CALCUL
DISPLA INPUT	-	no	no	yes	yes	yes
DISPLA CALCUL	no	-	no	yes	yes	yes
DISPLA POSTBU	no	no	-	yes	yes	yes
VELOCI	yes	yes	yes	-	yes	yes
STRESS INPUT	yes	yes	yes	yes	-	no
STRESS CALCUL	yes	yes	yes	yes	no	-

13.3.1.2 Incremental Adaptation of Initial Conditions

The START STEPS commands apply incremental adaptation to the initial conditions, for instance initial stresses. You should use this option only if the internal forces following from initial stresses $\mathbf{f}_{\text{int}}^0$ are not in equilibrium with the initial external load $\mathbf{f}_{\text{ext}}^0$. Due to this option, DIANA first temporarily adds the initial out-of-balance force to the initial external load. Next, DIANA uses the load steps that you specify to remove this addition. The external loading in step *nstep* reads

$$\mathbf{f}_{\text{ext}} = \mathbf{f}_{\text{int}}^0 + \sum_{i=1}^{nstep} size_i \times (\mathbf{f}_{\text{ext}}^0 - \mathbf{f}_{\text{int}}^0) \quad (13.3)$$

The sum of explicitly specified step sizes must be equal to 1.

The initial external load is defined as the nonlinear load set *loset* [§ 13.3.1 p. 220]. If you do not specify START STEPS commands, then DIANA will apply the internal forces, following from the initial stresses, at once during the first step of the next EXECUT block.

Note that DIANA internally creates an additional load set for the load vector equal to $\mathbf{f}_{\text{ext}}^0 - \mathbf{f}_{\text{int}}^0$. This load set is assigned a load set number which is one higher than the highest user-defined load set number. This additional load set number will appear in logging and output.

syntax

```

BEGIN STEPS
[ OFF ]
[ RESTOR stepn ]
[ methodw ... ]
EXPLIC
ITERAT
ENERGY
AUTOMA
[ SAVE ... ]
END STEPS

```

RESTOR restores data for step *step* which must have been saved previously [§ 13.3.9 p. 245].

method indicates how the step sizes are to be chosen. Options and syntax are analogous [EXPLICIT] to load step control.

EXPLICIT explicitly specified step sizes [§ 13.3.2.1 p. 225].

ITERAT iteration based automatic step size control [§ 13.3.2.2 p. 226].

ENERGY energy based automatic step size control [§ 13.3.2.3 p. 226].

AUTOMA cutback based automatic step size control [§ 13.3.2.4 p. 227].

SAVE saves data of specified steps for future restart [§ 13.3.9 p. 245].

Default

file.dcf

```
*NONLIN
[ commands ]
BEGIN EXECUT
  START STEPS
  [ commands ]
END EXECUT
*END
```

Due to these commands DIANA will incrementally adapt the initial conditions in one step of size 1, which is equivalent to the following commands:

file.dcf

```
*NONLIN
[ commands ]
BEGIN EXECUT
  BEGIN START
    STEPS EXPLICIT SIZES 1.0
  END START
  [ commands ]
END EXECUT
*END
```

13.3.2 Load Steps

The LOAD commands specify the execution of load steps.

syntax

```
BEGIN LOAD
[ OFF ]
[ LOADNR=losetn ]
[ BEGIN STEPS
  [ RESTOR stepn ]
  [ methodw ... ]
  EXPLICIT
  ITERAT
  ENERGY
  AUTOMA
  [ SAVE ... ]
  END STEPS ]
END LOAD
```

LOADNR *loset* refers to a load set from table 'LOADS' [§ 2.3.8 p. 45]. Default is the lowest available load set number.

[STEPS] STEPS specifies how to apply step sizes. Via *method* you may specify the step sizes explicitly, or you may let DIANA determine them automatically.

RESTOR restores data for step *step* which must have been saved previously [§ 13.3.9 p. 245].

method indicates how the step sizes are to be chosen. [EXPLICIT]

EXPLICIT explicitly specified step sizes [§ 13.3.2.1 p. 225].

ITERAT iteration based automatic step size control [§ 13.3.2.2 p. 226].

ENERGY energy based automatic step size control [§ 13.3.2.3 p. 226].

AUTOMA cutback based automatic step size control [§ 13.3.2.4 p. 227].

SAVE saves data of specified steps for future restart [§ 13.3.9 p. 245].

In case time dependent material or geometrical properties are used, the current time is taken into account. The current time is zero by default [§ 13.3.1 p. 220].

Default

file.dcf

```
*NONLIN
[ commands ]
EXECUT LOAD
*END
```

Due to these commands DIANA will execute a load step with load set 1 and a factor of 1, applying the default integration scheme and solution procedure, but no stop criteria. The same would occur if you had given the following commands:

file.dcf

```
*NONLIN
[ commands ]
BEGIN EXECUT
  BEGIN LOAD
    LOADNR=lowest
  BEGIN STEPS
    EXPLICIT SIZES 1.0
  END STEPS
  END LOAD
END EXECUT
*END
```

13.3.2.1 Explicitly Specified Load Step Sizes

Via the EXPLICIT commands you may specify load step sizes explicitly.

syntax

```
BEGIN EXPLICIT
[ OFF ]
SIZES sizesr...
[ ARCLen ... ]
END EXPLICIT
```

SIZES *sizes* are explicitly specified load step sizes. The number of values specifies the number of steps to be executed.

ARCLen applies Arc-length control [§ 13.3.2.5 p. 227].

13.3.2.2 Iteration Based Adaptive Loading

The ITERAT commands cause automatic adaptive load increments, based on the number of iterations.

syntax

```

BEGIN ITERAT
[ OFF ]
[ INISIZ=isr ]
[ CONTIN ]
[ NITERA=nin ]
[ MAXSIZ=maxsr ]
[ MINSIZ=minsr ]
[ NSTEPS=nsn ]
[ GAMMA=gamr ]
[ ARCLEN ... ]
END ITERAT

```

[*is*=1.] INISIZ=*is* is the initial size for the first step.

CONTIN asks DIANA to ignore the initial step size *is* and to continue with the step size of the previous EXECUTE block.

[*ni*=6] NITERA=*ni* is the number of iterations (should be considered as optimal).

[*maxs*=10⁶] MAXSIZ=*maxs* is the upper limit of the step size.

[*mins*=10⁻³] MINSIZ=*mins* is the lower limit of the step size.

DIANA applies the specified limits in the zero-iteration of each step to determine a first estimation of the step size.

[*ns*=1] NSTEPS=*ns* is the number of steps [§ 46.1.5.3 p. 563].

[$\gamma = 0.5$] GAMMA=*gam* specifies the exponent γ from Eq. (46.33) on page 563 which we can now write as

$${}^{t+\Delta t}\Delta\lambda_0 = \frac{{}^t\Delta l}{\sqrt{\delta\mathbf{u}_0^T\delta\mathbf{u}_0}} \left(\frac{{}^n i}{{}^t N} \right)^\gamma \quad (13.4)$$

with ${}^t N$ the number of iterations from the previous step, this results in ${}^{t+\Delta t}\Delta\lambda_0$ the size increment for the current step.

ARCLEN applies Arc-length control [§ 13.3.2.5].

13.3.2.3 Energy Based Adaptive Loading

Due to the ENERGY commands DIANA will apply automatic adaptive load increments, based on energy.

syntax

```

BEGIN ENERGY
[ OFF ]
[ INISIZ=isr ]
[ CONTIN ]
[ MAXSIZ=maxsr ]
[ MINSIZ=minsr ]
[ NSTEPS=nsn ]
[ ARCLEN ... ]
END ENERGY

```

[*is*=1.] INISIZ=*is* is the initial size for the first step.

CONTIN asks DIANA to ignore the initial step size *is* and to continue with the step size of the previous EXECUTE block.

MAXSIZ=*maxs* is the upper limit of the step size. [*maxs*=10⁶]

MINSIZ=*mins* is the lower limit of the step size. [*mins*=10⁻³]

DIANA applies the specified limits in the zero-iteration of each step to determine a first estimation of the step size.

NSTEPS=*ns* is the number of steps [§ 46.1.5.3 p. 563]. [*ns*=1]

ARCLEN applies Arc-length control [§ 13.3.2.5 p. 227].

13.3.2.4 Cutback Based Adaptive Loading

The AUTOMA command causes cutback based automatic adaptive load increments. See § 46.1.5.4 on page 565 for background theory.

syntax

BEGIN AUTOMA

[OFF]

[SIZE=*size_r*]

[MINSIZ=*mins_r*]

[MAXSIZ=*maxs_r*]

[CUTBCK=*cutb_r*]

[MAXSTP=*maxstp_n*]

[ARCLEN ...]

END AUTOMA

SIZE=*size* is the total load increment. [*size*=1]

MINSIZ=*mins* is the lower limit of the step size, relative to *size*. [*mins*=10⁻⁴]

MAXSIZ=*maxs* is the upper limit of the step size, relative to *size*. [*maxs*=1]

CUTBCK=*cutb* is the factor by which the step size is scaled down in case of non-convergence. [*cutb*=0.25]

MAXSTP=*maxstp* is a limit to the number of steps that are taken.

ARCLEN applies Arc-length control [§ 13.3.2.5 p. 227].

13.3.2.5 Arc-length Control

The optional ARCLEN commands specify a selection of displacements to be accounted for in Arc-length iteration methods. This option is called ‘Indirect Displacement control’ and may be useful in case of local snap-through or snap-back behaviour.

Arc-length methods may only be used in combination with nodal or element loads, or with prescribed non-zero displacements, but not with time steps. Nodal or element loads and prescribed non-zero displacements may not be applied in the same load set. Arc-length methods adapt the loading during iterations in one load step. For adaptive loading in consecutive load steps use the adaptive loading options of the STEPS command [§ 13.3.2 p. 224]. See § 46.1.5.2 on page 562 for background theory.

syntax

```

BEGIN ARCLLEN
[ OFF ]
[ methodw ]
SPHERI
UPDATE
[ controlw ... ]
REGULA
CMOD
[ lodunlw ]
SIGN
PIVOTS
[ BEGIN AUTARC
[ OFF ]
[ STFPAR=stfparr ]
END AUTARC ]
END ARCLLEN

```

[UPDATE] *method* indicates the Arc-length method to be applied: SPHERI for the Spherical Path Arc-length method [§ 46.1.5.2 p. 562], UPDATE for the Updated Normal Plane Arc-length method [§ 46.1.5.2 p. 562].

[REGULA] *control* selects a set of degrees of freedom to be applied in Arc-length control: REGULA for regular indirect displacement control, CMOD for CMOD control.

lodunl specifies a loading–unloading method. In the Adaptive Arc-length methods, loading–unloading can be determined with various methods, see § 46.1.5.3 on page 564 for background theory.

[SIGN]

SIGN for automatic loading–unloading for the step sizes based on energy. The Updated Normal Plane method is recommended in this case [§ 13.3.2.5 p. 227].

PIVOTS for loading–unloading depending on existence of *negative pivots* in the triangulated stiffness matrix. This method may only be used in combination with a regular Newton–Raphson iteration process [§ 13.3.5.1 p. 236]. This method cannot be used in combination with an iterative solution procedure [§ 30.4 p. 423].

($0 < stfpar < 1$) AUTARC indicates that the arc-length method will not be activated immediately, but can become active when the model softens and its stiffness (compared to its initial stiffness) becomes smaller or equal to *stfpar* specified by the option STFPAR.

[*stfpar*=0.001]

Indirect Displacement Control

syntax

```

BEGIN REGULA
BEGIN SET
NODES nodesng...
TYPE typew
DIRECT dirn
ALPHA alphar
END SET
END REGULA

```

In regular indirect displacement Arc-length control the SET block defines a set of degrees of freedom [Eq. (46.31) p. 562]. You may specify multiple SET blocks.

NODES *nodes* is a series of nodes, specified by numbers and/or groups.

TYPE *type* specifies a degree of freedom type: TRANSL for translation or ROTATI for rotation.

DIRECT *dirnr* is a direction number referring to table 'DIRECT'.

ALPHA *alpha* and the direction determine the weight factors α .

file.dcf

```
*NONLIN
[ commands ]
BEGIN EXECUTE
  BEGIN LOAD
    LOADNR=2
  BEGIN STEPS
    BEGIN EXPLIC
      SIZES 0.5 0.3 0.1
    BEGIN ARCLEN
      UPDATE
    BEGIN REGULA
      BEGIN SET
        NODES 1 3 5 7-13
        TYPE TRANSL
        DIRECT 5
        ALPHA 1.0
      END SET
    END REGULA
  END ARCLEN
END EXPLIC
END STEPS
END EXECUTE
*END
```

CMOD control

syntax

```
BEGIN CMOD
  BEGIN SET
    BEGIN SIDE1
      NODES nodes1ng...
      TYPE type1w
      DIRECT dir1n
      ALPHA alpha1r
    END SIDE1
    BEGIN SIDE2
      NODES nodes2ng...
      TYPE type2w
      DIRECT dir2n
      ALPHA alpha2r
    END SIDE2
  END SET
[ BEGIN SET
  ...
  END SET ... ]
END CMOD
```

In CMOD Arc-length control the SET block defines related pairs of degrees of freedom [Eq. (46.32) p. 563]. You may specify multiple SET blocks, one for each crack mouth.

SIDE1 defines the degrees of freedom along the first side of the crack mouth.

NODES *nodes1* is a series of node numbers.

TYPE *type1* specifies a degree of freedom type: TRANSL for translation or ROTATI for rotation.

DIRECT *dir1* is a direction number referring to table 'DIRECT'.

ALPHA *alpha1* and the direction determine the weight factors α_1 for each of the degrees of freedom.

SIDE2 defines the degrees of freedom along the second side of the crack mouth.

NODES *nodes2* is a series of node numbers. There must be just as much nodes along both sides of the crack mouth.

TYPE *type2* specifies a degree of freedom type: TRANSL for translation or ROTATI for rotation.

DIRECT *dir2* is a direction number referring to table 'DIRECT'.

ALPHA *alpha2* and the direction determine the weight factors α_2 for each of the degrees of freedom.

file.dcf

```
*NONLIN
[ commands ]
BEGIN EXECUTE
  BEGIN LOAD
  LOADNR=2
  BEGIN STEPS
  BEGIN EXPLIC
  SIZES 0.5 0.3 0.1
  BEGIN ARCLEN
  UPDATE
  BEGIN CMOD
  BEGIN SET
    BEGIN SIDE1
      NODES 7
      TYPE  TRANS
      DIRECT 1
      ALPHA 1.0
    END SIDE1
    BEGIN SIDE2
      NODES 74
      TYPE  TRANS
      DIRECT 1
      ALPHA -1.0
    END SIDE2
  END SET
  BEGIN SET
    BEGIN SIDE1
      NODES 9
      TYPE  TRANSL
      DIRECT 2      1
      ALPHA 0.8
    END SIDE1
    BEGIN SIDE2
      NODES 75
      TYPE  TRANSL
      DIRECT 1
      ALPHA -0.9
    END SIDE2
  END SET
```



```

        END CMOD
        END ARCLLEN
        END EXPLIC
        END STEPS
        END LOAD
    END EXECUTE
*END

```

In the above example CMOD(1) is $1 \times$ the translation in direction 1 of node 7 minus $1 \times$ the translation in direction 1 of node 74, CMOD(2) is $0.8 \times$ the translation in direction 2 of node 9 minus $0.9 \times$ the translation in direction 1 of node 75.

13.3.3 Time Steps

The TIME commands specify the execution of time steps.

syntax

```

BEGIN TIME
[ OFF ]
[ BEGIN STEPS
  [ RESTOR stepn ]
  [ methodw ... ]
  EXPLIC
  ITERAT
  EXPONE
  AUTOMA
  [ SAVE ... ]
  END STEPS ]
END TIME

```

STEPS specifies how to apply step sizes.

[STEPS]

RESTOR restores data for step *step* which must have been saved previously [§ 13.3.9 p. 245].

method indicates how the step sizes are to be chosen.

[EXPLIC]

EXPLIC explicitly specified step sizes [§ 13.3.3.1 p. 232].

ITERAT iteration based automatic step size control [§ 13.3.3.2 p. 232].

EXPONE applies exponential time increments [§ 13.3.3.3 p. 233].

AUTOMA applies cutback based automatic time increments [§ 13.3.3.4 p. 233].

SAVE saves data of specified steps for future restart [§ 13.3.9 p. 245].

Default

file.dcf

```

*NONLIN
[ commands ]
EXECUT TIME
*END

```

Due to these commands DIANA will execute a time step with a factor of 1, applying the default integration scheme and solution procedure, but no stop criteria. The same would occur if you had given the following commands:

file.dcf

```

*NONLIN
[ commands ]
BEGIN EXECUT
  BEGIN TIME
    BEGIN STEPS
      EXPLIC SIZES 1.0
    END STEPS
  END TIME
END EXECUT
*END

```

13.3.3.1 Explicitly Specified Time Step Sizes

Via the EXPLICIT commands you may specify step sizes explicitly.

syntax

```

BEGIN EXPLIC
[ OFF ]
[ SIZES sizesr... ]
END EXPLIC

```

[SIZES 1.0] SIZES *sizes* are explicitly specified time step sizes, i.e., time increments. The number of values specifies the number of steps to be executed.

13.3.3.2 Iteration Based Adaptive Time Increments

The ITERAT commands cause automatic adaptive time increments, based on the number of iterations.

syntax

```

BEGIN ITERAT
[ OFF ]
[ INISIZ=isr ]
[ CONTIN ]
[ NITERA=nitrn ]
[ MAXSIZ=maxsr ]
[ MINSIZ=minsr ]
[ NSTEPS=nsn ]
[ GAMMA=gamr ]
END ITERAT

```

[*is*=1.] INISIZ=*is* is the initial size for the first step.

CONTIN asks DIANA to ignore the initial step size *is* and to continue with the step size of the previous EXECUTE block.

[*ni*=6] NITERA=*ni* is the number of iterations (should be considered as optimal).

[*maxs*=10⁶] MAXSIZ=*maxs* is the upper limit of the step size.

[*mins*=10⁻³] MINSIZ=*mins* is the lower limit of the step size.

DIANA applies the specified limits in the zero-iteration of each step to determine a first estimation of the step size.

[$ns=1$] **NSTEPS=ns** is the number of steps [§ 46.1.5.3 p. 563].

[$\gamma = 0.5$] **GAMMA=gam** specifies the exponent γ from Eq. (46.33) on page 563 which we can now write as

$${}^{t+\Delta t}\Delta\lambda_0 = \frac{{}^t\Delta l}{\sqrt{{}^\delta\mathbf{u}_0^\top {}^\delta\mathbf{u}_0}} \left(\frac{{}^ni}{{}^tN} \right)^{ga} \quad (13.5)$$

with tN the number of iterations from the previous step, this results in ${}^{t+\Delta t}\Delta\lambda_0$ the size increment for the current step.

13.3.3.3 Exponential Time Increments

The **EXPONE** commands invoke an exponential time increment algorithm. With this option DIANA determines the time increment Δt_i for time step i from

$$\Delta t_i = f^i \times \Delta t_0 \quad (13.6)$$

where Δt_0 is the initial time increment and f a multiplication factor. The initial time increment Δt_0 and the factor f are related to the total time increment Δt_{inc} by the power series

$$\Delta t_0 (1 + f + f^2 + f^3 + \dots + f^{n-1}) = \Delta t_0 \frac{f^n - 1}{f - 1} = t_{\text{inc}} \quad (13.7)$$

with n the number of time steps. To determine the time increment Δt_i for step i from Eq. (13.6) and Eq. (13.7) DIANA uses some input parameters, of which the first three are optional.

syntax

BEGIN EXPONE

[**OFF**]

[_____]

DELTAT=dt0_r

NSTEPS=ns_n

[**FACTOR=fac_r**]

[**TOTALT=tti_r**]

END EXPONE

DELTAT=dt0 is the initial time increment Δt_0 . If you specify this parameter, then DIANA determines an initial time increment close to Δt_0 that yields an integer number of steps n . ($0 < \Delta t_0 < \frac{1}{2}t_{\text{inc}}$)

NSTEPS=ns is the number of time steps n in the current **EXECUT** command block. ($n > 0$)

FACTOR=fac is the multiplication factor f which DIANA applies between two subsequent ($n = 10$)

time steps. ($f > 0$)

($f = 10$)

TOTALT=tti is the total time increment t_{inc} for the current **EXECUT** command block. ($t_{\text{inc}} > \Delta t_0$)

($t_{\text{inc}} = 1$)

13.3.3.4 Cutback Based Automatic Time Increments

The **AUTOMA** commands invoke a cutback based automatic time incremental algorithm [§ 46.1.5.4 p. 565]. If the **SDIRK2** time integration method is used, it is possible to obtain time accurate solutions, else it is equivalent to the cutback based automatic load stepping [§ 46.1.5.4 p. 565].

syntax

```

BEGIN AUTOMA
[ OFF ]
[ SIZE=sizer ]
[ MINSIZ=minsr ]
[ MAXSIZ=maxsr ]
[ CUTBCK=cutbr ]
[ MAXSTP=maxstpr ]
[ SDIRK2 { _____ } ]
          TOLREL=epsrelr
          TOLABS=epsabsr
END AUTOMA

```

- [*size*=1] SIZE=*size* is the total time increment.
- [*mins*=10⁻⁴] MINSIZ=*mins* is the lower limit of the step size, relative to *size*.
- [*maxs*=1] MAXSIZ=*maxs* is the upper limit of the step size, relative to *size*.
- [*cutb*=0.25] CUTBCK=*cutb* is the factor by which the step size is scaled down in case of non-convergence.

MAXSTP=*maxstp* is a limit to the number of steps that are taken.

SDIRK2 applies time integration error control according to the SDIRK2 method [§ 46.1.5.4 p. 565]. This method only makes sense if the solution depends on the time step size (dynamics, viscoelastic and viscoplastic behaviour). Two optional parameters customize the error criterion [Eq. (46.40) p. 565]:

- [$\epsilon_{\text{rel}} = 10^{-3}$] TOLREL=*epsrel* specifies the relative error criterion ϵ_{rel} .
- [$\epsilon_{\text{abs}} = 10^{-5}$] TOLABS=*epsabs* specifies the absolute error criterion ϵ_{abs} .

13.3.4 Physical Nonlinear Analysis Options

The PHYSIC commands overrule the use of physical nonlinear phenomena as specified in the input data file. If the PHYSIC commands are absent then DIANA uses all phenomena from the input data file.

syntax

```

BEGIN PHYSIC
[ OFF ]
[ BOND [ _____ ] ]
    REINFO reinfsng...
    OFF
[ SLIP [ _____ ] ]
    REINFO reinfsng...
    OFF
[ CLEAR [ OFF ] ]
[ SUPPRE { _____ } ]
    DISPLA
    STRAIN
    STRESS
    OFF
[ DRAINE [ OFF ] ]
[ LIQUEF [ OFF ] ]
END PHYSIC

```

BOND use the bond option even for bond-less reinforcement.

REINFO *reinfo* specifies a series of reinforcement numbers or groups or both for which the bond option must be switched on. This command overrules the **NOBOND** input option [Vol. *Material Library*].

OFF switches off the bond option for bond-less reinforcement, i.e., bonding depends on the presence of the **NOBOND** input option.

SLIP use the slip option for bond-less reinforcement.

REINFO *reinfo* specifies a series of reinforcement numbers or groups or both for which the slip option must be applied. The influence of loading, creep and/or shrinkage may cause changes in reinforcement strains and stresses in the time period between prestressing and grouting. The slip option provides a rather coarse method to model these changes. **DIANA** will calculate the mean value of the change in reinforcement strain for the whole reinforcement. This mean change of strain is then uniformly applied to the whole reinforcement. Effects of friction between the reinforcement and its mother elements are not taken into account. This immediately shows the limits to the applicability of this method. The slip option may be used to model a partly bonded reinforcement. Therefore you should identify the bonded and not bonded part of the reinforcement. Then you should replace the reinforcement with a series of bonded reinforcements and reinforcements without bonding, but with the slip option applied.

OFF switches off the slip option for bond-less reinforcement.

CLEAR resets the total displacement and rotation field to zero. This option is useful to discard the displacement field from previous phases and execute blocks.

SUPPRE suppress the superposition of current step results.

DISPLA for all degrees of freedom (translations, rotations, pressures, etc.).

STRAIN for strain related degrees of freedom (translations, rotations, special spline element degrees of freedom) and the total strain itself. This option is particularly useful for mixture analysis, to initialize pressures whilst keeping displacements and strains at zero level.

STRESS for stresses only. Note that **SUPPRE STRESS** is only applicable in combination with prescribed displacements. This option is particular useful in combination with the **DEFORM IMPORT** [§ 28.1.3 p. 402] loading in a phased geotechnical analysis for suppressing stresses when a new soil layer is placed.

DRAINE use drained behaviour even for elements with basically undrained behaviour. This command overrules the **UNDRAI** input option [Vol. *Material Library*].

OFF switches drained behaviour off, i.e., applies undrained behaviour. This is the default if you specify input item **UNDRAI** in table 'MATERI' [Vol. *Material Library*]. Undrained behaviour is particularly used during the loading of clay and peat. First the dead weight is applied in combination with drained behaviour, followed by the additional load in combination with undrained behaviour.

LIQUEF OFF deactivates all liquefaction models during the execution of the steps in the current **EXECUT** block [Vol. *Material Library*]. **DIANA** will use other constitutive models like (Modified) Mohr–Coulomb instead, provided that you have supplied the input for such a model in table 'MATERI'.

13.3.5 Equilibrium Iteration

The **ITERAT** commands define the equilibrium iteration process to be used for steps. See § 46.1.1 on page 552 for background theory.

syntax

```

BEGIN ITERAT
[ MAXITE= $m_i$  ]
[ CONTIN [ OFF ] ]
[ METHOD ... ]
[ LINESE ... ]
[ CONVER ... ]
END ITERAT

```

[$m_i=10$] MAXITE= m_i is the maximum number of iterations for each time or load step.

CONTIN for iteration with the Continuation method [§ 46.1.2 p. 557]. The OFF option switches off this method, which also is the default behaviour.

METHOD specifies the iteration method [§ 13.3.5.1].

LINESE invokes a Line Search algorithm to scale the incremental displacements [§ 13.3.5.2].

CONVER specifies the convergence criterion for the iteration process [§ 13.3.5.3].

Default iteration process. If you do not give any ITERAT commands, then the default is a Regular Newton–Raphson method with a maximum of five iterations. As convergence criteria, the norms of the force and displacement fields will be applied. This is equivalent to the following commands:

file.dcf

```

*NONLIN
[ commands ]
BEGIN EXECUT
[ commands ]
BEGIN ITERAT
MAXITE=5
METHOD NEWTON REGULA
BEGIN CONVER
FORCE NEWREF TERMIN TOLCON=1.E-2 TOLABT=1.E+4
DISPLA NEWREF TERMIN TOLCON=1.E-2 TOLABT=1.E+4
END CONVER
END ITERAT
[ commands ]
END EXECUT
*END

```

13.3.5.1 Iteration Method

With the METHOD commands you may choose one out of various iteration methods: Constant or Linear Stiffness, Newton–Raphson, or Quasi-Newton (Secant).

syntax

```

BEGIN METHOD
[ _____ ]
CONSTA
LINEAR
NEWTON [ _____ [ _____ ] ]
        REGULA TANGEN
        MODIFI  LINEAR
        PREVIO

```

```

SECANT  [ _____ [ _____ ] ]
        BROYDE  TANGEN
        BFGS    LINEAR
        CRISFI  PREVIO

```

END METHOD

CONSTA invokes the Constant Stiffness method [§ 46.1.1.3 p. 556]. This command may also be used to ‘freeze’ the tangent stiffness matrix.

LINEAR invokes the Linear Stiffness method [§ 46.1.1.3 p. 556]. This command initiates an explicit run to the linear elastic stiffness, which is particularly attractive in case of sudden unloading.

NEWTON invokes a Newton–Raphson iteration method [§ 46.1.1.1 p. 553]. The default Newton–Raphson method is Regular. [REGULA]

REGULA for Regular Newton–Raphson DIANA sets up the tangential stiffness before each iteration.

MODIFI for Modified Newton–Raphson, DIANA sets up the tangential stiffness before the first iteration and subsequently keeps it constant during the step.

By default, the Newton–Raphson methods set up the tangential stiffness before each iteration. An option specifies an alternative for the first iteration:

TANGEN sets up a new tangential stiffness.

LINEAR uses the linear stiffness.

PREVIO uses the stiffness of the last iteration of the previous step.

SECANT invokes a Quasi-Newton (Secant) iteration method [§ 46.1.1.2 p. 554]. [BFGS]

BROYDE Broyden’s method.

BFGS The BFGS method (the default).

CRISFI Crisfield’s method.

By default, the Quasi-Newton methods do not set up a new tangential stiffness at the start of each step, but the tangential stiffness of the last iteration of the previous step is reused. An option specifies an alternative start: [PREVIO]

TANGEN sets up a new tangential stiffness.

LINEAR uses the linear stiffness.

PREVIO uses the stiffness of the last iteration of the previous step.

13.3.5.2 Line Search

DIANA offers a Line Search algorithm to scale the incremental displacements in the iteration process automatically. As outlined in § 46.1.3 on page 557, this may stabilize the convergence behaviour or increase the convergence speed.

Line Search cannot be applied in contact analysis [§ 46.2.5 p. 571].

You may invoke a Line Search algorithm via the **LINESE** commands. The algorithm may be guided by one or more optional parameters. Usually the default values of these parameters are satisfactory.

syntax

```

BEGIN LINESE
[ OFF ]
[ ETAMAX=maxscr ]
[ ETAMIN=minscr ]
[ PSI=psir ]
[ DETA=detar ]
[ MAXLS=mlsn ]
END LINESE

```

$[\eta_{\max} = 1]$ ETAMAX=*maxsc* is the upper bound for the Line Search scaling factor η_{\max} . The default is equivalent to ‘no extrapolation’.

$[\eta_{\min} = 0.1]$ ETAMIN=*minsc* is the lower bound for the Line Search scaling factor η_{\min} .

$[\Psi = 0.8]$ PSI=*psi* is the stop criterion Ψ . Line Search stops if the energy change (residue \times displacement variation) is less than $\Psi \times$ the original value.

$[\Delta\eta = 0.1]$ DETA=*deta* is the stop criterion $\Delta\eta$. Line Search stops if the interval in a Regula-Falsi process is less than $\Delta\eta$.

$[mls = 5]$ MAXLS=*mls* is the maximum number of line searches in each global iteration.

Default

file.dcf

```

*NONLIN
[ commands ]
BEGIN EXECUT
[ commands ]
ITERAT LINESE
END EXECUT
*END

```

Due to these commands DIANA will apply a Line Search algorithm with default values for all parameters which is equivalent to the following commands:

file.dcf

```

*NONLIN
[ commands ]
BEGIN EXECUT
[ commands ]
BEGIN ITERAT
LINESE ETAMAX=1.0 ETAMIN=0.1 PSI=0.8 DETA=0.1 MAXLS=5
END ITERAT
END EXECUT
*END

```

13.3.5.3 Convergence Criteria

The CONVER commands specify the convergence criteria for the equilibrium iteration process. See § 46.1.4 on page 558 for theoretical backgrounds. Multiple CONVER commands may be specified to use more than one criterion simultaneously. DIANA terminates the iteration process if one of the specified criteria is satisfied.

syntax

```

BEGIN CONVER
[ SIMULT ]
[ critw ] [ OFF ] [ referw ] [ noconvw ] { _____ } ...
FORCE          NEWREF    TERMIN    TOLCON=tcr
DISPLA          OLDREF    CONTIN     TOLABT=tar
ENERGY
RESIDU
END CONVER

```

SIMULT will let DIANA assume convergence when all specified criteria are satisfied simultaneously.

crit is the name of the criterion: **FORCE** based on the external loading and out-of-balance forces (the default), **DISPLA** based on displacements, **ENERGY** based on internal energy, or **RESIDU** based on the out-of-balance force vector. [FORCE]
[DISPLA]

OFF switches off the criterion **crit**.

refer specifies whether or not DIANA must determine a new reference norm: with option **NEWREF** a new norm is determined at the start of each step (the default), with option **OLDREF** no norm will be determined and the previously stored norm will be used. [NEWREF]

*The reference norm must at least be determined once, before it is used.
Only in cases where a new reference norm would result in an extremely
high or low value, it can be advantageous to use the old reference norm.
In other words: option OLDREF must be used with great care.*

noconv indicates what to do if no convergence occurs within the maximum number of iterations. With option **TERMIN** DIANA stops the analysis run, no further steps will be executed (the default). However, further steps may be executed in subsequent runs. With option **CONTIN** the analysis run will be continued, you should check the relevance of the analysis results carefully in this case. [TERMIN]

Two parameters specify the tolerance values. The default values depend on the used criterion as indicated in the examples below.

TOLCON=tc is the tolerance for convergence. When the norm has become less than **tc** × the reference norm, DIANA assumes sufficient accuracy, stops the iteration process and continues with the next step.

TOLABT=ta is the tolerance for divergence. When the norm exceeds a value of **ta** × the reference norm, DIANA assumes divergence and aborts the nonlinear analysis.

Default criteria

file.dcf

```

[ commands ]
BEGIN EXECUT
[ commands ]
BEGIN ITERAT
  CONVER
END ITERAT
END EXECUT
*END

```

If, like in the above commands, you only give the **CONVER**, or if you omit this command completely, then DIANA will simultaneously apply the criteria based on external loading and on displacements. This is equivalent to the following commands:

file.dcf

```
[ commands ]
BEGIN EXECUT
[ commands ]
BEGIN ITERAT
  CONVER FORCE DISPLA
END ITERAT
END EXECUT
*END
```

Default tolerances*file.dcf*

```
*NONLIN
[ commands ]
BEGIN EXECUT
[ commands ]
BEGIN ITERAT
  CONVER FORCE DISPLA ENERGY
END ITERAT
END EXECUT
*END
```

If, like in the above commands, you do not specify parameters for the tolerances, then the default values depend on the type of the criterion as shown below.

file.dcf

```
*NONLIN
[ commands ]
BEGIN EXECUT
[ commands ]
BEGIN ITERAT
  BEGIN CONVER
    FORCE  NEWREF TERMIN TOLCON=1.E-2 TOLABT=1.E+4
    DISPLA NEWREF TERMIN TOLCON=1.E-2 TOLABT=1.E+4
    ENERGY NEWREF TERMIN TOLCON=1.E-4 TOLABT=1.E+4
  END CONVER
END ITERAT
END EXECUT
*END
```

13.3.6 Stop Criteria for Step Execution

If you specify a stop criterion via a **STOP** command, then DIANA will stop the execution of the load or time steps in the current **EXECUT** command block. The analysis can be continued in a subsequent **EXECUT** command block. You may specify a stop criterion based on loading, or based on resulting strain or stress. By default, DIANA will not apply a stop criterion.

syntax

```
BEGIN STOP
[ OFF ]
[ LOAD ... ]
[ RESULT ... ]
END STOP
```

LOAD specifies stop criteria based on loading [§ 13.3.6.1].

RESULT specifies stop criteria based on results [§ 13.3.6.2].

Multiple criteria*file.dcf*

```

*NONLIN
[ commands ]
BEGIN EXECUT
[ commands ]
BEGIN STOP
  BEGIN LOAD
    TOTAL 3.567
    INCREM 0.27
  END LOAD
  BEGIN RESULT
    STRAIN PRINCI INTPNT MINVAL=-3.5E-3
  END RESULT
END STOP
END EXECUT
*END

```

If, like in the above commands, you specify multiple **STOP** commands then DIANA will use the criteria simultaneously.

13.3.6.1 Loading Based Stop

Criteria based on loading are useful in combination with Arc-length methods [§ 13.3.2.5 p. 227] or adaptive loading [§ 13.3.3.2 p. 232]. If a loading based stop criterion is specified then DIANA takes the number of steps from the **SIZES** command as a maximum number of steps. The stop criterion will stop the execution of the current **EXECUTE** command block. The analysis will continue with the next block (if any).

syntax

```

BEGIN LOAD
[ OFF ]
{ _____ }
TOTAL totlodr
INCREM inclodr
SIGN
END LOAD

```

TOTAL stops execution of steps when a total load *totlod* is reached.

INCREM stops execution of steps when the incremental load is less than *inclod*.

SIGN stops execution of steps when the sign of the load vector changes. This criterion is for use in combination with automatic loading-unloading or loading-unloading based on negative pivots [§ 13.3.2.5 p. 228].

13.3.6.2 Result Based Stop

With a stop criterion based on analysis results, DIANA will stop the execution of the steps in the current **EXECUT** command block if the value of certain element or nodal results exceeds a specified extreme.

syntax

```

BEGIN RESULT
[ OFF ]
[ BEGIN SELECT
...
END SELECT ]
itemw [typew] [formw] [operw] [compw] [locaw] _____ ...

```

```

STRAIN
STRESS
DISPLA
VELOCI
ACCELE
FORCE
NODFOR
END RESULT

```

```

MAXVAL= $max_r$ 
MINVAL= $min_r$ 

```

SELECT selects certain parts of the model for which the stop criterion must be applied. For complete syntax description see § 3.6.2 on page 59. By default DIANA applies the specified results criterion on the entire model.

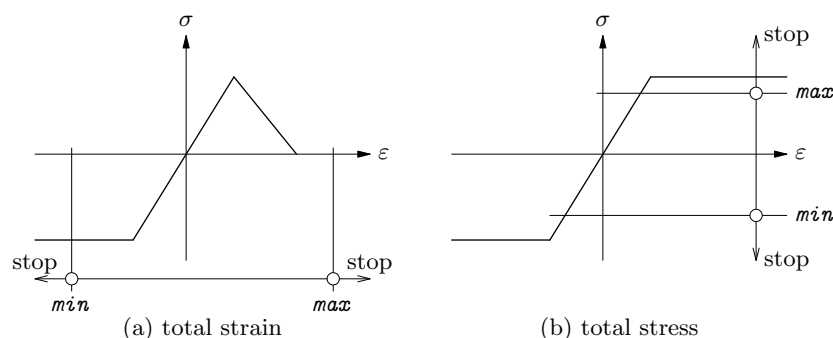


Figure 13.1: Result based stop criteria

STRAIN applies a result stop criterion based on the total Green–Lagrange strain [Fig. 13.1-a].² The specifiers for operation (transformation), component selection and location, *oper*, *comp* and *loca*, are described in § 13.4.2 on page 248.

STRESS applies a result stop criterion based on the total Cauchy stress [Fig. 13.1b].³ The specifiers for operation (transformation), component selection and location, *oper*, *comp* and *loca*, are described in § 13.4.3 on page 253.

DISPLA applies a result stop criterion based on displacements. The specifiers for type, formulation, operation (transformation), and component selection, *type*, *form*, *oper*, and *comp*, are described in § 7.4.1 on page 126.

VELOCI applies a result stop criterion based on velocities. The specifiers for type, formulation, operation (transformation), and component selection, *type*, *form*, *oper*, and *comp*, are described in § 7.4.2 on page 128.

ACCELE applies a result stop criterion based on accelerations. The specifiers for type, formulation, operation (transformation), and component selection, *type*, *form*, *oper*, and *comp*, are described in § 7.4.3 on page 129.

FORCE applies a result stop criterion based on nodal forces. The specifiers for type, formulation, operation (transformation), and component selection, *type*, *form*, *oper*, and *comp*, are described in § 13.4.7 on page 270.

NODFOR applies a result stop criterion based on nodal element forces. The specifiers for type, formulation, operation (transformation), and component selection, *type*, *form*, *oper*, and *comp*, are described in § 13.4.8 on page 271.

Be sure to select a result which is available in the model selection, or else the stop criterion will be void.

²The specifiers TOTAL GREEN for type and formulation are optional.

³The specifiers TOTAL CAUCHY for type and formulation are optional.

MAXVAL *max* is the maximum value that the strain or stress may reach before DIANA stops execution of steps.

MINVAL *min* is the minimum value that the strain or stress may reach before DIANA stops execution of steps.

file.dcf

```
*NONLIN
[ commands ]
BEGIN EXECUTE
[ commands ]
BEGIN STOP
  BEGIN RESULT
    SELECT ELEMEN CONCRETE
    STRAIN PRINCI INTPNT MINVAL=-3.5E-3
  END RESULT
END STOP
END EXECUTE
*END
```

Due to the commands in the example above DIANA will apply a stop criterion based on a maximum principal strain of -3.5×10^{-3} in the integration points of elements in group CONCRETE. When the criterion is satisfied, DIANA will terminate the analysis job and issue the following message on the standard output file:

file.out

```
EXECUTION STOPPED ON STRAIN TOTAL GREEN PRINCI INTPNT CRITERION
IN ELEMENT 324 THE VALUE IS -0.3635E-02; CRITERION IS -0.3500E-02
CONTINUED ANALYSIS POSSIBLE
```

13.3.7 Logging

You may ask DIANA to produce a certain amount of log information during step execution via the LOGGIN commands.

syntax

```
BEGIN LOGGIN
[ OFF ]
[ REPORT [ verbow ] [ whenw ] ]
          BRIEF EXECUT
          FULL STEP
          ITERAT

[ PLASTI [ OFF ] ]
[ CRACKI [ OFF ] ]
[ REACTI [ OFF ] ]
END LOGGIN
```

OFF turns logging off, i.e., DIANA will not give any log information. When logging is on (the default), you may customize the logging information via the following commands:

[ON]

REPORT indicates what to be logged and when.

verbo indicates the verbosity, i.e., the amount of log information to be produced: BRIEF gives a brief summary, FULL gives full log information including a summary.

[BRIEF]

when indicates when log information will be produced: EXECUT at the termination [STEP] of the last step, STEP at the termination of each of the executed steps, or ITERAT at the termination of each iteration.

[ON] PLASTI turns the logging of the plastic points in the model on or off.

[ON] CRACKI turns the logging of the cracks in the model on or off.

[ON] REACTI turns the logging of the cumulative reaction forces and moments in the model on or off.

Default

file.dcf

```
*NONLIN
[ commands ]
BEGIN EXECUT
[ commands ]
END EXECUT
*END
```

If, like in the above example, you do not specify LOGGIN commands then DIANA will briefly log the occurrence of plasticity and cracking, and cumulative reaction forces and moments after each load step. This is also achieved via the following commands:

file.dcf

```
*NONLIN
[ commands ]
BEGIN EXECUT
[ commands ]
BEGIN LOGGIN
  REPORT BRIEF STEP
  PLASTI
  CRACKI
  REACTI
END LOGGIN
END EXECUT
*END
```

Comprehensive

file.dcf

```
*NONLIN
[ commands ]
BEGIN EXECUT
[ commands ]
BEGIN LOGGIN
  REPORT FULL ITERAT
  PLASTI
  CRACKI
  REACTI
END LOGGIN
END EXECUT
*END
```

These commands ask for the most comprehensive logging: full information of plasticity and cracking, including a summary, and the cumulative reaction forces and moments at the end of each iteration.

13.3.8 Initial State References

In geotechnical analyses, the initial state may be set during initialization of the nonlinear analysis by means of the REFERE commands. Due to these commands, DIANA will evaluate the initial state at the end of the current EXECUT block, after the postprocessing of the last step of this block.

*syntax*BEGIN REFERE

[OFF]

{ _____ }

WEIGHT

PRESSU

STRESS

END REFERE

WEIGHT causes initial state procedures, like application of the lateral pressure ratio K_0 , to be applied using the results of the last step. See the section on input of the initial stress ratio in Volume *Material Library*,

If the initial state is a result of some nonlinear steps, you may use the WEIGHT command to let DIANA (re)apply initial state procedures.

PRESSU applies the results of the last step to compute the initial values used for computation of the excess pore pressure ratio r_p [§ 13.4.3 p. 255] and the excess pore pressure. See also Part VII on flow–stress analysis. [PRESSU]

STRESS saves the stress state of the last step to be used by engineering liquefaction analysis [Ch. 15 p. 287] or engineering creep analysis [Ch. 16 p. 291].

Default*file.dcf*

*NONLIN

[commands]

BEGIN EXECUT

[commands]

REFERE

END EXECUT

*END

These commands cause the last step of the current EXECUT block to be selected as the (new) reference for the determination of the initial geotechnical state. The same would occur if you had given the following commands:

file.dcf

*NONLIN

[commands]

BEGIN EXECUT

[commands]

REFERE PRESSU

END EXECUT

*END

13.3.9 Save/Restore Steps

With the SAVE and RESTOR commands you may save or restore executed steps.

Note that the SAVE/RESTOR commands to save or restore executed steps in a nonlinear structural analysis are different from the FILOS SAVE/RESTOR commands to save and restore items from the FILOS file [§ 3.2.2 p. 49].

syntax

```
[ RESTOR stepn ]
[ SAVE [            ] ]
      stepsn...
      ALL
      LAST
      CONVER
```

RESTOR restarts the analysis starting from a previously saved step, where *step* is an absolute step number. This step must have been saved previously via the SAVE command.

[ALL] SAVE indicates to save steps.

steps are absolute step numbers of the steps to be saved.

ALL will save all steps that are executed.

LAST will save the last step executed.

CONVER will save the last converged step.

13.4 Output of Analysis Results

You may indicate the analysis results to be output via the OUTPUT command block. Depending on the level where this block is specified, the selection is valid for all executed steps [Ch. 13 p. 211], or for the steps of the current EXECUT block [§ 13.3 p. 219]. For the general, analysis type independent options for output of analysis results see § 3.6 on page 55.

syntax

```
BEGIN OUTPUT [devicew] [outoptw...] [params]
[ OFF ]
[ BEGIN SELECT
  [ ... ]
  [ STEPS ... ]
END SELECT ]
[ LAYOUT ... ]
itemw ...
DISPLA
VELOCI
ACCELE
STRAIN
STRESS
STATUS
PRESSU
TEMPER
CONCEN
MATURI
FORCE
NODFOR
ELMFOR
PARAME
FRACTU
FSPRES
END OUTPUT
```

SELECT command block to customize the batch output.

... for model selection see § 3.6.2 on page 59, for stress- and strain transformation § 3.6.2.3 on page 60.

STEPS selects load or time steps for output [§ 13.4.1 p. 248].

LAYOUT optional commands to customize the layout and style of tabular output [§ 3.6.4 p. 64].

item is the name of the analysis result to be output. See § 3.6.1 on page 56 for complete syntax of this command.

DISPLA for displacements, syntax and output is analogous to linear transient analysis [§ 7.4.1 p. 126].

VELOCI for velocities, syntax and output is analogous to linear transient analysis [§ 7.4.2 p. 128].

ACCELE for accelerations, syntax and output is analogous to linear transient analysis [§ 7.4.3 p. 129].

STRAIN for strains [§ 13.4.2 p. 248].

STRESS for stresses [§ 13.4.3 p. 253].

STATUS for status [§ 13.4.4 p. 260].

PRESSU for pore pressure [§ 13.4.5 p. 268].

TEMPER for temperature [§ 13.4.6 p. 269].

CONCEN for concentration [§ 13.4.6 p. 269].

MATURI for maturity [§ 13.4.6 p. 269].

FORCE for nodal forces and moments [§ 13.4.7 p. 270].

NODFOR for element nodal forces [§ 13.4.8 p. 271].

ELMFOR for internal element forces [§ 13.4.9 p. 272].

PARAME for model parameters [§ 13.4.10 p. 274].

FRACTU for Linear Elastic Fracture Mechanics analysis parameters of crack tip elements [§ 13.4.11].

FSPRES for dynamic pressures of fluid-structure interface elements [§ 13.4.12].

file.dcf

```
*NONLIN
[ commands ]
BEGIN EXECUTE
[ commands ]
BEGIN OUTPUT TABULA
  SELECT STEPS 2 4 6 MAX
  DISPLA TOTAL TRANSL GLOBAL
END OUTPUT
END EXECUTE
*END
```

These commands could give the following tabular output.

file.tb

Analysis type	NONLIN					
Extreme results	MAXIMUM					
Result	DISPLA TOTAL TRANSL					
Axes	GLOBAL					
Extreme values	TDtX	2.932E-02	Stepnr	44	Nodnr	6
	TDtY	9.231E-03	Stepnr	92	Nodnr	9
	TDtZ	0.000E+00	Stepnr	1	Nodnr	1
Nodnr	TDtX	TDtY	TDtZ			
1	1.349E-02 (100)	0.000E+00 (1)	0.000E+00 (1)			
2	1.349E-02 (100)	0.000E+00 (1)	0.000E+00 (1)			
3	1.349E-02 (100)	0.000E+00 (1)	0.000E+00 (1)			

4	1.288E-02 (99)	-7.838E-05 (1)	0.000E+00 (1)
5	8.943E-03 (100)	-7.838E-05 (1)	0.000E+00 (1)
6	2.932E-02 (44)	3.531E-03 (93)	0.000E+00 (1)
7	2.108E-03 (20)	-5.283E-05 (1)	0.000E+00 (1)
8	6.586E-04 (39)	5.793E-03 (94)	0.000E+00 (1)
9	1.422E-02 (45)	9.231E-03 (92)	0.000E+00 (1)

This tabular output shows a maximum value for displacement component u_X at node 6 to be 0.02932 for step number 44. The maximum value for u_X at node 1 is 0.01349 for step 100. Notice that DIANA writes the maximum value for all selected output points just below the header paragraph.

13.4.1 Step Selection

The STEPS command selects steps for output of analysis results.

syntax

```

STEPS [ _____ ] { _____ }
      stepsn...    MIN
      ALL           MAX
      LAST

```

steps are numbers of selected steps.

[ALL] ALL will produce output for all steps that are executed (the default).

LAST selects the final step.

steps are absolute step numbers, and ALL/LAST refer to all steps/the last step of an EXECUT block.

DIANA will output the extreme values that occurred up to and including the selected steps if you specify one of the options:

MIN for the minimum values,

MAX for the maximum values.

For step-wise analyses output of extreme values cannot be used in combination with the option LAYOUT COMBIN [§ 3.6.4.1 p. 65] to assemble various results in one table for tabular output.

13.4.2 Strains

syntax

```

STRAIN [ typew ] [ formw ] [ operw ] { compw } { locaw } { optiw }
      TOTAL  GREEN  LOCAL  INTPNT  AVERAG
      ELASTI  FORCE  GLOBAL  NODES   ...
      PLASTI  MOMENT PRINCI  CENTER
      CREEP   DISFOR VONMIS
      CRACK   DISMOM REAXES
      CRKSUM  TRACTI VOLUME
      CRKWDT          DISSEI
      TEMPER          VOID
      CONCEN          JANFOR
      SHRINK          JANMOM

```

STRAIN specifies strains as output item. Table 13.2 on the next page outlines the availability and applicability of the various strain output options for each of the element families.

Table 13.2: AVAILABILITY OF STRAIN OUTPUT FOR NONLINEAR ANALYSIS

<i>item</i>	STRAIN	truss	beam	pl. stress	pl. strain	axisymm.	plate bend.	fl. shell	cu. shell	solid	str. interface	spring	p. mass	e. reinfo.
<i>type</i>	TOTAL	a	a	a	a	a	ae	ae	a	a	a	a	-	a
	ELASTI	a	a	a	a	a	-	-	a	a	-	-	-	i
	PLASTI	a	a	a	a	a	-	-	a	a	-	-	-	i
	CREEP	a	a	a	a	a	-	-	a	a	-	-	-	-
	CRACK	ad	ad	ad	ad	ad	-	-	ad	ad	-	-	-	-
	CRKSUM	ad	ad	ad	ad	ad	-	-	ad	ad	-	-	-	-
	CRKWDT	ad	ad	ad	ad	ad	-	-	ad	ad	-	-	-	-
	TEMPER	a	a	a	a	a	-	-	a	a	-	-	-	i
	CONCEN	a	a	a	a	a	-	-	a	a	-	-	-	-
<i>form</i>	SHRINK	a	a	a	a	a	-	-	a	a	-	-	-	-
	GREEN	a	a	a	a	a	-	ae	a	a	-	-	-	a
	FORCE	-	c	-	-	-	-	-	-	-	-	a	-	g
	MOMENT	-	c	-	-	-	-	-	-	-	-	a	-	h
	DISFOR	-	-	-	-	-	a	a	-	-	-	-	-	-
	DISMOM	-	-	-	-	-	a	a	-	-	-	-	-	-
	TRACTI	-	-	-	-	-	-	-	-	-	a	-	-	g
	DISSEI	-	-	-	-	-	-	-	-	-	f	-	-	-
<i>oper</i>	VOID	-	-	-	a	a	-	-	-	a	-	-	-	-
	LOCAL	a	a	a	a	a	a	a	a	a	a	a	-	a
	GLOBAL	a	a	a	a	a	a	a	a	a	-	a	-	g
	PRINCI	a	a	a	a	a	a	a	a	a	-	-	-	-
	VONMIS	a	a	a	a	a	a	a	a	a	-	-	-	-
	REAXES	-	-	a	-	-	ae	ae	a	a	-	-	-	-
	VOLUME	a	a	a	a	a	a	a	a	a	-	-	-	-
	JANFOR	-	-	-	-	-	-	-	-	-	j	-	-	-
<i>loca</i>	JANMOM	-	-	-	-	-	-	-	-	-	j	-	-	-
	INTPNT	a	k	a	a	a	a	a	a	a	a	-	-	i
	NODES	a	a	a	a	a	a	a	a	a	-	a	-	g
<i>center</i>	CENTER	a	a	a	a	a	a	a	a	a	a	-	-	-
<i>opti</i>	AVERAG	a	a	a	a	a	-	-	a	a	-	-	-	-

(a) All elements. (b) Not for class-I. (c) For all beam elements, class-II and class-III only in combination with local coordinate system, results are derived from **STRESS FORCE** and **STRESS MOMENT** and translated assuming linear material behaviour in the beam elements. (d) Only in integration points. (e) Only linear elastic. (f) Only for plane structural interface elements. (g) Only for bond-slip reinforcements. (h) Only for bond-slip reinforcements modeled by beam elements. (i) Not for bond-slip reinforcements. (j) Only for line interface to shell elements. (k) Only for class-II and class-III beam elements. (-) Not available or not suitable.

[TOTAL] *type* specifies the strain type.

TOTAL for total strains.

ELASTI for elastic strains.

PLASTI for plastic strains.

CREEP for creep strains of metal creep or transient creep models. Note that this output is not available for the creep of concrete under long term load modeled with viscoelasticity.

CRACK for crack strains in each crack individually. For total strain crack models the crack strain ε_{cr} is defined as:

$$\varepsilon_{cr} = \varepsilon_{nst} - \frac{\sigma}{E} \quad (13.8)$$

CRKSUM for summed crack strains over all cracks in an element using the following procedure: the crack strains at each integration point due to each crack are transformed to the element axes system and then summed up for all cracks at that integration point.

CRKWDT for crack width (= CRKSUM * crack bandwidth h_{cr}).

TEMPER for thermal strains.

CONCEN for concentration strains.

SHRINK for volumetric shrinkage strains $\varepsilon_{\text{vol}}^{\text{sh}}$. Shrinkage strains are always expressed as positive.

[GREEN] *form* specifies the strain formulation.

GREEN for Green–Lagrange strains [§ 13.4.2.1].

FORCE for deformations due to normal and shear forces [§ 13.4.2.2].

MOMENT for curvatures due to bending moments [§ 13.4.2.2].

DISFOR for generalized strains [§ 13.4.2.2].

DISMOM for distributed curvatures due to bending moments [§ 13.4.2.2].

TRACTI for tractions in structural interface elements [§ 13.4.2.3].

DISSEI for distributed seismic moments in plane structural interface elements [§ 13.4.2.4].

VOID for void-ratio [§ 13.4.2.5].

[GLOBAL] *oper* specifies an operation to be performed on the primary strains [§ 3.6.1 p. 57]. Two specific operations are available for line interface to shell elements with the Janssen material model:

JANFOR for averaged relative displacements in local *xyz* directions [§ 13.4.2.3].

JANMOM for relative rotations around the tangential direction. [§ 13.4.2.3].

comp selects strain components for output. Default is all available components.

loca specifies the location for the strains to be output [§ 3.6.1 p. 58].

opti are additional options.

AVERAG is specific for nonlinear analysis with the plasticity fraction model [Vol. *Material Library*], it gives the average value of all the fractions of the fraction model.

... see § 3.6.1 on page 58 for other options.

13.4.2.1 Green–Lagrange Strains

Total strains				<i>comp</i> ...								
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	XX	YY	ZZ	XY	YZ	ZX	1	2	3
STRAIN TOTAL	GREEN	LOCAL		E _{xx}	E _{yy}	E _{zz}	G _{xy}	G _{yz}	G _{zx}			
				ε_{xx}	ε_{yy}	ε_{zz}	γ_{xy}	γ_{yz}	γ_{zx}			
STRAIN TOTAL	GREEN	GLOBAL		EXX	EYY	EZZ	GXY	GYZ	GZX			
				ε_{XX}	ε_{YY}	ε_{ZZ}	γ_{XY}	γ_{YZ}	γ_{ZX}			
STRAIN TOTAL	GREEN	PRINCI								E1	E2	E3
										ε_1	ε_2	ε_3
STRAIN TOTAL	GREEN	REAXES								E1RA	E2RA	
										ε_1^a	ε_2^a	
STRAIN TOTAL	GREEN	VOLUME								Evol		
										ε_{vol}		

Primary elastic strains				<i>comp</i> ...								
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	XX	YY	ZZ	XY	YZ	ZX	1	2	3
STRAIN ELASTI	GREEN	LOCAL		E _{exx}	E _{eyy}	E _{ezz}	G _{exy}	G _{eyz}	G _{ezx}			
				ε_{xx}^e	ε_{yy}^e	ε_{zz}^e	γ_{xy}^e	γ_{yz}^e	γ_{zx}^e			
STRAIN ELASTI	GREEN	GLOBAL		EeXX	EeYY	EeZZ	GeXY	GeYZ	GeZX			
				ε_{XX}^e	ε_{YY}^e	ε_{ZZ}^e	γ_{XY}^e	γ_{YZ}^e	γ_{ZX}^e			
STRAIN ELASTI	GREEN	PRINCI								Ee1	Ee2	Ee3
										ε_1^e	ε_2^e	ε_3^e
STRAIN ELASTI	GREEN	REAXES								Ee1RA	Ee2RA	
										$\varepsilon_1^{\text{ea}}$	$\varepsilon_2^{\text{ea}}$	
STRAIN ELASTI	GREEN	VOLUME								Eevol		
										$\varepsilon_{\text{vol}}^e$		

Primary plastic strains				<i>comp</i> ...						
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	XX	YY	ZZ	XY	YZ	ZX	1 2 3
STRAIN PLASTI	GREEN	LOCAL		Ep _{xx} ϵ_{xx}^p	Ep _{yy} ϵ_{yy}^p	Ep _{zz} ϵ_{zz}^p	Gp _{xy} γ_{xy}^p	Gp _{yz} γ_{yz}^p	Gp _{zx} γ_{zx}^p	
STRAIN PLASTI	GREEN	GLOBAL		Ep _{XX} ϵ_{XX}^p	Ep _{YY} ϵ_{YY}^p	Ep _{ZZ} ϵ_{ZZ}^p	Gp _{XY} γ_{XY}^p	Gp _{YZ} γ_{YZ}^p	Gp _{ZX} γ_{ZX}^p	
STRAIN PLASTI	GREEN	PRINCI								Ep1 ϵ_1^p Ep2 ϵ_2^p Ep3 ϵ_3^p
STRAIN PLASTI	GREEN	REAXES								Ep1RA ϵ_1^{pa} Ep2RA ϵ_2^{pa}
STRAIN PLASTI	GREEN	VOLUME								Epvol ϵ_{vol}^p

Primary creep strains				<i>comp</i> ...						
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	XX	YY	ZZ	XY	YZ	ZX	1 2 3
STRAIN CREEP	GREEN	LOCAL		Ec _{xx} ϵ_{xx}^c	Ec _{yy} ϵ_{yy}^c	Ec _{zz} ϵ_{zz}^c	Gc _{xy} γ_{xy}^c	Gc _{yz} γ_{yz}^c	Gc _{zx} γ_{zx}^c	
STRAIN CREEP	GREEN	GLOBAL		Ec _{XX} ϵ_{XX}^c	Ec _{YY} ϵ_{YY}^c	Ec _{ZZ} ϵ_{ZZ}^c	Gc _{XY} γ_{XY}^c	Gc _{YZ} γ_{YZ}^c	Gc _{ZX} γ_{ZX}^c	
STRAIN CREEP	GREEN	PRINCI								Ec1 ϵ_1^c Ec2 ϵ_2^c Ec3 ϵ_3^c
STRAIN CREEP	GREEN	REAXES								Ec1RA ϵ_1^{ca} Ec2RA ϵ_2^{ca}
STRAIN CREEP	GREEN	VOLUME								Ecvol ϵ_{vol}^c

Von Mises strains				
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	
STRAIN TOTAL	GREEN	VONMIS		Eeq ϵ_{eq}
STRAIN ELASTI	GREEN	VONMIS		Eeeq ϵ_{eq}^e
STRAIN PLASTI	GREEN	VONMIS		Epeq ϵ_{eq}^p
STRAIN CREEP	GREEN	VONMIS		Eceq ϵ_{eq}^c

Crack strains				<i>comp</i> ...		
<i>item</i>	<i>type</i>	<i>form</i>	<i>loca</i>	NN	NS	NT
STRAIN CRACK	GREEN	INTPNT		Ek _{nn} ϵ_{nn}^{cr}	Gk _{ns} γ_{ns}^{cr}	Gk _{nt} γ_{nt}^{cr}

Summed crack strains				<i>comp</i> ...						
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	XX	YY	ZZ	XY	YZ	ZX	1 2 3
STRAIN CRKSUM	GREEN	LOCAL		Ek _{xx} ϵ_{xx}^k	Ek _{yy} ϵ_{yy}^k	Ek _{zz} ϵ_{zz}^k	Gk _{xy} γ_{xy}^k	Gk _{yz} γ_{yz}^k	Gk _{zx} γ_{zx}^k	
STRAIN CRKSUM	GREEN	GLOBAL		Ek _{XX} ϵ_{XX}^k	Ek _{YY} ϵ_{YY}^k	Ek _{ZZ} ϵ_{ZZ}^k	Gk _{XY} γ_{XY}^k	Gk _{YZ} γ_{YZ}^k	Gk _{ZX} γ_{ZX}^k	
STRAIN CRKSUM	GREEN	PRINCI								Ek1 ϵ_1^k Ek2 ϵ_2^k Ek3 ϵ_3^k
STRAIN CRKSUM	GREEN	VONMIS		Ekeq ϵ_{eq}^k						

Crack width				<i>comp</i> ...								
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	XX	YY	ZZ	XY	YZ	ZX	1	2	3
STRAIN	CRKWDT	GREEN	LOCAL	Ecwx	Ecwy	Ecwz	Gcwxy	Gcwyz	Gcwzx			
				ε_{xx}^{cw}	ε_{yy}^{cw}	ε_{zz}^{cw}	γ_{xy}^{cw}	γ_{yz}^{cw}	γ_{zx}^{cw}			
STRAIN	CRKWDT	GREEN	GLOBAL	EcwXX	EcwYY	EcwZZ	GcwXY	GcwYZ	GcwZX			
				ε_{XX}^{cw}	ε_{YY}^{cw}	ε_{ZZ}^{cw}	γ_{XY}^{cw}	γ_{YZ}^{cw}	γ_{ZX}^{cw}			
STRAIN	CRKWDT	GREEN	PRINCI							Ecw1	Ecw2	Ecw3
										ε_1^{cw}	ε_2^{cw}	ε_3^{cw}
STRAIN	CRKWDT	GREEN	VONMIS	Ecweq								
				ε_{eq}^{cw}								

Thermal strains				<i>comp</i> ...					
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	XX	YY	ZZ	XY	YZ	ZX
STRAIN	TEMPER	GREEN	LOCAL	Etmpxx	Etmpyy	Etmpzz	Gtmpxy	Gtmpyz	Gtmpzx
				ε_{xx}^T	ε_{yy}^T	ε_{zz}^T	γ_{xy}^T	γ_{yz}^T	γ_{zx}^T
STRAIN	TEMPER	GREEN	GLOBAL	EtmpXX	EtmpYY	EtmpZZ	GtmpXY	GtmpYZ	GtmpZX
				ε_{XX}^T	ε_{YY}^T	ε_{ZZ}^T	γ_{XY}^T	γ_{YZ}^T	γ_{ZX}^T

Concentration strains				<i>comp</i> ...					
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	XX	YY	ZZ	XY	YZ	ZX
STRAIN	CONCEN	GREEN	LOCAL	Econxx	Econyy	Econzz	Gconxy	Gconyz	Gconzx
				ε_{xx}^C	ε_{yy}^C	ε_{zz}^C	γ_{xy}^C	γ_{yz}^C	γ_{zx}^C
STRAIN	CONCEN	GREEN	GLOBAL	EconXX	EconYY	EconZZ	GconXY	GconYZ	GconZX
				ε_{XX}^C	ε_{YY}^C	ε_{ZZ}^C	γ_{XY}^C	γ_{YZ}^C	γ_{ZX}^C

Shrinkage strains		
<i>item</i>	<i>type</i>	<i>form</i>
STRAIN	SHRINK	GREEN
		Eshrvo
		ε_{vol}^{sh}

13.4.2.2 Deformations and Curvatures

Force deformations				<i>comp</i> ...		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	X	Y	Z
STRAIN	TOTAL	FORCE	LOCAL	Px		
				Δu_x		
STRAIN	TOTAL	FORCE	GLOBAL	PX	PY	PZ
				Δu_X	Δu_Y	Δu_Z

Generalized strains				<i>comp</i> ...							
<i>itm</i>	<i>type</i>	<i>form</i>	<i>oper</i>	XX	YY	ZZ	XY	YZ	ZX	1	2
STRAIN	TOTAL	DISFOR	LOCAL	Pxx	Pyy	Pzz	Pxy	Pyz	Pzx		
				Ψ_{xx}	Ψ_{yy}	Ψ_{zz}	Ψ_{xy}	Ψ_{yz}	Ψ_{zx}		
STRAIN	TOTAL	DISFOR	REAXES							P1RA	P2RA
										Ψ_1^a	Ψ_2^a

Curvatures				<i>comp</i> ...		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	X	Y	Z
STRAIN	TOTAL	MOMENT	LOCAL	Kx	Ky	Kz
				κ_x	κ_y	κ_z

Distributed curvatures				<i>comp</i> ...					
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	XX	YY	ZZ	XY	1	2
STRAIN	TOTAL	DISMOM	LOCAL	Kxx	Kyy	Kzz	Kxy		
				κ_{xx}	κ_{yy}	κ_{zz}	κ_{xy}		
STRAIN	TOTAL	DISMOM	REAXES					K1RA	K2RA
								κ_1^a	κ_2^a

13.4.2.3 Relative Displacements of Interface Elements

Note that the labels for the results in the local xyz directions are dependent on the interface element type [Vol. Element Library]. Output items related to the normal direction(s) are indicated with ‘N’ and items related to the shear direction(s) are indicated with ‘S’ in their name followed by the letter indicating the corresponding element direction.

Relative displacements				comp ...		
item	type	form	oper	X	Y	Z
STRAIN TOTAL TRACTI LOCAL				DUNx	DUNy	DUNz
				DUSx	DUSy	DUSz
				Δu_x	Δu_y	Δu_z
STRAIN TOTAL TRACTI GLOBAL				DUX	DUY	DUZ
				Δu_X	Δu_Y	Δu_Z
STRAIN TOTAL TRACTI JANFOR				DUJx	DUJy	DUJz
				Δu_x	Δu_y	Δu_z
STRAIN TOTAL TRACTI JANMOM				DRJy		
				$\Delta \phi_x$		

Elastic relative displacements				comp ...		
item	type	form	oper	X	Y	Z
STRAIN ELASTI TRACTI LOCAL				DUENx	DUENy	DUENz
				DUESx	DUESy	DUESz
				Δu_x^e	Δu_y^e	Δu_z^e
STRAIN ELASTI TRACTI GLOBAL				DUEX	DUEY	DUEZ
				Δu_X^e	Δu_Y^e	Δu_Z^e

Plastic relative displacements				comp ...		
item	type	form	oper	X	Y	Z
STRAIN PLASTI TRACTI LOCAL				DUPNx	DUPNy	DUPNz
				DUPSx	DUPSy	DUPSz
				Δu_x^p	Δu_y^p	Δu_z^p
STRAIN PLASTI TRACTI GLOBAL				DUPX	DUPY	DUPZ
				Δu_X^p	Δu_Y^p	Δu_Z^p

13.4.2.4 Distributed Seismic Moment

Distributed seismic moment [§ 47.1.4]						
item	type	form				
STRAIN TOTAL DISSEI				PSTOT	PSNEG	PSPOS
				\mathcal{P}_S	\mathcal{P}_{S-}	\mathcal{P}_{S+}

13.4.2.5 Void Ratio

The void ratio can be output for elements with the Modified Mohr–Coulomb or Cam-clay material models [Vol. Material Library].

Void ratio			
item	type	form	
STRAIN TOTAL VOID	VOID		e

13.4.3 Stresses

syntax

STRESS	[<i>type_w</i>]	[<i>form_w</i>]	[<i>oper_w</i>]	{ <i>comp_w</i> }	{ <i>loca_w</i> }	{ <i>opti_w</i> }
	TOTAL	CAUCHY	LOCAL		INTPNT	NOBOND
	EFFECT	PIOLAK	GLOBAL		NODES	CENTAX
	CRACK	FORCE	PRINCI		CENTER	...
		MOMENT	VONMIS			
		DISFOR	INVARI			
		DISMOM	REAXES			
		TRACTI	MAXSHR			
		GRADIE	ERATIO			
		SHEAR	CRKIND			
			BIAXFE			
			MOHRCO			
			HOEKBR			
			FRICTI			
			REFGLB			
			REFINV			
			JANFOR			
			JANMOM			

STRESS specifies stresses as output item. Table 13.3 on the next page outlines the availability and applicability of the various strain output options for each of the element families.

[TOTAL] *type* specifies the stress type.

TOTAL for total stresses.

EFFECT for effective stresses.

CRACK for crack stresses.

In porous materials, like soil, load is carried by effective stresses and the pressure of the fluid in the pores. The pore pressure result can be output by the item PRESSU [§ 13.4.5 p. 268]. The stresses between the grains of the soil are the effective stresses.

Note, that conform the sign convention in DIANA where compressive stresses are negative, the total stresses are the effective stresses minus the pore pressures.

When no pore pressure is available, such as in structural elements, the total and effective stresses are equal.

[CAUCHY] *form* specifies the stress formulation.

CAUCHY for Cauchy stresses [§ 13.4.3.1].

PIOLAK for Piola–Kirchhoff stresses [§ 13.4.3.2].

FORCE for concentrated force or tractions [§ 13.4.3.3].

MOMENT for concentrated moment [§ 13.4.3.4].

DISFOR for distributed force [§ 13.4.3.3].

DISMOM for distributed moment [§ 13.4.3.4].

TRACTI for tractions in structural interface and contact elements [§ 13.4.3.5].

GRADIE for gradients of stresses in reinforcement bars [§ 4.2.4.5 p. 87].

SHEAR for shear stress in the reinforcement mother element connection [§ 4.2.4.6 p. 87].

[GLOBAL] *oper* specifies an operation to be performed on the primary stresses [§ 3.6.1 p. 57]. A specific operation is available for the maximum shear stress:

Table 13.3: AVAILABILITY OF STRESS OUTPUT FOR NONLINEAR ANALYSIS

<i>item</i>	STRESS	truss	beam	pl. stress	pl. strain	axisymm.	plate bend.	fl. shell	cu. shell	solid	str. interface	contact	spring	p. mass	e. reinfo.	comp. line	comp. surf.
<i>type</i>	TOTAL	a	a	a	a	a	ac	ac	a	a	a	a	a	a	a	a	a
	EFFECT	-	-	-	a	a	-	-	-	a	a	-	-	-	-	-	-
	CRACK	a	a	a	a	a	-	-	a	a	-	-	-	-	-	-	-
<i>form</i>	CAUCHY	a	a	a	a	a	ac	ac	a	a	-	-	-	-	a	-	-
	PIOLAK	a	a	a	a	a	-	-	a	a	-	-	-	-	-	-	-
	FORCE	a	a	-	-	-	-	-	-	-	-	a	-	m	a	-	-
	MOMENT	-	a	-	-	-	-	-	-	-	-	a	-	g	a	-	-
	DISFOR	-	-	a	k	l	ac	ac	a	-	-	-	-	-	-	-	a
	DISMOM	-	-	-	k	l	ac	ac	a	-	-	-	-	-	-	-	a
	TRACTI	-	-	-	-	-	-	-	-	-	a	a	-	-	f	-	-
	GRADIE	-	-	-	-	-	-	-	-	-	-	-	-	-	b	-	-
SHEAR	-	-	-	-	-	-	-	-	-	-	-	-	-	b	-	-	
<i>oper</i>	LOCAL	a	a	a	a	a	a	a	a	a	a	a	-	a	a	a	a
	GLOBAL	a	a	a	a	a	a	a	a	a	a	-	d	-	f	-	-
	PRINCI	a	a	a	a	a	a	a	a	a	-	-	-	-	-	-	-
	VONMIS	a	a	a	a	a	a	a	a	a	-	-	-	-	-	-	-
	INVARI	-	-	a	a	a	-	-	-	a	-	-	-	-	-	-	-
	REAXES	-	-	a	-	-	ac	ac	a	a	-	-	-	-	-	-	-
	MAXSHR	-	-	a	a	a	-	-	a	a	-	-	-	-	-	-	-
	ERATIO	-	-	-	a	a	-	-	-	a	-	-	-	-	-	-	-
	CRKIND	ai	hi	ai	ai	ai	-	-	ai	ai	-	-	-	-	-	-	-
	BIAEFE	a	a	a	a	a	-	-	a	a	-	-	-	-	-	-	-
	MOHRCO	-	-	a	a	-	-	-	-	a	-	-	-	-	-	-	-
	HOEKBR	-	-	a	a	-	-	-	-	a	-	-	-	-	-	-	-
	FRICTI	-	-	-	-	-	-	-	-	-	a	-	-	-	-	-	-
	REFGLB	-	-	a	a	-	-	-	-	a	-	-	-	-	-	-	-
	REFINV	-	-	a	a	-	-	-	-	a	-	-	-	-	-	-	-
JANFOR	-	-	-	-	-	-	-	-	-	-	j	-	-	-	-	-	
JANMOM	-	-	-	-	-	-	-	-	-	-	j	-	-	-	-	-	
<i>loca</i>	INTPNT	a	e	a	a	a	a	a	a	a	a	-	-	h	-	-	-
	NODES	a	a	a	a	a	a	a	a	a	-	-	a	-	f	a	a
	CENTER	a	a	a	a	a	a	a	a	a	a	-	-	-	-	-	-
<i>opti</i>	NOBOND	-	a	-	-	-	-	-	a	-	-	-	-	-	-	-	-
	CENTAX	-	-	-	-	-	-	-	-	-	-	-	-	-	-	a	a

(a) All elements. (b) Only for bar reinforcements. (c) Only linear elastic. (d) Only for base springs. (e) Only for class-II and class-III beams. (f) Only for bond-slip reinforcements. (g) Only for bond-slip reinforcements modeled by beam elements. (h) Not for bond-slip reinforcements. (i) Only linear elastic and viscoelastic. (j) Only for line interface to shell elements. (k) For infinite shells only. (l) For shells of revolution only. (m) Only for bar reinforcements and bond-slip reinforcements. (-) Not available or not suitable.

MAXSHR gives the maximum shear stress $\tau_{\max} = \frac{\sigma_1 - \sigma_3}{2}$ with σ_1 and σ_3 the highest and lowest principal stress respectively. For plane stress elements the maximum shear stress τ_{\max} is defined as $\tau_{\max} = \frac{\sigma_1 - \sigma_2}{2}$ because the third principal stress σ_3 is zero by definition.

A specific operation is available for excess pore water pressure ratio:

ERATIO is specific for the excess pore water pressure ratio r_p in geotechnical analysis which is a function of the effective isotropic stress invariant p' and may be regarded as an utilization ratio for the excess pore pressure:

$$r_p = 1 - \frac{p'}{p'_{\text{ref}}} \quad (13.9)$$

DIANA derives the value for p'_{ref} from the initial stress field as specified in a START command [§ 13.3.1 p. 220].

A specific operation is available for crack indices:

CRKIND gives the crack indices I_{cr} (the crack indicator) and F_{tu} (the tensile strength utilization) which correspond to indications whether a crack arises or not:

$$I_{cr}(t) \geq \gamma_{cr} \quad \text{with} \quad I_{cr}(t) = \frac{f_t(t)}{\sigma_I(t)} \quad (13.10)$$

$$F_{tu}(t) \leq \beta_{cr} \quad \text{with} \quad F_{tu}(t) = \frac{\sigma_I(t)}{f_t(t)} \quad (13.11)$$

where f_t is the tensile strength of concrete, σ_I is the maximal principal stress and γ_{cr} and β_{cr} are safety factors. In order to avoid an infinite value of the crack indicator, DIANA will automatically set I_{cr} to 100 if $\sigma_I \leq 0.01f_t$. And in order to avoid negative values of the tensile strength utilization, DIANA will automatically set F_{tu} to 0 if $\sigma_I \leq 0$.

To determine the crack indices DIANA needs the values of the tensile strength in time $f_t(t)$. You must specify these as a material property [Vol. *Material Library*].

A specific operation is available for Safety Factors for concrete under static and dynamic loading conditions with reference to a biaxial failure envelope:

BIAXFE gives the following Safety Factors for concrete under static and dynamic loading conditions with reference to a biaxial failure envelope:

- $FS_{static\ usual} = R_{static\ usual}/r$
- $FS_{static\ unusual} = R_{static\ unusual}/r$
- $FS_{dynamic\ unusual} = R_{dynamic\ unusual}/r$
- $FS_{dynamic\ extreme} = R_{dynamic\ extreme}/r$

Where r is the distance from the origin to the actual stress point (σ_1, σ_2) :

$$r = \sqrt{(\sigma_1^2 + \sigma_2^2)} \quad (13.12)$$

To determine the Safety Factors DIANA needs additional material properties. For the required parameters and additional information on the Safety Factors, see Volume *Material Library*.

Note that all Safety Factors are limited to 100.

A specific operation is available for shear capacity of stress against Mohr–Coulomb failure criterion:

MOHRCO gives the shear capacity of stress against Mohr–Coulomb failure criterion [§ 47.2.9.1 p. 582]. To determine the shear capacity DIANA needs the the initial cohesion c (COHESI) and initial friction angle ϕ (PHI) of the Mohr–Coulomb material model [Vol. *Material Library*].

A specific operation is available for shear capacity of stress against Hoek–Brown failure criterion:

HOEKBR gives the shear capacity of stress against Hoek–Brown failure criterion [§ 47.2.9.2 p. 582]. To determine the shear capacity DIANA needs the unconfined compressive strength σ_{ci} of the (intact) rock sample (COMSTR), Hoek–Brown constant m_b (HOEKMB), and the Hoek–Brown constant s (HOEKS) material properties of the Hoek–Brown rock plasticity material model [Vol. *Material Library*].

A specific operation is available for shear capacity of stress against Coulomb friction failure criterion for structural interface elements:

FRICTI gives the shear capacity of stress against Coulomb friction criterion for structural interface elements [§ 47.2.9.3 p. 583]. To determine the shear capacity DIANA needs the initial cohesion c (COHESI) and initial friction angle ϕ (PHI) of the Coulomb friction interface material model [Vol. *Material Library*].

Two specific operations are available for stress changes with respect to the last stress state that has been marked with the **REFERE STRESS** command [§ 13.3.8 p. 244].

REFGLB gives the stress change as tensor in the global coordinate system with respect to the last stress state that has been marked with the **REFERE STRESS** command.

REFINV gives the stress change as tensor in terms of stress invariants with respect to the last stress state that has been marked with the **REFERE STRESS** command.

Two specific operations are available for line interface to shell elements with the Janssen material model:

JANFOR for distributed forces in local xyz directions [§ 13.4.3.5].

JANMOM for the distributed moment around the tangential direction. [§ 13.4.3.5].

comp selects stress components for output. Default is all available components.

loca specifies the location for the stresses to be output [§ 3.6.1 p. 58].

opti are additional options

NOBOND includes the contribution of prestress in posttensioned reinforcements to element forces and moments [Vol. *Element Library*].

CENTAX will automatically shift the specified composed line or composed surface to the centroidal axis of the structure and output the moment and/or distributed moment with respect to the shifted composed line and/or surface.

... see § 3.6.1 on page 58 for other options.

13.4.3.1 Cauchy Stresses

Primary stresses				<i>comp</i> ...								
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	XX	YY	ZZ	XY	YZ	ZX	1	2	3
STRESS TOTAL	CAUCHY	LOCAL		Sxx	Syy	Szz	Sxy	Syz	Szx			
				σ_{xx}	σ_{yy}	σ_{zz}	σ_{xy}	σ_{yz}	σ_{zx}			
STRESS EFFECT	CAUCHY	LOCAL		SExx	SEyy	SEzz	SExy	SEyz	SEzx			
				σ_{exx}	σ_{eyy}	σ_{ezz}	σ_{exy}	σ_{eyz}	σ_{ezx}			
STRESS TOTAL	CAUCHY	GLOBAL		SXX	SYX	SZZ	SXY	SYZ	SZX			
				σ_{XX}	σ_{YY}	σ_{ZZ}	σ_{XY}	σ_{YZ}	σ_{ZX}			
STRESS EFFECT	CAUCHY	GLOBAL		SEXX	SEYY	SEZZ	SEXY	SEYZ	SEZX			
				σ_{eXX}	σ_{eYY}	σ_{eZZ}	σ_{eXY}	σ_{eYZ}	σ_{eZX}			
STRESS TOTAL	CAUCHY	PRINCI								S1	S2	S3
										σ_1	σ_2	σ_3
STRESS EFFECT	CAUCHY	PRINCI								SE1	SE2	SE3
										σ_{e1}	σ_{e2}	σ_{e3}
STRESS TOTAL	CAUCHY	REAXES								S1RA	S2RA	
										σ_1^a	σ_2^a	

Von Mises stress				
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	
STRESS TOTAL	CAUCHY	VONMIS		Seq
				σ_{eq}
STRESS EFFECT	CAUCHY	VONMIS		SEeq
				σ_{eeq}

Stress invariants				<i>comp</i> ...
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	P Q LODE
STRESS TOTAL	CAUCHY	INVARI		P Q Lode
				p q θ
STRESS EFFECT	CAUCHY	INVARI		P Q Lode
				p' q θ

Maximum shear stress			
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>
STRESS TOTAL	CAUCHY	MAXSHR	Tmax τ_{\max}
STRESS INITIA	CAUCHY	MAXSHR	T0max τ_{\max}^0

Excess pore pressure			
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>
STRESS EFFECT	CAUCHY	ERATIO	SEer r_p

Crack stresses				<i>comp</i> ...		
<i>item</i>	<i>type</i>	<i>form</i>	<i>loca</i>	NN	NS	NT
STRESS CRACK	CAUCHY	INTPNT		Sknn σ_{nn}^{cr}	Skns σ_{ns}^{cr}	Sknt σ_{nt}^{cr}

Crack index				<i>comp</i> ...	
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	ICR	FTU
STRESS TOTAL	CAUCHY	CRKIND		Icr I_{cr}	Ftu F_{tu}

Biaxial failure envelope				<i>comp</i> ...			
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	STUS	STUN	DYUN	DYEX
STRESS TOTAL	CAUCHY	BIAXFE		FSstus FS_{stus}	FSstun FS_{stun}	FSdyun FS_{dyun}	FSdyex FS_{dyex}

Shear capacity [§ 47.2.9]				<i>comp</i> ...	
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	SHRCAP	
STRESS TOTAL	CAUCHY	MOHRCO		SHRCAP ψ	
STRESS EFFECT	CAUCHY	MOHRCO		SHRCAP ψ_e	
STRESS TOTAL	CAUCHY	HOEKBR		SHRCAP ψ	
STRESS EFFECT	CAUCHY	HOEKBR		SHRCAP ψ_e	
STRESS TOTAL	TRACTI	FRICTI		SHRCAP ψ	
STRESS EFFECT	TRACTI	FRICTI		SHRCAP ψ_e	

Primary stress changes				<i>comp</i> ...					
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	XX	YY	ZZ	XY	YZ	ZX
STRESS TOTAL	CAUCHY	REFGLB		dSXX $\Delta\sigma_{XX}$	dSYY $\Delta\sigma_{YY}$	dSZZ $\Delta\sigma_{ZZ}$	dSXY $\Delta\sigma_{XY}$	dSYZ $\Delta\sigma_{YZ}$	dSZX $\Delta\sigma_{ZX}$
STRESS EFFECT	CAUCHY	REFGLB		dSEX $\Delta\sigma_{eXX}$	dSEY $\Delta\sigma_{eYY}$	dSEZ $\Delta\sigma_{eZZ}$	dSEXY $\Delta\sigma_{eXY}$	dSEYZ $\Delta\sigma_{eYZ}$	dSEZX $\Delta\sigma_{eZX}$

Stress change invariants				<i>comp</i> ...		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	P	Q	LODE
STRESS TOTAL	CAUCHY	REFINV		P Δp	Q Δq	Lode $\Delta\theta$
STRESS EFFECT	CAUCHY	REFINV		P $\Delta p'$	Q Δq	Lode $\Delta\theta$

13.4.3.2 Piola–Kirchhoff Stresses

Primary stresses				<i>comp</i> ...								
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	XX	YY	ZZ	XY	YZ	ZX	1	2	3
STRESS TOTAL	PIOLAK	LOCAL		SPxx	SPyy	SPzz	SPxy	SPyz	SPzx			
				S_{xx}	S_{yy}	S_{zz}	S_{xy}	S_{yz}	S_{zx}			
STRESS EFFECT	PIOLAK	LOCAL		SPExx	SPEyy	SPEzz	SPExy	SPEyz	SPEzx			
				S_{exx}	S_{eyy}	S_{ezz}	S_{exy}	S_{eyz}	S_{ezx}			
STRESS TOTAL	PIOLAK	GLOBAL		SPXX	SPYY	SPZZ	SPXY	SPYZ	SPZX			
				S_{XX}	S_{YY}	S_{ZZ}	S_{XY}	S_{YZ}	S_{ZX}			
STRESS EFFECT	PIOLAK	GLOBAL		SPEXX	SPEYY	SPEZZ	SPEXY	SPEYZ	SPEZX			
				S_{eXX}	S_{eYY}	S_{eZZ}	S_{eXY}	S_{eYZ}	S_{eZX}			
STRESS TOTAL	PIOLAK	PRINCI								SP1	SP2	SP3
										S_1	S_2	S_3
STRESS EFFECT	PIOLAK	PRINCI								SPE1	SPE2	SPE3
										S_{e1}	S_{e2}	S_{e3}
STRESS TOTAL	PIOLAK	REAXES								SP1RA	SP2RA	
										S_1^a	S_2^a	

13.4.3.3 Forces

Concentrated forces				<i>comp</i> ...		
<i>type</i>	<i>type</i>	<i>form</i>	<i>oper</i>	X	Y	Z
STRESS TOTAL FORCE	LOCAL			Nx	Qy	Qz
				N_x	Q_y	Q_z
STRESS TOTAL FORCE	GLOBAL			NX	NY	NZ
				N_X	N_Y	N_Z

Distributed forces				<i>comp</i> ...							
<i>type</i>	<i>type</i>	<i>form</i>	<i>oper</i>	XX	YY	ZZ	XY	YZ	ZX	1	2
STRESS TOTAL DISFOR	LOCAL			Nxx	Nyy	Nzz	Nxy	Qyz	Qxz		
				n_{xx}	n_{yy}	n_{zz}	n_{xy}	q_{yz}	q_{xz}		
STRESS TOTAL DISFOR	REAXES									N1RA	N2RA
										n_1^a	n_2^a

13.4.3.4 Moments

Concentrated moments				<i>comp</i> ...		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	X	Y	Z
STRESS TOTAL MOMENT	LOCAL			Mx	My	Mz
				M_x	M_y	M_z
STRESS TOTAL MOMENT	GLOBAL			MX	MY	MZ
				M_X	M_Y	M_Z

Distributed moments				<i>comp</i> ...						
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	XX	YY	ZZ	XY	1	2	
STRESS TOTAL DISMOM	LOCAL			Mxx	Myy	Mzz	Mxy			
				m_{xx}	m_{yy}	m_{zz}	m_{xy}			
STRESS TOTAL DISMOM	REAXES							M1RA	M2RA	
								m_1^a	m_2^a	

13.4.3.5 Traction

Note that the labels for the results in the local xyz directions are dependent on the interface element type [Vol. Element Library]. Output items related to the normal direction(s) are indicated with ‘N’ and items related to the shear direction(s) are indicated with ‘S’ in their name followed by the letter indicating the corresponding element direction.

Traction				<i>comp</i> ...		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	X	Y	Z
STRESS TOTAL	TRACTI	LOCAL		STNx	STNy	STNz
				STSx	STSy	STSz
				t_x	t_y	t_z
STRESS TOTAL	TRACTI	GLOBAL		STX	STY	STZ
				t_X	t_Y	t_Z
STRESS EFFECT	TRACTI	LOCAL		STENx	STENy	STENz
				STESx	STESy	STESz
				t_{ex}	t_{ey}	t_{ez}
STRESS EFFECT	TRACTI	GLOBAL		STEX	STEY	STEZ
				t_{eX}	t_{eY}	t_{eZ}
STRESS TOTAL	TRACTI	JANFOR		NJxx	QJxy	QJxz
				t_x	t_y	t_z
STRESS TOTAL	TRACTI	JANMOM		MJxx		
				m_z		

13.4.4 Status

The post-analysis result **STATUS** is used in DIANA to output the internal state parameters of the material model at the integration points of an element, for instance the cracks, the plastic points etc. For each internal state parameter, a syntax will be presented in the following.

13.4.4.1 Plasticity Status

The plasticity status indicates whether irreversible plastic strains have occurred at an integration point.

syntax

STATUS [*type_w*] { *loca_w* } { *opti_w* }

PLASTI INTPNT

STATUS PLASTI specifies the plasticity status as output item.

[INTPNT] *loca* specifies the location for the plasticity status to be output [§ 3.6.1 p. 58]. For plasticity status only output in the integration points is possible.

opti are additional options [§ 3.6.1 p. 58].

The following types of plasticity status can appear in the output:

1. The status is *elastic*, and has always been so.
2. The status is *plastic*. In this case the plastic strain is also available for output via the STRAIN PLASTI options [Table 13.2 p. 249].
3. The status is elastic but was *previously plastic*.
4. The status of a Cam-clay model is *critical* [Vol. Material Library].

In the output, the plasticity status is indicated with a text string which depends on the *device*.

Plasticity status						
<i>item</i>	<i>type</i>	<i>device</i>	1	2	3	4
STATUS	PLASTI					
		TABULA	ELASTIC	PLASTIC	PREVIOUSLY PLASTIC	CRITICAL
		FEMVIE	Elastic	Plastic	Pr.plast	Critical
		FXPLUS		PL	E!	CR

13.4.4.2 Crack Status

The cracking status indicates whether a tension cut-off limit has been exceeded at an integration point.

syntax

STATUS [*type_w*] { *loca_w* } { *opti_w* }

CRACK INTPNT

STATUS CRACK for crack status.

loca specifies the location for the status to be output [§ 3.6.1 p. 58]. For cracking status [INTPNT] only output in the integration points is possible.

opti are additional options [§ 3.6.1 p. 58].

The following types of crack status can appear in the output (indicating the branch on the tension softening diagram [Fig. 13.2]):

1. The status is *no crack* if there is no crack yet.
2. The status is *open* for cracks on the fully open loading branch.
3. The status is *closed* for fully closed cracks, the material is elastic again in compression.
4. The status is *active* for cracks on the partially open loading branch.
5. The status is *inactive* for cracks on the fully open unloading branch.
6. The status is *partial* for cracks on the partially open (secant) unloading branch.

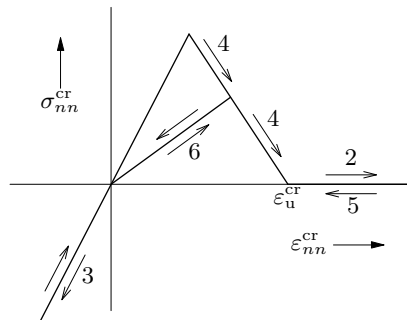


Figure 13.2: Crack status – linear tension softening

In the output, the crack status is indicated with a text string which depends on the *device*.

Crack status			1	2	3	4	5	6
<i>item</i>	<i>type</i>	<i>device</i>						
STATUS	CRACK							
	TABULA	NO CRACK YET		FULLY OPEN LOADING	CLOSED	PARTIALLY OPEN LOADING	FULLY OPEN UNLOADING	PARTIALLY OPEN UNLOADING
	FEMVIE		Fully_l	Closed	Partia_l	Fully_u	Partia_u	
	FXPLUS		O	C	P	O!	P!	

Crack display. In the iDIANA Results environment you may use the PRESENT SYMBOL command to display the crack status [Vol. iDIANA]. To get the crack pattern you must display the crack strains via the PRESENT DISC command.

13.4.4.3 Damage Index

The Japan Society of Civil Engineers Concrete Standard Specification for Concrete Structures 2012 (JSCE 2012) [49] allows to assess the performance of the structure by checking the Damage Index DI from a nonlinear finite element analysis. The damage index DI is defined in JSCE 2012 as:

$$DI = \sqrt{\left(\frac{\varepsilon_{xx} - \varepsilon_{yy}}{2}\right)^2 + \left(\frac{\gamma_{xy}}{2}\right)^2} \quad (13.13)$$

for membrane elements, and as

$$DI = \sqrt{\frac{(\varepsilon_{xx} - \varepsilon_{yy})^2 + (\varepsilon_{yy} - \varepsilon_{zz})^2 + (\varepsilon_{zz} - \varepsilon_{xx})^2}{6} + \frac{(\gamma_{xy}^2 + \gamma_{yz}^2 + \gamma_{zx}^2)}{4}} \quad (13.14)$$

for three-dimensional elements. For the multi-directional fixed crack model, Total Strain crack models, the Kotsovos concrete model, and the Maekawa–Fukuura concrete model the Damage Index can be calculated and output with the following command:

syntax

```
STATUS [ typew ] { locaw } { optiw }
          DAMIND   INTPNT
                   AVJSCE
```

DAMIND gives status output of the JSCE 2012 Damage Index as defined in Eqs. (13.13) and (13.14).

loca specifies the location for the Damage Index to be output [§ 3.6.1 p. 58]. The Damage Index can be output in the integration points or averaged over a region around the integration points according to the Japan Society of Civil Engineers (JSCE) [§ 3.6.2.4 p. 61].

[INTPNT]

opti are additional options [§ 3.6.1 p. 58].

JSCE 2012 Damage Index		
<i>item</i>	<i>type</i>	
STATUS	DAMIND	DI
		DI

13.4.4.4 Normalized Cumulative Energy

The Japan Society of Civil Engineers Concrete Standard Specification for Concrete Structures 2012 (JSCE 2012) [49] allows to assess the performance of the structure by checking the normalized cumulative energy from a nonlinear finite element analysis. For the Total Strain crack models, the Kotsovos concrete model, and the Maekawa–Fukuura concrete model the normalized cumulative energy can be calculated and output. The normalized cumulative energy W_n at step n is calculated incrementally by the following equation:

$$W_n = W_{n-1} + \frac{1}{2f} (\sigma_{ij}^{n-1} + \sigma_{ij}^n) (\varepsilon_{ij}^n - \varepsilon_{ij}^{n-1}) \quad (13.15)$$

Where W_{n-1} is the normalized cumulative energy in the previous step, f is the maximum stress that the material can resist under compressive loading and σ_{ij} and ε_{ij} are the stress tensor and strain tensor in the previous and current step, respectively. DIANA outputs both the normalized cumulative energy W_{n0} , which uses the user defined maximum compressive strength as maximum stress f and the normalized cumulative energy W_{n1} , which uses the maximum compressive strength that is being updated for lateral influences, as maximum stress f .

syntax

```
STATUS [ typew ] { formw } { locaw } { optiw }
      ENERGY   NRMCM   INTPNT
                        AVJSCE
```

ENERGY gives output of energy.

form specifies the energy formulation. NRMCM gives normalized cumulative energy. [NRMCM]

loca specifies the location for the status to be output [§ 3.6.1 p.58]. The normalized cumulative energy can be output in the integration points or averaged over a region around the integration points according to the Japan Society of Civil Engineers (JSCE) [§ 3.6.2.4 p.61]. [INTPNT]

opti are additional options [§ 3.6.1 p.58].

Normalized cumulative energy				
<i>item</i>	<i>type</i>	<i>form</i>		
STATUS	ENERGY	NRMCM	WNO	WN1
			W_{n0}	W_{n1}

13.4.4.5 Specific Kinetic Energy

For transient dynamic analyses the actual kinetic energy per volume can be calculated and output. The kinetic energy per volume is defined by the following equation:

$$W_k = \rho \cdot v \cdot v \quad (13.16)$$

where ρ is the density and v is the norm of the total velocity, respectively.

syntax

```
STATUS [ typew ] { formw } { locaw } { optiw }
      ENERGY   KINETI   INTPNT
```

ENERGY gives output of energy.

form specifies the energy formulation. KINETI gives specific kinetic energy.

loca specifies the location for the status to be output [§ 3.6.1 p.58]. For the specific kinetic energy only output at the integration points is possible. [INTPNT]

opti are additional options [§ 3.6.1 p.58].

Specific kinetic energy			
<i>item</i>	<i>type</i>	<i>form</i>	
STATUS	ENERGY	KINETI	WK
			W_k

13.4.4.6 Specific Stress Energy

For nonlinear analyses the actual stress energy per volume W_s can be calculated and output. The stress energy per volume W_s is defined by the following equation for continuum elements:

$$W_s = \frac{1}{2} (\sigma + \sigma_0) \varepsilon \quad (13.17)$$

Where σ is the effective stress tensor, σ_0 is the effective stress tensor at the reference state, and ε is the strain tensor relative to the reference state. The reference state is the start of an analysis or the beginning of a new construction stage, when the reference stresses and pore pressures are saved with the REFERE STRESS and/or REFERE POREPR commands [§ 13.3.8 p.244]. For interface elements the specific stress energy is not defined.

syntax

```
STATUS [ typew ] { formw } { locaw } { optiw }
        ENERGY   STRESS   INTPNT
```

ENERGY gives output of energy.

form specifies the energy formulation. STRESS gives specific stress energy.

[INTPNT] *loca* specifies the location for the status to be output [§ 3.6.1 p. 58]. For the specific stress energy only output at the integration points is possible.

opti are additional options [§ 3.6.1 p. 58].

Specific stress energy			
<i>item</i>	<i>type</i>	<i>form</i>	
STATUS	ENERGY	STRESS	WS
			W_s

13.4.4.7 Specific Pressure Energy

For nonlinear analyses the actual pressure energy per volume W_p can be calculated and output. The pressure energy per volume W_p is defined by the following equation for continuum elements:

$$W_p = \frac{1}{2} (p + p_0) \varepsilon_{\text{vol}} \quad (13.18)$$

Where p is the pore pressure, p_0 is the pore pressure at the reference state, and ε_{vol} is the volumetric strain relative to the reference state. The reference state is the start of an analysis or the beginning of a new construction stage, when the reference stresses and pore pressures are saved with the REFERE STRESS and/or REFERE POREPR commands [§ 13.3.8 p. 244]. For interface elements the specific pressure energy is not defined.

syntax

```
STATUS [ typew ] { formw } { locaw } { optiw }
        ENERGY   PRESSU   INTPNT
```

ENERGY gives output of energy.

form specifies the energy formulation. PRESSU gives specific pressure energy.

[INTPNT] *loca* specifies the location for the status to be output [§ 3.6.1 p. 58]. For the specific pressure energy only output at the integration points is possible.

opti are additional options [§ 3.6.1 p. 58].

Specific pressure energy			
<i>item</i>	<i>type</i>	<i>form</i>	
STATUS	ENERGY	PRESSU	WP
			W_p

13.4.4.8 Specific Gravitational Energy

For nonlinear analyses the field energy per volume related to a particle moving in a gravitational force field can be calculated and output. The gravitational energy per volume is defined by the following equation:

$$W_g = \rho g u \quad (13.19)$$

where ρ is the density, g is the gravity vector, and u is the total displacement, respectively.

syntax

```
STATUS [ typew ] { formw } { locaw } { optiw }
      ENERGY   GRAVIT   INTPNT
```

ENERGY gives output of energy.

form specifies the energy formulation. GRAVIT gives specific gravitational energy.

loca specifies the location for the status to be output [§ 3.6.1 p. 58]. For the specific [INTPNT] gravitational energy only output at the integration points is possible.

opti are additional options [§ 3.6.1 p. 58].

Specific gravitational energy			
<i>item</i>	<i>type</i>	<i>form</i>	
STATUS	ENERGY	GRAVIT	WG
			W_g

13.4.4.9 Liquefaction Status

Depending on the applied liquefaction model [Vol. *Material Library*], some status variables are optionally available for output at integration points.

syntax

```
STATUS [ typew ] { compw } { locaw } { optiw }
      TOWHAT   GAMMA   INTPNT
      BOWL     GSTAR
      USER
```

TOWHAT gives output of the liquefaction status parameter S_0 of the Towhata-Iai model.

BOWL gives output of the status variables of the Bowl model:

GAMMA gives the value of the Bowl internal variable Γ .

GSTAR gives the value of the Bowl internal variable G^* .

USER gives output of the internal parameters of the user-supplied liquefaction model [§ 13.4.4.14].

loca specifies the location for the status to be output [§ 3.6.1 p. 58]. For liquefaction [INTPNT] status only output at the integration points is possible.

opti are additional options [§ 3.6.1 p. 58].

13.4.4.10 Contact Status

syntax

```
STATUS [ typew ] { locaw } { optiw }
      CONTAC   NODES
```

STATUS CONTAC for contact status.

loca specifies the location for the status to be output [§ 3.6.1 p. 58]. For contact status [NODES] only output in the nodes is possible.

opti are additional options [§ 3.6.1 p. 58].

The following types of contact status can appear in the output:

1. The status is *no contact* when there is no contact.
2. The status is *stick* when there is friction between the contact elements.
3. The status is *slip* when the contact elements slip.

In the output, the contact status is indicated with a text string which depends on the *device*.

Contact status					
<i>item</i>	<i>type</i>	<i>device</i>	1	2	3
STATUS	CONTAC				
		TABULA	NO CONTACT	STICK	SLIP
		FEMVIE	No_conta	Stick	Slip
		FXPLUS		ST	SL

13.4.4.11 Kotsovos Concrete Model Status

The status output of the Kotsovos concrete model [Vol. *Material Library*] gives the elastic E_e , the plastic E_p , the total strain energy E_{tot} , and the compressive failure indicator in the integration points of the respective elements.

syntax

STATUS [*type_w*] { *loca_w* } { *opti_w* }

KOTSOV INTPNT

STATUS KOTSOV for the Kotsovos concrete model status output.

[INTPNT] *loca* specifies the location for the status to be output [§ 3.6.1 p. 58]. For the Kotsovos concrete model only status output in the integration points is possible.

opti are additional options [§ 3.6.1 p. 58].

Kostovos concrete model status					
<i>item</i>	<i>type</i>				
STATUS	KOTSOV	ENe	ENp	ENTot	CMPFAI
		E_e	E_p	E_{tot}	0 : no compressive failure
					1 : compressive failure

13.4.4.12 Engineering Masonry Model Status

The status output of the Engineering Masonry model [Vol. *Material Library*] indicates the type of failure that occurred in the integration points of the respective elements. The cumulative frictional shear strain, remaining cohesion, number of cracks, whether crushing has been reached, tensile strength of head joint, and actual Coulomb shear stress can be output, as well as an indicator whether the integration point has been linearized.

syntax

STATUS [*type_w*] { *loca_w* } { *opti_w* }

ENGMAS INTPNT

STATUS ENGMAS for the Engineering Masonry model status output:

CUMSHR gives the cumulative frictional shear strain γ_{cum} .

COHESI gives the remaining cohesion c in the integration point.

NCRACK gives the number of cracks in the integration point. In each integration point there can be maximum one crack in the bed joint, one crack in the head joint, and two cracks in the diagonal planes, thus 0, 1, 2, 3, or 4 cracks in total.

CRUSH gives the crushing status:

0 indicates that there is no crushing.

1 indicates that there is crushing.

ACTCRK gives the actual diagonal crack status:

0 indicates that there are no diagonal cracks active in this step.

1 indicates that the diagonal crack plane with angle $+\alpha$ is active in this step.

2 indicates that the diagonal crack plane with angle $-\alpha$ is active in this step.

FTALPH gives the tensile strength for the diagonal cracks $f_{t\alpha}$.

FTX gives the tensile strength for the head joint f_{tx} .

TAUMAX gives the Coulomb shear stress τ_{\max} .

OOPSHR gives the out-of-plane shear failure status:

0 indicates that there is no out-of-plane shear failure.

1 indicates that there is out-of-plane shear failure.

LINELM indicates whether the element has been linearized after failure:

0 indicates nonlinear behaviour.

1 indicates that the element has been linearized after failure.

loca specifies the location for the status to be output [§ 3.6.1 p. 58]. For the Engineering [INTPNT] Masonry model only status output in the integration points is possible.

opti are additional options [§ 3.6.1 p. 58].

Eng. Masonry <i>item type</i>											
STATUS ENGMAS	CUMSHR	COHESI	NCRACK	CRUSHD	ACTCRK	FTALPH	FTX	TAUMAX	OOPSHR	LINELM	
	γ_{cum}	c	0	0	0	$f_{t\alpha}$	f_{tx}	τ_{\max}	0	0	
			1	1	1				1	1	
			2		2						
			3								
			4								

13.4.4.13 Linearized Element Status

This status output indicates whether an integration point of the respective elements has been linearized after failure when using the Engineering Masonry model [Vol. *Material Library*]. Note that this output is equal to the last parameter of the Engineering Masonry model status output (STATUS ENGMAS) [§ 13.4.4.12].

syntax

STATUS [*type_w*] { *loca_w* } { *opti_w* }
LINELM INTPNT

STATUS LINELM for the linearized element status when using the Engineering Masonry model:

LINELM indicates whether the element has been linearized after failure:

0 indicates nonlinear behaviour.

1 indicates that the element has been linearized after failure.

loca specifies the location for the status to be output [§ 3.6.1 p. 58]. For the linearized [INTPNT] elements status only output in the integration points is possible.

opti are additional options [§ 3.6.1 p. 58].

Linearized element <i>item type</i>	
STATUS LINELM	LINELM
	0
	1

13.4.4.14 User-supplied Material Status

For the general user-supplied material model, and for the user-supplied liquefaction model, [Vol. *Material Library*], the stress and strain vector can be processed in the usual way: componentwise or in the principal directions. The same holds for the traction vectors of the user-supplied interface model. The vector of internal state variables for user-supplied models USRSTA can only be processed componentwise via the keyword STATUS USER.

syntax

```
STATUS [ typew ] { compw } { locaw } { optiw }
        USER      ITEM_  INTNPT
```

STATUS USER specifies the user-defined internal parameters as output item.

comp selects ‘components’ for output.

ITEM_ with _ is 01 to 99 for the components of the user-defined status variables USRSTA.

[INTNPT] *loca* specifies the location for the user status to be output [§ 3.6.1 p. 58]. For user status only output in the integration points is possible.

opti are additional options [§ 3.6.1 p. 58].

User-supplied material status			
<i>item</i>	<i>type</i>		
STATUS USER		ITEM01	ITEM02 ...
		<i>usrsta₁</i>	<i>usrsta₂</i> ...

13.4.5 Pore Pressures

In porous materials like soil, load is carried by effective stresses and the pressure of the fluid in the pores. In a nonlinear analysis the pressure in the pores (pore pressure) can be a loading (hydraulic head (HEAD), or pore pressure (PRESSU)), or the result of an undrained condition. The pore pressure load can be defined explicitly or be calculated with a preceding groundwater flow analysis. The undrained condition can be simulated by switching off the drained behaviour [§ 13.3.4 p. 235]. See also the UNDRAI input item [Vol. *Material Library*]. When a pore pressure loading is defined for elements with an undrained condition the pressure result is the sum of pore pressure loading and the excess pore pressure from the undrained condition. Pressure is a scalar type of result and therefore no component selection is allowed. You may select the pressure for output according to the following syntax:

syntax

```
PRESSU [ typew ] { locaw } { optiw }
        TOTAL    INTNPT
        MOHRCO   NODES
        HOEKBR   CENTER
        FRICTI
```

PRESSU specifies the calculated pore pressure result as output item.

[TOTAL] *type* specifies the type [§ 3.6.1 p. 57].

A specific type is available for hydrostatic pressure capacity of stress against the Mohr–Coulomb failure criterion:

MOHRCO gives the hydrostatic pressure capacity of stress against the Mohr–Coulomb failure criterion [§ 47.2.9.1 p. 582]. To determine the hydrostatic pressure capacity DIANA needs the initial cohesion *c* (COHESI) and initial friction angle ϕ (PHI) of the Mohr–Coulomb material model [Vol. *Material Library*].

A specific type is available for hydrostatic pressure capacity of stress against Mohr–Coulomb failure criterion:

HOEKBR gives the hydrostatic pressure capacity of stress against the Hoek–Brown failure criterion [§ 47.2.9.2 p. 582]. To determine the hydrostatic pressure capacity DIANA needs the unconfined compressive strength σ_{ci} of the (intact) rock sample (COMSTR), Hoek–Brown constant m_b (HOEKMB), and the Hoek–Brown constant s (HOEKS) material properties of the Hoek–Brown rock plasticity material model [Vol. *Material Library*].

A specific operation is available for hydrostatic pressure capacity of stress against the Coulomb friction failure criterion for structural interface elements:

FRICTI gives the hydrostatic pressure capacity of stress against Coulomb friction failure criterion [§ 47.2.9.3 p. 583]. To determine the hydrostatic pressure capacity DIANA needs the the initial cohesion c (COHESI) and initial friction angle ϕ (PHI) of the Coulomb friction interface material model [Vol. *Material Library*].

loca specifies the location for the pressure to be output [§ 3.6.1 p. 58].

[NODES]

opti are additional options [§ 3.6.1 p. 58].

Pore pressure		
<i>item</i>	<i>type</i>	
PRESSU TOTAL	P	
	<i>p</i>	

Hydrostatic pressure capacity [§ 47.2.9]		
<i>item</i>	<i>type</i>	
PRESSU MOHRCO		PPCAP
		χ
PRESSU FRICTI		PPCAP
		χ

13.4.6 Temperature, Concentration and Maturity

The temperature, concentration and maturity can be defined for each nodal point of an element as a function of time via input tables 'TEMPER', 'CONCEN', and 'MATURI' respectively [Vol. *Material Library*]. These data items can be selected for output according to the following syntax:

syntax

<i>item_w</i>	[<i>type_w</i>]	{ <i>loca_w</i> }	{ <i>opti_w</i> }
TEMPER	TOTAL	NODES	
CONCEN		INTPNT	
MATURI		CENTER	

item specifies the data item to be output for the elements.

TEMPER for the interpolated temperature at a certain time step.

CONCEN for the interpolated concentration at a certain time step.

MATURI for the interpolated maturity at a certain time step.

type specifies the type [§ 3.6.1 p. 57].

[TOTAL]

loca specifies the location for the temperature, concentration or maturity to be output [§ 3.6.1 p. 58].

[NODES]

opti are additional options [§ 3.6.1 p. 58].

<i>item</i>	<i>type</i>	
TEMPER	TOTAL	TE
		T
CONCEN	TOTAL	CO
		C
MATURI	TOTAL	MA
		M

13.4.7 External Forces, Reactions and Residuals

The nodal forces are energetically conjugate to the nodal degrees of freedom of the finite element system.⁴ In a nonlinear analysis, DIANA can output three types of nodal forces: the total *external forces*, the *support reactions*, and the *residual forces*.

syntax

FORCE [*type_w*] [*form_w*] [*oper_w*] { *comp_w* } { *opti_w* }

 EXTERN TRANSL LOCAL

 REACTI ROTATI GLOBAL

 RESIDU

FORCE specifies forces in the nodes as output item.

[REACTI] *type* specifies the type of the nodal forces.

 EXTERN for the externally applied forces \mathbf{f}_{ext} [Eq. (46.1) p. 551].

 REACTI for the reaction forces in all supported nodes. These are defined as the *out-of-balance* forces at the boundary conditions of the structure, i.e., the supported degrees of freedom: the difference between the externally applied forces \mathbf{f}_{ext} and the internal forces \mathbf{f}_{int} at the supports only [Eq. (46.5) p. 552].

 RESIDU for the residual forces \mathbf{g} , also called the *out-of-balance* forces. These are defined as the difference between the externally applied forces \mathbf{f}_{ext} and the internal resistance forces \mathbf{f}_{int} [Eq. (46.5) p. 552].

[TRANSL] *form* specifies the formulation [§ 3.6.1 p. 57]

[GLOBAL] *oper* specifies an operation to be performed on the forces and moments [§ 3.6.1 p. 57].

comp selects force or moment components for output. Default is all available components.

opti are additional options [§ 3.6.1 p. 58].

Total forces				<i>comp</i> ...		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	X	Y	Z
FORCE	EXTERN	TRANSL	LOCAL	FTx	FTy	FTz
				F_x	F_y	F_z
FORCE	EXTERN	TRANSL	GLOBAL	FTX	FTY	FTZ
				F_X	F_Y	F_Z
FORCE	EXTERN	ROTATI	LOCAL	MTx	MTy	MTz
				M_x	M_y	M_z
FORCE	EXTERN	ROTATI	GLOBAL	MTX	MTY	MTZ
				M_X	M_Y	M_Z

⁴If the types of the nodal degrees of freedom are translations and rotations then the nodal forces are translational forces (concentrated loads) and rotational forces (concentrated moments) respectively.

Support reactions				<i>comp</i> ...		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	X	Y	Z
FORCE	REACTI	TRANSL	LOCAL	FBx F_x^B	FBy F_y^B	FBz F_z^B
FORCE	REACTI	TRANSL	GLOBAL	FBX F_X^B	FBY F_Y^B	FBZ F_Z^B
FORCE	REACTI	ROTATI	LOCAL	MBx M_x^B	MBY M_y^B	MBz M_z^B
FORCE	REACTI	ROTATI	GLOBAL	MBX M_X^B	MBY M_Y^B	MBZ M_Z^B

Residual forces				<i>comp</i> ...		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	X	Y	Z
FORCE	RESIDU	TRANSL	LOCAL	FRx F_x^R	FRy F_y^R	FRz F_z^R
FORCE	RESIDU	TRANSL	GLOBAL	FRX F_X^R	FRY F_Y^R	FRZ F_Z^R
FORCE	RESIDU	ROTATI	LOCAL	MRx M_x^R	MRy M_y^R	MRz M_z^R
FORCE	RESIDU	ROTATI	GLOBAL	MRX M_X^R	MRY M_Y^R	MRZ M_Z^R

13.4.8 Nodal Element Forces

syntax

NODFOR [*type_w*] [*form_w*] [*oper_w*] { *comp_w* } { *opti_w* }

ELEMEN TRANSL LOCAL

REINFO ROTATI GLOBAL

TOTAL

DAMPIN

INERTI

NODFOR specifies internal nodal element forces and moments, nodal element damping forces and moments, and nodal element inertia forces and moments in the nodes as output item. This command gives the contributions of the element or reinforcement internal nodal forces and moments (or both), element damping forces and moments, or element inertia forces and moments to a certain node. A selection of elements which form a ‘section’ of the model gives the total internal forces and moments that act on that ‘section’.

type specifies the forces type.

[TOTAL]

ELEMEN for the internal nodal force and moment contribution of elements only.

REINFO for the internal nodal force and moment contribution of embedded reinforcements only.

TOTAL for the internal nodal force and moment contribution of both elements and embedded reinforcements.

DAMPIN for nodal element damping forces and moments.

INERTI for nodal element inertia forces and moments.

form specifies the formulation [§ 3.6.1 p. 57].

[TRANSL]

oper specifies an operation (transformation) to be performed on the forces and moments [§ 3.6.1 p. 57].

[GLOBAL]

comp selects force or moment components for output. Default is all available components.

[all]

opti are additional options [§ 3.6.1 p. 58].

Internal nodal total forces [§ 47.3.1]				<i>comp</i> ...		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	X	Y	Z
NODFOR	TOTAL	TRANSL	LOCAL	FNTx F_x^{nt}	FNTy F_y^{nt}	FNTz F_z^{nt}
NODFOR	TOTAL	TRANSL	GLOBAL	FNTX F_X^{nt}	FNTY F_Y^{nt}	FNTZ F_Z^{nt}
NODFOR	TOTAL	ROTATI	LOCAL	MNTx M_x^{nt}	MNTy M_y^{nt}	MNTz M_z^{nt}
NODFOR	TOTAL	ROTATI	GLOBAL	MNTX M_X^{nt}	MNTY M_Y^{nt}	MNTZ M_Z^{nt}

Internal nodal element forces [§ 47.3.1]				<i>comp</i> ...		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	X	Y	Z
NODFOR	ELEMEN	TRANSL	LOCAL	FNEx F_x^{ne}	FNEy F_y^{ne}	FNEz F_z^{ne}
NODFOR	ELEMEN	TRANSL	GLOBAL	FNEX F_X^{ne}	FNEY F_Y^{ne}	FNEZ F_Z^{ne}
NODFOR	ELEMEN	ROTATI	LOCAL	MNEx M_x^{ne}	MNEy M_y^{ne}	MNEz M_z^{ne}
NODFOR	ELEMEN	ROTATI	GLOBAL	MNEX M_X^{ne}	MNEY M_Y^{ne}	MNEZ M_Z^{ne}

Internal nodal reinforcement forces [§ 47.3.1]				<i>comp</i> ...		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	X	Y	Z
NODFOR	REINFO	TRANSL	LOCAL	FNRx F_x^{nr}	FNRy F_y^{nr}	FNRz F_z^{nr}
NODFOR	REINFO	TRANSL	GLOBAL	FNRX F_X^{nr}	FNRY F_Y^{nr}	FNRZ F_Z^{nr}
NODFOR	REINFO	ROTATI	LOCAL	MNRx M_x^{nr}	MNRy M_y^{nr}	MNRz M_z^{nr}
NODFOR	REINFO	ROTATI	GLOBAL	MNRX M_X^{nr}	MNRY M_Y^{nr}	MNRZ M_Z^{nr}

Nodal element damping forces				<i>comp</i> ...		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	X	Y	Z
NODFOR	DAMPIN	TRANSL	LOCAL	FDx F_x^{d}	FDy F_y^{d}	FDz F_z^{d}
NODFOR	DAMPIN	TRANSL	GLOBAL	FDX F_X^{d}	FDY F_Y^{d}	FDZ F_Z^{d}
NODFOR	DAMPIN	ROTATI	LOCAL	MDx M_x^{d}	MDy M_y^{d}	MDz M_z^{d}
NODFOR	DAMPIN	ROTATI	GLOBAL	MDX M_X^{d}	MDY M_Y^{d}	MDZ M_Z^{d}

Nodal element inertia forces				<i>comp</i> ...		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	X	Y	Z
NODFOR	INERTI	TRANSL	LOCAL	FMx F_x^{m}	FMY F_y^{m}	FMZ F_z^{m}
NODFOR	INERTI	TRANSL	GLOBAL	FMX F_X^{m}	FMY F_Y^{m}	FMZ F_Z^{m}
NODFOR	INERTI	ROTATI	LOCAL	MMx M_x^{m}	MMY M_y^{m}	MMZ M_z^{m}
NODFOR	INERTI	ROTATI	GLOBAL	MMX M_X^{m}	MMY M_Y^{m}	MMZ M_Z^{m}

13.4.9 Element Forces

syntax

```
ELMFOR [ typew ] [ formw ] [ operw ] { compw } { optiw }
      ELEMEN  TRANSL  GLOBAL
      REINFO  ROTATI
      TOTAL
      DAMPIN
      INERTI
```

ELMFOR specifies internal element forces and moments, element damping forces and moments, or element inertia forces and moments in the nodes of an element as output item. This command gives the contributions of the element or reinforcement internal forces and moments (or both), element damping forces and moments, or element inertia forces and moments to a certain node of an element.

type specifies the forces type. [TOTAL]

ELEMEN for internal force and moment contribution of elements only.

REINFO for internal force and moment contribution of embedded reinforcements only.

TOTAL for internal force and moment contribution of both elements and embedded reinforcements.

DAMPIN for element damping forces and moments.

INERTI for element inertia forces and moments.

form specifies the formulation [§ 3.6.1 p. 57]. [TRANSL]

oper specifies an operation (transformation) to be performed on the forces and moments [§ 3.6.1 p. 57]. [GLOBAL]

comp selects force or moment components for output. Default is all available components. [all]

opti are additional options [§ 3.6.1 p. 58].

Internal element forces				<i>comp</i> ...		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	X	Y	Z
ELMFOR TOTAL	TRANSL	GLOBAL		FETX F_X^{et}	FETY F_Y^{et}	FETZ F_Z^{et}
ELMFOR TOTAL	ROTATI	GLOBAL		METX M_X^{et}	METY M_Y^{et}	METZ M_Z^{et}
ELMFOR ELEMEN	TRANSL	GLOBAL		FEEX F_X^{ee}	FEEY F_Y^{ee}	FEEZ F_Z^{ee}
ELMFOR ELEMEN	ROTATI	GLOBAL		MEEX M_X^{ee}	MEEY M_Y^{ee}	MEEZ M_Z^{ee}
ELMFOR REINFO	TRANSL	GLOBAL		FERX F_X^{er}	FERY F_Y^{er}	FERZ F_Z^{er}
ELMFOR REINFO	ROTATI	GLOBAL		MERX M_X^{er}	MERY M_Y^{er}	MERZ M_Z^{er}

Element damping forces				<i>comp</i> ...		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	X	Y	Z
ELMFOR DAMPIN	TRANSL	GLOBAL		FEDX F_X^{ed}	FEDY F_Y^{ed}	FEDZ F_Z^{ed}
ELMFOR DAMPIN	ROTATI	GLOBAL		MEDX M_X^{ed}	MEDY M_Y^{ed}	MEDZ M_Z^{ed}

Element inertia forces				<i>comp</i> ...		
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	X	Y	Z
ELMFOR INERTI	TRANSL	GLOBAL		FEMX F_X^{em}	FEMY F_Y^{em}	FEMZ F_Z^{em}
ELMFOR INERTI	ROTATI	GLOBAL		MEMX M_X^{em}	MEMY M_Y^{em}	MEMZ M_Z^{em}

13.4.10 Model Parameters

For some models DIANA will calculate, or assume, default values for appropriate material or geometry parameters. In most cases you can overrule these values by specifying the parameters explicitly in the input data. Typical examples are the crack bandwidth in models for cracking, the area under the stress-strain curve which is equal to the crack energy G_f divided by the crack bandwidth h_{cr} for several tension softening curves of the Total Strain crack models, the preconsolidation stress in models for clay, and the reduction and safety factor for the D-min soil model [Vol. *Material Library*], or spatial functions attached to geometry or material parameters, e.g. depth dependent Young's moduli, Poisson's ratios, mass densities [Vol. *Material Library*], or thickness [Vol. *Element Library*]. Sometimes, these parameters may be useful in postprocessing. Via the **OUTPUT** block you can ask DIANA to output certain parameters, being the values that you have explicitly specified or else the calculated (default) values.

syntax

PARAME	[<i>type_w</i>]	[<i>loc_a_w</i>]	[<i>oper_w</i>]	{ <i>comp_w</i> }	{ <i>opti_w</i> }
	BANDWI	INTPNT	LOCAL		
	SEAREA	NODES	GLOBAL		
	PRECON				
	DMIN				
	YOUNG				
	POISON				
	DENSIT				
	YLDSTR				
	COHESI				
	PHI				
	PSI				
	LAMBDA				
	TENSTR				
	GF1				
	COMSTR				
	DSNX				
	DSSX				
	DSNY				
	DSSY				
	DSNZ				
	DSSZ				
	MAXSHR				
	GAMMAR				
	TAUFAC				
	PMLX				
	PMLY				
	PMLZ				
	THICK				
	ECCENT				

PARAME for output of a model parameter. Table 13.4 on the next page outlines the availability and applicability of the various model parameters output for each of the element families.

type specifies which parameter must be output.

BANDWI gives the crack bandwidth h_{cr} . This value is applied in various models for concrete cracking [Vol. *Material Library*]. If direct input (**CRACKB**) or the Rots' element based method (**CBSPEC ROTS**) is used, the crack bandwidth model parameter is available at every integration point regardless of presence of any crack at that integration point. If Govindjee's projection method (**CBSPEC GOVIND**) is being used, the crack bandwidth model parameter is available only

Table 13.4: AVAILABILITY OF MODEL PARAMETER OUTPUT

<i>item</i>	PARAMETER	truss	beam	pl. stress	pl. strain	axisymm.	plate bend.	fl. shell	cu. shell	solid	str. interface	contact	spring	p. mass	e. reinfo.	comp. line	comp. surf.
<i>type</i>	BANDWI	-	b	a	c	a	-	-	a	a	-	-	-	-	-	-	-
	SEAREA	-	b	a	c	a	-	-	a	a	-	-	-	-	-	-	-
	PRECON	-	-	-	c	d	-	-	a	-	-	-	-	-	-	-	-
	DMIN	-	-	-	c	d	-	-	a	-	-	-	-	-	-	-	-
	YOUNG	a	a	a	a	a	a	a	a	a	-	-	-	-	a	-	-
	POISON	a	a	a	a	a	a	a	a	a	-	-	-	-	e	-	-
	DENSIT	a	a	a	a	a	a	a	a	a	-	-	-	-	e	-	-
	YLDSTR	a	a	a	a	a	-	-	a	a	-	-	-	-	a	-	-
	COHESI	-	-	-	a	a	-	-	a	a	-	-	-	-	f	-	-
	PHI	-	-	-	a	a	-	-	a	a	-	-	-	-	f	-	-
	PSI	-	-	-	a	a	-	-	a	a	-	-	-	-	-	-	-
	LAMBDA	-	-	-	a	a	-	-	a	-	-	-	-	-	-	-	-
	TENSTR	a	g	a	a	a	-	-	a	a	-	-	-	-	-	-	-
	GF1	a	g	a	a	a	-	-	a	a	-	-	-	-	-	-	-
	COMSTR	a	g	a	a	a	-	-	a	a	-	-	-	-	-	-	-
	DSNX	-	-	-	-	-	-	-	-	a	-	-	-	-	-	-	-
	DSSX	-	-	-	-	-	-	-	-	a	-	-	-	-	e	-	-
	DSNY	-	-	-	-	-	-	-	-	a	-	-	-	-	e	-	-
	DSSY	-	-	-	-	-	-	-	-	a	-	-	-	-	-	-	-
	DSNZ	-	-	-	-	-	-	-	-	a	-	-	-	-	e	-	-
	DSSZ	-	-	-	-	-	-	-	-	a	-	-	-	-	-	-	-
	MAXSHR	-	-	-	-	-	-	-	-	-	-	-	-	-	f	-	-
	GAMMAR	-	-	-	a	a	-	-	-	a	-	-	-	-	-	-	-
	TAUFAC	-	-	-	a	a	-	-	-	a	-	-	-	-	-	-	-
	PMLX	-	-	-	a	-	-	-	-	a	-	-	-	-	-	-	-
	PMLY	-	-	-	a	-	-	-	-	a	-	-	-	-	-	-	-
	PMLZ	-	-	-	a	-	-	-	-	a	-	-	-	-	-	-	-
	THICK	-	-	a	h	i	a	a	a	-	j	-	-	-	-	-	-
	ECCENT	-	a	-	-	-	-	-	a	-	-	-	-	-	-	-	-

(a) All elements. (b) Only Class-III beams. (c) Only for regular plane strain. (d) Not for axisymmetric shell. (e) Only for bond-slip reinforcements and pile foundations. (f) Only for pile foundations. (g) Not for Class-I beams. (h) Only for infinite shells. (i) Only for axisymmetric shells. (j) Only for line interfaces to shells and two-dimensional line interfaces between membranes. (-) Not available or not suitable.

at the crack integration points, because Govindjee's method requires the crack orientation to compute the crack bandwidth. In case of multiple cracks in an integration point, the average crack bandwidth is output. Note that the crack bandwidth h model parameter is only available in integration points.

SEAREA gives the area under the stress-strain curve, which is equal to the crack energy G_f divided by the crack bandwidth h_{cr} . This output item is only available for Total Strain crack models with the linear tension softening curve based on crack energy (LINEAR), softening according to Hordijk et al. (HORDYK), exponential tension softening (EXPONE), softening according to Japan Society of Civil Engineers (JSCESO), CEB-FIP Model Code 1990 tension softening (MC1990), fib Model Code for Concrete Structures 2010 tension softening (MC2010), and the model for tensile failure of fiber reinforced concrete as defined by the *fédération internationale du béton*/International Federation for Structural Concrete (fib) working groups (FRCCON) [Vol. *Material Library*].

PRECON gives the preconsolidation stress p'_c . This value is applied in various models for clay and can be input as data item PRECON [Vol. *Material Library*].

DMIN gives the reduction factor R and the safety factor F_S for the D-min soil model [Vol. *Material Library*].

YOUNG gives the Young's modulus E .

POISON gives the Poisson's ratio ν .

DENSIT gives the mass density ρ . For all regular structural elements ρ is the mass density per *unit volume*. For regular structural interface elements ρ is the

mass density *per unit area*. For fluid–structure interface elements ρ is the mass density of the fluid ρ_f .

YLDSTR gives the yield stress σ_y for Von Mises and Tresca plasticity.

COHESI gives the cohesion c for Mohr–Coulomb and Drucker–Prager plasticity, and for Coulomb friction and nonlinear elastic friction interface materials.

PHI gives the friction angle ϕ in the used units (radians or degrees) for Mohr–Coulomb, Drucker–Prager, and Egg Cam-clay plasticity, and for Coulomb friction and nonlinear elastic friction interface materials.

PSI gives the dilatancy angle ψ in the used units (radians or degrees) for Mohr–Coulomb and Drucker–Prager plasticity, and for Coulomb friction interface materials.

LAMBDA gives the hardening parameter λ for Egg Cam-clay plasticity.

TENSTR gives the tensile strength f_t for the Total Strain crack model and the Maekawa–Fukuura concrete model.

GF1 gives the Mode-I fracture energy G_f^I for the Total Strain crack model and the Maekawa–Fukuura concrete model.

COMSTR gives the compressive strength f_c for the Total Strain crack model and the Maekawa–Fukuura concrete model.

DSNX gives the linear stiffness modulus D_{11} , which sets the relation between the normal traction t_{nx} in the element x direction and the normal relative displacement Δu_{nx} in the element x direction, for interface materials.

DSSX gives the linear stiffness modulus D_{11} , which sets the relation between the shear traction t_{sx} in the element x direction and the shear relative displacement Δu_{sx} in the element x direction, for interface models.

DSNY gives the linear stiffness modulus D_{22} , which sets the relation between the normal traction t_{ny} in the element y direction and the normal relative displacement Δu_{ny} in the element y direction, for interface materials.

DSSY gives the linear stiffness modulus D_{22} , which sets the relation between the shear traction t_{sy} in the element y direction and the shear relative displacement Δu_{sy} in the element y direction, for interface models.

DSNZ gives the linear stiffness modulus D_{33} , which sets the relation between the normal traction t_{nz} in the element z direction and the normal relative displacement Δu_{nz} in the element z direction, for interface materials.

DSSZ gives the linear stiffness modulus D_{33} , which sets the relation between the shear traction t_{sz} in the element z direction and the shear relative displacement Δu_{sz} in the element z direction, for interface models.

MAXSHR gives the absolute maximum value of the force per length that the pile shaft–soil interface can transfer f_{\max} for pile foundations.

GAMMAR *gammar* is the characteristic shear strain γ_r for the Hardin–Drnevich model or for the Ramberg–Osgood model.

TAUFAC *tf* is an optional factor to the shear–stress values in the multilinear diagram of shear strains γ and the corresponding shear stresses τ (GAMTAU) or to the ratio of secant and initial shear stiffness values in the multilinear diagram of shear strains γ and the corresponding ratios of secant shear stiffness and initial stiffness G/G_0 (GAMBET).

PMLX gives the perfectly matched layer (PML) parameter in X direction.

PMLY gives the perfectly matched layer (PML) parameter in Y direction.

PMLZ gives the perfectly matched layer (PML) parameter in Z direction.

THICK gives the thickness t in the respective element nodes.

ECCENT gives the eccentricity e in the respective element nodes.

loca specifies the location of the parameter to be output. For crack bandwidth, the area under the stress-strain curve, and the preconsolidation stress this is always, and by default, the element integration point. For the D-min soil material parameters the location of the output can be either nodes or integration points. By default the model parameters for the D-min model are in the nodes. For the other material parameters the location of the output can be either nodes or integration points. By default the model parameters for these material parameters are output in the element integration points. For thickness and eccentricity, the output is always in the nodes of the elements.

oper is an operation on the result item to be performed prior to its output:

LOCAL for transformation to local *xyz* directions,

GLOBAL for transformation to global *XYZ* directions,

comp selects model parameter components for output. Default is all available components. [all]

opti are additional options [§ 3.6.1 p. 58].

Model parameters			<i>comp</i> ...		
<i>item</i>	<i>type</i>	<i>oper</i>	X	Y	Z
PARAME BANDWI			Hcr		
			h_{cr}		
PARAME SEAREA			Gf/hcr		
			G_f/h_{cr}		
PARAME PRECON			Pc		
			p'_c		
PARAME DMIN			FR	FS	
			R	F_S	
PARAME YOUNG			YOUNG		
			E		
PARAME POISON			POISON		
			ν		
PARAME DENSIT			DENSIT		
			ρ		
PARAME YLDSTR			YLDSTR		
			σ_y		
PARAME COHESI			COHESI		
			c		
PARAME PHI			PHI		
			ϕ		
PARAME PSI			PSI		
			ψ		
PARAME LAMBDA			LAMBDA		
			λ		
PARAME TENSTR			TENSTR		
			f_t		
PARAME GF1			GF1		
			G_f^I		
PARAME COMSTR			COMSTR		
			f_c		
PARAME DSNX			DSNX		
			D_{11}		
PARAME DSSX			DSSX		
			D_{11}		
PARAME DSNY			DSNY		
			D_{22}		
PARAME DSSY			DSSY		
			D_{22}		
PARAME DSNZ			DSNZ		
			D_{33}		
PARAME DSSZ			DSSZ		
			D_{33}		
PARAME MAXSHR			MAXSHR		
			f_{max}		
PARAME THICK			THICK		
			t		
PARAME ECCENT LOCAL			ECCENx	ECCENy	ECCENZ
			e_x	e_y	e_z
PARAME ECCENT GLOBAL			ECCENX	ECCENY	ECCENZ
			eX	eY	eZ

13.4.11 Fracture Mechanics Parameters

For crack tip elements [Vol. *Element Library*] DIANA can calculate and output the parameters for Linear Elastic Fracture Mechanics analysis (LEFM), based on displacements, stresses and strains resulting from a nonlinear analysis. See Chapter 57 for background theory.

syntax

FRACTU

FRACTU specifies the LEFM parameters as output item. No further options apply for this output. The LEFM parameters are always output as two ‘components’.

LEFM parameters		
<i>item</i>		
FRACTU	K	G
	K_I	G_I

K_I is the Mode-I stress intensity factor [Eq. (57.2) p. 639]. The stress intensity factor can only be calculated if the material parameters and the thickness (for plane stress elements) of all the elements in the crack front are identical.

G_I is the Mode-I energy release rate [Eq. (57.8) p. 640].

13.4.12 Dynamic Pressures

For fluid–structure interface elements [Vol. *Element Library*] DIANA can calculate and output the dynamic pressures. Positive dynamic pressures indicate additional pressure on the structure, while negative dynamic pressures, i.e. suction pressures, reduce the pressure on the structure. See [§ 48.5 p. 595] for background theory.

syntax

FSPRES	[<i>type_w</i>]	[<i>loca_w</i>]	{ <i>opti_w</i> }
	TOTAL	NODES	

FSPRES specifies the dynamic pressures of the fluid–structure interface elements as output item.

type specifies the dynamic pressure type. [TOTAL]

TOTAL for the total dynamic pressures of the fluid–structure interface elements.

loca specifies the location for the dynamic pressures to be output [§ 3.6.1 p. 58]. [NODES]

opti are additional options [§ 3.6.1 p. 58].

Dynamic pressures			
<i>item</i>	<i>type</i>	<i>loca</i>	
FSPRES	TOTAL	NODES	PRfs
			p_{fs}

Chapter 14

Strength Reduction Analysis

In DIANA strength reduction method is implemented as a separate module named REDUCT. The main output of this type of analysis is the factor of safety. See Chapter 51 on page 611 for background theory.

Strength reduction method is typically used for the assessment of slope stability where dominantly a Mohr-Coulomb or similar material model is used. Presently in DIANA only the Mohr-Coulomb and Drucker-Prager models, as well as the Coulomb friction for interface elements are considered. Future extension may include the Modified Mohr-Coulomb and Hoek-Brown models.

A new material parameter **STRRED** is introduced to be used in table of the input file along with the other material parameters of the Mohr-Coulomb or Drucker-Prager model, or Coulomb friction model for interface elements. This will allow the users to specify only a certain part of the model to be subjected to strength reduction instead of the whole model. For syntax description see Volume *Material Library*.

The strength reduction analysis ***REDUCT** must be preceded by a standard nonlinear static analysis done with application ***NONLIN** [§ 13 p. 211]. Therefore, it does not need to invoke the modules **FILOS** and **INPUT**. The command sequence for Module REDUCT is as follows.

syntax

```
*REDUCT  
[ EXECUT ... ]  
[ OUTPUT ... ] ...  
*END
```

EXECUT executes steps [§ 14.1 p. 281].

OUTPUT selects analysis results for output [§ 13.4 p. 246]. Syntax wise output block of ***REDUCT** is the same as for ***NONLIN**. Output items are written to the output device for specified step numbers, but unlike ***NONLIN** the step numbers are interpreted as the iteration number in the loop for determination of factor of safety. It worths mentioning here that the factor of safety is the main output item of this kind of strength reduction analysis. This information is written in the standard DIANA output file *file.out*.

Unlike ***NONLIN**, model evaluation and nonlinear analysis type specification are not required since ***REDUCT** continues from the state of the database left behind by the preceding nonlinear analysis.

14.1 Step Execution

With the **EXECUT** commands you ask DIANA to execute steps. Unlike ***NONLIN**, in case of ***REDUCT** a command file may contain only one **EXECUT** block. Apart from a few exceptions most of the commands are same as ***NONLIN**. When necessary details of some commands

are given in the referred sections. The commands that are the same as *NONLIN are referred to appropriate sections of *NONLIN.

syntax

```

BEGIN EXECUT
[ OFF ]
[ BEGIN SAFETY
  [ INIFS=inifsr ]
  [ INCFS=incfsr ]
  [ TOLFS=tolfsr ]
END SAFETY ]
[ BEGIN FINDEQ
  [ STEPS ... ]
  [ ITERAT ... ]
  [ SOLVE ... ]
  [ STOP ... ]
  [ LOGGIN ... ]
END FINDEQ ]
END EXECUT

```

SAFETY specifies settings for the factor of safety:

<p>[INIFS=1.0] (0 < INIFS ≤ 20) [INCFS=0.1] (0 < INCFS ≤ 20) [TOLFS=0.01] (0 < TOLFS < 1)</p>	<p>INIFS specifies the initial factor of safety. INCFS specifies the increment of the factor of safety. TOLFS specifies the tolerance for the factor of safety.</p>
---	---

FINDEQ specifies settings for the equilibrium process:

STEPS specifies the stepping process to be applied [§ 14.1.1 p. 283].
 ITERAT specifies the equilibrium iteration process to be applied [§ 14.1.2 p. 284].
 SOLVE customizes the settings for the solution method [Ch. 30 p. 421].
 STOP specifies a stop criterion for step execution [§ 14.1.3 p. 285].
 LOGGIN customizes the logging information that will be output during execution of the steps [§ 13.3.7 p. 243].

If no EXECUT block is specified, then it is filled out with the following default settings.

file.dcf

```

*REDUCT
BEGIN EXECUT
  BEGIN SAFETY
    INIFS=1.0
    INCFS=0.1
    TOLFS=0.01
  END SAFETY
  BEGIN FINDEQ
    STEPS EXPLIC SIZES 1.0
    BEGIN ITERAT
      MAXITE=50
      METHOD SECANT BFGS
      BEGIN CONVER
        FORCE NEWREF TOLCON=1.E-3 TOLABT=1.E+4
      END CONVER
    END ITERAT
  END FINDEQ
END EXECUT
*END

```

14.1.1 Steps

The **STEPS** commands specify the execution of load steps. However unlike ***NONLIN** the user does not need to specify any load case. The load vector is constructed as an internal force vector resulting from the stress difference between the stress at the end of the initial nonlinear analysis and the reduced stress due to the reduction of strength parameters cohesion (c) and friction angle (ϕ), for further details see [Ch. 51]. In order to remove the out-of-balance force incrementally DIANA first temporarily adds the internal force due to the reduced stress σ (resulting from the reduction of the strength parameters) to the initial external load. Next, DIANA uses the steps that you specify to remove this addition. The external loading in step *nstep* reads

$$\mathbf{f}_{\text{ext}} = \mathbf{f}_{\text{int}} + \sum_{i=1}^{nstep} \mathbf{size}_i \times (\mathbf{f}_{\text{ext}}^0 - \mathbf{f}_{\text{int}}) \quad (14.1)$$

where $\mathbf{f}_{\text{ext}}^0 = \int \mathbf{B}^T \sigma_0$ and $\mathbf{f}_{\text{int}} = \int \mathbf{B}^T \sigma$. The external force $\mathbf{f}_{\text{ext}}^0$ and the stress σ_0 result from the preceding nonlinear analysis. The sum of explicitly specified step sizes must be equal to 1. The initial external load is defined as the nonlinear load set *loset* [§ 13.3.1 p. 220]. Note that DIANA internally creates an additional load set for the load vector equal to $\mathbf{f}_{\text{ext}}^0 - \mathbf{f}_{\text{int}}$. This load set is assigned a load set number which is one higher than the highest user-defined load set number. This additional load set number will appear in logging and output. This setup of external force vector is similar to incremental adaptation of initial conditions [§ 13.3.1.2 p. 223].

If the total load factor exceeds 1, DIANA will automatically stop further stepping, because the total load factor has to be 1. On the other hand, if the total load factor is less than 1, DIANA will automatically do additional steps to reach the load factor 1. Therefore, explicit step size of 1.0 is used by default.

syntax

BEGIN STEPS

*method*_w ...

EXPLIC

ITERAT

AUTOMA

END STEPS

method indicates how the step sizes are to be chosen. Via *method* you may specify the [EXPLIC]
step sizes explicitly, or you may let DIANA determine them automatically. [STEPS]

EXPLIC explicitly specified step sizes [§ 14.1.1.1 p. 283].

ITERAT iteration based automatic step size control [§ 14.1.1.2 p. 284].

AUTOMA cutback based automatic step size control [§ 14.1.1.3 p. 284].

14.1.1.1 Explicitly Specified Step Sizes

Via the **EXPLIC** commands you may specify load step sizes explicitly.

syntax

BEGIN EXPLIC

[OFF]

SIZES *sizes*_{r...}

END EXPLIC

SIZES *sizes* are explicitly specified load step sizes. The number of values specifies the number of steps to be executed.

14.1.1.2 Iteration Based Adaptive Loading

The ITERAT commands cause automatic adaptive load increments, based on the number of iterations.

syntax

```

BEGIN ITERAT
[ OFF ]
[ INISIZ= $is_r$  ]
[ NITERA= $nitr_n$  ]
[ MAXSIZ= $maxs_r$  ]
[ MINSIZ= $mins_r$  ]
[ NSTEPS= $ns_n$  ]
[ GAMMA= $gam_r$  ]
END ITERAT

```

[$is=1$.] INISIZ= is is the initial size for the first step.

[$ni=6$] NITERA= ni is the number of iterations (should be considered as optimal).

[$maxs=10^6$] MAXSIZ= $maxs$ is the upper limit of the step size.

[$mins=10^{-3}$] MINSIZ= $mins$ is the lower limit of the step size.

DIANA applies the specified limits in the zero iteration of each step to determine a first estimation of the step size.

[$ns=1$] NSTEPS= ns is the number of steps [§ 46.1.5.3 p. 563].

[$\gamma = 0.5$] GAMMA= gam specifies the exponent γ . For further information see [§ 13.3.2.2 p. 226].

14.1.1.3 Cutback Based Adaptive Loading

The AUTOMA command causes cutback based automatic adaptive load increments. See § 46.1.5.4 on page 565 for background theory.

syntax

```

BEGIN AUTOMA
[ OFF ]
[ SIZE= $size_r$  ]
[ MINSIZ= $mins_r$  ]
[ MAXSIZ= $maxs_r$  ]
[ CUTBCK= $cutb_r$  ]
[ MAXSTP= $maxstp_n$  ]
END AUTOMA

```

[$size=1$] SIZE= $size$ is the total load increment.

[$mins=0.1$] MINSIZ= $mins$ is the lower limit of the step size, relative to $size$.

[$maxs=1$] MAXSIZ= $maxs$ is the upper limit of the step size, relative to $size$.

[$cutb=0.5$] CUTBCK= $cutb$ is the factor by which the step size is scaled down in case of non-convergence.

MAXSTP= $maxstp$ is a limit to the number of steps that are taken.

14.1.2 Equilibrium Iteration

The ITERAT commands define the equilibrium iteration process to be used for steps. See § 46.1.1 on page 552 for background theory.

syntax

```

BEGIN ITERAT
[ MAXITE= $mi_n$  ]
[ METHOD ... ]
[ LINESE ... ]
[ CONVER ... ]
END ITERAT

```

MAXITE= mi is the maximum number of iterations for each time or load step. [$mi=50$]

METHOD specifies the iteration method [§ 13.3.5.1 p. 236]. In contrary to *NONLIN by default a Quasi-Newton (Secant) iteration method will be used for strength reduction analysis. [SECANT]

LINESE invokes a Line Search algorithm to scale the incremental displacements [§ 13.3.5.2 p. 237].

CONVER specifies the convergence criterion for the iteration process [§ 13.3.5.3].

If you do not give any ITERAT commands, then the default is a Quasi-Newton (Secant) method with a maximum of fifty iterations. As convergence criteria, the norms of the force with tolerance 0.001 will be applied.

14.1.3 Stop Criteria for Step Execution

If you specify a stop criterion via a STOP command, then DIANA will stop the execution of steps if the stop criterion is met. In the context of the strength reduction analysis, it means that structural failure has occurred. You may specify a stop criterion based on resulting strain or stress.

syntax

```

BEGIN STOP
[ RESULT ... ]
END STOP

```

RESULT specifies stop criteria based on results [§ 13.3.6.2].

Chapter 15

Engineering Liquefaction Analysis

In DIANA the engineering liquefaction method is implemented as a separate module named ELIQUE. For more information see the corresponding background theory in Chapter 52 on page 613.

A new material parameter SHRRED is introduced to be used in table 'MATERI' of the input file. This will allow the user to specify only a certain part of the model to be subjected to reduction of shear modulus, which is necessary for an engineering liquefaction analysis, instead of the whole model. For the description of the material parameter SHRRED see Volume *Material Library*.

An engineering liquefaction analysis *ELIQUE must be preceded by a standard non-linear static analysis done with application *NONLIN [Ch. 13 p. 211]. Therefore, it does not need to invoke the modules FILOS and INPUT. The command sequence for Module ELIQUE is as follows.

syntax

```
*ELIQUE
[STEPS ...]
[ITERAT ...]
[SOLVE ...]
[LOGGIN ...]
[OUTPUT ...] ...
*END
```

STEPS specifies the stepping process to be applied [§ 15.1 p. 288].

ITERAT specifies the equilibrium iteration process to be applied [§ 13.3.5 p. 235].

SOLVE customizes the settings for the solution method [Ch. 30 p. 421].

LOGGIN customizes the logging information that will be output during execution of the steps [§ 13.3.7 p. 243].

OUTPUT selects analysis results for output [§ 13.4 p. 246]. Syntax wise the output block of *ELIQUE is the same as for *NONLIN.

The command sequence is quite similar to *NONLIN. But some differences are also noticeable. Unlike *NONLIN, model evaluation and nonlinear analysis type specification are not required since *ELIQUE continues from the state of the database left behind by the preceding nonlinear analysis. Therefore, the settings that were active in the preceding nonlinear analysis remain active in the engineering liquefaction analysis. For instance, if geometric nonlinearity was active in the preceding *NONLIN analysis, it is also accounted for in the *ELIQUE analysis. With STEPS commands the user asks DIANA to execute

steps. Unlike ***NONLIN**, no **EXECUT** block is defined. Apart from such exceptions most of the commands are same as ***NONLIN**. When necessary, details of some commands are given in the referred sections. The commands that are the same as ***NONLIN** are referred to the appropriate sections of ***NONLIN**. If no command is specified after ***ELIQUE**, then it is filled out with the following default settings.

file.dcf

```

*ELIQUE
  STEPS EXPLIC SIZES 1.0
  BEGIN ITERAT
    MAXITE=10
    METHOD NEWTON REGULA
    BEGIN CONVER
      FORCE NEWREF TOLCON=1.E-2 TOLABT=1.E+4
      DISPLA NEWREF TOLCON=1.E-2 TOLABT=1.E+4
    END CONVER
  END ITERAT
*END

```

15.1 Steps

The **STEPS** commands specify the execution of load steps. However unlike ***NONLIN**, the user does not need to specify any load case. The load vector is constructed as an internal force vector, resulting from the stress difference between the stress at the end of the initial nonlinear analysis, and the reduced stress due to the reduction of shear modulus. For further details see the background theory in Chapter 52 on page 613. In order to remove the out-of-balance force incrementally, DIANA first temporarily adds the internal force due to the reduced stress σ (resulting from the reduction of the shear modulus) to the initial external load. Next, DIANA uses the steps that you specify to remove this addition. The external loading in step *nstep* reads

$$\mathbf{f}_{\text{ext}} = \mathbf{f}_{\text{int}} + \sum_{i=1}^{nstep} \text{size}_i \times (\mathbf{f}_{\text{ext}}^0 - \mathbf{f}_{\text{int}}) \quad (15.1)$$

where $\mathbf{f}_{\text{ext}}^0 = \int \mathbf{B}^T \sigma_0$ and $\mathbf{f}_{\text{int}} = \int \mathbf{B}^T \sigma$. The external force $\mathbf{f}_{\text{ext}}^0$ and the stress σ_0 result from the preceding nonlinear analysis. The sum of explicitly specified step sizes must be equal to 1. The initial external load is defined as the nonlinear load set *loset* [§ 13.3.1 p. 220]. Note that DIANA internally creates an additional load set for the load vector equal to $\mathbf{f}_{\text{ext}}^0 - \mathbf{f}_{\text{int}}$. This load set is assigned to a load set number which is one higher than the highest user-defined load set number. This additional load set number will appear in logging and output. This setup of external force vector is similar to incremental adaptation of initial conditions [§ 13.3.1.2 p. 223].

If the total load factor exceeds 1, DIANA will automatically stop further stepping, because the total load factor has to be 1. On the other hand, if the total load factor is less than 1, DIANA will automatically do additional steps to reach the load factor 1. Therefore, explicit step size of 1.0 is used by default.

syntax

```

BEGIN STEPS
[ OFF ]
  methodw ...
  EXPLIC
  ITERAT
  ENERGY
  AUTOMA
END STEPS

```

method indicates how the step sizes are to be chosen. Via *method* you may specify the step sizes explicitly, or you may let DIANA determine them automatically.

[EXPLICIT]

[STEPS]

EXPLICIT explicitly specified step sizes [§ 13.3.2.1 p. 225].

ITERAT iteration based automatic step size control [§ 15.1.1 p. 289].

ENERGY iteration based automatic step size control [§ 15.1.1 p. 289].

AUTOMA cutback based automatic step size control [§ 13.3.3.4 p. 233].

15.1.1 Iteration Based Adaptive Loading

The ITERAT commands cause automatic adaptive load increments, based on the number of iterations.

syntax

BEGIN ITERAT

[OFF]

[INISIZ=*is*_{*r*}]

[NITERA=*ni*_{*t*}*r*_{*n*}]

[MAXSIZ=*maxs*_{*r*}]

[MINSIZ=*mins*_{*r*}]

[NSTEPS=*ns*_{*n*}]

[GAMMA=*gam*_{*r*}]

END ITERAT

INISIZ=*is* is the initial size for the first step.

[*is*=1.]

NITERA=*ni* is the number of iterations (should be considered as optimal).

[*ni*=6]

MAXSIZ=*maxs* is the upper limit of the step size.

[*maxs*=10⁶]

MINSIZ=*mins* is the lower limit of the step size.

[*mins*=10⁻³]

DIANA applies the specified limits in the zero iteration of each step to determine a first estimation of the step size.

NSTEPS=*ns* is the number of steps [§ 46.1.5.3 p. 563].

[*ns*=1]

GAMMA=*gam* specifies the exponent γ . For further information see [§ 13.3.2.2 p. 226].

[γ = 0.5]

15.1.2 Energy Based Adaptive Loading

Due to the ENERGY commands DIANA will apply automatic adaptive load increments, based on energy.

syntax

BEGIN ENERGY

[OFF]

[INISIZ=*is*_{*r*}]

[MAXSIZ=*maxs*_{*r*}]

[MINSIZ=*mins*_{*r*}]

[NSTEPS=*ns*_{*n*}]

[ARCLN ...]

END ENERGY

INISIZ=*is* is the initial size for the first step.

[*is*=1.]

MAXSIZ=*maxs* is the upper limit of the step size.

[*maxs*=10⁶]

MINSIZ=*mins* is the lower limit of the step size.

[*mins*=10⁻³]

DIANA applies the specified limits in the zero-iteration of each step to determine a first estimation of the step size.

[*ns*=1] NSTEPS=*ns* is the number of steps [§ 46.1.5.3 p. 563].

ARCLEN applies Arc-length control [§ 13.3.2.5 p. 227].

Chapter 16

Engineering Creep Analysis

In DIANA the engineering creep method is implemented as a separate module named ECREEP. For more information see the corresponding background theory in Chapter 53 on page 615.

A couple of new material parameters, YOUNLT and POISLT which stand for long term Young's modulus and Poison's ratio respectively, are introduced to be used in table 'MATERI' of the input file. This will allow the users to specify only a certain part of the model to be subjected to the effect of long term Young's modulus and Poison's ratio, which is necessary for engineering creep analysis instead of the whole model. For the description of these material parameters see Volume *Material Library*.

An engineering creep analysis *ECEEP must be preceded by a standard nonlinear static analysis done with application *NONLIN [Ch. 13 p. 211]. Therefore, it does not need to invoke the modules FILOS and INPUT. The settings that were active in the preceding nonlinear analysis remain active in the engineering creep analysis. For instance, if geometric nonlinearity was active in the preceding *NONLIN analysis, it is also accounted for in the *ECEEP analysis. The command sequence for Module ECREEP is as follows.

syntax

```
*ECEEP
[STEPS ...]
[ITERAT ...]
[SOLVE ...]
[LOGGIN ...]
[OUTPUT ...] ...
*END
```

STEPS specifies the stepping process to be applied [§ 15.1 p. 288].

ITERAT specifies the equilibrium iteration process to be applied [§ 13.3.5 p. 235].

SOLVE customizes the settings for the solution method [Ch. 30 p. 421].

LOGGIN customizes the logging information that will be output during execution of the steps [§ 13.3.7 p. 243].

OUTPUT selects analysis results for output [§ 13.4 p. 246]. Syntax wise the output block of *ECEEP is the same as for *NONLIN.

The command sequence is exactly the same as for an engineering liquefaction analysis, see Chapter 15 on page 287. If no command is specified after *ECEEP, then it is filled out with the following default settings.

file.dcf

```
*ECREEP
STEPS EXPLIC SIZES 1.0
BEGIN ITERAT
  MAXITE=10
  METHOD NEWTON REGULA
  BEGIN CONVER
    FORCE NEWREF TOLCON=1.E-2 TOLABT=1.E+4
    DISPLA NEWREF TOLCON=1.E-2 TOLABT=1.E+4
  END CONVER
END ITERAT
*END
```

Part V

Stability Analysis

Chapter 17

Introduction to Stability Analysis

In this chapter we will discuss how to perform a stability analysis. The user actions needed depend on the type of analysis. Currently one type of analysis is available: *Euler stability* analysis.

The Euler stability analysis may be followed by a *perturbation analysis* to investigate the postbuckling behaviour. For this perturbation a small number of buckling modes is taken into account. This approach results in a potential energy function which is defined in terms of amplitudes of the selected modes. The postbuckling displacement field is solved by applying a *continuation analysis* using a stepwise generalized Newton–Raphson scheme.

17.1 Euler Stability Analysis

Euler stability analysis gives information about ‘linearized stability’ of a structure. It tells whether solutions from linear elastic analysis are stable or whether small disturbances to those solutions exist, requiring no extra external energy. This type of stability analysis does not allow for any physical nonlinearities, geometrical nonlinear (i.e., large deformation) effects are only partly taken into account. However, often it is a relatively simple and effective method to get a fair impression of a structure’s buckling modes. Background theory is given in § 54.1 on page 617. In order to perform an Euler stability analysis with DIANA you must take the following actions:

1. Prepare input data like for a linear analysis [Ch. 4 p. 71]. If appropriate, add initial imperfections [§ 17.4].
2. Perform a stability analysis. An Euler stability analysis is accomplished by Module EULER as described in Chapter 18.
3. Output the buckling modes (optional) [§ 18.2.3 p. 303].

17.2 Perturbation Analysis

Perturbation analysis gives information about the initial postbuckling behaviour of the structure. Postbuckling deformation fields can be calculated and effects resulting from interaction of selected buckling modes can be analysed. This type of stability analysis does not allow for any physical nonlinearities, geometrical nonlinear effects (large deformation) are only taken into account partly. See § 18.3.1 on page 305 for commands and § 54.3 on page 621 for background theory. In order to perform a perturbation analysis with DIANA you must take the following actions:

1. Perform a complete Euler stability analysis as described in the previous section.
2. Select the perturbation modes.

3. Solve for the second order fields.
4. Compute the potential coefficients A_{ijk} and A_{ijkl} .
5. Plot the second order fields.

17.3 Continuation Analysis

Continuation analysis gives information about the equilibrium paths in the initial post-buckling region. Continuation analysis must always be preceded by a perturbation analysis. See § 18.3.2 on page 308 for commands and § 54.3 on page 621 for background theory. In order to perform a continuation analysis with DIANA you must take the following actions:

1. Perform a complete perturbation analysis as described in the previous section.
2. Solve the reduced set of equilibrium equations.
3. Plot the postbuckling displacement fields.

17.4 Input of Initial Imperfections

Many of DIANA's elements may be applied in Euler stability analysis with initial imperfections [Vol. *Element Library*]. Any imperfection on an element should be small in comparison with the size of the elements, for instance less than 1 % for isoparametric elements and less than 0.1 % for class-I and class-II beams. Thus imperfections on an isoparametric element of size 50 should be less than 0.5. This is required to get an accurate estimation of the influence of imperfections.

syntax

```
'INIVAR'
DISPLA fieldn
1      5 6
noden   typew dirnrn displar
1      5 6
/ nodesng... /
      typew dirnrn displar
1      5 6
/ nodesn... /
      typew dirnrn / displsr... /
```

'INIVAR' is the general table heading for input of initial values for variables.

DISPLA is the subtable heading for initial (strain-free) displacements. Number *field* is the field number for reference from the analysis commands for imperfections [§ 18.2.2 p. 302].

node is a single node number. *nodes* is a series of nodes, which must be specified between slashes. Depending on the format, *nodes* may comprise numbers or groups or both.

type is the type of variable: TR for translation or RO for rotation.

dirnr specifies the direction, referring to table 'DIRECT' [§ 1.5 p. 11].

displa is the initial displacement. *displs* is a series of initial displacements, one for each node in *nodes*, it must be specified between slashes.

Omission of *displa* or *displs* induces random displacements in combination with the random imperfections option [§ 18.2.2 p. 302].

```

file.dat
'INIVAR'
DISPLA 3
  1      TR 3    0.001
/ 10-12 / TR 2 / 0.002(3) /
  4      RO 5    0.00005

```

This example specifies initial displacement field number 3. The initial translation of node 1 is 0.001 in direction 3. The initial translation of nodes 10 to 12 is 0.002 in direction 2. The initial rotation of node 4 is 0.00005 around direction 5. All other translations and rotations are initially equal to zero.

17.5 Applicability of Elements

This section outlines which of the stability analysis options may be applied for the various structural elements. Table 17.1 outlines the applicability of the various options for stabil-

Table 17.1: STABILITY ANALYSIS FOR STRUCTURAL ELEMENTS

	Truss	Beam	Pl. stress	Plate bend.	Pl. strain	Axisymm.	Fl. shell	Cu. shell	Solid
Euler	*	*	fk	-	g	g	*	*	g
Initial displacements	*	c	fk	-	gj	gh	*	*	g
Imperfections	a	cd	fk	-	gj	gh	*	*	g
Perturbation	b	e	-	-	-	-	i	l	-

(*) All elements. (a) Only enhanced elements. (b) Not for cable elements. (c) Not for 2-D elements. (d) Not for class-II. (e) Not for class-I. (f) Not for orthotropic geometry. (g) Not for rubber. (h) Not for shell of revolution. (i) Only spline elements and analytically integrated flat shell element T18FSH. (j) Not for infinite shell. (k) Not for elements with drilling rotation. (l) Not for linear elements. (-) Not applicable.

ity analysis for structural elements: *Euler* stability analysis, *initial displacements*, initial *imperfections* and *perturbation* and continuation analysis. See Volume *Element Library* for description of basic variables and input of these elements.

Truss elements. All truss elements support Euler stability analysis and can take nonlinear effects of initial displacements into account. Only the enhanced truss elements accept imperfections. The cable elements do not support perturbation and continuation analysis.

Beam elements. All beam elements support Euler stability analysis. The class-I beam elements do not support perturbation and continuation analysis. The two-dimensional class-III beam elements do not account for nonlinear effects of initial displacements or imperfections.

Plane stress elements. The regular plane stress elements support Euler stability analysis. These elements also account for nonlinear effects due to initial displacements and imperfections. None of the plane stress elements support perturbation and continuation analysis.

Plane strain elements. The regular plane strain elements and the infinite shell elements support Euler stability analysis. These elements also account for nonlinear effects due to initial displacements and imperfections. None of the plane strain elements support perturbation and continuation analysis.

Axisymmetric elements. The regular solid ring elements and the shells of revolution support Euler stability analysis. These elements also account for nonlinear effects due to initial displacements and imperfections. None of the axisymmetric elements support perturbation and continuation analysis.

Plate bending elements. The plate bending elements do not support stability analysis.

Flat shell elements. All flat shell elements can be used in stability analysis. These elements accept initial displacements and imperfections. Spline elements and the analytically integrated flat shell element T18FSH are also suitable for perturbation and continuation analysis.

Curved shell elements. All curved shell elements support Euler stability analysis. These elements also account for nonlinear effects due to initial displacements and imperfections. Only the higher-order curved shell elements support perturbation and continuation analysis.

Solid elements. The regular solid elements support Euler stability analysis. These elements also account for nonlinear effects due to initial displacements and imperfections. None of the solid elements support perturbation and continuation analysis.

Structural interface elements. The structural interface elements do not support stability analysis.

Spring elements. The spring elements do not apply for stability analysis but can be used in finite element models for such an analysis.

Mass elements. The point mass elements do not apply for stability analysis but can be used in finite element models for such an analysis.

Embedded reinforcements. In stability analysis, the influence of embedded reinforcements is accounted for via the element stiffness.

Chapter 18

Euler Stability Analysis

In order to perform an Euler stability analysis with DIANA you must take the following actions:

1. Invoke Module FILOS to initialize an analysis database [§ 3.2 p. 48].
2. Invoke Module INPUT to read the finite element model into the database [§ 3.3 p. 50].
3. Invoke Module EULER to perform an Euler stability analysis.

The primary tasks for Module EULER are invoked via the following command sequence.

syntax

```
*EULER
[ MODEL ... ]
[ EIGEN [ OFF ] ... ]
[ REDUCE [ OFF ] ... ]
[ CONTIN [ OFF ] ... ]
*END
```

MODEL evaluates and assembles the finite element model [§ 18.1 p. 299].

EIGEN solves the eigenvalue problem in a stability analysis [§ 18.2 p. 300].

REDUCE performs a postbuckling perturbation analysis [§ 18.3.1 p. 305].

CONTIN performs a postbuckling continuation analysis [§ 18.3.2 p. 308].

Default. If you only give the *EULER command, then DIANA will perform a normal Euler stability analysis, i.e., as if you specified the following.

file.dcf

```
*EULER
MODEL
EIGEN
*END
```

18.1 Model Evaluation

The MODEL commands customize the evaluation of the finite element model prior to the actual Euler stability analysis.

syntax

BEGIN MODEL

[OFF]

[EVALUA ...]

[ASSEMB ...]

END MODEL

EVALUA checks and evaluates geometric and material properties for elements [§ 3.4 p. 51].

ASSEMB assembles the elements, i.e., creates appropriate system degrees of freedom [§ 3.5 p. 54].

18.2 Buckling Modes

To solve the eigenvalue problem in an Euler stability analysis, i.e., to determine the buckling modes, you must give commands in the EIGEN block.

syntax

BEGIN EIGEN

[OFF]

[STABIL ...]

[IMPERF ...]

[EXECUT ...]

[OUTPUT ...] ...

END EIGEN

OFF suppresses the execution of a buckling analysis. This may be useful to save computing time if you want to perform a postbuckling analysis [§ 18.3], while the previously determined buckling modes still reside on the FILOS file.

STABIL specifies the type of stability analysis [§ 18.2.1].

IMPERF specifies the imperfections [§ 18.2.2].

EXECUT specifies how to solve the eigenvalue problem. In particular this involves the type of the solution procedure. Commands are identical to those for eigenvalue analysis with Module EIGEN [§ 31.3 p. 432].

OUTPUT specifies the buckling modes to be output [§ 18.2.3 p. 303].

file.dcf

EULER*BEGIN** EIGEN

STABIL LOAD=1

EXECUT NMODES=2

OUTPUT DISPLA

END EIGEN***END**

The first command starts Module EULER. Then, the commands in the EIGEN block ask for buckling analysis where the STABIL command specifies the type of stability analysis. The LOAD=1 parameter asks for buckling eigenmodes to be calculated for linear elastic load set 1. The EXECUT command specifies how to execute the buckling analysis. Parameter NMODES=2 selects the first two modes to be determined (if existing at all). Finally the OUTPUT command specifies the output to be produced: the buckling eigenmodes (as displacements).

18.2.1 Stability Analysis

The **STABIL** command block within the **EIGEN** command block of Module **EULER** specifies how to setup the system of equations for stability analysis.

syntax

```

BEGIN STABIL
[ OFF ]
[ LOAD=loadsetn ]
[ DISPLA [ OFF ] ]
[ NLPREB ]
END STABIL

```

OFF suppresses the setup of the system of equations. This may save computing time when the system of equations is still available on the **FILOS** file from a previous stability analysis.

LOAD=loadset uses the linear elastic stresses of the specified load set for the geometric stress–stiffness matrix \mathbf{K}_G . Default is the lowest available load set number. The load set number corresponds to a load set in input table '**LOADS**' [§ 2.3 p. 31].

DISPLA specifies that the initial displacement matrix \mathbf{K}_{LL} must be incorporated in the system of equations [Eq. (54.18) p. 618] and. solves the eigenvalue problem

$$\left(\mathbf{K}_{L0} + \lambda_{\text{crit}} (\mathbf{K}_{LL}(\mathbf{u}_{\text{lin}}) + \mathbf{K}_G(\mathbf{u}_{\text{lin}})) \right) \delta \mathbf{U} = 0$$

OFF causes the initial displacement matrix to be neglected, i.e., only the linear stiffness matrix \mathbf{K}_{L0} and the geometrical stress–stiffness matrix \mathbf{K}_G are considered in solving the eigenvalue problem

$$\left(\mathbf{K}_{L0} + \lambda_{\text{crit}} \mathbf{K}_G(\mathbf{u}_{\text{lin}}) \right) \delta \mathbf{U} = 0$$

By default the initial displacement matrix \mathbf{K}_{LL} is being neglected.

NLPREB specifies that the stability (buckling) analysis is done on top of the deformed configuration achieved by a previous nonlinear analysis. The following eigenvalue problem is solved with **DISPLA ON**:

$$\left(\mathbf{K}_T + \lambda_{\text{crit}} (\mathbf{K}_{LL}(\mathbf{u}_{\text{lin}}) + \mathbf{K}_G(\mathbf{u}_{\text{lin}})) \right) \delta \mathbf{U} = 0$$

The following eigenvalue problem is solved with **DISPLA OFF**:

$$\left(\mathbf{K}_T + \lambda_{\text{crit}} \mathbf{K}_G(\mathbf{u}_{\text{lin}}) \right) \delta \mathbf{U} = 0$$

Where \mathbf{K}_T is the tangential stiffness matrix at the end of the previous nonlinear analysis.

For curved shell elements, the contribution from prebuckling nonlinearity is also accounted for in the subsequent perturbation and continuation analysis when **NLPREB** is specified.

*Note that tangent stiffness matrices are only available when the nonlinear analysis (*NONLIN) involves a Newton-Raphson iteration method (METHOD NEWTON) [§ 13.3.5.1 p. 236]. No tangent stiffness matrices are being constructed when the nonlinear analysis is done with other iteration methods, e.g. Constant Stiffness, Linear stiffness, or secant (Quasi-Newton) method.*

file.dcf

```
BEGIN EIGEN
  STABIL DISPLA OFF
END EIGEN
```

file.dcf

```
BEGIN EIGEN
  BEGIN STABIL
    LOAD=2
    DISPLA ON
  END STABIL
END EIGEN
```

18.2.2 Imperfections

The IMPERF command block applies an imperfection pattern.

Any imperfections setup in a buckling analysis, are automatically taken into account in subsequent nonlinear analyses. This allows for postbuckling analysis including critical imperfections.

Imperfections can be specified as:

- Scaled buckling modes [§ 18.2.2.1].
- Random imperfection pattern [§ 18.2.2.2].
- User-specified imperfection pattern [§ 18.2.2.3].

See § 54.2 on page 620 for background theory.

18.2.2.1 Scaled Buckling Modes

The commands for application of scaled buckling modes as imperfection pattern are as follows.

Scaled buckling modes imperfection

syntax

```
BEGIN IMPERF
[ OFF ]
  BEGIN BUCKLI
    [ MODE=buckl1n [ buckl2n ... bucklnn ] / ]
    [ MAX=size1r [ size2r ... sizenr ] / ]
  END BUCKLI ]
END IMPERF
```

OFF suppresses the application of an imperfection pattern.

BUCKLI asks for one or more buckling modes as imperfection pattern.

MODE= *buckl1* to *buckln* specify the buckling modes to be used for the imperfection pattern. Generally, the most appropriate imperfection field is the lowest buckling mode which DIANA takes by default.

(*buckli* ≤ *nmodes*)

[*buckl1* = 1]

MAX= *size1* to *sizen* specify the largest translations or rotations for each scaled buckling mode *buckli*.

[*size1*=0.01]

18.2.2.2 Random Imperfection Pattern

The commands for application of a random imperfection pattern are as follows.

Random imperfection pattern *syntax*

```

BEGIN IMPERF
[ OFF ]
BEGIN RANDOM
  [ FIELD=fieldn ]
  [ MAX=sizer ]
END RANDOM ]
END IMPERF

```

OFF suppresses the application of an imperfection pattern.

RANDOM asks for a random imperfection pattern. Note that the random pattern is reproducible, i.e., each analysis run will produce the same random pattern.

FIELD= *field* refers to a field number from table 'INIVAR' [§ 17.4 p. 296]. Default is the lowest available field number. The node-type-direction combinations in this table specify the degrees of freedom that will have a random imperfection. The values *displa* or *displs* for the displacements must not be input in this case.

MAX= *size* indicates that the imperfections in the finite element model will not exceed the value that you specify with parameter *size*. But generally, none of the imperfections will exactly equal this value. [*size*=0.01]

18.2.2.3 User-specified Imperfection Pattern

The commands for application of a user-specified imperfection pattern are as follows.

User-specified imperfection pattern *syntax*

```

BEGIN IMPERF
[ OFF ]
BEGIN USER
  [ FIELD=fieldn ]
  [ FACTOR=facr ]
END USER ]
END IMPERF

```

OFF suppresses the application of an imperfection pattern.

USER takes a user-specified imperfection pattern.

FIELD= *field* refers to a field number from table 'INIVAR' [§ 17.4 p. 296]. Default is the lowest available field number. The node-type-direction combinations in this table specify the degrees of freedom that will have an imperfection.

FACTOR= *fac* is the multiplication factor for the displacement values from the selected field of table 'INIVAR' to get the actual imperfection. [*fac* = 1]

18.2.3 Output of Buckling Modes and Values

DIANA can output the buckling modes as displacements via the OUTPUT command block. For the general, analysis type independent options for output of analysis results see § 3.6 on page 55.

syntax

```

BEGIN OUTPUT [devicew] [outoptw...] [params]
[ OFF ]
[ BEGIN SELECT
  [ ... ]
  [ MODES _____ ]
      modesn...
      ALL
END SELECT ]
[ LAYOUT ... ]
itemw [typew] [formw] [operw] { compw } { optiw }
DISPLA  BUCKLI  TRANSL  LOCAL
          ROTATI  GLOBAL
END OUTPUT

```

SELECT customizes the batch output selection.

... for model selection see § 3.6.2 on page 59.

[ALL] MODES selects specific buckling modes for output: *modes* is a series of mode numbers, ALL selects all calculated modes.

LAYOUT specifies optional commands to customize the layout and style of tabular output [§ 3.6.4.1 p. 65].

item is the name of the analysis result to be output. See § 3.6 on page 55 for complete syntax of this command.

DISPLA for displacements, which are the only available results of a stability analysis.

[BUCKLI] *type* specifies the displacement type [§ 3.6.1 p. 57]: BUCKLI] for buckling modes.

Default

file.dcf

```

*EULER
[ commands ]
BEGIN EIGEN
  OUTPUT
END EIGEN
*END

```

If you only give a single OUTPUT command, like in the above example, or if you do not give the OUTPUT command at all, then DIANA gives output of the translational displacements in global XYZ orientation, as if you had given the following commands.

file.dcf

```

*EULER
[ commands ]
BEGIN EIGEN
  BEGIN OUTPUT
    DISPLA BUCKLI TRANSL GLOBAL
  END OUTPUT
END EIGEN
*END

```

Buckling modes. DIANA normalizes the buckling mode ϕ_i so much that the largest translation component u has the value of 1.

item	type	form	oper	comp ...		
				X	Y	Z
DISPLA	BUCKLI	TRANSL	LOCAL	Dtx	Dty	Dtz
				u_x	u_y	u_z
DISPLA	BUCKLI	TRANSL	GLOBAL	DtX	DtY	DtZ
				u_X	u_Y	u_Z
DISPLA	BUCKLI	ROTATI	LOCAL	Drx	Dry	Drz
				ϕ_x	ϕ_y	ϕ_z
DISPLA	BUCKLI	ROTATI	GLOBAL	DrX	DrY	DrZ
				ϕ_X	ϕ_Y	ϕ_Z

Buckling values. The appropriate buckling value is output together with the required buckling mode. The buckling values correspond with the parameter λ_i . The lowest buckling value is the most critical buckling value. If a load \mathbf{f} is applied to the structure and if the resulting stresses are used for the geometric stress stiffness matrix, the lowest buckling value represents the ratio $\mathbf{f}_{\text{crit}}/\mathbf{f}$. Thus it indicates the factor of the load \mathbf{f} that can be applied on the structure before buckling occurs. However, the validity of this statement is restricted by the assumptions made in Euler buckling theory [§ 54.1 p.617].

Negative buckling values indicate that buckling occurs if the direction of the external load is switched. Hence, such modes do not really occur if the load as specified in the data file is applied on the structure.

18.3 Postbuckling Analysis

To perform a postbuckling analysis with Module EULER you must give REDUCE and CONTIN commands. Note that a postbuckling analysis can only be performed if the results of a previous buckling analysis are available on the FILOS file.

syntax

```

BEGIN REDUCE
[ OFF ]
...
END REDUCE
BEGIN CONTIN
[ OFF ]
...
END CONTIN

```

REDUCE specifies how to perform the perturbation analysis [§ 18.3.1]. The OFF option suppresses the execution of a perturbation analysis. This may save computing time when the reduced system of equations is still available from a previous perturbation analysis.

CONTIN specifies how to perform the continuation analysis [§ 18.3.2]. The OFF option suppresses the execution of a continuation analysis.

18.3.1 Perturbation Analysis

To specify the perturbation analysis, i.e., the reduction of the full set of equations, you must give commands in the REDUCE block.

syntax

```

BEGIN REDUCE
[ BEGIN EXECUT
  [ OFF ]
  [ SELECT MODES            ]
                    modes n...
                    ALL
  [ LOADFA=lambdar ]
  [ MAXAMP=ampr ]
  [ UNCPL ]
  END EXECUT ]
[ BEGIN OUTPUT [devicew] [outoptw...] [params]
  [ OFF ]
  [ SELECT ... ]
  [ LAYOUT ... ]
  DISPLA [typew] [formw] [operw] { compw } { optiw }
          SECMOD   TRANSL   LOCAL
          ROTATI   GLOBAL

  END OUTPUT ]
END REDUCE

```

EXECUT defines how to execute the perturbation analysis.

- OFF suppresses the execution of a perturbation analysis. This may save computing time when you only want to get output of a previous perturbation analysis.
- [ALL] SELECT MODES selects buckling modes for the perturbation analysis. The modes must have been determined previously [§ 18.2 p. 300]. You may explicitly specify is a set of *modes*, or you may require a perturbation analysis for ALL determined modes. All modes is also the default if you do not select modes.
- [$\lambda_p = 0.95$] LOADFA=*lambda* specifies the relative load factor λ_p of the perturbation point [Eq. (54.37) p. 621].
- MAXAMP=*amp* specifies that the selected buckling modes will be scaled in the perturbation analysis such that the maximum displacement amplitude of each selected buckling model will be equal to *amp*.
- UNCPL specifies that the second order modes are considered uncoupled, and the cross-correlated postbuckling coefficients will not be computed in the subsequent continuation analysis.
- OUTPUT specifies the desired output of the perturbation analysis. For the general, analysis type independent options for output of analysis results see § 3.6 on page 55.
- SELECT customizes the batch output selection. Appropriate for perturbation analysis is node selection [§ 3.6.2 p. 59].
- LAYOUT optional commands to customize the layout and style of tabular output [§ 3.6.4.1 p. 65].
- [SECMOD] DISPLA SECMOD will output the second order displacements [§ 18.3.1.1].

Default

file.dcf

```

*EULER
[ commands ]
BEGIN REDUCE
[ commands ]
  OUTPUT
END REDUCE
*END

```

If you only give a single OUTPUT command, like in the above example, or if you do not give the OUTPUT command at all, then DIANA gives output of the translational displacements in global XYZ orientation, as if you had given the following commands.

file.dcf

```
*EULER
[ commands ]
BEGIN REDUCE
[ commands ]
BEGIN OUTPUT
  DISPLA SECMOD TRANSL GLOBAL
END OUTPUT
END REDUCE
*END
```

Example.

file.dcf

```
BEGIN REDUCE
  BEGIN EXECUT
    SELECT MODES 1 2
    LOADFAC=0.91
  END EXECUT
  BEGIN OUTPUT
    DISPLA SECMOD
  END OUTPUT
END REDUCE
```

file.dcf

```
BEGIN REDUCE
  EXECUT SELECT MODES 1 2 LOADFAC=0.91
  OUTPUT DISPLA
END REDUCE
```

18.3.1.1 Output of Second Order Displacements

The SECMOD type specifier gives output of displacements of a perturbation analysis.

<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	<i>comp ...</i>		
				X	Y	Z
DISPLA	SECMOD	TRANSL	LOCAL	D2tx	D2ty	D2tz
				u_x	u_y	u_z
DISPLA	SECMOD	TRANSL	GLOBAL	D2tX	D2tY	D2tZ
				u_X	u_Y	u_Z
DISPLA	SECMOD	ROTATI	LOCAL	D2rx	D2ry	D2rz
				ϕ_x	ϕ_y	ϕ_z
DISPLA	SECMOD	ROTATI	GLOBAL	D2rX	D2rY	D2rZ
				ϕ_X	ϕ_Y	ϕ_Z

Default*file.dcf*

```

*EULER
[ commands ]
BEGIN REDUCE
[ commands ]
OUTPUT
END REDUCE
*END

```

If you only give a single **OUTPUT** command, like in the above example, or if you do not give the **OUTPUT** command at all, then DIANA gives output of the translational displacements in global *XYZ* orientation, as if you had given the following commands.

file.dcf

```

*EULER
[ commands ]
BEGIN REDUCE
[ commands ]
BEGIN OUTPUT
DISPLA SECMOD TRANSL GLOBAL
END OUTPUT
END REDUCE
*END

```

18.3.2 Continuation Analysis

To specify the continuation analysis you must give commands within the **CONTIN** block.

syntax

```

BEGIN CONTIN
  BEGIN EXECUT
    [ OFF ]
    [ COEFF [ _____ ] ]
      ON
      OFF
    [ BEGIN START
      [ LAMBDA=lambdar ]
      [ AMPLIT amplr... ]
    END START ]
    [ AMPIMP ampir... ]
    [ STEPS { _____ } ]
      NSTEPS=nstepsn
      SIZE=sizer
    [ NORM [ DISPLA ] CONVER=epsr ]
    [ DYBUCK [ FACTOR=facr ] ]
  END EXECUT
  [ BEGIN OUTPUT [devicew] [outoptw...] [params] ]
    [ OFF ]
  [ BEGIN SELECT
    [ ... ]
    [ STEPS _____ ]
      stepsn...
      ALL

```

```

        LAST
    END SELECT ]
[ LAYOUT ... ]
DISPLA [ typew ] [ formw ] [ operw ] { compw } { optiw }
        POSTBU   TRANSL   LOCAL
                ROTATI   GLOBAL

    END OUTPUT ]
END CONTIN

```

EXECUT specifies how to perform the continuation analysis.

OFF suppresses the execution of the continuation analysis.

COEFF specifies whether postbuckling coefficients will be computed (ON) or not (OFF). [COEFF ON]

START specifies the starting point in the load factor–mode amplitude space. Parameter LAMBDA=*lambda* is an estimation for the load factor λ . AMPLIT *ampl* are estimations for the respective mode amplitudes, one value for each selected mode. [$\lambda = 1$]
[*ampl* = 0]

AMPIMP *ampi* are the respective imperfection amplitudes for each selected mode. [*ampi* = 0]

STEPS specifies the execution of steps in the continuation procedure. Parameter NSTEPS=*nsteps* indicates the number of steps, SIZE=*size* is the step size. For dynamic buckling analysis the steps are considered as time steps. [*nsteps* = 1]
[*size* = 0.01]

NORM specifies the convergence criterion for the iteration process. The tolerance ε is indicated by parameter CONVER=*eps*. In postbuckling analysis the only available criterion is on the displacement norm, therefore you may omit the DISPLA option. [$\varepsilon = 10^{-5}$]
[DISPLA]

DYBUCK specifies that an additional dynamic coefficient will be computed. This coefficient can be used to estimate the dynamic buckling load of the structure under dynamic loading specified by a time–load diagram (table 'TIMELO' [§ 6.5.3 p. 108]). FACTOR specifies an additional multiplication factor *fac* for the time–load diagram (table 'TIMELO'). For more information see the background theory [§ 54.3 p. 621]. [*fac* = 1]

OUTPUT specifies the desired output of the perturbation analysis. For the general, analysis type independent options for output of analysis results see § 3.6 on page 55.

SELECT customizes the batch output selection.

... is a model selection [§ 3.6.2 p. 59]. Appropriate for continuation analysis is node selection

STEPS selects steps for output of analysis results. Where *steps* are numbers of selected steps, ALL will produce output for all steps that are executed (the default), LAST selects the final step. [ALL]

LAYOUT optional commands to customize the layout and style of tabular output [§ 3.6.4.1 p. 65].

DISPLA POSTBU will output the postbuckling displacements [§ 18.3.2.1]. [POSTBU]

Default

file.dcf

```

*EULER
[ commands ]
BEGIN CONTIN
[ commands ]
OUTPUT
END CONTIN
*END

```

If you only give a single OUTPUT command, like in the above example, or if you do not give the OUTPUT command at all, then DIANA gives output of the translational displacements in global XYZ orientation, as if you had given the following commands.

file.dcf

```

*EULER
[ commands ]
BEGIN CONTIN
[ commands ]
BEGIN OUTPUT
  DISPLA POSTBU TRANSL GLOBAL
END OUTPUT
END CONTIN
*END

```

Load factor. The postbuckling load factor is output together with the postbuckling displacement field. This factor corresponds with the quotient λ/λ_1 where λ is the actual postbuckling load factor and λ_1 is the lowest buckling value. The load corresponding to the actual postbuckling configuration is defined by

$$f_{pb} = \lambda_{crit} = \lambda \times \lambda_1 \times f \quad (18.1)$$

Example.

file.dcf

```

BEGIN CONTIN
BEGIN EXECUT
  BEGIN START
    LAMBDA=1.0
    AMPLIT 0.0 0.0
  END START
  STEPS NSTEPS=10 SIZE=1.0
  NORM DISPLA CONVER=0.001
END EXECUT
BEGIN OUTPUT
  DISPLA POSTBU
END OUTPUT
END CONTIN

```

In the above example a postbuckling continuation analysis of ten steps is executed, starting in the initial buckling point. A step size of 1.0 is applied while the convergence criterion is 0.001.

18.3.2.1 Output of Postbuckling Displacements

The POSTBU type specifier gives output of displacements of a postbuckling continuation analysis. As outlined in the background theory for postbuckling analysis [§ 54.3 p. 621], DIANA calculates the total displacements field \mathbf{u}_{pb} for the actual postbuckling configuration according to

$$\mathbf{u}_{pb} = \lambda \mathbf{u}_{lin} + a_i \phi_i + a_i a_j \mathbf{u}_{ij} \quad (18.2)$$

where \mathbf{u}_{ij} is called the second order displacement vector and a_i should be interpreted as amplitude of the respective mode.

<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	<i>comp ...</i>		
				X	Y	Z
DISPLA	POSTBU	TRANSL	LOCAL	DPtx	DPty	DPtz
				u_x	u_y	u_z
DISPLA	POSTBU	TRANSL	GLOBAL	DPtX	DPtY	DPtZ
				u_X	u_Y	u_Z
DISPLA	POSTBU	ROTATI	LOCAL	DPrx	DPrY	DPrz
				ϕ_x	ϕ_y	ϕ_z
DISPLA	POSTBU	ROTATI	GLOBAL	DPrX	DPrY	DPrZ
				ϕ_X	ϕ_Y	ϕ_Z

Part VI

Potential Flow Analysis

Chapter 19

Introduction to Potential Flow Analysis

DIANA's flow modules support the following applications of potential flow analysis:

- Heat flow and concentration flow [Ch. 21],
- Groundwater flow [Ch. 22],
- Beam cross-section analysis [Ch. 23],
- Reynolds flow [Ch. 24].

See Volume *Analysis Examples* for real-life examples of potential flow analysis and Chapter 55 for background theory.

Convection-diffusion problems. Using the options described for heat flow it is of course possible to solve analogous other convection-diffusion problems.

Staggered flow–stress analysis. The potential field, resulting from a potential flow analysis may be input for a structural analysis. This implies a so-called staggered flow–stress analysis. Modules HEATSS, HEATR, GROWSS, and GROWTR can automatically convert a finite element model for structural analysis to a model for potential flow analysis. Modules LINSTA, NONLIN, EIGEN, EULER, MODAL, FREQUE, SPECTR, and HFTD convert the finite element model back to a structural model and use the flow analysis results as input, e.g. loading input [Ch. 26].

Fluid–structure interaction. By coupling of a potential flow analysis to a dynamical structural analysis, see Part III, DIANA can model the added mass and for modules FREQUE and HFTD also the added damping by the fluid or gas which surrounds the structure.

Chapter 20

Input for Potential Flow Analysis

This chapter describes all input that can be specified for a potential flow analysis. The chapters on a particular type of potential flow analysis describe which of this input data is actually required, for instance Chapter 21 for heat flow analysis. For a description of general input of a finite element mesh see Chapter 1. For input of elements in table 'ELEMEN' see Volume *Element Library*. For input of material properties in table 'MATERI' see Volume *Material Library*. See also Volume *Getting Started* for a description of the DIANA input data format.

20.1 Nodal Potentials

20.1.1 Fixed Nodal Potentials

Depending in the type of analysis, nodes with a fixed nodal potential must be specified in table 'FIXTEM', 'FIXHEA' or 'FIXPOT'. Non-zero values for fixed potentials must be input via table 'BOUNDA' [§ 20.2 p. 317].

Any node specified in one of these tables but not in table 'BOUNDA', will have a fixed potential of zero.

A special case occurs in a *beam cross-section analysis* [Ch. 23]: you must specify one, and only one, fixed node in table 'FIXPOT' when doing such analysis.

syntax

<i>tabnam_w</i>	
'FIXTEM'	
'FIXHEA'	
'FIXPOT'	
NAME <i>name_s</i>	
1	80
<i>node_n</i>	
1	80
/ <i>nodes_{ng...}</i> /	

NAME *name* is the name for the fixed nodal potential set and may be used for identification of and reference to the fixed nodal potential set. A name specification line must follow immediately after the table heading. The name specification may be respecified at the start of a new subtable.

tabnam is the table heading for input of nodes with a fixed potential: 'FIXTEM' for fixed temperatures in heat flow analysis, or 'FIXHEA' for fixed heads in groundwater flow

analysis, or 'FIXPOT' for fixed potentials in potential flow analyses other than heat or groundwater flow (for instance concentration flow).

The nodes may be specified with a single number *node* (one per input line) or with a series of *nodes* between slashes which comprises numbers or groups or both.

The single number format and the series format may also be combined like in the following example.

file.dat

```
'FIXTEM'
NAME CONSTANT
  10
 / 15-19 /
```

In this example node 10 has a fixed temperature and nodes 15 to 19 have fixed temperatures as well.

20.1.2 Initial Nodal Potentials

You may start a transient analysis with an initial potential field. In a nonlinear steady-state state analysis you may specify an estimation for the first solution of the potential field, which also may be considered as an initial field. There are two ways to specify such an initial potential field: (1) directly via input table 'INIVAR' or (2) as a potential field found from a previous analysis. If you do not specify an initial field in either way, then DIANA assumes that all the nodal potentials are initially equal to zero.

There are three forms of input syntax for table 'INIVAR': (1) one nodal potential per line, (2) nodes in a series of numbers or groups or both, with one potential valid for all the nodes in the series, (3) nodes in a series of numbers, with a series of potentials: one value for each node.

syntax

```
'INIVAR'
subtabw fieldn
TEMPER
HEAD
POTENT
```

```
1 5 6 80
noden phir
```

```
1 5 6 80
/ nodesng... /
    phir
```

```
1 5 6 80
/ nodesn... /
    /phisr... /
```

'INIVAR' is the table heading for initial nodal potentials.

subtab is the subtable name, depending on the type of analysis: **TEMPER** for initial temperatures in heat flow analysis, or **HEAD** for initial heads in groundwater flow analysis, or **POTENT** for initial potentials in potential flow analysis other than heat or groundwater flow (for instance concentration flow).

The *field* number is used for reference from the analysis commands.

node is a single node number. *nodes* is a series of nodes, which must be specified between slashes. Depending on the format, *nodes* may comprise numbers or groups or both.

phi is the value of the initial temperature T_0 , or the initial potential ϕ_0 ; *phis* is a series of initial temperatures or potentials, one for each node in *nodes*, it must be specified between slashes.

file.dat

```
'INIVAR'
TEMPER 3
  1  0.001
/ 4 7 10-30(5) / 0.00005
/ 34 37 40-70(5) / / 0.001(2) 0.0005(7) /
```

20.2 Boundary Conditions

There are two classes of boundary conditions, each of which is specified in a separate subtable of table 'BOUNDA'.

Nodal potentials: prescribed potential on nodes or boundary elements.

Discharges: prescribed discharge Q (source) on nodes or ordinary elements.

You may specify various cases. Each case may have nodal or element boundary conditions or both. In the analysis, each case represents a set of nodal potentials for the nodes where the potential is prescribed, and a set of total equivalent prescribed nodal fluxes for the other nodes.

If you do not specify a condition for a certain boundary, then DIANA assumes that this boundary is completely insulated, i.e., the flux through it is equal to zero.

syntax

```
'BOUNDA'
CASE casen
[NAME names]

NODAL
1 5 6 80
nodal boundary conditions

ELEMEN
1 5 6 12 13 80
element boundary conditions
```

'BOUNDA' is the table heading for boundary conditions input.

CASE *case* is the number of a new boundary case. A case number specification line must follow immediately after the table heading. The case number may be respecified at the start of a new subtable.

NAME *name* is the name for the boundary case and may be used for identification of the boundary case set. A name specification line must follow immediately after the boundary case number specification.

NODAL is the subtable for nodal boundary conditions [§ 20.2.1].

ELEMEN is the subtable for element boundary conditions. See Volume *Element Library* for input syntax.

20.2.1 Nodal Boundary Conditions

Nodal boundary conditions are input in subtable NODAL. There are three forms of input syntax: (1) one nodal condition per line, (2) nodes in a series of numbers or groups or both, with one condition value valid for all the nodes in the series, (3) nodes in a series of numbers and/or groups, with a series of condition values: one value for each node.

syntax

'BOUNDA'			
NODAL			
1	5	6	80
$node_n$	$type_c$	$value_r$	
1	5	6	80
/ $nodes_{ng...}$ /			
	$type_c$	$value_r$	

NODAL is the subtable heading for nodal boundary conditions.

$node$ is a single node number, $nodes$ is a series of nodes, it must be specified between slashes and may comprise numbers or groups or both.

$type$ is the type of the boundary condition: T for prescribed temperature in heat flow analysis, H for prescribed head in groundwater flow analysis, Q for prescribed discharge (point source), or P for prescribed potential in flow analysis other than heat or groundwater flow.

$value$ is the value of the boundary condition; $values$ is a series of condition values, one for each node in $nodes$.

file.dat

'BOUNDA'			
CASE 1			
NODAL			
2	Q	0.01	
4	T	-0.12	
/ 20-35 /			
	T	0.25	

20.2.2 Transient Boundary Conditions

For transient analysis, the boundary conditions specified in table 'BOUNDA' [§ 20.2 p. 317] must be related to the time. For each case from table 'BOUNDA', linear diagrams can be specified for the multiplication factor of the boundary conditions of that case in a time interval: $t_1 \leq t < t_2$. The total boundary conditions on a certain time are a superposition of all the multiplications of the interpolated factors with the boundary conditions belonging to the specified cases. You may specify the diagram completely, i.e., times and multiplication factors, in table 'TIMEBO'. Alternatively, you may specify only the times in table 'TIMEBO' and import the multiplication factors from an external file.

Complete table

syntax

'TIMEBO'			
1			80
BOUNDA $case_n$			
TIMES $times_{r...}$ /			
FACTOR $f_{r...}$ /			

'TIMEBO' is the table heading for time dependent boundary conditions.

BOUNDA $case$ is the number of the boundary case, active in the interval. This number refers to a boundary case specified in table 'BOUNDA' [§ 20.2 p. 317].

TIMES $times$ are the times t for the corresponding multiplication factors. Times must be specified in increasing order.

($t_{i+1} \geq t_i$)

FACTOR f are the multiplication factors for the corresponding times.

You may specify multiple sets of times and factors, optionally with a different case number for each set. Within one set, the number of factors f must always match the number of times in *times* as shown in the following example.

file.dat

```
'TIMEBO'
BOUNDA 2
TIMES 0.00 0.10 /
FACTOR 2.5 3.6 /
BOUNDA 1
TIMES 0.13 0.16 0.16 0.45 /
FACTOR 6.4 4.2 2.5 0.0 /
BOUNDA 3
TIMES 0.45:0.501(0.01) /
FACTOR 4.8 5.3 7.9 7.8 6.2 1.4 /
```

In this example no boundary case is active from $t = 0.10$ to $t = 0.13$.

Factors imported

syntax

```
'TIMEBO'
```

1		5	6						80
---	--	---	---	--	--	--	--	--	----

```
BOUNDA casen
TIMES timesr... /
FACTOR IMPORT files
      [SKIP nlinn]
      [ ]
      SCALE sfacr
      PEAK pvalr
```

'TIMEBO' is the table heading for the time dependent boundary conditions.

BOUNDA *case* is the number of the boundary case, active during the following times. The case number *case* refers to a boundary case specified in table 'BOUNDA' [§ 20.2 p. 317].

TIMES *times* are the times t for the corresponding load factors. Times must be specified in increasing order. ($t_{i+1} \geq t_i$)

FACTOR IMPORT asks DIANA to read the multiplication factors from an external file named *file*. Factors on this file may be separated by spaces, commas, tabs or newlines. There are two additional options.

SKIP causes DIANA to skip the first *nlin* lines before starting to read factors. [*nlin* = 0]

SCALE specifies a scale factor *sfac*. DIANA will multiply each factor from the external file by this factor before applying the boundary condition. [*sfac* = 1]

PEAK asks DIANA to scale the values from the external file such that the greatest absolute value of the load factors is equal to *pval*. ($pval > 0$)

You may specify multiple sets of times and file, optionally with a different load number for each set. Within one set, DIANA reads as many factors from *file* as there are times in *times*.

```

file.dat
'TIMEBO'
BOUNDA 4
TIMES 0.00 0.10 0.13 0.16 0.16 0.45 /
FACTOR IMPORT "bound.log"
      SKIP 2
      SCALE 3.

```

In this example DIANA will read six factors from external file `bound.log`. This file could be like shown below.

```

bound.log
Logging boundary
Thu Jan 6 15:42:15 CET 2000
1 3 4 5.2 3.2 1.6

```

The first two lines will be skipped. The actual factors for transient boundary case 4 will be: $f_1 = 3 \times 1 = 3$, $f_2 = 3 \times 3 = 9$, etc. Note that in this example the factor suddenly drops at time $t = 0.16$ from $3 \times 5.2 = 15.6$ to $3 \times 3.2 = 9.6$.

Value formats. All regular Fortran formats are allowed for the factors on the external file. Integers (without decimal point) are interpreted as reals. Exponents of ten may be specified via E or D format. For instance, 45, 45.0, 45.0E0, and 45.0D0 all represent the real value of 45.0 for the scale factor.

20.3 Linear Constraints

Linear constraints are user specified linear *dependencies* between degrees of freedom of the system of equations (temperatures, pressures, potentials). These dependencies are specified in input table 'TYINGS', and in DIANA terms called *tyings*.

Some examples of the application of tyings are: symmetry, mesh refinement, connection in case of incompatibility. Before you go into detail on these topics, keep in mind what NAFEMS [60] says on this point:

“Constraint equations or displacement transformations, as required by the analysis system, must be formulated with extreme care and geometric consistency to the full analysis accuracy. It is better to truncate the nodal geometry data to ensure transcription than to make errors in physically meaningless digits. If in any doubt, always seek expert advice!”

In general, you should always note when applying tyings whether they have a physical meaning or not. In particular, tyings depend on the system degrees of freedom that will arise in the nodes during the composition of the finite element model. If you apply tyings that are physically meaningless, this will often become clear from the absence of equilibrium and/or from inexplicable answers.

20.3.1 Nodes and Degrees of Freedom

Generally speaking, a tying consists of a degree of freedom in a *master* node, in the figures of this section drawn as ●, and one or more degrees of freedom in *slave* nodes, drawn as ○. Figure 20.1 shows the notation convention for degrees of freedom for tyings. As for

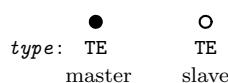


Figure 20.1: Tying degrees of freedom

rigid supports, degrees of freedom for tyings are specified by means of a *type*: TE for temperature, PR for pressure, or PO for potential.

A slave degree of freedom cannot be a master in another tying, nor can it be supported.

The solution of the system of equations yields the solution for the degrees of freedom of the master. DIANA derives the solution for the slave degrees of freedom from the master's.

20.3.2 Single- and Multi-point Tyings

Tyings may be input in two formats: *single-point* or *multi-point*.

20.3.2.1 Single-point

In single-point format you must explicitly specify all nodes for each tying separately.

20.3.2.2 Multi-point

In multi-point format the tying is specified as a connection between a master edge and a slave edge, where these edges consist of straight line sections [Fig. 20.2]. The principle is that you specify the corner nodes of these edges, the so-called *vertices*, indicated with double circles. You also specify a set of master and slave nodes in arbitrary order, the

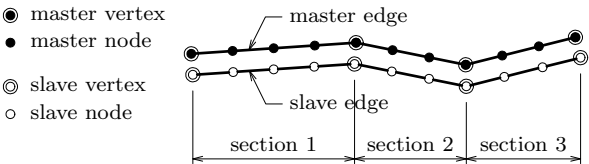


Figure 20.2: Multi-point tying

single circles.

The master and slave nodes also comprise the vertices themselves, so you must specify the vertices twice.

If the master and slave nodes are on the straight edge sections then DIANA determines their sequence [Fig. 20.2]. DIANA ignores specified nodes which are not a straight edge section.

Conditions. For multi-point tyings, you must specify as many master vertices as slave vertices. For each edge section, the amount of (accepted) master and slave nodes depends on the type of tying.

20.3.3 General Input Syntax

Tyings must be input in one single table 'TYINGS', divided in separate subtables for the various types of tyings. These subtables, listed below, may appear repeatedly and in arbitrary order.

				<i>syntax</i>	
'TYINGS'					
NAME <i>name</i> _s					
<hr/>					
EQUAL					
EQUMPC					
<hr/>					
1	5	6	80		
<i>Equalities ...</i>					
<hr/>					
BETWEE					
BTWMPC					
<hr/>					

1	5	6		80
---	---	---	--	----

Interconnections ...

FIX
FIXMPC

1	5	6		80
---	---	---	--	----

General connections ...

NAME *name* is the name for the linear constraint set and may be used for identification of and reference to the linear constraint set. A name specification line must follow immediately after the table heading. The name specification may be respecified at the start of a new subtable.

Equalities describe equality of degrees of freedom [§ 20.3.4]. Subtable EQUAL is for single-point tyings, EQUMPC for multi-point tyings.

Interconnections describe the interconnection of a node (and its degrees of freedom) to two other nodes [§ 20.3.5]. Subtable BETWEE is for single-point tyings, BTWMPC for multi-point tyings.

General connections describe linear dependency between one degree of freedom and one or more other degrees of freedom [§ 20.3.6]. Subtable FIX is for single-point tyings, FIXMPC for multi-point tyings.

The input syntax for the various types of tyings is described in the referred sections which also comprise some instructive examples.

20.3.4 Equalities

The most common use of tyings is the equalization of degrees of freedom, or more specifically: the constraint that one or more displacements are equal. Equality tyings may be input in single- or multi-point format.

20.3.4.1 Single-point Equalities

Equalities in single-point format must be specified in subtable EQUAL with one or more slave nodes, combined with a single master node.

syntax

```
'TYINGS'
NAME names
EQUAL typew {typew ...}
1      5 6
snoden  mnoden
1      5 6
/snodesng... /
      mnoden
```

type describe one or more degrees of freedom [§ 20.3.1 p. 320].

snode is the number of the slave node.

snodes is a series of slave nodes.

mnode is the number of the master node.

General example. The following input is a general example of the single-point equality tyings.

file.dat

```

'TYINGS'
: (a) two nodes with same temperature
NAME TWOSAME
EQUAL TE
  34  57
: (b) range of nodes having the same temperature
NAME EQUALRANGE
EQUAL TE
  / 18-21 / 35

```

The first tying describes that the temperature is the same for node 34 and 57. In the second tying, temperatures of nodes 18 to 21 are the same as for node 35.

20.3.4.2 Multi-point Equalities

Equalities in multi-point format [§ 20.3.2 p. 321] must be specified in subtable EQUMPC with series of slave vertices and nodes, combined with series of master vertices and nodes [Fig. 20.3]. For each edge section, the amount of master nodes n must be equal to the

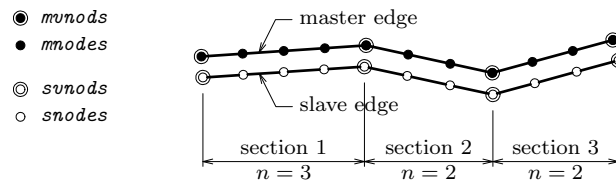


Figure 20.3: Multi-point equality

amount of slave nodes, i.e., each slave node is connected to one master node.

syntax

```

'TYINGS'
NAME names
EQUMPC typew { typew ... }

```

```

1      5 6
/ snods n... /
      / mnods n... /
/ snodes ng... /
      / mnodes ng... /

```

type specify one or more degrees of freedom [§ 20.3.1 p. 320].

snods is a series of nodes indicating the slave vertices.

mnods is a series of nodes indicating the master vertices.

snodes is a series of slave nodes.

mnodes is a series of master nodes.

file.dat

```

'TYINGS'
NAME EQUALPRESSURE
EQUMPC PR
/ 16 26 /      / 1 11 /
/ 26 23 21 18 16 / / 11 9 6 4 1 /

```

20.3.5 Interconnection

An interconnection of tying connects one node (the slave), to two others (the masters). Interconnection tyings may be input in single- or multi-point format. An interconnection tying indicates that a slave node lies in between two master nodes.

20.3.5.1 Single-point Interconnection

Interconnections in single-point format must be specified in subtable **BETWEE** with a single slave node and two master nodes [Fig. 20.4]. By default DIANA assumes that the

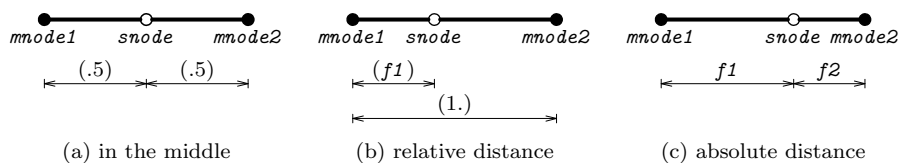


Figure 20.4: Single-point interconnection

slave node is just in the middle of the two masters [Fig. 20.4a]. You may overrule this assumption by specifying the position of the slave node explicitly [Fig. 20.4bc].

syntax

```
'TYINGS'
NAME names
BETWEE typew { typew ... }
1      5 6
snoden mnode1n mnode2n { f1r { f2r } }
```

type specify one or more degrees of freedom [§ 20.3.1 p. 320].

snode is the number of the slave node.

[$f1 = 0.5$]
($0 < f1 < 1$) *mnode1*, *mnode2* are the two master nodes. The optional factors *f1* and *f2* specify the position of the slave node relative to the master nodes. A single factor *f1* specifies the relative distance from the slave node to the first master node [Fig. 20.4b]. Two factors *f1* and *f2* specify the absolute distances from the slave node to the masters [Fig. 20.4c]. If $f1 + f2$ is not equal to the distance between the master nodes, then *snode* is positioned in between them proportionally to *f1* and *f2* to determine the linear dependency.

Mesh refinement. A typical application of interconnection tyings is *Mesh refinement* as shown in the input for the mesh of Figure 20.5.

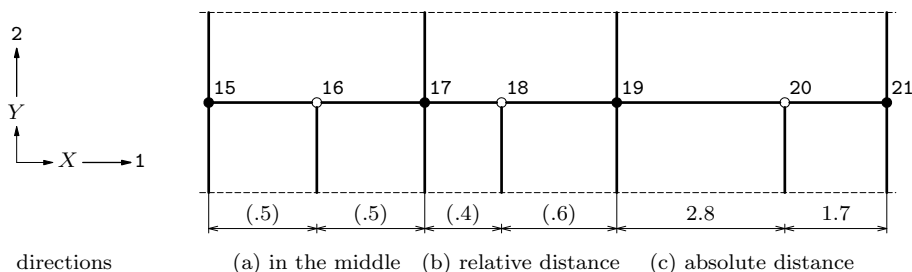


Figure 20.5: Mesh refinement

file.dat

```
'TYINGS'
NAME "Between potentials"
BETWEEN PO
: (a) just in the middle
```

```

16 15 17
: (b) relative distance
18 17 19 .4
: (c) absolute distance
20 19 21 2.8 1.7

```

20.3.5.2 Multi-point Interconnection

Interconnections in multi-point format [§ 20.3.2 p. 321] must be specified in subtable BTWMP with series of slave vertices and nodes, combined with series of master vertices and nodes [Fig. 20.6]. If there are n master nodes between two vertices and k slave nodes

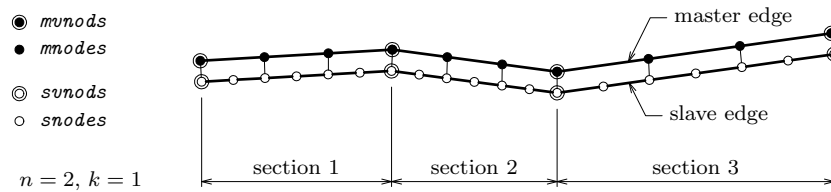


Figure 20.6: Multi-point interconnection

between each pair of master nodes then there must be $n + k \times (n + 1)$ nodes on each section of a slave edge. The values of n and k must be the same for each section.

syntax

```

'TYINGS'
NAME names
BTWMP typew {typew ...}
1      5 6
/ snodsn... /
      / mnodsn... /
/ snodesng... /
      / mnodesng... /

```

type specify one or more degrees of freedom [§ 20.3.1 p. 320].

snods is a series of nodes indicating the slave vertices.

mnods is a series of nodes indicating the master vertices.

snodes is a series of slave nodes.

mnodes is a series of master nodes.

file.dat

```

'TYINGS'
NAME "MPC between pressures"
BTWMP PR
/ 13 9 / / 26 28 /
/ 9 10 11 12 13 / / 26 27 28 /

```

20.3.6 General Connection

If none of the previous types of tying is applicable, you may specify a tying in its most general way: a linear relation between one slave degree of freedom and one or more master degrees of freedom. These types of tyings, called *general connections*, may be input in single- or multi-point format.

20.3.6.1 Single-point General Connection

General connections in single-point format must be specified in subtable FIX with a single slave node connected to one or more master nodes.¹

syntax

```
'TYINGS'
NAME names
FIX stypew
1      5 6
snoden
      mnoden mtypew {facr} ...
```

stype specify one slave degree of freedom [§ 20.3.1 p. 320].

snode is the number of the slave node.

mnode is a master node which must be followed by the description of its degree of freedom [§ 20.3.1 p. 320]. Factor *fac* is an optional multiplication factor: $slave = fac \times master$. You may specify multiple masters for each slave node, each one on a new line!

[*fac* = 1]

Single master example. The input data below presents two simple examples of general connections.

file.dat

```
'TYINGS'
NAME FIXEDTEMP
FIX TE
    52  53 TE -2.
FIX TE
    75  76 TE 2.
```

Multiple Masters Example. The input data below keeps the temperature difference between two pairs of nodes the same. In formula: $T_1 - T_2 = T_3 - T_4$.

file.dat

```
'TYINGS'
NAME "Multiple master nodes"
FIX TE
    1  2 TE 1.
    3 TE 1.
    4 TE -1.
```

20.3.6.2 Multi-point General Connection

General connections in multi-point format [§ 20.3.2 p. 321] must be specified in subtable FIXMPC with series of slave vertices and nodes, combined with series of master vertices and nodes [Fig. 20.7]. For each edge section, the amount of master nodes *n* must be equal to the amount of slave nodes, i.e., each slave node is connected to one master node.

¹The REMAKE option of Module INPUT [§3.3 p.50] always delivers single-point general connection tyings regardless of the original specification.

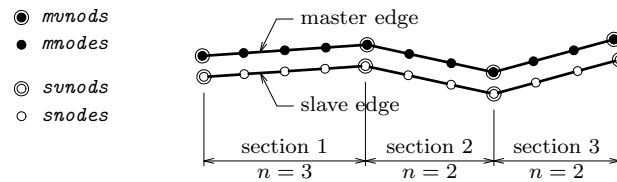


Figure 20.7: Multi-point general connection

syntax

```
'TYINGS'
FIXMPC stypew
1      5 6
/ svnodsn... /
           / mnodsn... /
/ snodesng... /
           / mnodesng... / mtypw {facr}
```

stype specify one slave degree of freedom [§ 20.3.1 p. 320].

svnods is a series of nodes indicating the slave vertices.

mnods is a series of nodes indicating the master vertices.

snodes is a series of slave nodes.

mnodes is a series of master nodes. Each series of master nodes must be followed by *mtyp*, the description of a degree of freedom [§ 20.3.1 p. 320]. DIANA will apply this degree of freedom for all master nodes in the series. Factor *fac* is an optional multiplication factor: $slave = fac \times master$. This factor will also be applied for all master nodes in the series. [fac = 1]

file.dat

```
'TYINGS'
NAME "Fixed MPC pressures"
FIXMPC PR
/ 16 26 /           / 1 11 /
/ 16 18 21 23 26 / / 1 4 6 9 11 / PR 1.0
```

Chapter 21

Heat Flow Analysis

This chapter is a user guide to heat flow analysis, which is a dedicated form of potential flow analysis.¹ In order to perform a heat flow analysis with DIANA you must take the following actions:

1. Invoke Module FILOS to initialize an analysis database [§ 3.2 p. 48].
2. Invoke Module INPUT to read the finite element model into the database [§ 3.3 p. 50]. See §21.1 for a description of input data required for heat flow analysis.
3. Either invoke Module HEATSS to perform a steady-state analysis [§ 21.2 p. 332], or Module HEATTR to perform a transient analysis [§ 21.3 p. 337].
4. You may get plots of the analysis results in the postprocessing working environment of iDIANA [Vol. iDIANA].

The minimum sets of commands for complete steady-state and transient heat flow analysis are shown below.

Steady-state

file.dcf

```
*FILOS  
INITIA  
*INPUT  
*HEATSS  
*END
```

Transient

file.dcf

```
*FILOS  
INITIA  
*INPUT  
*HEATTR  
*END
```

The next section gives some information on general aspects of modelling for a heat flow analysis.

21.1 General Modelling Aspects

This section describes the general aspects for heat flow analysis: shape and dimensions of the finite element mesh, conductivity, capacity, sources, fixed temperatures, boundary convection and radiation, initial temperature and flow fields, Finally we give an overview of special features: hydration heat, cooling pipes, solidification or evaporation, subsequent stress analysis, phased analysis, and a convective field.

¹The modules described in this chapter may also be applied for concentration flow analysis.

21.1.1 Meshing

The shape and dimensions of the analysis domain are modeled by a finite element mesh of *general flow continuum elements*: triangles, quadrilaterals, wedges, or bricks. Available are continuum elements for two-dimensional models, for axisymmetric models, and for three-dimensional models. See Volume *Element Library* for more information.

21.1.2 Conductivity and Capacity

The conductivity and capacity of continuum elements are specified in table 'MATERI'. Specification is possible as a constant property, and besides also as a multilinear function of temperature, time, or both. The conductivity may also be anisotropic. See Volume *Material Library* for more information.

21.1.3 Heat Sources and Fixed Temperatures

Sources and temperatures can be prescribed using table 'BOUNDA', and scaled as a multilinear function of time, using table 'TIMEBO' [§ 20.2.2 p. 318]. Point sources can be specified as boundary conditions via subtable NODAL of table 'BOUNDA' [§ 20.2.1 p. 317], while line, plane, and volume sources can be specified as boundary conditions for continuum elements via subtable ELEMEN [Vol. *Element Library*]. Temperature dependent sources (point, lines, planes) can be modeled by a nonlinear conduction coefficient in boundary elements or interface elements [Vol. *Material Library*]. Prescribed temperatures at nodes can be specified via table 'FIXPOT' [§ 20.1.1 p. 315] and subtable NODAL of table 'BOUNDA' [§ 20.2.1 p. 317].

21.1.4 Convection and Radiation at Boundaries

Convection or radiation at boundaries of the continuum model can be modeled by the convection coefficient of boundary elements or the conduction coefficient of interface elements [Vol. *Material Library*]. Line shaped elements are used at the faces of a two-dimensional continuum model and triangular or quadrilateral elements at the faces of a three-dimensional model. The convection, conduction and radiation coefficient are specified as material properties [Vol. *Material Library*]. Radiation may also be simulated by defining an appropriate temperature dependent convection coefficient.

In Volume *Analysis Examples* you may find two examples of NAFEMS [61] benchmarks run with DIANA: *nafet4* which shows a linear steady-state heat flow analysis with convection, and *nafet2* which shows a nonlinear steady-state heat flow with radiation.

21.1.5 Conduction and Convection Coefficient

Interface elements. Input of the conduction coefficient for interface elements is possible as a constant property or as a multilinear function of the temperature difference, time, or both, where the difference refers to the internal and the external temperature [Vol. *Material Library*]. The external temperature is defined by temperatures in the nodes of the external face. Prescribed external temperatures are input via subtable NODAL of table 'BOUNDA' [§ 20.2.1 p. 317].

Boundary elements. Input of the convection coefficient for boundary elements is possible as a constant property and as a multilinear function of boundary temperature and/or time [Vol. *Material Library*]. The external temperature is now specified in subtable ELEMEN of table 'BOUNDA' [Vol. *Element Library*].

Continuum elements. Input of the conduction coefficient for continuum elements is possible as a constant property and as a multilinear function of boundary temperature and/or time [Vol. *Material Library*]. Example *rodht* in Volume *Analysis Examples* shows a nonlinear transient heat flow using temperature dependent conductivity and capacity.

21.1.6 Temperature and Flow Fields

Table 'INIVAR' is used for input of a temperature field [§ 20.1.2 p. 316]. In a nonlinear steady-state analysis, a specified temperature field may be applied as a first estimate of the solution. In a transient analysis, it may be applied as an initial temperature field. For all heat flow analyses, DIANA will evaluate the model, conductivity and boundary conditions by default. In a transient analysis, capacity is also evaluated by default.

A steady-state analysis with Module HEATSS [§ 21.2 p. 332], or execution of several time steps in a transient analysis with Module HEATTR [§ 21.3 p. 337], results in computation of temperature and flow fields. Both these modules can be used for linear and iterative nonlinear analysis, i.e., for constant and for time/temperature dependent properties respectively. Transient analysis includes capacity contributions and initial temperatures, if they are available. The output of temperature and fluxes can be specified via the OUTPUT command block.

21.1.7 Special Features

21.1.7.1 Hydration Heat

Heat generation based on the degree of reaction is available for continuum elements, to model chemical reactions like cement hydration. Heat generation characteristics and dependency of conductivity and capacity on this degree of reaction, are specified as material properties with multilinear diagrams [Vol. *Material Library*]. Intended usage and initialization of the degree of reaction are specified as subcommands of a nonlinear analysis with Module HEATTR [§ 21.3.2.1 p. 340]. The output of resulting degrees of reaction during the transient nonlinear analysis can be specified via the OUTPUT command block. Example *hydrat* in Volume *Analysis Examples* shows a nonlinear transient analysis of heat generation caused by cement hydration.

21.1.7.2 Equivalent Age

As an alternative to the degree of reaction, DIANA can calculate the equivalent age as a maturity variable to quantify the progress of the hydration reaction. For such calculation you must specify special material characteristics [Vol. *Material Library*]. With a special command for Module HEATTR you may invoke an equivalent age calculation [§ 21.3.2.1 p. 340]. The output of the resulting new maturity variable during the nonlinear transient analysis can be obtained via the OUTPUT command block [§ 21.4.6 p. 346].

21.1.7.3 Cooling Pipes

Cooling pipes are used to reduce temperature gradients due to cement hydration. Input of fluid properties and the conduction coefficient of the pipe lining is specified as a material property [Vol. *Material Library*]. To model the heat convection implicitly or explicitly, DIANA offers a set of *cooling pipe elements* [Vol. *Element Library*]. These element types are usually combined with three-dimensional general flow continuum elements.

The implicit element L4HT has separate nodes for the fluid temperature. Usage results in nonsymmetric matrices which DIANA will setup automatically. A transient analysis results in computation of temperatures of both concrete and fluid [§ 21.3 p. 337].

The explicit element L2HT uses internal fluid temperatures, which are adapted by Module HEATTR after each time step. A pipe composed from explicit elements needs additional input from table 'COOLPI'. Internal temperatures can be obtained via the OUTPUT command block.

21.1.7.4 Solidification or Evaporation

A material generally consumes heat at the temperatures of solidification or evaporation. This heat consumption can be modeled inside a transient nonlinear analysis [§ 21.3.2.1 p. 340], by specification of a temperature dependent multilinear enthalpy function as a material property [Vol. *Material Library*]. DIANA derives a nonlinear capacity from this

function. Example `solidi` in *Volume Analysis Examples* shows a nonlinear transient heat flow analysis of solidification.

21.1.7.5 Subsequent Stress Analysis

DIANA's Module `NONLIN` allows you to couple a thermal stress analysis to a previous heat flow analysis. This is a so-called staggered analysis [Ch. 26], where Module `NONLIN` generates the input of load, temperature and maturity for a structural analysis from the results of the heat analysis. Part IV in this volume describes how to perform the subsequent nonlinear structural analysis.

21.1.7.6 Phased Analysis

DIANA's Module `PHASE` enables the combination of changing geometry and boundary conditions with transient potential flow analysis [Ch. 29]. The module is used, for example, to analyse cement hydration during phased concrete casting.

21.1.7.7 Convective Field

Temperature transport by for example groundwater flow can be analysed by application of a convective field, additional to the conduction. DIANA can derive the convective field from manual input, or by importing the flux field resulting from a preceding groundwater flow analysis on the same mesh [Vol. *Material Library*]. The conduction of the material is input as constant diffusivity. A flux multiplication factor, equal to the volumetric capacity of the fluid fraction is also material input. Example `conflow` in *Volume Geotechnical Analysis* shows an analysis of temperature transport by forced convection.

21.2 Steady-state Analysis

Module `HEATSS` is used to perform a steady-state heat flow analysis. Basically there are two types of steady-state analysis: *linear* if the material properties are constant or *nonlinear* if they are variable.

syntax

```
*HEATSS
[ MODEL [ OFF ] ... ]
[ EXECUT [ _____ ] ... ]
      LINEAR
      NONLIN
      OFF
[ OUTPUT [ OFF ] ... ] ...
```

`MODEL` to evaluate the finite element model [§ 21.2.1].

[`LINEAR`] `EXECUT` to execute the steady-state heat flow analysis. The `LINEAR` option asks for a linear steady-state analysis [§ 21.2.2], the `NONLIN` option for a nonlinear steady-state analysis [§ 21.2.3].

`OUTPUT` to specify analysis results for output [§ 21.4].

Default

file.dcf

```
*HEATSS
*END
```

If, like in the above commands, you only give the `*HEATSS` command, then DIANA will execute a complete linear steady-state analysis with the default output. This is equivalent to the following commands.

file.dcf

```

*HEATSS
BEGIN MODEL
  EVALUA
  ASSEMB
  MATRIX CONDUCT
  BOUNDA
END MODEL
BEGIN EXECUT
  LINEAR
  SOLVE
END EXECUT
BEGIN OUTPUT
  TEMPER
  FLUX LOCAL NODES
END OUTPUT
*END

```

21.2.1 Model Evaluation

The MODEL commands customize the evaluation of the finite element model prior to the actual analysis.

syntax

```

BEGIN MODEL
[ EVALUA [ OFF ] ... ]
[ ASSEMB [ OFF ] ]
[ MATRIX [ CONDUCT ] [ OFF ] ]
[ BOUNDA [ OFF ] ]
END MODEL

```

EVALUA to check and evaluate geometric and material properties for elements [§ 3.4 p. 51].

ASSEMB to assemble the elements, i.e., creating appropriate system degrees of freedom [§ 3.5 p. 54].

MATRIX CONDUCT to setup the element conductivity matrices. The **OFF** option switches off the creation of conductivity matrices. This may save computing time if the matrices are still available on the FILOS file and appropriate.

BOUNDA to convert boundary conditions to right-hand-side nodal flux vectors. The **OFF** option switches off the creation of right-hand-side nodal flux vectors, which may be useful if these are still available on the FILOS file.

21.2.2 Linear Steady-state Analysis

A linear steady-state analysis will do if temperature does not affect the material properties. Otherwise you must perform a nonlinear steady-state analysis [§ 21.2.3].

syntax

```

BEGIN EXECUT
[ LINEAR ]
[ SOLVE ... ]
END EXECUT

```

LINEAR will perform a linear steady-state heat flow analysis.

SOLVE customizes the settings for the solution method [Ch. 30 p. 421].

21.2.3 Nonlinear Steady-state Analysis

A nonlinear heat flow analysis is required if the material properties vary with the temperature. You may perform such an analysis via the `NONLIN` commands.

syntax

```

BEGIN EXECUT
  BEGIN NONLIN
    [ BOUNDA { _____ } ]
      CASE=casen
      FACTOR=facr
    [ ITERAT ... ]
    [ ESTIMA ... ]
  END NONLIN
[ SOLVE ... ]
END EXECUT

```

`NONLIN` will perform a nonlinear steady-state heat flow analysis.

`BOUNDA` specifies the boundary conditions.

Parameter `CASE=case` refers to a boundary case number in table 'BOUNDA' [§ 20.2 p. 317]. Default is the lowest available case number. Parameter `FACTOR=fac` specifies a multiplication factor.

[*fac*=1.0]

`ITERAT` specifies the iteration method to be applied in the nonlinear solution procedure [§ 21.2.3.1].

`ESTIMA` specifies a first estimation for the temperature field [§ 21.2.3.2].

`SOLVE` customizes the settings for the solution method [Ch. 30 p. 421].

Default

file.dcf

```

*HEATSS
[ commands ]
BEGIN EXECUT
  NONLIN
END EXECUT
[ commands ]
*END

```

If, like in the above commands, you give a single `NONLIN` command, then DIANA will perform a nonlinear steady-state analysis without a first estimation of the temperature field, i.e., a zero temperature field is the default, and with non-scaled boundary conditions from case 1. The default iteration procedure will be applied. This is equivalent to the following commands.

file.dcf

```

*HEATSS
[ commands ]
BEGIN EXECUT
  BEGIN NONLIN
    BOUNDA CASE=lowest FACTOR=1.0
    ITERAT
    ESTIMA OFF
  END NONLIN
  SOLVE
END EXECUT
[ commands ]
*END

```

21.2.3.1 Iteration Method

With the ITERAT commands you may customize the iteration process that DIANA will apply in the nonlinear solution procedure. See also § 55.1.3.2 on page 629 for some background theory on these iteration methods.

syntax

```

BEGIN ITERAT
[ MAXITE= $mi_n$  ]
[ METHOD [ NEWTON ] _____ ]
                        REGULA
                        MODIFI
[ [ CONVER [ TEMPER ] ] TOLCON= $eps_r$  ]
END ITERAT

```

MAXITE= mi is the maximum number of iterations. [$mi=5$]

METHOD specifies the iteration method: NEWTON indicates a Newton–Raphson iteration scheme (the only option), i.e., the conductivity matrix will be updated.

MODIFI invokes the Modified Newton–Raphson method.

REGULA invokes the Regular Newton–Raphson method (the default). [REGULA]

CONVER specifies the convergence criterion for the iteration process: TEMPER indicates a convergence criterion on the norm of the temperature field (the only option).

Parameter TOLCON= eps is the tolerance ϵ on the reference norm. When the norm of the incremental temperature field has become less than $\epsilon \times$ the reference norm DIANA assumes sufficient accuracy and stops the iteration process [Eq. (55.32) p. 629]. [$\epsilon = 10^{-6}$]

Default

file.dcf

```

*HEATSS
[ commands ]
BEGIN EXECUT
  BEGIN NONLIN
    [ commands ]
    ITERAT
    [ commands ]
  END NONLIN
END EXECUT
[ commands ]
*END

```

If, like in the above commands, you give a single ITERAT command, then DIANA will apply a Regular Newton–Raphson iteration scheme with at most five iterations. As convergence criterion, the norm of the temperatures will be applied with a tolerance of 10^{-6} . This is equivalent to the following commands.

file.dcf

```

*HEATSS
[ commands ]
BEGIN EXECUT
  BEGIN NONLIN
    [ commands ]
    BEGIN ITERAT
      MAXITE=5
      METHOD NEWTON REGULA
      CONVER TEMPER TOLCON=1.E-6
    END ITERAT
  [ commands ]

```

```

END NONLIN
END EXECUT
[ commands ]
*END

```

21.2.3.2 First Estimation of Temperature

With the **ESTIMA** commands you may specify a temperature field which DIANA will apply as a first estimation of the solution.

syntax

```

BEGIN ESTIMA
[ OFF ]
[ TEMPER [OFF] [ _____ ]
                        INPUT { _____ }
                              FIELD=fieldn
                              FACTOR=facr
                        CALCUL { _____ }
                              CASE=casen
                              FACTOR=facr
]
END ESTIMA

```

TEMPER indicates that the estimation is for a temperature field (the only possibility).

There are two methods to specify the temperature field. For both options parameter **FACTOR=*fac*** is an optional multiplication factor.

[*fac*=1]

INPUT indicates a start with temperatures as specified in table 'INIVAR' [§ 20.1.2 p. 316] (the default). Parameter **FIELD=*field*** refers to a field number in this table. Default is the lowest available field number.

[INPUT]

CALCUL indicates a start with temperatures as calculated in a steady-state analysis. If temperatures are available from a previous steady-state analysis, these temperatures are used. Otherwise a linear steady-state analysis is performed. Parameter **CASE=*case*** refers to a boundary case number in table 'BOUNDAR' [§ 20.2 p. 317]. Default is the lowest available case number.

Default

file.dcf

```

*HEATSS
[ commands ]
BEGIN EXECUT
  BEGIN NONLIN
    [ commands ]
    ESTIMA
    [ commands ]
  END NONLIN
END EXECUT
[ commands ]
*END

```

If, like in the above commands, you give a single **ESTIMA** command, then DIANA will apply the potentials of the lowest field number in input table 'INIVAR' as a first estimation of the temperature field. This is equivalent to the following commands.

file.dcf

```

*HEATSS
[ commands ]
BEGIN EXECUT
  BEGIN NONLIN
    [ commands ]
    BEGIN ESTIMA
      TEMPER INPUT FIELD=lowest FACTOR=1.0
    END ESTIMA
    [ commands ]
  END NONLIN
END EXECUT
[ commands ]
*END

```

21.3 Transient Analysis

Module HEATTR is used to perform a transient heat flow analysis. In order to perform such analysis, you must additionally specify the time dependency of boundary conditions in table 'TIMEBO' [§ 20.2.2].

After the preliminary tasks like model evaluation and setup of the system of equations, a transient analysis basically involves two steps: (1) initiation of a linear or nonlinear transient analysis, (2) solution of the equations for the specified time steps. You may perform these steps in one DIANA job or in various subsequent jobs.

syntax

```

*HEATTR
[ MODEL [ OFF ] ... ]
[ INITIA [ OFF ] ... ]
[ EXECUT [ OFF ] ... ]
[ OUTPUT [ OFF ] ... ] ...

```

MODEL to evaluate the finite element model [§ 21.3.1].

INITIA to initiate a transient heat flow analysis [§ 21.3.2].

EXECUT to execute time steps in the transient heat flow analysis [§ 21.3.3].

OUTPUT to specify analysis results for output [§ 21.4].

21.3.1 Model Evaluation

The **MODEL** commands customize the evaluation of the finite element model prior to the actual analysis.

syntax

```

BEGIN MODEL
[ EVALUA [ OFF ] ... ]
[ ASSEMB [ OFF ] ]
[ MATRIX [ OFF ] ... ]
[ BOUNDA [ OFF ] ]
END MODEL

```

EVALUA to check and evaluate geometric and material properties for elements [§ 3.4 p. 51].

ASSEMB to assemble the elements, i.e., creating appropriate system degrees of freedom.

MATRIX to setup the element conductivity and capacity matrices [§ 21.3.1.1].

BOUNDA to convert boundary conditions to right-hand-side nodal flux vectors. The OFF option switches off the creation of right-hand-side nodal flux vectors, which may be useful if these are still available on the FILOS file.

21.3.1.1 Conductivity and Capacity Matrices

By default, DIANA will set up the element conductivity and capacity matrices each time that you invoke Module HEATTR. You may explicitly ask DIANA to setup the matrices via MATRIX commands.

syntax

```
BEGIN MATRIX
[ OFF ]
[ CONduc [ OFF ] ]
[ CAPACI [ OFF ] [ _____ ] ]
                        CONSIS
                        LUMPED
END MATRIX
```

OFF switches off the creation of matrices. This may save computing time if these are still available on the FILOS file and appropriate.

CONduc sets up the element conductivity matrices.

CAPACI sets up the element capacity matrices. You may specify the type of matrices with [CONsis] CONSIS for consistent matrices (the default), or with LUMPED for lumped matrices.

Default

file.dcf

```
*HEATTR
BEGIN MODEL
  MATRIX
END MODEL
[ commands ]
*END
```

If, like in the above commands, you give a single MATRIX command, then DIANA will setup the conductivity matrices and the consistent capacity matrices, which would also occur if you did not give the MATRIX command at all. This is equivalent to the following commands.

file.dcf

```
*HEATTR
BEGIN MODEL
  BEGIN MATRIX
    CONduc
    CAPACI CONSIS
  END MATRIX
END MODEL
[ commands ]
*END
```

21.3.2 Initiate Transient Analysis

The INITIA commands initiate the transient analysis. The main purpose of this task is to evaluate the initial state of the model, i.e., prior to the execution of the first time step.

syntax

```

BEGIN INITIA
[ TEMPER [ OFF ] [ _____ ]
                        INPUT { _____ }
                              FIELD=fieldn
                              FACTOR=facr
                        CALCUL { _____ }
                              CASE=casen
                              FACTOR=facr
[ SOLVE ... ]
[ TIME=stimer ]
[ _____ ]
  LINEAR
  NONLIN ...
END INITIA

```

TEMPER indicates an initial temperature field. There are two methods to specify the temperature field. For both options parameter FACTOR=*fac* is an optional multiplication factor. [INPUT]
[*fac*=1]

INPUT indicates a start with temperatures as specified in table 'INIVAR' [§ 20.1.2 p. 316]. Parameter FIELD=*field* refers to a field number in this table. Default is the lowest available field number.

CALCUL indicates a start with temperatures as calculated in a steady-state analysis. If temperatures are available from a previous steady-state analysis, these temperatures are used. For a nonlinear transient analysis, DIANA first checks if a nonlinear steady-state solution with matching boundary case number and load factor is available. Secondly, DIANA looks for an existing linear steady-state solution and scales it if necessary. Otherwise a linear steady-state analysis is performed. Parameter CASE=*case* refers to a boundary case number in table 'BOUNDATA' [§ 20.2 p. 317]. Default is the lowest available case number.

SOLVE customizes the settings for the solution method to be applied for the linear steady-state analysis [Ch. 30 p. 421].

TIME *stime* is a user specified starting time. [*stime*=0]

LINEAR initiates a linear analysis, i.e., with constant material properties. [LINEAR]

NONLIN initiates a nonlinear analysis, i.e., with time or temperature dependent material properties [§ 21.3.2.1].

Default

file.dcf

```

*HEATTR
[ commands ]
INITIA
[ commands ]
*END

```

If, like in the above commands, you give a single INITIA command, then DIANA will initiate a linear analysis, i.e., with constant material properties. Note that no initial temperature field will be apply by default. This is equivalent to the following commands.

file.dcf

```

*HEATTR
[ commands ]
BEGIN INITIA
  LINEAR
END INITIA
[ commands ]
*END

```

21.3.2.1 Nonlinear Transient Heat Flow Analysis

A nonlinear heat flow analysis is required if the material properties vary with the temperature or the time. You may initiate such an analysis via the **NONLIN** commands.

syntax

```

BEGIN NONLIN
[ HYDRAT [ _____ ] ]
      DGRINI=r0r
      OFF
[ EQUAGE [ _____ ] ]
      EQAINI=te0r
      OFF
END NONLIN

```

[$r_0 = 0.01$] **HYDRAT** initiates a nonlinear transient analysis with heat generation due to cement hydration.
 ($0 \leq r_0 \leq 1$) Parameter **DGRINI= r_0** specifies the initial degree of reaction r_0 . The **OFF** option suppresses the application of heat generation due to hydration, as if you omitted the **HYDRAT** command.

[$t_{eq0} = 0$] **EQUAGE** invokes the calculation of the equivalent age t_{eq} . Parameter **EQAINI= $te0$** specifies the initial equivalent age t_{eq0} . The **OFF** option suppresses the calculation of the equivalent age, as if you omitted the **EQUAGE** command.

21.3.3 Time Steps Execution

With the **EXECUT** commands you ask DIANA to execute time steps in a transient heat flow analysis. A command file may contain one or more **EXECUT** blocks. Details of various commands are given in the referred sections.

syntax

```

BEGIN EXECUT
[ OFF ]
[ ALPHA=alphar ]
[ SIZES sizesr... ]
[ NONLIN ... ]
[ SOLVE ... ]
[ SAVE [ STEPS _____ ] ]
      stepsn...
      ALL
      LAST
      NONE
END EXECUT

```

($0 \leq \alpha \leq 1$) **ALPHA** *alpha* is the time integration parameter α . Usual methods are: Euler forward ($\alpha = 0$), Crank–Nicolson ($\alpha = \frac{1}{2}$), Galerkin ($\alpha = \frac{2}{3}$) and Euler backward ($\alpha = 1$) which is the default. See § 55.1.3.1 on page 628 for some background theory on time integration.

SIZES *sizes* are explicitly specified time step sizes, i.e., time increments. The number of values specifies the number of steps to be executed. [SIZES 1.0]

NONLIN specifies options for nonlinear transient analysis [§ 21.3.3.1].

SOLVE customizes the settings for the solution method [Ch. 30 p. 421].

SAVE specifies the steps for which analysis results must be saved for a subsequent structural analysis in a staggered flow–stress analysis [Ch. 26]. By default, in a model for staggered flow–stress analysis, DIANA will save the results for all executed steps. [ALL]

STEPS *steps* indicates the steps for which the analysis results must be saved. Alternatively you may save **ALL** steps, only the **LAST** step, or **NONE** of the steps. The **NONE** option drops the possibility of a subsequent structural analysis!

Default

file.dcf

```
*HEATTR
[ commands ]
EXECUT
*END
```

Due to these commands DIANA will execute a time step with a factor of 1, applying the default solution procedure, and giving the default output of analysis results. The same would occur if you had given the following commands.

file.dcf

```
*HEATTR
[ commands ]
BEGIN EXECUT
  ALPHA=1.0
  SIZES 1.0
  SOLVE
END EXECUT
BEGIN OUTPUT
  TEMPER
  FLUX LOCAL NODES
  REACTI NODES
  INTTMP NODES
END OUTPUT
*END
```

Degrees of reaction (**REACTI**) will only be output in case of hydration heat analysis [§ 21.4.5 p. 345]. Internal temperatures (**INTTMP**) will only be output in case of nonlinear transient analysis with cooling pipe elements [§ 21.4.7 p. 346].

21.3.3.1 Nonlinear Analysis Options

Via the **NONLIN** commands you may customize a nonlinear transient analysis, provided that such analysis has been initiated previously [§ 21.3.2.1 p. 340].

syntax

```
BEGIN NONLIN
[ BEGIN ITERAT
  [ MAXITE= $mi_n$  ]
  [ METHOD [ NEWTON ] _____ ]
  REGULA
```

```

MODIFI
[ CONVER [ TEMPER ] [ TOLCON=epsr ] ]
END ITERAT ]
[ HYDRAT [ _____ ] ]
STEP
ITERAT
END NONLIN

```

ITERAT specifies the iteration method to be applied in the nonlinear solution procedure. See also § 55.1.3.2 on page 629 for some background theory on these iteration methods.

[*mi*=5] MAXITE=*mi* is the maximum number of iterations.

METHOD specifies the iteration method: NEWTON for a Newton–Raphson iteration scheme (the only option), i.e., the conductivity matrix will be updated.

MODIFI invokes the Modified Newton–Raphson method.

[REGULA] REGULA invokes the Regular Newton–Raphson method (the default).

CONVER specifies the convergence criterion for the iteration process: TEMPER indicates a convergence criterion on the norm of the temperature field (the only option).

[$\epsilon = 10^{-6}$] Parameter TOLCON=*eps* is the tolerance ϵ on the reference norm. When the norm of the incremental temperature field has become less than $\epsilon \times$ the reference norm DIANA assumes sufficient accuracy and stops the iteration process [Eq. (55.32) p. 629].

HYDRAT indicates when the degree of reaction, and thus the element heat production, must be updated. This option only applies for hydration heat analysis.

[STEP] STEP for update after each time step (the default).

ITERAT for update within the time step after each iteration.

Default

file.dcf

```

*HEATTR
[ commands ]
BEGIN EXECUT
[ commands ]
NONLIN
END EXECUT
*END

```

If, like in the above example, you only give a single NONLIN command then DIANA will perform a nonlinear analysis applying a Regular Newton–Raphson iteration scheme with at most five iterations. As convergence criterion, the norm of the temperatures will be applied with a tolerance of 10^{-6} . In case of an hydration heat analysis, the heat production will be updated after each time step. This is equivalent to the following commands.

file.dcf

```

*HEATTR
[ commands ]
BEGIN EXECUT
BEGIN NONLIN
[ commands ]
BEGIN ITERAT
MAXITE=5
METHOD NEWTON REGULA
CONVER TEMPER TOLCON=1.E-6
END ITERAT

```



```

    HYDRAT STEP
  END NONLIN
END EXECUT
*END

```

21.4 Output of Analysis Results

You can get output of analysis results from a heat flow analysis via the `OUTPUT` commands. For the general, analysis type independent options for output of analysis results see § 3.6 on page 55.

syntax

```

BEGIN OUTPUT [devicew] [outoptw...] [params]
[ OFF ]
[ BEGIN SELECT
  [ ... ]
  [ BOUNDA            {        } ]
             casesn... MIN
             ALL      MAX
  [ STEPS ... ]
END SELECT ]
[ LAYOUT ... ]
itemw ...
TEMPER
FLUX
FLOW
REACTI
EQUAGE
INTTMP
END OUTPUT

```

`SELECT` command block to customize the batch output.

... for model selection see § 3.6.2 on page 59.

`BOUNDA` specifies a selection of boundary cases in linear steady-state analysis: *cases* is a series of case numbers referring to input table 'BOUNDA' [§ 20.2 p. 317], `ALL` selects all four boundary cases.

`MIN` selects the minimum value of the (selected) boundary cases to be output, `MAX` selects the maximum value.

`STEPS` selects time steps for output of transient heat flow analysis [§ 21.4.1 p. 344].

`LAYOUT` optional commands to customize the layout and style of tabular output [§ 3.6.4 p. 64].

item is the name of the analysis result to be output.

`TEMPER` for nodal temperatures [§ 21.4.2 p. 344].

`FLUX` for element fluxes [§ 21.4.3 p. 344].

`FLOW` for nodal flows [§ 21.4.4 p. 345].

`REACTI` for degrees of reaction [§ 21.4.5 p. 345], only available in hydration heat analysis.

`EQUAGE` for equivalent age [§ 21.4.6 p. 346], only available in hydration heat analysis.

`INTTMP` for internal temperature of cooling pipe elements [§ 21.4.7 p. 346], only available in transient nonlinear heat flow analysis with cooling pipe elements.

21.4.1 Step Selection

The **STEPS** command selects time steps for output of analysis results of a transient heat flow analysis.

syntax

```
STEPS [ _____ ] { _____ }
      stepsn...    MIN
      ALL           MAX
      LAST
```

steps are numbers of selected steps. These numbers are relative to the steps of the preceding **EXECUT** command block.

[ALL] **ALL** will produce output for all steps that are executed (the default).

LAST selects the final step.

DIANA will output the extreme values that occurred up to and including the selected steps if you specify one of the options:

MIN for the minimum values,

MAX for the maximum values.

*For step-wise analyses output of extreme values cannot be used in combination with the option **LAYOUT COMBIN** [§ 3.6.4.1 p. 65] to assemble various results in one table for tabular output.*

21.4.2 Temperatures

The temperature field is the solution vector of the system of equations for the heat flow analysis. Temperatures are scalar values calculated in the nodes, component selection is not applicable.

syntax

```
TEMPER { optiw }
```

TEMPER gives the temperatures T .

opti are additional options [§ 3.6.1 p. 58].

<i>item</i>	
TEMPER	PTE
	T

21.4.3 Fluxes

In a heat flow analysis, fluxes are calculated for elements. They can be output in the integration points, element center point, or in the element nodes. **DIANA** also writes the integrated fluxes over groups of boundary elements to the standard output file *file.out*.

syntax

```
FLUX [ operw ] { compw } [ locaw ] { optiw }
      LOCAL          INTPNT
      GLOBAL          NODES
      BOUNDA          CENTER
```

FLUX gives the fluxes q [Eq. (55.2) p. 627].

[GLOBAL] *oper* specifies an operation to be performed on the fluxes.

LOCAL transforms fluxes in continuum elements to local *xyz* directions.

GLOBAL transforms fluxes in continuum elements to global *XYZ* directions.

BOUNDA gives the total discharge Q for boundary elements. This rate of flow is a scalar value, therefore component selection is not applicable.

loca specifies the location of the fluxes to be output [§ 3.6.1 p. 58].

[NODES]

opti are additional options [§ 3.6.1 p. 58].

<i>item</i>	<i>oper</i>	<i>comp</i> ...		
		X	Y	Z
FLUX	LOCAL	FLx	FLy	FLz
		q_x	q_y	q_z
FLUX	GLOBAL	FLX	FLY	FLZ
		q_X	q_Y	q_Z
FLUX	BOUNDA	q		
		Q		

21.4.4 Flows

In a heat transfer analysis nodal flows are calculated. Nodal flows are scalar values, component selection is not applicable.

syntax

```
FLOW [ typew ] { optiw }
```

```
  REACTI
```

```
  RESIDU
```

```
  EXTERN
```

FLOW gives the nodal flows.

type specifies the type of the nodal flow.

[REACTI]

REACTI for the reaction flow Q in all nodes with fixed temperature.

RESIDU for the residual flow ΔQ , also called the out-of-balance flow. This is defined as the difference between the externally applied flow Q and the internal resistance flow.

EXTERN for the externally applied flow Q .

opti are additional options [§ 3.6.1 p. 58].

<i>item</i>	<i>type</i>	
FLOW	REACTI	FLB
		Q
FLOW	RESIDU	FLR
		ΔQ
FLOW	EXTERN	FLE
		Q

21.4.5 Degrees of Reaction

In hydration analysis DIANA calculates the degrees of reaction r [§ 21.1.7.1 p. 331], which can be output in the element nodes and integration points.

syntax

```

REACTI [ locaw ] { optiw }
        INTPNT
        NODES
        CENTER

```

REACTI gives the degrees of reaction r .

[NODES] *loca* specifies the location of the degrees of reaction to be output [§ 3.6.1 p. 58].

opti are additional options [§ 3.6.1 p. 58].

<i>item</i>	
REACTI	DGR
	r

21.4.6 Equivalent Age

In hydration heat analysis DIANA can calculate the equivalent age t_{eq} [§ 21.3.2.1 p. 340]. This result can be output in the element integration points or in the element nodes.

syntax

```

EQUAGE [ locaw ] { optiw }
        INTPNT
        NODES
        CENTER

```

EQUAGE gives the equivalent age t_{eq} .

[NODES] *loca* specifies the location of the equivalent age to be output [§ 3.6.1 p. 58].

opti are additional options [§ 3.6.1 p. 58].

<i>item</i>	
EQUAGE	EqA
	t_{eq}

21.4.7 Internal Temperature of Cooling Pipe

The internal fluid temperatures of cooling pipe elements can be output in the element nodes.

syntax

```

INTTMP [ locaw ] { optiw }
        NODES

```

INTTMP gives the internal temperature T_{int} .

[NODES] *loca* specifies the location of the temperature to be output [§ 3.6.1 p. 58].

opti are additional options [§ 3.6.1 p. 58].

<i>item</i>	
INTTMP	PTE
	T_{int}

Chapter 22

Groundwater Flow Analysis

With DIANA you can perform two types of groundwater flow analysis: detailed groundwater flow with partially saturated areas [§ 22.1], and regional groundwater flow in aquifers [§ 22.2].

22.1 Detailed Groundwater Flow

For a detailed analysis of horizontal and vertical components of groundwater flow with partly saturated areas DIANA offers some special nonlinear groundwater flow options. First we describe the general topics: shape and dimensions, hydraulic conductivity and storativity, sources and sinks, and seepage faces. Finally we give an overview of special features: turbulence, contamination or thermal transport, subsequent stress analysis, and phased analysis.

22.1.1 Meshing

The shape and dimensions of the analysis domain are modeled by a mesh of continuum *groundwater flow elements*: triangles, quadrilaterals, wedges, and bricks. Available are continuum elements for two-dimensional models, for axisymmetric models, and for three-dimensional models. See Volume *Element Library* for details. DIANA assumes the direction of gravity in global $-Y$ direction for two-dimensional models, and in global $-Z$ direction for three-dimensional models.

22.1.2 Hydraulic Conductivity and Storativity

The hydraulic conductivity and storativity of continuum elements are derived from input of material properties. The elastic storativity and saturated hydraulic conductivity are specified as constants. The hydraulic conductivity can be isotropic or anisotropic.

Nonlinear hydraulic conductivity and storage are applied in partially saturated areas. DIANA mostly derives the nonlinear phreatic storativity from the porosity and a pressure dependent saturation, which is specified by the user as a multilinear function. It is also possible to specify directly a pressure dependent phreatic storativity. A relative hydraulic conductivity can be specified as a multilinear function of the saturation or the pressure. See Volume *Material Library* for details. See also example `sandcol` in Volume *Geotechnical Analysis*, a benchmark of a transient nonlinear analysis of phreatic level rise in a sand column.

22.1.3 Sources and Sinks

Sources and sinks are input via table 'BOUNDA'. Their values can be scaled as a multilinear function of time, using table 'TIMEBO' [§ 20.2.2 p. 318]. Point sources can be specified as boundary conditions via subtable NODAL of table 'BOUNDA' [§ 20.2.1 p. 317], while line, plane, and volume sources can be specified as boundary conditions for continuum elements via subtable ELEMEN [Vol. *Element Library*]. Sources which depend on the hydraulic head

(point, lines, planes) can be modeled by a nonlinear hydraulic conduction coefficient in boundary elements or interface elements [Vol. *Material Library*]. Prescribed hydraulic heads at nodes can be specified via table 'FIXHEA' [§ 20.1.1 p. 315] and subtable NODAL of table 'BOUNDA' [§ 20.2.1 p. 317].

22.1.4 Seepage Faces

At a seepage face the water flows out above the external phreatic level. DIANA has special boundary elements available to model seepage faces [Vol. *Element Library*]. Line shaped elements are used at the faces of a two-dimensional continuum model and plane shaped elements at the faces of a three-dimensional model. The external phreatic level is specified in table 'BOUNDA' [§ 20.2 p. 317]. DIANA imposes a zero pressure condition at the seepage face by use of a penalty hydraulic conduction coefficient. The default value may be adapted via material input [Vol. *Material Library*]. See also example *seepfce* in Volume *Geotechnical Analysis*, a benchmark of nonlinear steady-state analysis with partly unsaturated flow and a seepage face.

22.1.5 Hydraulic Head and Flow Fields

Table 'INIVAR' is used for input of an initial hydraulic head [§ 20.1.2 p. 316]. DIANA will evaluate the model, hydraulic conductivity and boundary conditions by default.

A subsequent steady-state analysis with Module GROWSS [§ 22.3 p. 350], or execution of several time steps with module GROWTR [§ 22.4 p. 355], results in computation of fields for hydraulic head and flux. Partly saturated parts always require an iterative nonlinear analysis [§ 22.3.3 p. 352] [§ 22.4.2 p. 357]. In transient analysis, storage contributions and initial fields are included.

The output of hydraulic heads and fluxes can be obtained via commands in the OUTPUT block. Output of the pressure component of the hydraulic head and of discharges is also possible.

22.1.6 Special Features

22.1.6.1 Turbulence

Turbulence can be included in a nonlinear analysis by input of the *Forchheimer* constants as a material property [Vol. *Material Library*].

22.1.6.2 Contamination Transport

Contamination transport is analysed similar to a temperature analysis with an imported convective field from a groundwater flow analysis [§ 21.1.7.7]. Example *conflow* in Volume *Geotechnical Analysis* shows a steady-state analysis with convection of contaminants.

22.1.6.3 Stability and Consolidation

DIANA's Module NONLIN allows you to analyse for example the influence of groundwater flow on the mechanical stability, by coupling a separate groundwater flow analysis to a subsequent effective stress analysis, a so-called staggered analysis [Ch. 26]. The module generates structural loads in the input table 'LOADS' from the computed hydraulic heads.

DIANA's Module NONLIN also allows you to analyse for example consolidation of soil, by simultaneous solution of pore fluid flow and stress, a so-called mixture analysis [Ch. 27].

22.1.6.4 Phased Analysis

DIANA's Module PHASE enables the combination of changing geometry and boundary conditions with transient groundwater flow analysis [Ch. 29]. The module is used for example to predict the transient changes in groundwater flow by introduction of sheet pilings, sinks, excavation etc.

22.2 Regional Groundwater Flow

A two-dimensional regional analysis of groundwater flow can predict the horizontal flow through aquifers, using the Dupuit assumption. These aquifers can be interacting by interjacent resistance layers (aquicludes). Nonlinear transmissivity and phreatic storage is introduced in unconfined aquifers. The direction of gravity is always assumed in global $-Z$ direction, e.g. normal to the plane of the aquifer(s).

In Volume *Geotechnical Analysis* you may find two examples of aquifer analysis. **Caquif** shows a linear steady-state analysis in a confined aquifer. **Aquifs** shows a nonlinear steady-and transient analysis of a combination of a semi-confined aquifer, an intermediate resistance layer and an unconfined aquifer.

22.2.1 Meshing

Aquifers are modeled by two-dimensional quadrilateral and triangular aquifer elements. For these elements you must specify the top and bottom level. Interjacent resistance layers are modeled by *flow interface elements*. See Volume *Element Library* for details.

22.2.2 Hydraulic Conductivity and Storativity

The saturated hydraulic conductivity, the elastic storativity and the porosity (phreatic storativity) are input for aquifers. A hydraulic conduction coefficient is input for the resistance layers. Input is specified as material properties, see Volume *Material Library* for details.

22.2.3 Sources and Sinks

Sources and sinks are input via table 'BOUNDA'. Their values can be scaled as a multilinear function of time, via 'TIMEBO' [§ 20.2.2 p. 318]. Point sources can be specified as boundary conditions via subtable NODAL of table 'BOUNDA' [§ 20.2.1 p. 317], while line and plane sources can be specified on the aquifer elements via subtable ELEMEN. Prescribed hydraulic heads at nodes can be specified via table 'FIXPOT' [§ 20.1.1 p. 315] and subtable NODAL of table 'BOUNDA' [§ 20.2.1 p. 317].

22.2.4 Hydraulic Head and Flow Fields

Table 'INIVAR' is used for input of an initial hydraulic head [§ 20.1.2 p. 316]. DIANA will evaluate the model, hydraulic conductivity and boundary conditions by default. A subsequent steady-state analysis with Module GROWSS [§ 22.3], or execution of several time steps with module GROWTR [§ 22.4 p. 355], results in computation of fields for hydraulic head and flux. Unconfined aquifers always require an iterative nonlinear analysis [§ 22.3.3 p. 352] [§ 22.4.2 p. 357]. In transient analysis are storage contributions and initial fields included. The output of hydraulic heads and fluxes can be obtained via the OUTPUT command block.

22.2.5 Special Features

22.2.5.1 Contamination Transport

Contamination transport is analysed similar to a temperature analysis with an imported convective field from a groundwater flow analysis.

22.2.5.2 Phased Analysis

DIANA's Module PHASE enables the combination of changing geometry and boundary conditions with transient potential flow analysis. The module is used, for example, to model the initial effects to groundwater flow due to introduction of sheet piling, sinks, excavation etc. [Ch. 29].

22.3 Steady-state Analysis

Module GROWSS is used to perform a steady-state groundwater flow analysis. Basically there are two types of steady-state analysis: *linear* if the material properties are constant or *nonlinear* if they are variable.

syntax

```
*GROWSS
[ MODEL [ OFF ] ... ]
[ EXECUTE [ _____ ] ... ]
      LINEAR
      NONLIN
      OFF
[ OUTPUT [ OFF ] ... ] ...
```

MODEL to evaluate the finite element model [§ 22.3.1].

[LINEAR] EXECUT to execute the steady-state groundwater flow analysis. The LINEAR option asks for a linear steady-state analysis [§ 22.3.2], the NONLIN option for a nonlinear steady-state analysis [§ 22.3.3].

OUTPUT to specify analysis results for output [§ 22.5].

Default

file.dcf

```
*GROWSS
*END
```

If, like in the above commands, you only give the *GROWSS command, then DIANA will execute a complete linear steady-state analysis with the default output. This is equivalent to the following commands.

file.dcf

```
*GROWSS
BEGIN MODEL
  EVALUA
  ASSEMB
  MATRIX CONDUCT
  BOUNDA
END MODEL
BEGIN EXECUT
  LINEAR
  SOLVE
END EXECUT
BEGIN OUTPUT
  HEAD TOTAL
  HEAD PRESSU
  FLUX LOCAL NODES
END OUTPUT
*END
```

22.3.1 Model Evaluation

The MODEL commands customize the evaluation of the finite element model prior to the actual analysis.

syntax

```

BEGIN MODEL
[ EVALUA [ OFF ] ... ]
[ ASSEMB [ OFF ] ]
[ MATRIX [ OFF ] ... ]
[ BOUNDA [ OFF ] ]
END MODEL

```

EVALUA to check and evaluate geometric and material properties for elements [§ 3.4 p. 51].

ASSEMB to assemble the elements, i.e., creating appropriate system degrees of freedom [§ 3.5 p. 54].

MATRIX to setup the element conductivity matrices [§ 22.3.1.1].

BOUNDA to convert boundary conditions to right-hand-side nodal flux vectors. The **OFF** option switches off the creation of right-hand-side nodal flux vectors, which may be useful if these are still available on the FILOS file.

22.3.1.1 Conductivity Matrices

By default, DIANA will set up the element conductivity matrices each time that you invoke Module GROWSS. You may customize the setup of conductivity matrices via the **MATRIX** commands.

syntax

```

BEGIN MATRIX
  BEGIN CONDUCT
    [ OFF ]
    [ CONVEC [ OFF ] {                      } ]
                        CASE=casen
                        FACTOR=facr
  END CONDUCT
END MATRIX

```

CONDUCT will set up the conductivity matrices. The **OFF** option switches off the creation of conductivity matrices, this may save computing time if the matrices are still available on the FILOS file and appropriate.

CONVEC will set up the full nonsymmetric conductivity matrices for convective fields as calculated, and stored on the FILOS file, during a previous potential flow analysis.¹ The **OFF** option switches off the creation of these special conductivity matrices.

CASE=case specifies the case number to be selected from the calculated element fluxes if the previous calculation was linear steady-state. The case number refers to the case number in table 'BOUNDA' [§ 20.2 p. 317]. Default is the lowest available case number.

FACTOR=fac takes into account the components of the rate of flow of the previous groundwater flow analysis. In this case **fac** is the multiplication factor for the calculated flow velocity.

[*fac*=1.0]

22.3.2 Linear Steady-state Analysis

A linear steady-state analysis will do if potential does not affect the material properties. Otherwise you must perform a nonlinear steady-state analysis [§ 22.3.3].

¹Example **conflow** illustrates an application of this option [Vol. *Geotechnical Analysis*].

syntax

```

BEGIN EXECUT
  LINEAR
  [ SOLVE ... ]
END EXECUT

```

LINEAR will perform a linear steady-state groundwater flow analysis.

SOLVE customizes the settings for the solution method [Ch. 30 p. 421].

22.3.3 Nonlinear Steady-state Analysis

A nonlinear groundwater flow analysis is required if the material properties vary with the potential, for instance with unconfined aquifers. You may perform such an analysis via the NONLIN commands.

syntax

```

BEGIN EXECUT
  BEGIN NONLIN
  [ BOUNDA { _____ } ]
    CASE=casen
    FACTOR=facr
  [ ITERAT ... ]
  [ ESTIMA ... ]
  END NONLIN
  [ SOLVE ... ]
END EXECUT

```

NONLIN will perform a nonlinear steady-state groundwater flow analysis.

BOUNDA specifies the boundary conditions.

Parameter CASE=*case* refers to a boundary case number in table 'BOUNDA' [§ 20.2 p. 317]. Default is the lowest available case number. Parameter FACTOR=*fac* specifies a multiplication factor.

[*fac*=1.0]

ITERAT specifies the iteration method to be applied in the nonlinear solution procedure [§ 22.3.3.1].

ESTIMA specifies a first estimation for the potential field [§ 22.3.3.2].

SOLVE customizes the settings for the solution method [Ch. 30 p. 421].

Default

file.dcf

```

*GROWSS
[ commands ]
BEGIN EXECUT
  NONLIN
END EXECUT
[ commands ]
*END

```

If, like in the above commands, you give a single NONLIN command, then DIANA will perform a nonlinear steady-state analysis without a first estimation of the potential field and with non-scaled boundary conditions from the lowest case number. The default iteration procedure will be applied. This is equivalent to the following commands.

file.dcf

```

*GROWSS
[ commands ]
BEGIN EXECUT
  BEGIN NONLIN
    BOUNDA CASE=lowest FACTOR=1.0
    ITERAT
    ESTIMA OFF
  END NONLIN
  SOLVE
END EXECUT
[ commands ]
*END

```

22.3.3.1 Iteration Method

With the ITERAT commands you may customize the iteration process that DIANA will apply in the nonlinear solution procedure. See also § 55.1.3.2 on page 629 for some background theory on these iteration methods.

syntax

```

BEGIN ITERAT
[ MAXITE=min ]
[ METHOD [ NEWTON ] _____ ]
                        REGULA
                        MODIFI
[ [ CONVER [ POTENT ] ] TOLCON=tcn ]
END ITERAT

```

MAXITE=*mi* is the maximum number of iterations.

[mi=5]

METHOD specifies the iteration method: NEWTON indicates a Newton–Raphson iteration scheme (the only option), i.e., the conductivity matrix will be updated.

MODIFI invokes the Modified Newton–Raphson method.

REGULA invokes the Regular Newton–Raphson method (the default).

[REGULA]

CONVER specifies the convergence criterion for the iteration process: POTENT indicates a convergence criterion on the norm of the potential field (the only option).

Parameter TOLCON=*tc* is the tolerance on the reference norm. When the norm of the incremental potential field has become less than *tc* × the reference norm DIANA assumes sufficient accuracy and stops the iteration process.

[tc=10⁻³]

Default

file.dcf

```

*GROWSS
[ commands ]
BEGIN EXECUT
  BEGIN NONLIN
    [ commands ]
    ITERAT
    [ commands ]
  END NONLIN
END EXECUT
[ commands ]
*END

```

If, like in the above commands, you give a single ITERAT command, then DIANA will apply a Regular Newton–Raphson iteration scheme with at most five iterations. As convergence criterion, the norm of the potentials will be applied with a tolerance of 10⁻³. This is equivalent to the following commands.

file.dcf

```

*GROWSS
[ commands ]
BEGIN EXECUT
  BEGIN NONLIN
    [ commands ]
    BEGIN ITERAT
      MAXITE=5
      METHOD NEWTON REGULA
      CONVER POTENT TOLCON=1.E-3
    END ITERAT
  [ commands ]
END NONLIN
END EXECUT
[ commands ]
*END

```

22.3.3.2 First Estimation of Potentials

With the ESTIMA commands you may specify a hydraulic head field which DIANA will apply as a first estimation of the solution.

syntax

```

BEGIN ESTIMA
[ OFF ]
[ POTENT [ OFF ] [ _____ ] ]
      INPUT { _____ }
              FIELD=fieldn
              FACTOR=facr
      CALCUL { _____ }
              CASE=casen
              FACTOR=facr
END ESTIMA

```

POTENT indicates that the estimation is for a hydraulic head field (the only possibility).

There are two methods to specify the hydraulic head field. For both options parameter FACTOR=*fac* is an optional multiplication factor.

[*fac*=1]

INPUT indicates a start with hydraulic heads as specified in table 'INIVAR' [§ 20.1.2 p. 316]. Parameter FIELD=*field* refers to a field number in this table. Default is the lowest available field number.

CALCUL indicates a start with hydraulic heads as calculated in a steady-state analysis. If hydraulic heads are available from a previous steady-state analysis, these hydraulic heads are used. Otherwise a linear steady-state analysis is performed. Parameter CASE=*case* refers to a boundary case number in table 'BOUNDAR' [§ 20.2 p. 317]. Default is the lowest available case number.

Default

file.dcf

```

*GROWSS
[ commands ]
BEGIN NONLIN
[ commands ]
  ESTIMA
[ commands ]
END NONLIN
[ commands ]
*END

```

If, like in the above commands, you give a single **ESTIMA** command, then DIANA will apply the potentials of the lowest field number in input table 'INIVAR' as a first estimation of the potential field. This is equivalent to the following commands.

file.dcf

```
*GROWSS
[ commands ]
BEGIN NONLIN
[ commands ]
BEGIN ESTIMA
  POTENT INPUT FIELD=lowest FACTOR=1.0
END ESTIMA
[ commands ]
END NONLIN
[ commands ]
*END
```

22.4 Transient Analysis

Module GROWTR is used to perform a transient groundwater flow analysis. In order to perform such analysis, you must additionally specify the time dependency of boundary conditions in table 'TIMEBO' [§ 20.2.2].

After the preliminary tasks like model evaluation and setup of the system of equations, a transient analysis basically involves two steps: (1) initialization of initial potentials and boundary conditions, (2) solution of the equations for the specified time steps. You may perform these steps in one DIANA job or in various subsequent jobs.

syntax

```
*GROWTR
[ MODEL [ OFF ] ... ]
[ INITIA [ OFF ] ... ]
[ EXECUTE [ OFF ] ... ]
[ OUTPUT [ OFF ] ... ] ...
```

MODEL to evaluate the finite element model [§ 22.4.1].

INITIA to initiate a transient groundwater flow analysis [§ 22.4.2].

EXECUT to execute time steps in the transient groundwater flow analysis [§ 22.4.3].

OUTPUT to specify analysis results for output [§ 22.5].

22.4.1 Model Evaluation

The **MODEL** commands customize the evaluation of the finite element model prior to the actual analysis.

syntax

```
BEGIN MODEL
[ EVALUA [ OFF ] ... ]
[ ASSEMB [ OFF ] ]
[ MATRIX [ OFF ] ... ]
[ BOUNDA [ OFF ] ]
END MODEL
```

EVALUA to check and evaluate geometric and material properties for elements [§ 3.4 p. 51].

ASSEMB to assemble the elements, i.e., creating appropriate system degrees of freedom [§ 3.5 p. 54].

MATRIX to setup the element conductivity and storativity matrices [§ 22.4.1.1].

BOUNDA to convert boundary conditions to right-hand-side nodal flux vectors. The OFF option switches off the creation of right-hand-side nodal flux vectors, which may be useful if these are still available on the FILOS file.

22.4.1.1 Conductivity and Storativity Matrices

By default, DIANA will set up the element conductivity and storativity matrices each time that you invoke Module GROWTR. You may explicitly ask DIANA to setup the matrices, or customize their setting up, via the MATRIX commands.

syntax

```
BEGIN MATRIX
[ OFF ]
[ CONduc [ OFF ] ]
[ STORAT [ OFF ] [ _____ ] ]
      CONSIS
      LUMPED
END MATRIX
```

OFF switches off the creation of conductivity and storativity matrices. This may save computing time if the matrices are still available on the FILOS file and appropriate.

CONduc sets up the element conductivity matrices.

STORAT sets up the element storativity matrices. You may specify the type of matrices with CONSIS for consistent matrices (the default), or with LUMPED for lumped matrices.

Default

file.dcf

```
*GROWTR
BEGIN MODEL
  MATRIX
END MODEL
[ commands ]
*END
```

If, like in the above commands, you give a single MATRIX command, then DIANA will setup the conductivity matrices and the consistent storativity matrices, which would also occur if you did not give the MATRIX command at all. This is equivalent to the following commands.

file.dcf

```
*GROWTR
BEGIN MODEL
  BEGIN MATRIX
    CONduc
    STORAT CONSIS
  END MATRIX
END MODEL
[ commands ]
*END
```

22.4.2 Initiate Transient Analysis

The INITIA commands initiate the transient analysis. The main purpose of this task is to evaluate the initial state of the model, i.e., prior to the execution of the first time step.

syntax

```

BEGIN INITIA
[ POTENT [ OFF ] [ _____ ]
                        INPUT { _____ }
                              FIELD=fieldn
                              FACTOR=facr
                        CALCUL { _____ }
                              CASE=casen
                              FACTOR=facr

[ SOLVE ... ]
[ TIME=stimer ]
[ _____ ]
  LINEAR
  NONLIN
END INITIA

```

POTENT indicates an initial hydraulic head field. There are two methods to specify the hydraulic head field. For both options parameter FACTOR=*fac* is an optional multiplication factor. [*fac*=1]

INPUT indicates a start with hydraulic heads as specified in table 'INIVAR' [§ 20.1.2 p. 316]. Parameter FIELD=*field* refers to a field number in this table. Default is the lowest available field number.

CALCUL indicates a start with hydraulic heads as calculated in a steady-state analysis. If hydraulic heads are available from a previous steady-state analysis, these hydraulic heads are used. For a nonlinear transient analysis, DIANA first checks if a nonlinear steady-state solution with matching boundary case number and load factor is available. Secondly, DIANA looks for an existing linear steady-state solution and scales it if necessary. Otherwise a linear steady-state analysis is performed. Parameter CASE=*case* refers to a boundary case number in table 'BOUNDAR' [§ 20.2 p. 317]. Default is the lowest available case number.

SOLVE customizes the settings for the solution method to be applied for the linear steady-state analysis [Ch. 30 p. 421].

TIME *stime* is a user specified starting time. [*stime*=0]

LINEAR initiates a linear analysis, i.e., with constant material properties. [LINEAR]

NONLIN initiates a nonlinear analysis. A nonlinear analysis is required if the material properties vary with the potentials, for instance with unconfined aquifers.

Default

file.dcf

```

*GROWTR
[ commands ]
INITIA
[ commands ]
*END

```

If, like in the above commands, you give a single INITIA command, then DIANA will apply the potentials of the lowest field number in input table 'INIVAR' as an initial potential field. This is equivalent to the following commands.

file.dcf

```

*GROWTR
[ commands ]
BEGIN INITIA
  POTENT INPUT FIELD=lowest FACTOR=1.0
  LINEAR
END INITIA
[ commands ]
*END

```

22.4.3 Time Steps Execution

With the EXECUT commands you ask DIANA to execute time steps in a transient groundwater flow analysis. A command file may contain one or more EXECUT blocks. Details of various commands are given in the referred sections.

syntax

```

BEGIN EXECUT
[ OFF ]
[ ALPHA=alphar ]
[ SIZES sizesr... ]
[ NONLIN ... ]
[ SOLVE ... ]
[ SAVE [ STEPS            ] ]
                        stepsn...
                        ALL
                        LAST
                        NONE
END EXECUT

```

ALPHA=*alpha* is the time integration parameter α . is the time integration parameter α . Usual methods are: Euler forward ($\alpha = 0$), Crank–Nicolson ($\alpha = \frac{1}{2}$), Galerkin ($\alpha = \frac{2}{3}$) and Euler backward ($\alpha = 1$) which is the default. See also § 55.1.3.1 on page 628 for some background theory on time integration.

[*sizes*=1.0] SIZES *sizes* are explicitly specified time step sizes, i.e., time increments. The number of values specifies the number of steps to be executed.

NONLIN specifies options for nonlinear transient analysis [§ 22.4.3.1].

SOLVE customizes the settings for the solution method [Ch. 30 p. 421].

SAVE specifies the steps for which analysis results must be saved for a subsequent structural analysis in a staggered flow–stress analysis [Ch. 26]. By default, in a model for staggered flow–stress analysis, DIANA will save the results for all executed steps.

[ALL] STEPS *steps* indicates the steps for which the analysis results must be saved. Alternatively you may save ALL steps, only the LAST step, or NONE of the steps. The NONE option drops the possibility of a subsequent structural analysis!

Default

file.dcf

```

*GROWTR
[ commands ]
EXECUT
*END

```

Due to these commands DIANA will execute a time step with a factor of 1, applying the default solution procedure, and giving the default output of analysis results. The same would occur if you had given the following commands.

file.dcf

```

*GROWTR
[ commands ]
BEGIN EXECUT
  ALPHA=1.0
  SIZES 1.0
  SOLVE
END EXECUT
BEGIN OUTPUT
  HEAD TOTAL
  HEAD PRESSU
  FLUX LOCAL NODES
END OUTPUT
*END

```

22.4.3.1 Nonlinear Analysis Options

Via the **NONLIN** commands you may customize a nonlinear transient analysis, provided that such analysis has been initiated previously [§ 22.4.2 p. 357].

syntax

```

BEGIN NONLIN
[ BEGIN ITERAT
  [ MAXITE= $mi_n$  ]
  [ METHOD [ NEWTON ] _____ ]
                                REGULA
                                MODIFI
[ [ CONVER [ POTENT ] ] TOLCON= $tc_n$  ]
  END ITERAT ]
END NONLIN

```

ITERAT specifies the iteration method to be applied in the nonlinear solution procedure. See also § 55.1.3.2 on page 629 for some background theory on these iteration methods.

MAXITE= mi is the maximum number of iterations. [$mi=5$]

METHOD specifies the iteration method: **NEWTON** for a Newton–Raphson iteration scheme (the only option), i.e., the conductivity matrix will be updated.

MODIFI invokes the Modified Newton–Raphson method.

REGULA invokes the Regular Newton–Raphson method (the default). [REGULA]

CONVER specifies the convergence criterion for the iteration process: **POTENT** indicates a convergence criterion on the norm of the potential field (the only option).

Parameter **TOLCON= tc** is the tolerance on the reference norm. When the norm of the incremental potential field has become less than $tc \times$ the reference norm [$tc=10^{-3}$] DIANA assumes sufficient accuracy and stops the iteration process.

Default

file.dcf

```

*GROWTR
[ commands ]
BEGIN EXECUT
[ commands ]
NONLIN
END EXECUT
*END

```

If, like in the above example, you only give a single **NONLIN** command then DIANA will perform a nonlinear analysis applying a Regular Newton–Raphson iteration scheme with at most five iterations. As convergence criterion, the norm of the potentials will be applied with a tolerance of 10^{-3} . Moreover, a default set of analysis results will be output. This is equivalent to the following commands.

file.dcf

```
*GROWTR
[ commands ]
BEGIN EXECUT
  BEGIN NONLIN
    [ commands ]
    BEGIN ITERAT
      MAXITE=5
      METHOD NEWTON REGULA
      CONVER POTENT TOLCON=1.E-3
    END ITERAT
  END NONLIN
END EXECUT
*END
```

22.5 Output of Analysis Results

You can get output of analysis results from a groundwater flow analysis via the **OUTPUT** commands. For the general, analysis type independent options for output of analysis results see § 3.6 on page 55.

syntax

```
BEGIN OUTPUT [devicew] [outoptw...] [params]
[ OFF ]
[ BEGIN SELECT
  [ ... ]
  [ BOUNDA _____ { } ]
               casesn... MIN
               ALL      MAX
  [ STEPS ... ]
END SELECT ]
[ LAYOUT ... ]
itemw ...
HEAD
FLUX
FLOW
PRESSU
END OUTPUT
```

SELECT command block to customize the batch output.

... for model selection see § 3.6.2 on page 59.

BOUNDA specifies a selection of boundary cases in linear steady–state analysis: *cases* is a series of case numbers referring to input table 'BOUNDA' [§ 20.2 p. 317], **ALL** selects all four boundary cases.

MIN selects the minimum value of the (selected) boundary cases to be output, **MAX** selects the maximum value.

STEPS selects time steps for output of transient analysis [§ 22.5.1 p. 361].

LAYOUT optional commands to customize the layout and style of tabular output [§ 3.6.4 p. 64].

item is the name of the analysis result to be output.

HEAD for hydraulic heads [§ 22.5.2 p. 361].

FLUX for element fluxes [§ 22.5.3 p. 362].

FLOW for nodal flows [§ 22.5.4 p. 362].

PRESSU for pore pressures [§ 22.5.5 p. 363].

22.5.1 Step Selection

The STEPS command selects time steps for output of analysis results of a transient ground-water flow analysis.

syntax

```
STEPS [ _____ ] { _____ }
      stepsn...  MIN
      ALL          MAX
      LAST
```

steps are numbers of selected steps.

ALL will produce output for all steps that are executed (the default).

[ALL]

LAST selects the final step.

DIANA will output the extreme values that occurred up to and including the selected steps if you specify one of the options:

MIN for the minimum values,

MAX for the maximum values.

For step-wise analyses output of extreme values cannot be used in combination with the option LAYOUT COMBIN [§ 3.6.4.1 p. 65] to assemble various results in one table for tabular output.

The interpretation of the above options depends on the position of the current OUTPUT block. At the module command level (*GROWTR [§ 22.4 p. 355]), *steps* are absolute step numbers, and ALL/LAST refer to all steps/the last step of the current analysis. At the EXECUT command level [§ 22.4.3 p. 358], *steps* are relative step numbers, and ALL/LAST refer to all steps/the last step of the current EXECUT block.

22.5.2 Hydraulic Heads

The hydraulic head field is the solution vector of the system of equations for the ground-water flow analysis. This represents the fluid column height in the nodes as scalar values, therefore component selection is not applicable.

syntax

```
HEAD [ typew ] { optiw }
      TOTAL
      PRESSU
```

HEAD gives the hydraulic head ϕ .

type indicates the type of the hydraulic heads to be output.

TOTAL gives the total hydraulic head, or pressure potential $\phi = \phi_p$ (the default). [TOTAL]

PRESSU gives the pressure head $\phi_h = \phi_p + \hat{\mathbf{g}} \cdot (\mathbf{x} - \mathbf{x}_{\text{ref}})$ in detailed groundwater flow analysis, with $\hat{\mathbf{g}}$ the unit vector of the gravity vector \mathbf{g} , for instance if $\mathbf{g} = (1, -1, 0)$ then $\hat{\mathbf{g}} = (\sqrt{0.5}, -\sqrt{0.5}, 0)$; \mathbf{x} the global XYZ coordinates of the output location; and \mathbf{x}_{ref} the reference position from table 'MODEL' [§ 1.2 p. 6].

opti are additional options [§ 3.6.1 p. 58].

Note that DIANA assumes that for axisymmetric and two-dimensional models the gravity direction \mathbf{g} is in the global Y direction, and for three-dimensional models in the global Z direction if not explicitly specified in table 'MODEL'.

<i>item</i>	<i>type</i>	
HEAD	TOTAL	H
		ϕ_p
HEAD	PRESSU	HPR
		ϕ_h

22.5.3 Fluxes

In a groundwater flow analysis, fluxes are calculated for elements. They can be output in the integration points, element center point, or in the element nodes. DIANA also writes the integrated fluxes over groups of boundary elements to the standard output file *file.out*.

syntax

FLUX	[<i>oper_w</i>]	{ <i>comp_w</i> }	[<i>loca_w</i>]	{ <i>opti_w</i> }
	LOCAL		INTPNT	
	GLOBAL		NODES	
	BOUNDA		CENTER	

FLUX gives the fluxes q [Eq. (55.2) p. 627].

[GLOBAL] *oper* specifies an operation to be performed on the fluxes.

LOCAL transforms fluxes in continuum elements to local xyz directions.

GLOBAL transforms fluxes in continuum elements to global XYZ directions.

BOUNDA gives the total discharge Q for boundary elements. This rate of flow is a scalar value, therefore component selection is not applicable.

[NODES] *loca* specifies the location of the fluxes to be output [§ 3.6.1 p. 58].

opti are additional options [§ 3.6.1 p. 58].

<i>item</i>	<i>oper</i>	<i>comp ...</i>		
		X	Y	Z
FLUX	LOCAL	FLx	FLy	FLz
		q_x	q_y	q_z
FLUX	GLOBAL	FLX	FLY	FLZ
		q_X	q_Y	q_Z
FLUX	BOUNDA	q		
		Q		

22.5.4 Flows

In a groundwater flow analysis nodal flows are calculated. Nodal flows are scalar values, component selection is not applicable.

syntax

```
FLOW [ typew ] { optiw }
      REACTI
      RESIDU
      EXTERN
```

FLOW gives the nodal flows.

type specifies the type of the nodal flow.

[REACTI]

REACTI for the reaction flow Q in all nodes with fixed pressure potentials.

RESIDU for the residual flow ΔQ , also called the out-of-balance flow. This is defined as the difference between the externally applied flow Q and the internal resistance flow.

EXTERN for the externally applied flow Q .

opti are additional options [§ 3.6.1 p. 58].

<i>item</i>	<i>type</i>	
FLOW	REACTI	FLB
		Q
FLOW	RESIDU	FLR
		ΔQ
FLOW	EXTERN	FLE
		Q

22.5.5 Pore Pressures

The pore pressure field can be derived from the pressure head field by multiplication of the fluid density ρ_f and the gravity acceleration g . As this result is a scalar no component selection is available.

syntax

```
PRESSU PORE { optiw }
```

PRESSU PORE gives the pore pressure $p = \phi_h \cdot \rho_f \cdot g$.

*Note that a default fluid density ρ_f of 1000 kg/m³ is being used when parameter **DENSFL** is not defined in the material table of the respective elements. The gravity acceleration g is defined as 9.81 m/s². When different units are used in the model, table 'UNITS' must be defined.*

opti are additional options [§ 3.6.1 p. 58].

<i>item</i>	<i>type</i>	
PRESSU	PORE	P
		p

Chapter 23

Beam Cross-section Analysis

Via Module BCROSS, which actually performs a simple steady-state potential flow analysis, DIANA can determine cross-sectional properties and shear stress distributions of beam cross-sections of arbitrary composition and with arbitrary shape. The implemented theory assumes free warping. In order to perform a beam-cross-section analysis with DIANA you must take the following actions:

1. Invoke Module FILOS to initialize an analysis database [§ 3.2 p. 48].
2. Invoke Module INPUT to read the finite element model into the database [§ 3.3 p. 50]. See §23.1 for a description of input data required for beam cross-section analysis.
3. Invoke Module BCROSS to perform a the actual beam cross-section analysis [§ 23.2 p. 366].
4. You may perform interactive postprocessing of results with DianaIE [§ 3.6.3 p. 63], which is the default output device when the analysis is neither started from ¡DIANA nor FX⁺, or with ¡DIANA [§ 3.6.5 p. 66], which is the default output device when the analysis is started from ¡DIANA, or with FX⁺ [§ 3.6.6 p. 67], which is the default output device when the analysis is started from FX⁺. Alternatively, tabulated output is available [§ 3.6.4 p. 64].

See also the examples `crosls` and `croslc` in Volume *Analysis Examples*, respectively for a homogeneous and an inhomogeneous cross-section. The minimum set of commands for a complete beam cross-section analysis is as shown below.

file.dcf

```
*FILOS
INITIA
*INPUT
*BCROSS
*END
```

23.1 General Modelling Aspects

This section describes the general aspects for beam cross-section analysis: shape and dimensions of the finite element mesh, and definition of elastic properties.

23.1.1 Meshing

The shape and dimensions of a beam cross-section are modeled by a mesh of two-dimensional, quadrilateral or triangular *cross-section elements* [Vol. *Element Library*]. To avoid singularity of the system of equations you must specify an arbitrary node with a fixed potential in table 'FIXPOT' [§ 20.1.1 p. 315].

23.1.2 Elastic Properties

The Young's moduli and Poisson's ratios for different material sections of the cross-section are specified as material properties [Vol. *Material Library*].

23.2 Analysis Commands

Module BCROSS determines the beam cross-section properties via a dedicated steady-state potential flow analysis.

syntax

```
*BCROSS
[ MODEL [ OFF ] ... ]
[ EXECUT [ OFF ] ... ]
[ OUTPUT [ OFF ] ... ] ...
```

MODEL to evaluate the finite element model [§ 23.2.1].

EXECUT to execute the beam cross-section analysis [§ 23.2.2].

OUTPUT for output of shear stress distributions due to shear forces and torsional moment [§ 23.3].

23.2.1 Model Evaluation

The MODEL commands customize the evaluation of the finite element model prior to the actual analysis.

syntax

```
BEGIN MODEL
[ EVALUA [ OFF ] ... ]
[ ASSEMB [ OFF ] ]
[ MATRIX [ CONDOC ] [ OFF ] ]
[ BOUNDA [ OFF ] ... ]
END MODEL
```

EVALUA to check and evaluate geometric and material properties for elements [§ 3.4 p. 51].

ASSEMB to assemble the elements, i.e., creating appropriate system degrees of freedom [§ 3.5 p. 54].

MATRIX CONDOC to setup the element conductivity matrices. The OFF option switches off the creation of conductivity matrices. This may save computing time if the matrices are still available on the FILOS file and appropriate.

BOUNDA to generate the required boundary cases and convert these to right-hand-side nodal flux vectors [§ 23.2.1.1].

23.2.1.1 Right-hand-side Nodal Flux Vectors

The steady-state potential flow analysis, dedicated to a cross-section analysis with Module BCROSS requires the following boundary cases:

- case 1** to simulate a shear force Q_y ,
- case 2** to simulate a shear force Q_z ,
- case 3** to simulate a torsion moment M_x ,
- case 4** the previous three cases combined.

By default, DIANA will assume unit values for the cross-sectional loading (Q_y , Q_z , M_x), while setting up the four boundary cases. However, you may overrule the default unit values by specifying parameters. In either case, the boundary cases will be converted to right-hand-side nodal flux vectors.

syntax

BEGIN BOUNDA

[OFF]

[QY= qy_r]

[QZ= qz_r]

[MX= mx_r]

END BOUNDA

OFF switches off the creation of the nodal flux vectors. This may save computing time if these vectors are still available on the FILOS file and appropriate.

QY= qy specifies the value for Q_y . [$Q_y = 1$]

QZ= qz specifies the value for Q_z . [$Q_z = 1$]

MX= mx specifies the value for M_x . [$M_x = 1$]

Default

file.dcf

*BCROSS

[commands]

BEGIN MODEL

BOUND A

END MODEL

[commands]

*END

If, like in the above commands, you give a single BOUNDA command, then DIANA will setup the right-hand-side nodal flux vectors, with unit values for the cross-sectional loading, which would also occur if you did not give the BOUNDA command at all or if you would have given the following commands.

file.dcf

*BCROSS

[commands]

BEGIN MODEL

BEGIN BOUNDA

QY=1.0

QZ=1.0

MX=1.0

END BOUNDA

END MODEL

[commands]

*END

23.2.2 Cross-section Analysis Execution

The EXECUT commands cause the actual beam cross-section analysis to be executed.

syntax

BEGIN EXECUT

[SOLVE ...]

END EXECUT

SOLVE customizes the settings for the solution method [Ch. 30 p. 421].

23.3 Output of Analysis Results

Module BCROSS gives two types of analysis results: cross-sectional properties [§ 23.3.1], and shear stresses [§ 23.3.2].

23.3.1 Cross-sectional Properties

Module BCROSS always prints the cross-sectional properties on the standard output file. Table 23.1 shows the available properties, with the label that precedes them on the output

Table 23.1: OUTPUT OF CROSS-SECTION PROPERTIES

	Label	Item	Formula	ISO unit
Moments of inertia	Iy	I_y	$\int_A z^2 dA$	m ⁴
	Iz	I_z	$\int_A y^2 dA$	m ⁴
	It	I_t		m ⁴
Extreme fibers	y _{max}	y_{\max}		m
	y _{min}	y_{\min}		m
	z _{max}	z_{\max}		m
	z _{min}	z_{\min}		m
Sections moduli	Wy _{max}	$W_{y,\max}$	I_y/z_{\min}	m ³
	Wy _{min}	$W_{y,\min}$	I_y/z_{\max}	m ³
	Wz _{max}	$W_{z,\max}$	I_z/y_{\min}	m ³
	Wz _{min}	$W_{z,\min}$	I_z/y_{\max}	m ³
Center, axes and area	Xc	X_c		m
	Yc	Y_c		m
	Alpha	α		rad
	A	A		m ²
Reciprocal shear shape factors	ky	k_y	$1/S_y$	
	kz	k_z	$1/S_z$	
Shear center	ey	e_y		m
	ez	e_z		m
Rigidity and stiffness (inhomogeneous)	E.Iy		$\sum_{i=1}^n E_{(i)} I_{y(i)}$	N·m ²
	E.Iz		$\sum_{i=1}^n E_{(i)} I_{z(i)}$	N·m ²
	G.It		$\sum_{i=1}^n G_{(i)} I_{t(i)}$	N·m ²
	E.A		$\sum_{i=1}^n E_{(i)} A_{(i)}$	N

line.¹ Figure 23.1 on the facing page shows some properties of an arbitrary cross-section. Note that angle α has a negative value in this example.

23.3.2 Shear Stresses

You can get output of shear stresses from a cross-section analysis via the **OUTPUT** commands. For the general, analysis type independent options for output of analysis results see § 3.6 on page 55.

syntax

```

BEGIN OUTPUT [devicew] [outoptw...] [params]
[ OFF ]
[ BEGIN SELECT
[ ... ]
[ CASES _____ ]
      casesn...
```

¹See also the output of example crosrec in Volume *Analysis Examples*.

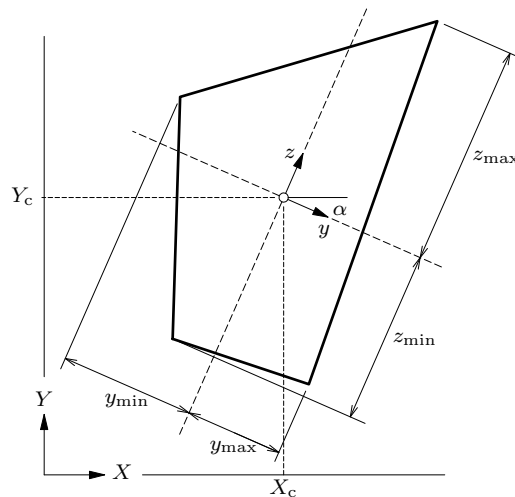


Figure 23.1: Cross-section properties

```

      ALL
    END SELECT ]
  [ LAYOUT ... ]
  SHEAR [ operw ] { compw } [ locaw ] { optiw } ...
        LOCAL          INTPNT
        GLOBAL         NODES
                     CENTER
END OUTPUT

```

SELECT command block to customize the batch output.

... for model selection see § 3.6.2 on page 59.

CASES specifies a selection of boundary cases [§ 23.2.1.1]: *cases* is a series of case numbers between 1 and 4, ALL selects all four boundary cases.

LAYOUT optional commands to customize the layout and style of tabular output [§ 3.6.4 p. 64].

SHEAR indicates output of shear stresses as analysis result due to the standard boundary cases [§ 23.2.1.1].

oper specifies an operation to be performed on the shear stresses. [LOCAL]

LOCAL transforms shear stresses to local *xyz* directions.

GLOBAL transforms shear stresses to global *XYZ* directions.

loca specifies the location of the shear stresses to be output [§ 3.6.1 p. 58]. [NODES]

opti are additional options [§ 3.6.1 p. 58].

<i>item</i>	<i>oper</i>	<i>comp ...</i>	
		X	Y
SHEAR	LOCAL	Sx	Sy
		σ_{zx}	σ_{zy}
SHEAR	GLOBAL	SX	SY
		σ_{ZX}	σ_{ZY}

The shear stresses can be used as input in a linear static analysis to define the shear distribution for beam elements.

Chapter 24

Reynolds Flow – Lubrication Analysis

DIANA enables modelling of lubrication between rotating bearings etc. by an implementation of the Reynolds equation. This equation assumes a thin film of viscous, Newtonian fluid between moving boundaries. Via Module REYNOL, which actually performs a simple linear steady-state potential flow analysis, DIANA can determine the pressure field of a lubrication problem. In order to perform a lubrication analysis with DIANA you must take the following actions:¹

1. Invoke Module FILOS to initialize an analysis database [§ 3.2 p. 48].
2. Invoke Module INPUT to read the finite element model into the database [§ 3.3 p. 50]. See §24.1 for a description of input data required for lubrication analysis.
3. Invoke Module REYNOL to perform a the actual lubrication analysis [§ 24.2 p. 372].
4. You may get plots of the analysis results in the postprocessing working environment of iDIANA [Vol. iDIANA].

The minimum set of commands for a complete lubrication analysis is as shown below.

file.dcf

```
*FILOS  
INITIA  
*INPUT  
*REYNOL  
*END
```

24.1 General Modelling Aspects

24.1.1 Meshing

The film is modeled by quadrilateral or triangular *lubrication elements* [Vol. *Element Library*]. The varying thickness of the film is input in table 'GEOMET'.

24.1.2 Viscosity

The dynamic viscosity η is input as a material property [Vol. *Material Library*].

¹See also the example lubric in Volume *Analysis Examples*.

24.1.3 Boundary Conditions

Prescribed pressures at the boundaries are input in table 'FIXPOT' [§ 20.1.1 p. 315]. Non-zero values for these prescribed pressures must be specified in subtable NODAL of table 'BOUNDAL' [§ 20.2.1 p. 317].

24.1.4 Velocity

The velocity difference between the two faces of the element is input in subtable ELEMEN of table 'BOUNDAL' [Vol. *Element Library*].

24.2 Analysis Commands

Module REYNOL determines the pressure field via a dedicated linear steady-state potential flow analysis.

syntax

```
*REYNOL
[ MODEL [ OFF ] ... ]
[ EXECUT [ OFF ] ... ]
[ OUTPUT [ OFF ] ... ] ...
```

MODEL to evaluate the finite element model [§ 24.2.1].

EXECUT to execute the lubrication analysis [§ 24.2.2].

OUTPUT for output of the pressure field [§ 24.3].

24.2.1 Model Evaluation

The MODEL commands customize the evaluation of the finite element model prior to the actual analysis.

syntax

```
BEGIN MODEL
[ EVALUA [ OFF ] ... ]
[ ASSEMB [ OFF ] ]
[ MATRIX [ CONDUCT ] [ OFF ] ]
[ BOUNDAL [ OFF ] ]
END MODEL
```

EVALUA to check and evaluate geometric and material properties for elements [§ 3.4 p. 51].

ASSEMB to assemble the elements, i.e., creating appropriate system degrees of freedom [§ 3.5 p. 54].

MATRIX CONDUCT to setup the element conductivity matrices. The OFF option switches off the creation of conductivity matrices. This may save computing time if the matrices are still available on the FILOS file and appropriate.

BOUNDAL to convert the boundary cases from input table 'BOUNDAL' to right-hand-side nodal flux vectors. The OFF option switches off the conversion of boundary cases. This may save computing time if these are still available on the FILOS file and appropriate.

24.2.2 Lubrication Analysis Execution

The EXECUT commands cause the actual lubrication analysis to be executed.

syntax

```
BEGIN EXECUT
[ SOLVE ... ]
END EXECUT
```

SOLVE customizes the settings for the solution method [Ch. 30 p. 421].

24.3 Output of Analysis Results

You can get output of analysis results from a lubrication analysis via the **OUTPUT** commands. For the general, analysis type independent options for output of analysis results see § 3.6 on page 55.

syntax

```
BEGIN OUTPUT [devicew] [outoptw...] [params]
[ OFF ]
[ BEGIN SELECT
  [ ... ]
  [ CASES _____ ]
    casesn...
    ALL
  END SELECT ]
[ LAYOUT ... ]
  itemw ...
  PRSNOD
  FLUX
END OUTPUT
```

SELECT command block to customize the batch output.

... for model selection see § 3.6.2 on page 59.

CASES specifies a selection of boundary cases [§ 20.2 p. 317]: *cases* is a series of case numbers referring to table 'BOUNDARY'. The ALL option selects all boundary cases.

LAYOUT optional commands to customize the layout and style of tabular output [§ 3.6.4 p. 64].

item is the name of the analysis result to be output.

PRSNOD for nodal pressures [§ 24.3.1 p. 373].

FLUX for element fluxes [§ 24.3.2 p. 374].

24.3.1 Nodal Pressures

The pressure field is the solution vector of the system of equations for the lubrication analysis. Pressures are scalar values calculated in the nodes, component selection is not applicable.

syntax

```
PRSNOD { optiw }
```

PRSNOD the nodal pressures *p*.

opti are additional options [§ 3.6.1 p. 58].

<i>item</i>	
PRSNOD	PRP
	<i>p</i>

24.3.2 Fluxes

In a lubrication analysis, fluxes are calculated for elements. They can be output in the integration points, element center point or in the element nodes.

syntax

```

FLUX [ operw ] { compw } [ locaw ] { optiw }
      LOCAL          INTPNT
      GLOBAL         NODES
                      CENTER

```

FLUX gives the fluxes q .

[GLOBAL] *oper* specifies an operation to be performed on the fluxes.

LOCAL transforms fluxes in to local xy directions.

GLOBAL transforms fluxes to global XYZ directions.

[NODES] *loca* specifies the location of the fluxes to be output [§ 3.6.1 p. 58].

opti are additional options [§ 3.6.1 p. 58].

<i>item</i>	<i>oper</i>	<i>comp</i> ...		
		X	Y	Z
FLUX	LOCAL	FLx	FLy	FLz
		q_x	q_y	q_z
FLUX	GLOBAL	FLX	FLY	FLZ
		q_X	q_Y	q_Z

Part VII

Coupled Flow–Stress Analysis

Chapter 25

Introduction to Flow–Stress Analysis

This chapter introduces the applicability of coupled flow–stress analysis, see also Chapter 56 for background theory. In coupled flow–stress analysis the interaction may be one- or two-directional.

If the flow only affects the deformation and not vice versa, the interaction is *one-directional*. Typical examples are a geotechnical (static) stability analysis [§ 25.1] or a structural analysis with thermal load [§ 25.2]. To solve one-directional interaction a so-called *staggered analysis* is most convenient: the flow field is calculated first and acts as an external load for the following stress analysis. DIANA’s Modules LINSTA and NONLIN can automatically convert the model from flow to structural analysis. Chapter 26 formally describes how to use DIANA for staggered analysis. See also Volume *Analysis Examples* and Volume *Geotechnical Analysis* for some examples of staggered analysis.

If the flow affects the deformation and vice versa, the interaction is *two-directional*. A typical example is a geotechnical transient consolidation analysis. To solve two-directional interaction, a simultaneous calculation of flow and stress is most convenient [§ 25.3]. DIANA offers the *mixture analysis* for this type of interaction problems where the regular structural elements can be extended to *mixture elements* via a special option. Chapter 27 formally describes how to use DIANA for mixture analysis. See also example *consol* in Volume *Geotechnical Analysis* which shows a typical mixture analysis.

25.1 Staggered Solution with Pore Pressure

The one-directional coupled flow–stress analysis or staggered analysis is performed with the usual DIANA modules for potential flow and structural analysis. Modules LINSTA and NONLIN will automatically transform the hydraulic heads from a potential flow analysis to input for the structural analysis by generation of input for table ‘LOADS’.

Staggered analysis is applicable to analyse the effect of pore fluid on deformation and effective stress: *an imposed flow causes deformation, but a mechanical load does not cause flow*. The implementation of flow analysis and load transformation account for unsaturated areas.

All effective stress based models for elasticity and plasticity are applicable in combination with a pore fluid load. The available models cover the elastoplastic behaviour of the usual soil types. Darcy’s law is used to calculate hydraulic heads in the porous part of the mesh.

Anisotropic properties for permeability and stiffness can be modeled. Permeability and hydrostatic capacity are nonlinear in unsaturated areas or areas with turbulent flow.

The implementation allows plane strain, axisymmetric, and three-dimensional analysis. The same mesh must be used for potential flow and structural analysis. Both linear and quadratic structural elements are interpreted as linear potential flow elements. Combination of elements with porous material (soil) and non-porous materials (structure) is feasible. It is possible to perform a static analysis, linear as well as nonlinear.

25.2 Staggered Solution with Temperature

The one-directional coupled flow–stress analysis or staggered analysis is performed with the usual DIANA modules for potential flow and structural analysis. Both linear and quadratic structural elements are interpreted as linear potential flow elements. Modules LINSTA and NONLIN will automatically convert the temperature from a potential flow analysis to input for the structural analysis by generation of input tables 'LOADS' or 'TEMPER'. The temperature analysis of hydrating concrete also yields a degree of reaction, from which DIANA generates a table 'MATURI'.

Staggered analysis is applicable to analyse the effect of temperature on deformation and stress: *an imposed temperature causes deformation, but a mechanical load does not cause thermal effects.*

All structural material models are applicable in combination with a temperature load. A nonlinear temperature dependency may be introduced for most of these models. The capacity and conductivity may be nonlinear.

The implementation allows plane strain, axisymmetric, and three-dimensional analysis. A mesh with linear or quadratic structural elements must be used for potential flow and structural analysis. Both linear and quadratic structural elements are interpreted as linear potential flow elements. Combination of elements with conducting material and insulating material is feasible. It is possible to perform a static or transient analysis, linear as well as nonlinear.

25.3 Simultaneous Solution with Pore Pressure

The two-directional coupled flow–stress analysis or mixture analysis is performed with the usual DIANA modules for structural analysis. User-selected structural elements are extended to so-called *mixture elements* by addition of a pore pressure potential to the displacements as nodal degree of freedom.

Mixture elements are applicable for the analysis of the interaction between deformation of porous media and pore fluid, according to the full Biot theory: *a mechanical load causes flow and an imposed flow causes deformation.* Furthermore Biot's theory includes undrained compressibility and differentiation between effective stress and intergranular stress.

Biot's theory does not account for inertial effects during rapid loading in dynamic analysis, e.g. the velocity of soil and fluid are not considered separately. Biot's theory assumes saturation and therefore is not directly suited for analysis with unsaturated areas.

All effective stress based models for elasticity and plasticity are applicable in combination with mixture elements. The available models cover the elastoplastic behaviour of the common soil types. Anisotropic properties for permeability and stiffness can be modeled.

The implementation allows plane strain, axisymmetric, and three-dimensional analysis. Combination of elements with porous material (soil) and non-porous material (structure) is feasible. The mixture continuum elements and mixture interface elements can be used with a wide range of structural elements and structural material models.

It is possible to perform a static and transient analysis, linear as well as physical nonlinear. Special options are available to initialize in-situ (overconsolidated) stresses and flow. Initial stress and pore pressure can also be derived from a preceding nonlinear analysis phase.

The elements have restricted applicability for dynamic analysis (no separate fluid velocities). Furthermore the elements are not applicable for eigenvalue analysis and modal analysis, because these analysis types are currently restricted to symmetrical matrices. Application of the elements is meaningless for buckling stability analysis and pure potential flow analysis.

Chapter 26

Staggered Analysis

If no measured potential field is available, the input for a structural analysis may be found from a preliminary potential flow analysis [Part VI]. The combination of a potential flow analysis and a subsequent structural analysis is also known as ‘staggered flow–stress analysis’.

DIANA will automatically convert the results of a potential flow analysis to input data for a structural analysis. A peculiarity in staggered flow–stress analysis is that the model for the flow analysis always must consist of linearly interpolated elements whereas for the structural analysis either linearly or quadratically interpolated elements must be applied. DIANA will automatically convert the elements whenever necessary.

26.1 How to Perform Staggered Analysis

A staggered flow–stress analysis comprises sequential execution of the following steps:

1. Prepare a finite element model, appropriate for flow and structural analysis [§ 26.2].
2. Perform a potential flow analysis [§ 26.3].
3. Perform structural analysis [§ 26.5].

We introduce these steps by showing two typical command sequences, respectively for a steady-state and for a transient heat flow analysis.

Steady-state

file.dcf

```
*FILOS
INITIA
*INPUT
*HEATSS
...
*LINSTA
OUTPUT ...
*END
```

First the complete model is read into a newly initialized FILOS file [Ch. 3]. Then a steady-state heat flow analysis is performed via Module HEATSS [§ 21.2 p. 332]. Module LINSTA will convert the results of the heat flow analysis, in this case temperatures, to input data for a linear static structural analysis and then perform such analysis [Ch. 4].

Transient

file.dcf

```
*FILOS
INITIA
*INPUT
*HEATTR
...
BEGIN EXECUTE
...
```

```

SAVE STEPS 1 4 5
END EXECUTE
*NONLIN
...
BEGIN EXECUTE
  BEGIN TIME
  EXPLIC SIZES ...
  END TIME
END EXECUTE
OUTPUT ...
*END

```

A transient heat flow analysis, here performed with Module HEATTR [§ 21.3 p. 337], must be combined with a transient nonlinear structural analysis. In the heat flow analysis you must save the results of specified steps via the **SAVE** command. A transient nonlinear structural analysis is then performed with Module NONLIN [Ch. 13], but first NONLIN will automatically convert the heat flow results to loads or temperatures and maturity for structural analysis.

26.2 Model Definition

A model for staggered flow–stress analysis basically comprises two domains: one for flow analysis and one for structural analysis. In practice these domains will overlap completely, or at least for a considerable part. We will call this overlap the ‘flow–stress domain’.

A staggered flow–stress analysis with DIANA requires that the complete model is defined prior to the potential flow analysis, i.e., the flow-only domain, the structural-only domain, and the flow–stress domain altogether on one input data file. The flow-only domain must be modeled with flow elements and the structural-only domain with structural elements. For both domains you must supply appropriate material properties. For the flow–stress domain please note the following:

- The flow–stress domain, where the elements are active in both potential flow and structural analysis, must be modeled with structural elements. The choice of elements is restricted to those listed in column ‘Flow–stress domain’ of Table 26.1 on the next page.
- The elements in the flow–stress domain that are linearly interpolated stay linearly interpolated also in the potential flow analysis. The elements in the flow–stress domain that are quadratically interpolated, however, are converted to linearly interpolated elements as indicated in column ‘Conversion to flow analysis’. Therefore in the potential flow analysis of a staggered analysis the mid-nodes of quadratic structural elements are meaningless.
- For the elements in the flow–stress domain you must specify appropriate material properties, not only for structural analysis but also for flow analysis [Vol. *Material Library*].
- DIANA will consider any structural element without properties for flow analysis to belong to the structural-only domain.
- Any structural element that cannot be converted to the flow domain, e.g. plate bending, flat, curved, infinite, and axisymmetric shell elements, will belong to the structural-only domain.

26.3 Potential Flow Analysis

The first analysis in a staggered flow–stress analysis is the potential flow analysis. This may be a heat flow or concentration flow analysis [Ch. 21], or a detailed groundwater flow analysis [§ 22.1 p. 347]. DIANA recognizes the type of analysis from the module that you

Table 26.1: ELEMENTS FOR STAGGERED FLOW-STRESS ANALYSIS

Flow-stress domain		Conversion to flow analysis		
		heat	concen.	groundwater
pl. stress	T6MEM	T3HT	T3HT	T3GW
	C8MEM	Q4HT	Q4HT	Q4GW
	CT12M	T3HT	T3HT	T3GW
	CQ16M	Q4HT	Q4HT	Q4GW
	T9MEM	T3HT	T3HT	T3GW
	Q12ME	Q4HT	Q4HT	Q4GW
pl. strain	T6EPS	T3HT	T3HT	T3GW
	Q8EPS	Q4HT	Q4HT	Q4GW
	CT12E	T3HT	T3HT	T3GW
	CQ16E	Q4HT	Q4HT	Q4GW
	CQ20E	Q4HT	Q4HT	Q4GW
	CQ22E	Q4HT	Q4HT	Q4GW
	CT18GE	T3HT	T3HT	T3GW
	CQ24GE	Q4HT	Q4HT	Q4GW
axisymm.	T6AXI	T3AHT	T3AHT	T3AGW
	Q8AXI	Q4AHT	Q4AHT	Q4AGW
	CT12A	T3AHT	T3AHT	T3AGW
	CQ16A	Q4AHT	Q4AHT	Q4AGW
	CQ20A	Q4AHT	Q4AHT	Q4AGW
	CQ22A	Q4AHT	Q4AHT	Q4AGW
solid	TE12L	TE4HT	TE4HT	TE4GW
	PY15L	PY5HT	PY5HT	PY5GW
	TP18L	TP6HT	TP6HT	TP6GW
	HX24L	HX8HT	HX8HT	HX8GW
	CTE30	TE4HT	TE4HT	TE4GW
	CPY39	PY5HT	PY5HT	PY5GW
	CTP45	TP6HT	TP6HT	TP6GW
	CHX60	HX8HT	HX8HT	HX8GW
	CHX64	HX8HT	HX8HT	HX8GW
interface	N4IF	IPT2H	IPT2H	IPT2H
	N6IF	IPT2H	IPT2H	IPT2H
	L8IF	IL4HT	IL4HT	IL4HT
	CL12I	IL4HT	IL4HT	IL4HT
	T18IF	IT6HT	IT6HT	IT6HT
	Q24IF	IQ8HT	IQ8HT	IQ8HT
	CT36I	IT6HT	IT6HT	IT6HT
base spring	CQ48I	IQ8HT	IQ8HT	IQ8HT
	SP6BA	IPT2H	IPT2H	IPT2H
boundary	SP12BA	IPT2H	IPT2H	IPT2H
	L4TM	B2HT	B2HT	B2GW
	CL6TM	B2HT	B2HT	B2GW
	T9TM	BT3HT	BT3HT	BT3GW
	Q12TM	BQ4HT	BQ4HT	BQ4GW
	CT18TM	BT3HT	BT3HT	BT3GW
boundary axisymm.	CQ24TM	BQ4HT	BQ4HT	BQ4GW
	L4TM	B2AHT	B2AHT	B2AGW
	CL6TM	B2AHT	B2AHT	B2AGW

invoke. Prior to the actual flow analysis, any structural element with material properties appropriate for the analysis type will be converted internally to a flow element as indicated in Table 26.1.

Transient analysis. For transient analysis DIANA can convert the time dependent flow results to input for structural analysis [§ 21.3.3 p.340] [§ 22.4.3 p.358]. By default conversion will be performed for all time steps. Alternatively you may select certain time steps for which conversion must be performed.

file.dcf

```

*HEATTR
commands
BEGIN EXECUT
  commands
  SAVE STEPS 1 3 4
END EXECUT

```

26.4 Conversion to Structural Analysis

In a staggered flow–stress analysis DIANA will automatically convert results of a potential flow analysis to input for a structural analysis. Note that results of linearly interpolated potential flow elements will be converted to input either for linear structural elements or for quadratic structural elements. Results of each boundary case in a steady-state flow analysis will be transformed to a corresponding load case for linear structural analysis. Results of saved steps in a transient flow analysis will be converted to temperature and maturity input for a transient nonlinear structural analysis.

The transformation of a groundwater flow model to a structural model also has some additional points of interest [§ 26.4.2].

26.4.1 Heat and Concentration Flow to Structural

Temperature or concentration load cause initial strains following from thermal or chemical expansion or shrinkage of the elements [Vol. *Element Library*]. In a nonlinear analysis the effect of the temperature or concentration on several material properties may be analysed in combination with other nonlinearities [Part IV].

For a possible transient nonlinear structural analysis DIANA will apply a maturity model to convert the transient heat flow analysis results. The type of the maturity model depends on what was determined during the heat flow analysis: the degree of reaction in hydration heat analysis, or the equivalent age [§ 21.3.2.1 p. 340]. This feature enables the analysis of chemical reactions like cement hydration (young hardening concrete) and vulcanization of rubber. See Volume *Material Library* for more information on maturity.

26.4.2 Groundwater Flow to Structural

After a detailed groundwater flow analysis [§ 22.1], DIANA will convert the resulting potential heights in the flow–stress domain to pressure input for structural analysis. This conversion requires the fluid density ρ_f which you must have specified as a material property [Vol. *Material Library*].

The conversion also requires the acceleration of gravity g . For the conversion to structural, DIANA will always assume $g = 9.81\text{m/s}^2$, in a two-dimensional model working in the negative Y direction and in a three-dimensional model in the negative Z direction. The gravity acceleration vector \mathbf{g} is applied to calculate the dead weight load \mathbf{f}_w Eq. (26.1) and the buoyancy Eq. (26.2).

If you specify the model in units other than SI, then you must specify the applied units [§ 1.1 p. 3]. Otherwise, the assumed acceleration of gravity will be applied incorrectly.

Pore fluid load. The pore pressure height field, resulting from a groundwater flow analysis, causes external loading for both linear and nonlinear analysis. The existence of pore fluid causes two loading types on a porous medium like soil: dead weight and buoyancy.

The dead weight load \mathbf{f}_w depends on the porosity n , the fluid density ρ_f , the acceleration of gravity \mathbf{g} and the degree of saturation S :

$$\mathbf{f}_w = n \rho_f S \mathbf{g} \quad (26.1)$$

This dead weight load is applied as a special element load, see **WEIGHT** input in Volume *Element Library*. The buoyancy is caused by the isotropic fluid stress σ_{ii} which depends on the pressure head ϕ_p and volumetric fluid weight γ_f :

$$\sigma_{ii} = -\phi_p \gamma_f = -\phi_p \rho_f |\mathbf{g}| \quad (26.2)$$

DIANA uses the pressure-saturation diagram and the porosity for both the potential flow analysis and the calculation of the structural pore fluid load. This load is applied as a special element load, see **PRESSU** input in Volume *Element Library*. You must specify the pressure-saturation diagram and the porosity as material properties, see the **SATURA** and **POROSI** input in Volume *Material Library*.

26.4.3 Checking the Conversion

In some cases it may be instructive to see how DIANA has converted the flow-stress domain to a model for structural analysis. Therefore you may invoke Module **LINSTA** or **NONLIN** to set up the structural model and then **INPUT** to ‘remake’ appropriate input tables.

file.dcf

```
*NONLIN
MODEL
TYPE OFF
EXECUT OFF
OUTPUT OFF
*INPUT
REMAKE TABLE LOADS
*END
```

26.5 Structural Analysis

After the flow analysis you may either invoke one of the structural analysis modules **LINSTA**, **NONLIN**, **EIGEN**, or **BALANC** to perform a structural analysis. Each of these modules will convert a potential flow model and its analysis results to a model for structural analysis.

Chapter 27

Mixture Analysis

This chapter describes how to use DIANA for mixture analysis. In this context, mixture analysis means the simultaneous solution of deformation and pore pressures for the mixture of a fluid and a porous solid like soil. Typically, transient mixture analysis is used to analyse consolidation or swelling caused by mechanical loading or unloading. In transient mixture analysis, two-sided interaction is introduced if porous media are subjected to short duration load and possess a low permeability compared to the loading rate. Static mixture analysis can also be used as a substitute for staggered analysis [Ch. 26], to calculate the deformation caused by steady-state pore fluid flow directly.

27.1 Model Input

The input of the element mesh, supports and mechanical load is as usual for structural analysis. You must extend structural elements to mixture elements by specifying additional pore pressure potentials. These mixture elements require additional material input. Mixture analysis requires input of pore pressure constraints. Special features are available to incorporate gravitational load.

27.1.1 Mixture Elements

Mixture elements are plane strain, axisymmetric, and solid continuum elements with corresponding structural interface elements as described in Volume *Element Library*, but with a special ‘mixture’ option. For continuum elements, due to this option, DIANA adds a scalar pore pressure potential to the set of element degrees of freedom in each element node. For interface elements this potential is added to the nodes at one or at both sides of the element.

27.1.1.1 Combination with Structural Options

All options of structural elements in static and transient analysis, like material models, can also be applied in combination with the mixture option. There is an exception for combination with *assumed strain options*: the constant shear and constant dilatation options can be combined with the mixture option, however:

Incompatible enhanced assumed strain modes (EAS) cannot be combined with the mixture option.

See Volume *Element Library* for more information on assumed strain options.

A combination of mixture elements with structural interface elements and non-mixture elements like shells or beams can be used to analyse for example soil–structure interaction.

27.1.1.2 Extension of Structural Elements

To extend a structural element to a mixture element you must specify the mixture option in input table 'DATA'. There is a slight difference between the input syntax for continuum elements and for interface elements.

Continuum elements										<i>syntax</i>
'DATA'										
1	5	6	12	13	80					
<i>datnr_n</i>		MIXTUR								

MIXTUR indicates that the mixture option applies for all continuum elements referring to data number *datnr*, i.e., DIANA will extend these elements to mixture elements [Fig. 27.1].

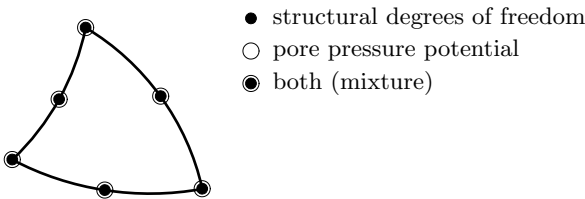


Figure 27.1: Addition of pore pressure potential to continuum elements

The combination of mixture elements with non-mixture elements is straightforward by using separate data numbers for mixture elements like in the following example.

						file.dat
'ELEMEN'						
1	Q8EPS	125	276	412	418	
...						
DATA						
/	1	3	5	/	1	
/	12	17	38	/	2	
'DATA'						
1	MIXTUR					
	NGAUS	3	3			
2	NGAUS	3	3			
...						

In this example input, elements 1, 3, and 5 will be extended to mixture elements. Elements 12, 17, and 18 will remain purely structural.

Interface elements										syntax
'DATA'										
1	5	6	12	13	80					
datnr _n		MIXTUR		[]						
				ONESID						
				TWO SID						

MIXTUR indicates that the mixture option applies for all interface elements referring to data number *datnr*, i.e., DIANA will extend these elements to mixture elements. An input option defines which side(s) of the interface element must be extended with the potential degree of freedom. With option **ONESID**, DIANA will only extend the first side to mixture, i.e., the side with the first node of the connectivity [Fig. 27.2-a]. Option **TWO SID** causes both sides to be extended to mixture [Fig. 27.2-b]. The **ONESID** option is typically applied for an interface between soil and structure. The **TWO SID** option is applied for an interface between soil and soil, to model hydraulic fracture in joints or connections in phased analysis [Ch. 29].

[ONESID]

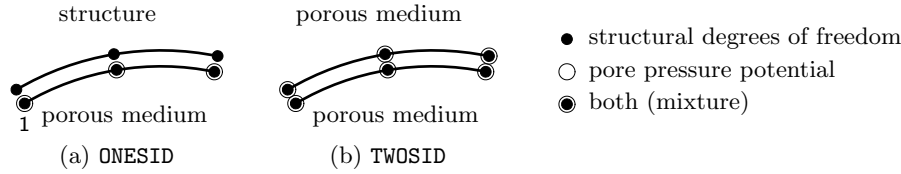


Figure 27.2: Addition of pore pressure potential to interface elements

file.dat

```

'ELEMEN'
  1  CL20I   61 62 63 64 65 71 72 73 74 75
...
DATA
/  1  3  5 / 1
/ 12 17 38 / 2
'DATA'
  1  MIXTUR ONESID
  2  MIXTUR TWOSID
...

```

In this example input, elements 1, 3, and 5 will be extended to mixture elements at side one only. Elements 12, 17, and 18 will be extended at both sides. The other elements will remain purely structural.

Material properties. For mixture elements, you must supplement the material properties for structural analysis with special properties to describe the properties of the porous medium. See Volume *Material Library* for input description.

27.1.2 Input of Pore Pressure Potential

DIANA uses the pore pressure potential ϕ as the basic degree of freedom in the nodes. You can only specify values for the pore pressure potential to input pore pressure constraints. During analysis, DIANA composes the pore pressure potential ϕ in a fluid pressure p and a gravitational part $-\rho_f \mathbf{g} (\mathbf{x} - \mathbf{x}_{\text{ref}})$ [§ 56.1.6 p. 633]. DIANA can supply output of the pore pressure potential ϕ , the pore pressure p , and the excess pore pressure p_e .

Input for the pore pressure potential is specified via the regular input tables for structural stress analysis [Ch. 2]. You must specify the nodes with an imposed pore pressure potential in table 'SUPPOR'. Constraints for the nodal pore pressure potential are specified in table 'LOADS' analogously to the constraints for a displacement: nonzero values in subtable DEFORM and nodal discharge in subtable NODAL.¹ Moreover, linear constraints for pore pressure potentials may be specified in table 'TYINGS'.

¹DIANA-10.1 does not support imposed specific discharge on element faces.

Imposed potential*syntax*

```

'SUPPOR'
NAME names
1      5 6
----- 80
noden    PR
1      5 6
----- 80
/ nodesng... /
      PR

```

NAME *name* is the name for the support set and may be used for identification of and reference to the support set. A name specification line must follow immediately after the table heading. The name specification may be respecified at the start of a new subtable.

node is a single node number, *nodes* is a series of nodes which must be specified between slashes and may comprise numbers or groups or both.

PR is the degree of freedom type that indicates an imposed pore pressure potential for the specified *node*(*s*).

Nonzero potential*syntax*

```

'LOADS'
DEFORM
1      5 6
----- 80
noden    PR pr
1      5 6
----- 80
/ nodesng... /
      PR pr
1      5 6
----- 80
/ nodesng... /
      PR / psr... /

```

node is a single node number, *nodes* is a series of nodes which must be specified between slashes and may comprise numbers or groups or both.

PR is the degree of freedom type which indicates a pore pressure potential.

p is the prescribed value of the pore pressure potential ϕ , *ps* is a series of values, one for each node in *nodes*.

Nodal discharge*syntax*

```

'LOADS'
NODAL
1      5  6      80
noden   PR qr
1      5  6      80
/ nodesng... /
      PR qr
1      5  6      80
/ nodesng... /
      PR / qsr... /

```

node is a single node number, *nodes* is a series of nodes which must be specified between slashes and may comprise numbers or groups or both.

PR is the degree of freedom type which indicates a pore pressure potential.

q is the value *Q* of the nodal discharge, *qs* is a series of values, one for each node in *nodes*.

Nonzero potential and nodal discharge are considered to be 'loads' and therefore must be related to a load case, like in the following example.

file.dat

```

'SUPPOR'
NAME PRESSURES
/ 3-27 / PR
'LOADS'
CASE 1
DEFORM
5      PR 2.57
CASE 2
NODAL
/ 55-60 / PR -12.36

```

Linear constraints. Linear constraints (tyings) are mainly used to keep pore pressure potentials of different nodes mutually equal. See § 2.2 on page 16 for general syntax description. For the pore pressure potential, the degree of freedom must be specified as type PR without a direction number, like in the following example.

file.dat

```

'TYINGS'
NAME "Equal pressures"
EQUAL PR
2      12
10     / 3-6 /

```

In the EQUAL subtable you must specify sets of a master and slave nodes.

27.1.3 Input of Weight Load

The gravity vector \mathbf{g} will generally cause a distributed dead weight load $\rho \mathbf{g}$ with ρ being the total mass density. In mixture analysis, the gravitational part of the pore pressure potential $\rho_f \mathbf{g} (\mathbf{x} - \mathbf{x}_{\text{ref}})$ induces an additional buoyancy force, which results in weight reduction for the porous medium and an hydrostatic normal force on the boundaries, with $\mathbf{x} - \mathbf{x}_{\text{ref}}$ the position vector.

DIANA includes the buoyancy force automatically, if you specify a gravity load and a fluid density. A regular gravity load specification, as described in § 2.3.2 on page 33, can be extended with the origin \mathbf{x}_{ref} of the position vector, which is used as the reference point for the total head definition [§ 1.2 p. 6].

syntax

```
'LOADS'
WEIGHT
```

WEIGHT is the subtable heading for dead weight load.

The gravity acceleration g , gravity direction, fluid density ρ_f , and the origin of the position vector \mathbf{x}_{ref} are defined in table 'MODEL' [§ 1.2 p. 6].

27.2 Preliminary Linear Analysis

In static analysis, the time derivatives are zero. This means that the two-sided coupling reduces to single-sided coupling: *flow influences stress but stress does not influence flow*.

An exclusive static linear elastic analysis must be performed, if one-directional interaction is analysed and if the model behaves linearly. A preliminary static linear elastic analysis for a transient analysis may be performed to check the model and to calculate an initial stress and flow field.

The commands for linear static analysis are as described in Chapter 4, with an additional option to output pore pressure potentials: the PRESSU option in the DISPLA command as shown below.

file.dcf

```
*FILOS
INITIA
*INPUT
*LINSTA
BEGIN OUTPUT
  DISPLA PRESSU PH
END OUTPUT
*END
```

The PH component of the PRESSU option gives output of the pore pressure potential ϕ [§ 27.4.1.1 p. 394].

27.3 Transient and Nonlinear Analysis

For the coupled analysis of stress and pore fluid flow, a nonsymmetric system of nodal displacements \mathbf{u} and pore pressure potentials ϕ is solved simultaneously. For direct time integration in transient analysis and/or analysis of nonlinearity, a stepwise method is available from Module NONLIN, with iteration in case of nonlinearity. You may use a static nonlinear analysis for instance to analyse the stability of a model subjected to ground water flow. A transient analysis is suitable for analysis of consolidation, liquefaction, or swelling of soil.

27.3.1 Initialization

The stepwise analysis is initialized for Module NONLIN with TYPE, and EXECUT START commands. The commands indicate whether there is nonlinearity in the model, whether a non-default time integration scheme must be used, and whether an initial stress field must be used. The following is an example of such commands.

file.dcf

```

*NONLIN
BEGIN TYPE
  PHYSIC
  METHOD HHT
END TYPE
BEGIN EXECUT
  BEGIN START
    LOAD LOADNR=1
    INITIA STRESS CALCUL LOAD=1 FACTOR=1.0
  END START
END EXECUT

```

The **PHYSIC** command in the **TYPE** block indicates physical nonlinear analysis.

Initial load. In geotechnical analysis it is common practice to initialize the drained stress and flow field, using the previously calculated elastic results due to dead weight load. Additionally, DIANA also allows you to derive the initial stress and (excess) pore pressure from a preceding nonlinear analysis phase. In new elements, the stress and pore pressure potential field are calculated from the specified linear load set with the **INITIA STRESS CALCUL LOAD=1 FACTOR=1.0** command. The corresponding dead weight load is applied with the **LOAD LOADNR=1** command. During stress initialization, DIANA adapts the initial stress in new elements according to the lateral pressure ratio K_0 .

Time integration. In mixture analysis, the **METHOD** command is primarily used to specify the direct time integration scheme, like in the above example with **HHT** for the Hilber–Hughes–Taylor scheme [§ 13.2.3 p. 217]. For theoretical backgrounds of time integration with respect to mixture analysis see § 56.2.2 on page 635.

27.3.2 Initial Equilibrium

During initialization with results of a linear analysis, in most cases no deformation will occur due to appliance of dead weight. However, in case of nonlinearity (plasticity, partial saturation) or if the initial stress field is no longer balanced by the load (phased analysis), then initial deformation will occur and the initial stress and (excess) pore pressure field will alter to restore equilibrium during following static load steps. These load steps can be performed with the **START STEPS** option of the **EXECUTE** command block [§ 13.3 p. 219].

Suppression of strain superposition

file.dcf

```

*NONLIN
...
BEGIN EXECUT
  BEGIN PHYSIC
    SUPPRE STRAIN
  END PHYSIC
  ...
END EXECUT

```

DIANA offers a special option to suppresses superposition of displacements and strains during the establishment of the initial equilibrium. The **SUPPRE STRAIN** command suppresses superposition of displacements and strains during the establishment of the initial equilibrium [§ 13.3.4 p. 234]. Only the stress state will be affected in this case.

Initial state determination

file.dcf

```

*NONLIN
...
BEGIN EXECUT
  BEGIN START

```

```

BEGIN STEPS
  EXPLIC SIZES 1.0
END STEPS
END START
REFERE PRESSU WEIGHT
BEGIN PHYSIC
  SUPPRE STRAIN
  DRAINE
END PHYSIC
END EXECUT

```

With the EXECUT START STEPS command block you can apply stepwise a possible unbalanced part of the initial load [§ 13.3.1.2 p. 223]. During the first step, DIANA will implicitly adapt the initial stress and pore pressure in case of initial nonlinearity. During all load steps, the behaviour is default completely drained, except for layers with material property UNDRAI specified in the input data. You can enforce drained behaviour in all layers by using the keyword DRAINE in the PHYSIC command block [§ 13.3.4 p. 235]. Due to the REFERE PRESSU WEIGHT command DIANA will re-apply the K_0 procedure to new elements after step 1, and use the pore pressure after step 1 as the reference p_{ref} for the calculation of excess pore pressures $p_e = p - p_{\text{ref}}$ [§ 13.3.8 p. 244].

27.3.3 Load and Time Steps

By default, elements with the ‘undrained’ material property will behave completely undrained during the execution of load and time steps. Elements without this property will behave drained during load steps and will show transient pore pressure dissipation during time steps. However, you may overrule the ‘undrained’ material property by specifying the DRAINE command for execution of time steps [§ 13.3.4 p. 235]. In this case, DIANA will switch off the undrained behaviour for all elements; even for elements with ‘undrained’ material property. The time dependency of the load sets, using multilinear diagrams, must be specified in input Table ‘TIMELO’

file.dat

```

'TIMELO'
LOAD 1
TIMES 1.0 1.1 1000.0 /
FACTOR 0.0 1.0 1.0 /

```

27.3.3.1 Static Load Steps

The LOAD command block within the EXECUT command block may be used to apply static load steps. During static load steps the behaviour is completely drained or completely undrained. The following is an example of such commands.

file.dcf

```

*NONLIN
...
BEGIN EXECUT
  BEGIN LOAD
    LOADNR=1
    BEGIN STEPS
      EXPLIC SIZES 0.2(5)
    END STEPS
  END LOAD
END EXECUT
BEGIN EXECUT
  BEGIN LOAD
    LOADNR=1
    BEGIN STEPS

```

```

    EXPLICIT SIZES 0.00000001
  END STEPS
END LOAD
BEGIN PHYSIC
  DRAINE
END PHYSIC
END EXECUT

```

The first EXECUT LOAD block uses the default behaviour (undrained in layers with the material property UNDRAI, drained in the other layers). The second EXECUT LOAD block uses drained behaviour in all layers.

27.3.3.2 Time Steps

Time steps are executed via the EXECUT TIME commands [§ 13.3.3 p. 231]. You must specify the PHYSIC DRAINE command to enforce pore pressure dissipation in mixture elements with the material property UNDRAI. The following is an example of such commands.

file.dcf

```

*NONLIN
...
BEGIN EXECUT
  BEGIN TIME
    BEGIN STEPS
      EXPLICIT SIZES 0.004(10) 0.008(10)
    END STEPS
  END TIME
BEGIN PHYSIC
  DRAINE
END PHYSIC
END EXECUT

```

27.4 Output of Analysis Results

In a DIANA Mixture analysis, the pore pressure potentials ϕ are the basic system degrees of freedom in the nodes, from which DIANA derives the total pore pressure p and excess pore pressure p_e . DIANA can also output the corresponding discharges in the nodes. For mixture elements with ‘undrained’ material property you can get pressures at element level. Also at element level, you can get the saturation and the Darcy flux.

syntax

```

[ *NONLIN ]
BEGIN OUTPUT [ device_w ] [ outopt_w... ] [ params ]
[ ... ]
  item_w ...
  DISPLA
  VELOC
  ACCELE
  PRESSU
  FORCE
  SATURA
  DARCYF
END OUTPUT

```

The following describes only the special commands for Module NONLIN to get output of mixture analysis results.

... See § 13.4 on page 246 for general output commands for Module NONLIN.

item is the name of the result item to be output. See § 3.6.1 on page 56 for complete syntax of this command.

DISPLA gives the pore pressure potentials ϕ [F/L²] which are the basic system degrees of freedom in the nodes [§ 27.4.1.1].

VELOCI specifies the first time derivative of the pore pressure potential ϕ [§ 27.4.1.1].

ACCELE specifies the second time derivative of the pore pressure potential ϕ [§ 27.4.1.1].

FORCE gives the corresponding discharges in the nodes [§ 27.4.2].

PRESSU gives the pressures at element level for mixture elements [§ 27.4.1.2].

SATURA gives the saturation at element level [§ 27.4.3].

DARCYF gives the Darcy flux at element level [§ 27.4.4].

27.4.1 Pressures

There are two types of pressures available for output: the basic degrees of freedom in the nodes and the pressures in elements.

27.4.1.1 Nodal Pressure Potentials

syntax

<i>item</i> _{<i>w</i>}	[<i>type</i> _{<i>w</i>}]	[<i>form</i> _{<i>w</i>}]	{ <i>comp</i> _{<i>w</i>} }	{ <i>opti</i> _{<i>w</i>} }
DISPLA	TOTAL	PRESSU	PH	...
VELOCI	INCREM			
ACCELE	PHASE			

DISPLA specifies the system degrees of freedom at the nodes as output item.

VELOCI specifies the first time derivative of the pore pressure potential ϕ at the nodes as output item.

ACCELE specifies the second time derivative of the pore pressure potential ϕ at the nodes as output item.

[TOTAL] *type* specifies the type: TOTAL for the total value of the system degrees of freedom, INCREM for the incremental values, PHASE for the phased values, i.e. the change between two consecutive phases in a phased mixture analysis.

form specifies the formulation of the system degrees of freedom: PRESSU for the pore pressure potential components. Without this formulation DIANA gives the translational displacements.

comp selects components for output.

PH for the pore pressure potential ϕ .

<i>item</i>	<i>type</i>	<i>form</i>	<i>comp</i> ... PH
DISPLA	TOTAL	PRESSU	TDprph ϕ
DISPLA	INCREM	PRESSU	IDprph $\Delta\phi$
DISPLA	PHASE	PRESSU	PDprph ϕ
VELOCI	TOTAL	PRESSU	TVprph $\dot{\phi}$
VELOCI	INCREM	PRESSU	IVprph $\Delta\dot{\phi}$
VELOCI	PHASE	PRESSU	PVprph $\dot{\phi}$
ACCELE	TOTAL	PRESSU	TAprph $\ddot{\phi}$
ACCELE	INCREM	PRESSU	IAprph $\Delta\ddot{\phi}$
ACCELE	PHASE	PRESSU	PAprph $\ddot{\phi}$

27.4.1.2 Element Pressures

For mixture elements, DIANA can give output of pore pressure components at the element level.

syntax

```

PRESSU [ typew ] { compw } { locaw } { optiw }
          TOTAL      PO          INTPNT  ...
                      EX          NODES
                      CENTER

```

PRESSU specifies the calculated pressure for undrained behaviour as output item.

type specifies the type: TOTAL applies for pressure due to undrained behaviour, the only possibility.

[TOTAL]

comp selects components for output.

PO for the total pore pressure:

- $p = K\Delta V$ (undrained)
- $p = \phi + \rho_f \mathbf{g}(\mathbf{x} - \mathbf{x}_{\text{ref}})$
- $p = \max(p, 0)$ (pressure dependent saturation)

EX for the excess pore pressure: $p_e = p - p_{\text{ref}}$.

<i>item</i>	<i>type</i>	<i>comp</i> ...	
		PO	EX
PRESSU	TOTAL	PRpo	PRex
		p	p_e

27.4.2 Nodal Discharges

syntax

```

FORCE [ typew ] [ formw ] { optiw }
      EXTERN  PRESSU  ...
      REACTI
      RESIDU

```

FORCE specifies the nodal discharge as output item. These nodal discharges correspond to the basic degrees of freedom.

[REACTI] *type* specifies the type: **EXTERN** for the externally applied discharges, **REACTI** for the reaction values (the constraints), **RESIDU** for the residuals (in each node).

form specifies the formulation of the discharges: **PRESSU** for the pressures. Without this formulation DIANA gives the translational forces.

<i>item</i>	<i>type</i>	<i>form</i>	
FORCE	EXTERN	PRESSU	QTpr Q
FORCE	REACTI	PRESSU	QBpr Q^B
FORCE	RESIDU	PRESSU	QRpr Q^R

27.4.3 Saturation

syntax

SATURA [*loca_w*] { *opti_w* }
 INTPNT ...
 NODES
 CENTER

SATURA specifies the element saturation as output item.

<i>item</i>	
SATURA	S S

27.4.4 Darcy Flux

syntax

DARCYF [*oper_w*] { *comp_w* } { *loca_w* } { *opti_w* }
 LOCAL ... INTPNT ...
 GLOBAL NODES
 CENTER

DARCYF specifies the Darcy flux in the element as output item.

<i>item</i>	<i>oper</i>	<i>comp</i> ...		
		X	Y	Z
DARCYF	LOCAL	Qx	Qy	Qz
		q_x	q_y	q_z
DARCYF	GLOBAL	QX	QY	QZ
		q_X	q_Y	q_Z

Part VIII

Phased Analysis

Introduction to phased analysis. Phased analysis enables modelling of phased construction. It determines the effects of construction history and shows the critical construction stages.

A phased analysis comprises several calculation phases. Between each phase the finite element model changes by addition or removal of elements and constraints. In each phase a separate analysis is performed, in which the results from previous phases are automatically used as initial values. These results are typically stresses, deformations, potentials, velocities etc.

The start of each phase can include input of the model part which is changed compared to previous input or added. At the start of each phase, you must select the active part of the model and specify the superposition of the nodal results (displacements or potentials) from previous phases. After the start you may perform a common analysis using regular DIANA analysis modules.

DIANA can perform phased analysis for linear, nonlinear and dynamic structural analysis [Ch. 28], and for transient potential flow analysis [Ch. 29]. Also a phased staggered flow–stress analysis is possible [Ch. 26].

Chapter 28

Phased Structural Analysis

In phased structural analysis the model may change from phase to phase. For instance, in each new phase elements and reinforcements may become active or inactive at the users' request, or supports may be removed or added. This chapter describes the impact of phased analysis on model input [§ 28.1] and initialization of a new phase [§ 28.2 p. 404]. Especially for novice users we explain how DIANA increments analysis results from phase to phase [§ 28.4 p. 409].

Before starting a phased structural analysis it is important to note that switching from linear to nonlinear analysis, or vice versa, is not allowed. So you must use one of the two following command sequences.

Linear analysis

file.dcf

```
*FILOS
INITIA
*INPUT
commands
*PHASE
commands
*LINSTA
commands
*INPUT      optional!
commands
*PHASE
commands
*LINSTA
commands
...
```

Nonlinear analysis

file.dcf

```
*FILOS
INITIA
*INPUT
commands
*PHASE
commands
*NONLIN
commands
*INPUT      optional!
commands
*PHASE
commands
*NONLIN
commands
...
```

28.1 Input of the Finite Element Model

In phased analysis you must start with reading the model data into the FILOS file with Module INPUT. After input of the model, but prior to the actual analysis of a phase, you must perform the initialization of the phase with Module PHASE as described in § 28.2 on page 404.

28.1.1 Read First Model

Via Module INPUT you must read the model for the first phase. The minimal input applies to the active part of the mesh and the actual boundary conditions. Input may additionally contain elements and nodes connected to inactive parts, which may be activated in subsequent phases.

file.dcf

```
*INPUT
```

28.1.2 Adapt Model

Each following phase it may be necessary to adapt the model on the FILOS file. For instance, changes in boundary conditions often require adaptation of the model data and changes in the active part of the mesh may require such adaptation if the new elements and nodes were not supplied before. However, you should note the following.

During a phased analysis the integration points of elements must keep their location.

Consequently, you may not alter element properties which affect the location of the integration points. For instance, the node coordinates, the cross-section of beam elements, and the thickness of curved shell elements must all remain unaltered during a phased analysis. Module INPUT offers three possibilities to adapt the model data on the FILOS file:

- The single READ command replaces an existing table by a new one.
- The APPEND option of the READ command appends additional data to an existing table.
- The DELETE command deletes a table from the FILOS file.

See § 3.3 on page 50 for formal syntax description of these commands. Phased analysis does not affect the already read tables, which are not appended or read again.

file.dcf

```
*INPUT
READ TABLE SUPPOR LOADS
READ APPEND TABLE COORDI ELEMEN
DELETE TABLE TYINGS
```

28.1.3 Import Deformation

You may import the displacements of nodes resulting from the previous phase as constraints (fixed displacements) in a subsequent phase. Then you must supply a special form of subtable DEFORM of table 'LOADS' which specifies how to import displacements.

syntax

```

'LOADS'
CASE casen
DEFORM IMPORT
      LODSET losetn
      NONLIN

```

1	5	6		80
---	---	---	--	----

```

snoden  [typew] [DI=dirnrn]
          [mnoden]

```

1	5	6		80
---	---	---	--	----

```

/ snodesng... /
    [typew] [DI=dirnrn]
    [/ mnodesng... /]

```

DEFORM is the heading for the subtable with fixed displacements. Keyword IMPORT indicates that the deformations come from the previous phase. LODSET specifies that deformations come from the linear static analysis in the previous phase, where *loset* is the load set number. NONLIN specifies that deformations come from the stepwise analysis inside the previous phase.

snode is a supported slave node for which the values of fixed displacement must be imported, *snodes* are a series of supported slave nodes specified by numbers and/or group names.

type is the selected type of the fixed displacements, TR for translations only, RO for rotations only. Default: all supported displacement types in the slave nodes.

DI=*dirnr* is the selected direction of the fixed displacements, *dirnr* refers to a direction in table 'DIRECT'. Default: all supported directions of the displacement type in the slave nodes.

mnode is the master node, *mnodes* are a series of master nodes. Imported displacements of the master nodes are transferred to fixed displacements of the slave nodes. Default: if no master nodes are specified, DIANA assumes that the master nodes and slave nodes are equal.

Example*file.dat*

```

'DIRECT'
  1   1.  0.  0.
  2   0.  1.  0.
  3   0.  0.  1.
'SUPPOR'
NAME CONSTRAINTS
  982  TR  2  RO  3
  983  TR  1  TR  2
  984  TR  1  TR  2  RO  3
  986  TR  1  TR  2  RO  3
'LOADS'
CASE 1
DEFORM IMPORT LODSET 1
  982  TR  DI=2
      981
  982  RO  DI=3
      981
  983
  984
      985
  986  TR

```

```

985
CASE 2
DEFORM IMPORT NONLIN
...

```

Table 'DIRECT' specifies the Model *XYZ*-directions. Load case 1 imports the displacements from load set 1 of the linear static analysis in the previous phase: $u_Y^{982} = u_Y^{981}$, $\phi_Z^{982} = \phi_Z^{981}$, all fixed translations of node 983 from node 983, all fixed translations and rotations of node 984 from node 985 and all fixed translations of node 986 from node 985. Load case 2 imports the displacements from the stepwise analysis in the previous phase.

28.1.4 Initial Velocity

Stepwise dynamic analysis is an extension of stepwise transient static analysis with momentum contributions using masses and nodal velocities. The initial nodal velocities of an element determine the initial momentum of the connected nodes. By default, DIANA uses the element velocities of the last phase in which the element was active as initial velocities of the active phase. DIANA applies the following decision scheme to determine the initial velocities in phased analysis, if non-default initial values are required.

- If specified, initial velocities of the element are taken from the table 'DATA'.

<i>syntax</i>									
'DATA'									
1	5	6	12	13					
		[VELMOD		$v1_r$	$v2_r$...	vn_r]		

VELMOD specifies initial element velocities: $v1$ for the first degree of freedom in model *XYZ*-orientation, $v2$ for the second, etc. up to vn for the last. The sequence of the degrees of freedom is as follows (with n is the number of element nodes):

$$\{u_X^1 u_Y^1 u_Z^1 u_X^2 u_Y^2 u_Z^2 \dots u_X^n u_Y^n u_Z^n \dots \phi_X^1 \phi_Y^1 \phi_Z^1 \phi_X^2 \phi_Y^2 \phi_Z^2 \dots \phi_X^n \phi_Y^n \phi_Z^n\}$$

In words: first *XYZ* translations for element nodes, then *XYZ* rotations for element nodes. Translations or rotations according to the basic displacements for the element. For instance only translations u for a Q8MEM element but translations u and rotations ϕ for a CQ40S element [Vol. *Element Library*]. Condition: one velocity for each degree of freedom.

This input is typically used whenever active elements have some well known initial velocities like a zero velocity.

- If the input table 'INIVAR' is present and VELMOD is not, then the initial velocities from this table will be used.
- If you do not specify an initial velocity, then DIANA uses the element velocities of the last phase in which the element was active, with zero initial velocities for new elements.

28.2 Phase Initialization

To initialize a new phase, you must give commands to Module PHASE.

<i>syntax</i>									
*PHASE									
[BEGIN ACTIVE									
		[ELEMEN		$elems_{s\dots}$...]...			
		[REINFO		$reins_{s\dots}$...]...			

```

[ SUPPOR suppss... ] ...
[ _____ ] ...
  FIXTEM ftemss...
  FIXHEA fheass...
  FIXPOT fpotss...
[ TYINGS tygss... ] ...
END ACTIVE ]
[ BEGIN RESFOR
[ ELEMEN elemss... ]
  FACTOR factorr
END RESFOR ] ...
SUPERP
  ALL
  NONE
  IMPORT

```

ACTIVE selects a certain part of the model to be active in the new phase. By default, DIANA activates the complete model, as read previously [§ 28.1]. With commands in an **ACTIVE** block you may select only a part of the model to be active.

ELEMEN specifies active elements where *elems* is a series of element sets names [§ 28.2.1]. If you do not specify active elements, then DIANA will assume all elements available from input to be active.

REINFO specifies active reinforcements where *reins* is a series of reinforcement set names [§ 28.2.2]. If you do not specify active reinforcements, then DIANA will assume all the reinforcement parts in active elements of all reinforcements available from input to be active.

SUPPOR specifies active support sets where *supps* is a series of support set names [§ 28.2.3]. If you do not specify active support sets, then DIANA will assume all the support sets available from input to be active.

FIXTEM specifies active fixed temperature sets in a heat flow analysis, where *ftems* is a series of fixed temperature set names [§ 28.2.4]. If you do not specify active fixed temperature sets, then DIANA will assume all the fixed temperature sets available from input to be active.

FIXHEA specifies active fixed head sets in a groundwater analysis, where *fheas* is a series of fixed head set names [§ 28.2.5]. If you do not specify active fixed head sets, then DIANA will assume all the fixed head sets available from input to be active.

FIXPOT specifies active fixed potential sets in potential flow analysis other than heat or groundwater flow, where *fpots* is a series of fixed potential set names [§ 28.2.6]. If you do not specify active fixed potential sets, then DIANA will assume all the fixed potential sets available from input to be active.

TYINGS specifies active tying sets where *tygs* is a series of tying set names [§ 28.2.7]. If you do not specify active tying sets, then DIANA will assume all the tying sets available from input to be active.

RESFOR selects a certain part of the inactive model. For this part the residual forces of the inactivated elements will act as an external load on the elements that are active in the new phase. By default no elements are selected. With commands in a **RESFOR** block you may select inactive parts of the model to contribute to the external forces on active elements. Note that the stiffness of the deactivated elements is not contributing in the new phase.

ELEMEN specifies the inactive elements for which the residual forces must act as an external load on the active elements. *elems* is a series of element set names of the inactive elements.

FACTOR specifies the fraction of the residual element forces that is applied as external load on active elements.

[ALL] **SUPERP** specifies how nodal results from the previous phase shall be superposed to the phased results. This command only applies for linear static analysis. In nonlinear analysis a similar effect can be achieved via the **SUPPRE** command [§ 13.3.4 p. 234].

ALL will add the incremental displacements (nodal analysis results) of a new phase to the total displacements of the previous phases.

NONE suppresses the superposition of incremental displacements to the total displacements in structural analysis. This is particularly useful to determine initial stresses due to dead weight load.

IMPORT suppresses the superposition of the incremental phased displacements as specified in table 'LOADS' [§ 28.1.3]. In this way, the correct total phased displacement is obtained while assembling parts of the model.

By default, DIANA will evaluate the changed elastic material properties for existing elements and renew their stiffness, mass and damping matrices in a phased analysis.

file.dcf

```
*PHASE
BEGIN ACTIVE
  ELEMEN SHELLS
  REINFO BAR1
END ACTIVE
SUPERP ALL
```

file.dcf

```
*PHASE
BEGIN ACTIVE
  ELEMEN SHELLS BEAMS
END ACTIVE
SUPERP NONE
```

28.2.1 Element Activation

The **ELEMEN** command block is a sub-block in the **ACTIVE** block to activate elements for phased analysis.

syntax

```
BEGIN ELEMEN [ _____ ]
               elemss...
               ALL
[ MATERI matnrn ]
END ELEMEN
```

[ALL] **ELEMEN** element activation: *elems* is a series of elements specified by element set names. ALL indicates all elements.

MATERI can be used to redefine the element material to *matnr* for the active elements.

28.2.2 Reinforcement Activation

The **REINFO** command block is a sub-block in the **ACTIVE** block to activate reinforcements for phased analysis.

syntax

```

BEGIN REINFO [ _____ ]
               reins s...
               ALL
               NONE
[ MATERI matnrn ]
END REINFO

```

REINFO reinforcement activation: *reins* is a series of reinforcements specified by reinforcement sets names. ALL indicates all reinforcement sets. NONE indicates no reinforcements at all. [ALL]

MATERI can be used to redefine the reinforcement material to *matnr* for the active reinforcements.

28.2.3 Support Activation

The SUPPOR command block is a sub-block in the ACTIVE block to activate supports for phased analysis.

syntax

```

BEGIN SUPPOR [ _____ ]
               supps s...
               ALL
               NONE
END SUPPOR

```

SUPPOR suppor activation: *supps* is a series of support set names. ALL indicates all supports. NONE indicates no supports at all. [ALL]

28.2.4 Fixed Temperature Activation

The FIXTEM command block is a sub-block in the ACTIVE block to activate fixed temperatures for phased heat flow analysis.

syntax

```

BEGIN FIXTEM [ _____ ]
               ftems s...
               ALL
               NONE
END FIXTEM

```

FIXTEM fixed temperature activation: *ftems* is a series of fixed temperature set names. ALL indicates all fixed temperatures. NONE indicates no fixed temperatures at all. [ALL]

28.2.5 Fixed Head Activation

The FIXHEA command block is a sub-block in the ACTIVE block to activate fixed heads for phased groundwater flow analysis.

syntax

```

BEGIN FIXHEA [ _____ ]
               fheas s...
               ALL
               NONE
END FIXHEA

```

FIXHEA fixed head activation: *fheas* is a series of fixed head set names. ALL indicates all fixed heads. NONE indicates no fixed heads at all. [ALL]

28.2.6 Fixed Potential Activation

The FIXPOT command block is a sub-block in the ACTIVE block to activate fixed potentials for phased flow analysis other than heat or groundwater flow.

syntax

```
BEGIN FIXPOT [ _____ ]
               fpotss...
               ALL
               NONE
END FIXPOT
```

- [ALL] FIXPOT fixed potential activation: *fpots* is a series of fixed potential set names. ALL indicates all fixed potentials. NONE indicates no fixed potentials at all.

28.2.7 Tying Activation

The TYINGS command block is a sub-block in the ACTIVE block to activate tyings for phased analysis.

syntax

```
BEGIN TYINGS [ _____ ]
               tygss...
               ALL
               NONE
END TYINGS
```

- [ALL] TYINGS tying activation: *tygs* is a series of tying set names. ALL indicates all tyings. NONE indicates no tyings at all.

28.2.8 Reinforcement

One reinforcement section generally consists of a number of particles, embedded in elements [Vol. *Element Library*]. DIANA generates element-by-element locations from global location input for new reinforcements, using the locations of all available elements. DIANA also performs a geometry evaluation for new particles of active reinforcements in active elements.

You may add new reinforcement to existing elements by activation in Module PHASE. DIANA will implicitly add particles of existing reinforcements in newly activated elements, provided that the previously generated element-by-element location already included these elements.

Changes in geometrical properties (location, cross-section) of existing reinforcements are not allowed. If you adapt any material properties of existing reinforcements in the input data, then DIANA will evaluate the changed elastic material properties to set up the stiffness contributions for all active reinforcement parts and renew the stiffness matrices of the active elements.

28.2.9 Displacement Constraints

When initializing for a new phase, DIANA checks and processes all specified supports, tyings and prescribed displacements from the appropriate tables on the FILOS file for the active nodes. In other words: in a new phase DIANA will create a completely new set of degrees of freedom. Input tables which affect the creation of degrees of freedom are 'SUPPOR', 'TYINGS' and 'LOADS'. For subtable DEFORM of table 'LOADS' you must realize the following.

In phased analysis, DIANA interprets the specified prescribed displacements as total loads.

You may add new constraints to the model by appending data to the appropriate tables on the FILOS file [§ 28.1.2].

28.2.10 Loads

When initializing for a new phase, DIANA checks and processes all specified external loads (concentrated nodal loads, element loads, weight) and internal loads (temperature and prestress) from the FILOS table 'LOADS' for the active nodes, elements and reinforcements.

In phased analysis, DIANA interprets both the specified external and internal loads as total loads.

You may add new loads to existing loads by appending data to the table 'LOADS' on the FILOS file. If you want to add, adapt or delete loads on currently active nodes, elements or reinforcements, you must reread or delete table 'LOADS'.

28.3 Output of Displacements

In linear and nonlinear phased analysis, there is a special option for the DISPLA output item to get output of the incremental displacements of a phase. See § 3.6.1 on page 56 for general syntax.

syntax

```
*modulew
  LINSTA
  NONLIN
  ...
BEGIN OUTPUT devicew [ options ]
  DISPLA [ typew ] ...
    PHASE
    ...
END OUTPUT
```

PHASE gives the displacement increment in the active phase only. If you do not specify this option then DIANA will output the superposed total displacements after the active phase.

28.4 Incrementing Analysis Results

To explain the basics of phased analysis we will first describe the procedure for an unchanged model, i.e., elements, reinforcements and supports active in the first phase will remain active in all subsequent phases, neither elements, nor reinforcements, nor supports will ever be added to the model during the phased analysis [§ 28.4.1]. Then we will explain the analysis procedure for a changing model [§ 28.4.3].

28.4.1 Unchanged Model

To illustrate the way of thinking for incremental loading in phased analysis we will use a very simple model: a bar fully clamped at one end, and for a certain analysis phase i loaded with an axial force load $F_{(i)}$ at the other end [Fig. 28.1].

Phase i . To achieve an equilibrium state, i.e., in the direct analysis of phase i , the force causes a displacement $u_{(i)}$ of the loaded end. From the displacement $u_{(i)}$ DIANA derives the strain $\varepsilon_{(i)}$ and then the stress $\sigma_{(i)}$. Formally we could denote the direct analysis process of phase i as

$$F_{(i)} \Rightarrow \begin{cases} u_{(i)} \\ \varepsilon_{(i)} \\ \sigma_{(i)} \end{cases} \quad (28.1)$$

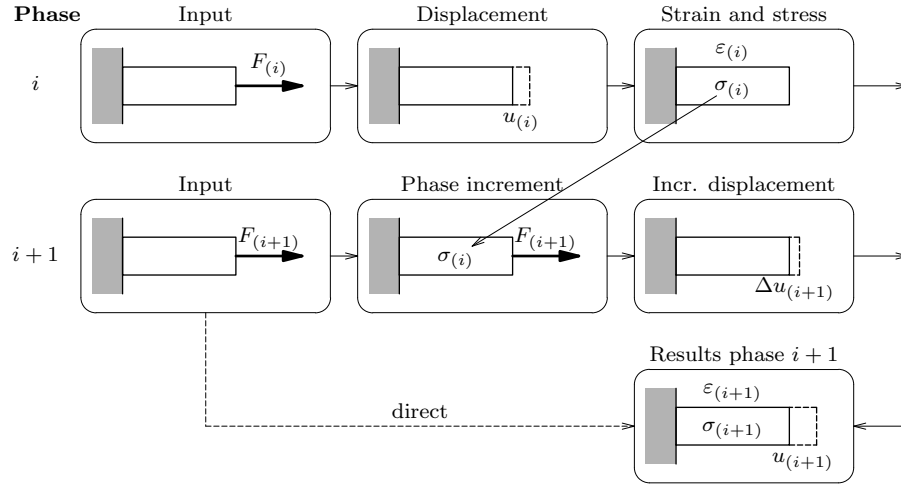


Figure 28.1: Phases with unchanged model

Phase $i+1$. Suppose that in the subsequent phase the load increases with $\Delta F_{(i+1)}$ to a total of $F_{(i+1)}$. When performing a phased analysis with DIANA it is important to know that

In each analysis phase, you must apply the total load on the model.

That is because during the analysis of the new phase, DIANA applies not only the total load $F_{(i+1)}$, but also the strain $\varepsilon_{(i)}$ and the stress $\sigma_{(i)}$ from the previous phase which could be considered as an initial tension stress, i.e., a prestress load. The force causes a displacement $u_{(i+1)}$, and the stress a displacement $-u_{(i)}$ which is a compression equal to the elongation in the previous phase. These two together give an incremental displacement $\Delta u_{(i+1)} = u_{(i+1)} - u_{(i)}$.

In phased analysis, the primary analysis results of a new phase are the incremental displacements.

From the incremental displacement $\Delta u_{(i+1)}$ DIANA derives the incremental strain $\Delta \varepsilon_{(i+1)}$ and the incremental stress $\Delta \sigma_{(i+1)}$. The primary analysis process of phase $i+1$ thus yields the *incremental* results

$$\text{Increment: } F_{(i+1)} - F_{\sigma_{(i)}} \Rightarrow \begin{cases} \Delta u_{(i+1)} \\ \Delta \varepsilon_{(i+1)} \\ \Delta \sigma_{(i+1)} \end{cases} \quad (28.2)$$

To terminate a phase in the analysis, DIANA adds the incremental results Eq. (28.2) to the total results of the previous phase Eq. (28.1), which yields the total results of the current phase:

$$\text{Total: } F_{(i+1)} \Rightarrow \begin{cases} u_{(i+1)} &= u_{(i)} + \Delta u_{(i+1)} \\ \varepsilon_{(i+1)} &= \varepsilon_{(i)} + \Delta \varepsilon_{(i+1)} \\ \sigma_{(i+1)} &= \sigma_{(i)} + \Delta \sigma_{(i+1)} \end{cases} \quad (28.3)$$

Direct analysis. Alternatively we could have obtained this total result directly in a regular analysis, without the phased analysis option, by applying the total load $F_{(i+1)}$ on the model

$$F_{(i+1)} \Rightarrow \begin{cases} u_{(i+1)} \\ \varepsilon_{(i+1)} \\ \sigma_{(i+1)} \end{cases} \quad (28.4)$$

This equation is very much like the one for the previous phase Eq. (28.1). It seems that the only advantage of phased analysis is that we can get output of intermediate and

incremental results. Is that really the only advantage of phased analysis? No, there is more: *in phased analysis with DIANA you may change the model from phase to phase* [§ 28.4.3].

Example. We will demonstrate the principle of total load input for each phase with the example of the bar which is loaded by a force $F = 2$ in phase 1 [Fig. 28.2]. We assume

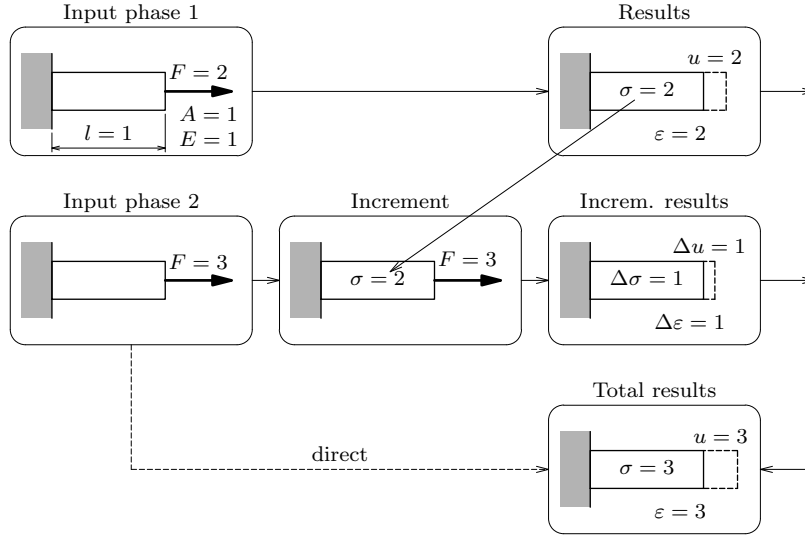


Figure 28.2: Increasing load F

that all relevant physical and material properties are equal to 1. The results of phase 1 are simply to calculate: the bar will be elongated by $u = 2$ and there is a uniform strain $\varepsilon = 2$ and stress $\sigma = 2$ along the bar axis.

In phase 2 the total load increases to $F = 3$ which we must apply on the model. DIANA will now analyse the incremental situation with the load $F = 3$ and an internal tension stress $\sigma = 2$ from the previous phase. Due to the load, the elongation will be 3 and due to the internal stress the shortening will be 2. In other words: the real load is the external load ($F = 3$) minus the internal load ($F_\sigma = 2$) which is equal to 1. This results in an incremental elongation $\Delta u = 1$, which yields an incremental strain $\Delta \varepsilon = 1$ and an incremental stress $\Delta \sigma = 1$.

DIANA now calculates the total results of phase 2 by adding the increments to the results of phase 1. This gives a total elongation $u = 3$, a total strain $\varepsilon = 3$ and a total stress $\sigma = 3$. Obviously, the same total results would have appeared in a regular, non-phased, analysis where the load $F = 3$ was applied directly on the model.

28.4.2 Initial Stress

We will now demonstrate that

In phased analysis with DIANA you must not only apply the total external load for each phase but also the total initial stress.

In other words: DIANA considers initial stresses as loading and, if appropriate, you must specify them again for each phase. To illustrate this we consider the same model as in the previous section, but with an additional initial stress σ^0 , i.e., a prestress load [Fig. 28.3]. For this example we assume specific properties of the bar: the length is l , the cross-section is A , and the Young's modulus of elasticity is E .

Phase 1. The initial displacements and strains for phase 1 are zero. Therefore the total displacements and strains are the same as the incremental ones. If the total stress for phase 1 is $\sigma_{(1)}$ then the incremental stress is

$$\Delta \sigma_{(1)} = \sigma_{(1)} - \sigma_{(1)}^0 \quad (28.5)$$

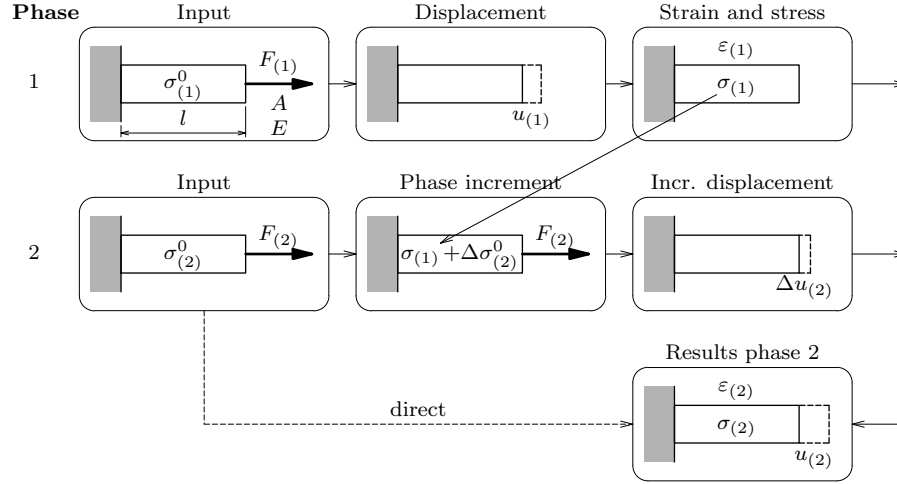


Figure 28.3: Phases with unchanged model and initial stress

If we denote the results from the force load with a subscript F and from the prestress with a subscript σ^0 , like u_F and u_{σ^0} for the displacements, then the results for phase 1 are

$$F_{(1)} + F_{\sigma_{(1)}^0} \Rightarrow \begin{cases} u_{(1)} = \Delta u_{(1)} = u_{F_{(1)}} + u_{\sigma_{(1)}^0} \\ \varepsilon_{(1)} = \Delta \varepsilon_{(1)} = \varepsilon_{F_{(1)}} + \varepsilon_{\sigma_{(1)}^0} \\ \sigma_{(1)} = \sigma_{(1)}^0 + \Delta \sigma_{(1)} = \sigma_{(1)}^0 + \Delta \varepsilon_{(1)} E \end{cases} \quad (28.6)$$

If we consider the results due to the prestress as ‘increments’ then this situation is comparable with that of phase $i + 1$ in the previous example Eq. (28.3). Substitution of the models’ properties in Eq. (28.6) gives

$$\begin{cases} u_{(1)} = \Delta u_{(1)} = \frac{F_{(1)} l}{EA} - \frac{\sigma_{(1)}^0 l}{E} \\ \varepsilon_{(1)} = \Delta \varepsilon_{(1)} = \frac{u_{(1)}}{l} \\ \sigma_{(1)} = \frac{F_{(1)}}{A} \end{cases} \quad (28.7)$$

Phase 2. We assume that in phase 2 the total external load is $F_{(2)}$ and the total stress from the previous phase is $\sigma_{(2)}^0$. The phase increment to be analysed now is: an internal stress $\sigma_{(1)}$ from the previous phase, a total force $F_{(2)}$, and an incremental stress

$$\Delta \sigma_{(2)}^0 = \sigma_{(2)}^0 - \sigma_{(1)}^0 \quad (28.8)$$

The new stress in phase 2 now becomes

$$\bar{\sigma}_{(2)}^0 = \sigma_{(1)} + \Delta \sigma_{(2)}^0 = \sigma_{(2)}^0 + \Delta \varepsilon_{(1)} E \quad (28.9)$$

This results in an incremental displacement $\Delta u_{(2)}$ from which DIANA determines the incremental strains and stresses:

$$\text{Increment: } F_{(2)} + F_{\Delta \sigma_{(2)}^0} + F_{\sigma_{(1)}} \Rightarrow \begin{cases} \Delta u_{(2)} \\ \Delta \varepsilon_{(2)} \\ \Delta \sigma_{(2)} \end{cases} \quad (28.10)$$

And substitution of the models' properties in Eq. (28.10) gives

$$\text{Increment: } \begin{cases} \Delta u_{(2)} = \frac{F_{(2)}l}{EA} - \frac{F_{(1)}l}{EA} - \frac{\sigma_{(2)}^0 l}{E} + \frac{\sigma_{(1)}^0 l}{E} \\ \Delta \varepsilon_{(2)} = \frac{F_{(2)}}{EA} - \frac{F_{(1)}}{EA} - \frac{\sigma_{(2)}^0}{E} + \frac{\sigma_{(1)}^0}{E} \\ \Delta \sigma_{(2)} = \frac{F_{(2)}}{A} - \frac{F_{(1)}}{A} - \sigma_{(2)}^0 + \sigma_{(1)}^0 \end{cases} \quad (28.11)$$

The total results for phase 2 are now equal to those of phase 1 Eq. (28.6), with the addition of the calculated increments Eq. (28.10):

$$\text{Total: } \begin{cases} u_{(2)} = u_{(1)} + \Delta u_{(2)} = u_{F_{(2)}} + u_{\sigma_{(2)}^0} \\ \varepsilon_{(2)} = \varepsilon_{(1)} + \Delta \varepsilon_{(2)} \\ \sigma_{(2)} = \bar{\sigma}_{(2)}^0 + \Delta \sigma_{(2)} = \sigma_{(1)} + \Delta \sigma_{(2)}^0 + \Delta \sigma_{(2)} \end{cases} \quad (28.12)$$

Substitution of Eq. (28.7) and Eq. (28.11) in Eq. (28.12) for the models' properties gives

$$\text{Total: } \begin{cases} u_{(2)} = \frac{F_{(2)}l}{EA} - \frac{\sigma_{(2)}^0 l}{E} \\ \varepsilon_{(2)} = \frac{F_{(2)}}{EA} - \frac{\sigma_{(2)}^0}{E} \\ \sigma_{(2)} = \frac{F_{(2)}}{A} \end{cases} \quad (28.13)$$

Direct. In the equations Eq. (28.13) for the total results of phase 2, all incremental values have disappeared. We would have obtained the same results when applying the force and prestress load of phase 2 directly to the model. This proves that the input of the total prestress in phase 2 yields the correct results.

Linear analysis. We will demonstrate the principle of total initial stress input for each phase with the example of the bar which is loaded by a force $F = 2$ and an initial stress $\sigma^0 = 1$ in phase 1 [Fig. 28.4]. Again we assume that all relevant physical and material

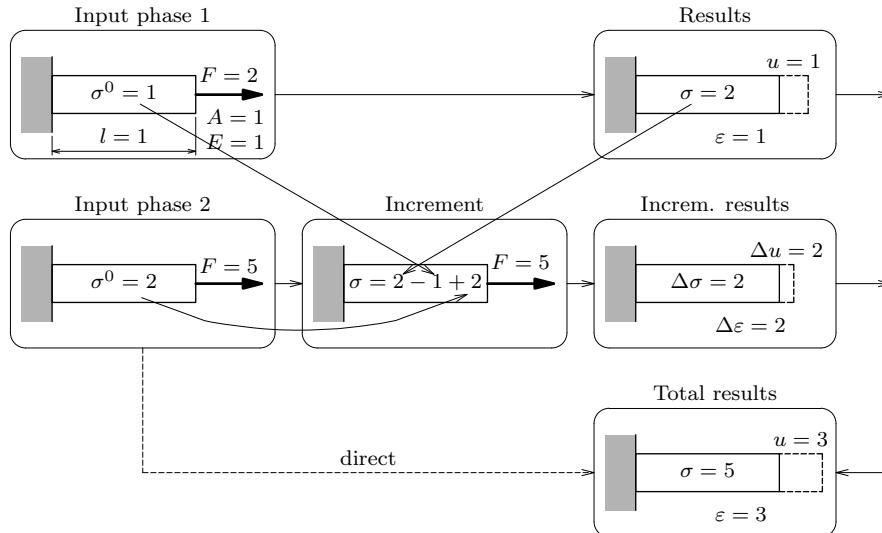


Figure 28.4: Increasing load F and initial stress σ^0

properties are equal to 1. In phase 1, the force F will cause an elongation of 2 and the prestress will shorten the bar by 1. This yields a total elongation of $u = u_F - u_{\sigma^0} = 2 - 1 = 1$ and a strain $\varepsilon = 1$. The total stress in phase 1 is $\sigma = 2$ and the incremental stress is $\Delta \sigma = 1$.

In phase 2 the external load increases to $F = 5$ and the stress from the previous phase $\sigma = 2$ (assuming that the prestress is still 1), which we both must apply on the model. DIANA will now analyse the incremental situation with the load $F = 5$ an internal tension stress $\sigma = 2$ from the previous phase and the incremental prestress $\Delta\sigma^0 = 1$, i.e., we increment the prestress from 1 to 2. Due to the load, the elongation will be 5 and due to the stress of the previous phase the shortening will be 3. This results in an incremental elongation $\Delta u = 2$, which yields an incremental strain $\Delta\varepsilon = 2$ and an incremental stress $\Delta\sigma = 2$.

According to Eq. (28.12) the total elongation after phase 2 is $u = 3$, the total strain $\varepsilon = 3$, and the total stress is equal to the prestress of phase 2 plus the stress of phase 1 minus the prestress of phase 1 plus the stress increment for phase 2: $\sigma = 2 + 2 - 1 + 2 = 5$. These total results are equal to those that we would have obtained in a direct, non-phased, analysis of the situation after phase 2.

28.4.3 Model Changes

In phased analysis you may change the model from phase to phase. For instance, you may declare some elements and/or reinforcements to become inactive or declare inactive elements to become active again, add or delete elements in a new phase,¹ change material properties, or remove existing supports or apply new supports.

When the model changes the basic principles for structural analysis are the same as described in the previous section. However, in this case the inactive elements do not contribute to the transit of force through the model, and strains and stresses in inactive elements do not change until these elements become active again.

Example. The scheme in Figure 28.5 shows what happens in phased analysis when in phase 2 an element is added to the model. Note that the total stresses and strains

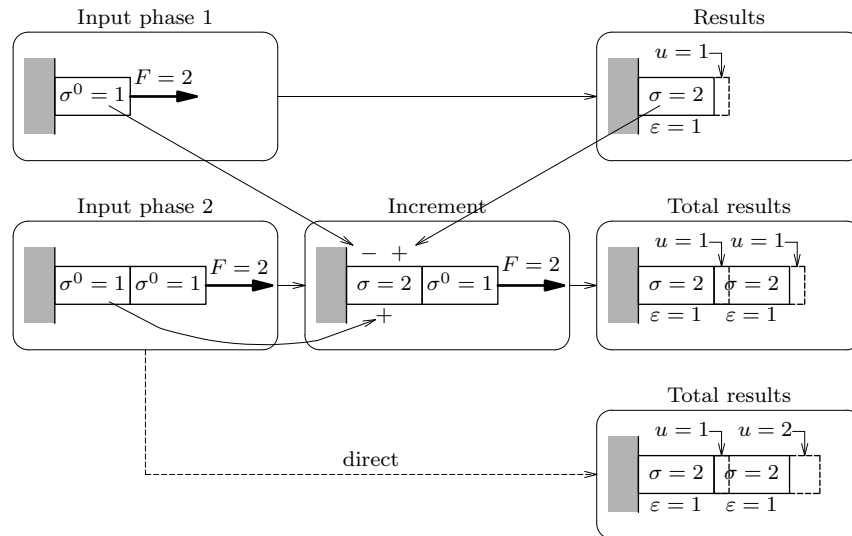


Figure 28.5: Addition of an element

for phased and direct analysis are alike for both elements. However, the total horizontal displacement u of the bar tip in phased analysis is half the displacement in direct analysis. This illustrates that in phased analysis, DIANA connects new elements to the *deformed* mesh of the previous phase.

¹It is not necessary that you define the complete model beforehand.

Chapter 29

Phased Potential Flow Analysis

This chapter describes the user aspects of phased analysis impact on input and execution of phased transient potential flow analysis. The used terminology is explained in the introduction to phased analysis [Part [VIII](#)].

Before starting a phased potential flow analysis it is important to note that switching from linear to nonlinear analysis, or vice versa, is not allowed.

Transient heat flow analysis

file.dcf

```
*FILOS
INITIA
*INPUT
commands
*PHASE
commands
*HEATTR
commands
*INPUT      optional!
commands
*PHASE
commands
*HEATTR
commands
...
```

Transient groundwater flow analysis

file.dcf

```
*FILOS
INITIA
*INPUT
commands
*PHASE
commands
*GROWTR
commands
*INPUT      optional!
commands
*PHASE
commands
*GROWTR
commands
...
```

29.1 Implementation Backgrounds

The steps described in the previous section can be involved in each phase of a potential flow analysis. Results from previous phases are stored at element level and used as initial values. In transient analysis with capacity contribution, initial values will influence the time dependent solution.

29.1.1 Basic Equations

$$\text{Constitutive law:} \quad \mathbf{q} = \mathbf{k} \nabla \phi \quad (29.1)$$

$$\text{Equilibrium law:} \quad \nabla \mathbf{q} = c \dot{\phi} + f \quad (29.2)$$

With \mathbf{q} the flux vector, \mathbf{k} the conductivity tensor, ϕ the potential, c the capacity and f volume sources. The Finite Element equations derived from the basic equations are:

$$\mathbf{M} \dot{\mathbf{u}} + \mathbf{K} \mathbf{u} = \mathbf{F} \quad (29.3)$$

With \mathbf{M} the capacity matrix, \mathbf{K} the conductivity matrix, ϕ the vector of nodal potentials and \mathbf{F} the resulting vector of nodal fluxes, derived from external (dis)charges and constraints.

29.1.2 Solution

The potential flow finite element equations are solved stepwise with direct time integration. This means that for each step Δt , the equation is solved at time $t + \alpha \Delta t$. Combination of the Finite Element equations Eq. (29.3) and direct time integration results in an equation of the following form.

$$\mathbf{K}^* \mathbf{u}^* = \mathbf{F}^* \quad (29.4)$$

With \mathbf{K}^* the effective matrix, \mathbf{u}^* the vector of nodal potentials, and \mathbf{F}^* the effective vector of nodal fluxes.

Special for phased analysis is that at the first time step in a new phase, the previously calculated values, stored at element level, contribute to \mathbf{F}^* . The following paragraphs for implicit and explicit time integration show how these initial values \mathbf{u}^t contribute.

29.1.2.1 Implicit Time Integration

The implicit time integration is characterized by

$$\boxed{\alpha > 0 \quad \text{and} \quad \mathbf{u}^* = \mathbf{u}^{t+\alpha\Delta t}} \quad (29.5)$$

With the time derivative:

$$\dot{\mathbf{u}}^{t+\alpha\Delta t} = \frac{\mathbf{u}^{t+\alpha\Delta t} - \mathbf{u}^t}{\alpha\Delta t} \quad (29.6)$$

The effective matrix:

$$\mathbf{K}^* = \mathbf{M}^{t+\alpha\Delta t} + \alpha\Delta t \mathbf{K}^{t+\alpha\Delta t} \quad (29.7)$$

The effective nodal flux vector:

$$\mathbf{F}^* = \alpha\Delta t \mathbf{F}^{t+\alpha\Delta t} + \mathbf{M}^{t+\alpha\Delta t} \mathbf{u}^t \quad (29.8)$$

In case of nonlinearity the iteration scheme is:

$$\mathbf{K}_i^* \mathbf{u}_{i+1}^* = \mathbf{F}_i^* \quad (29.9)$$

The extrapolation for $\alpha < 1$ is:

$$\mathbf{u}^{t+\Delta t} = \mathbf{u}^t + \dot{\mathbf{u}}^{t+\alpha\Delta t} \Delta t \quad (29.10)$$

29.1.2.2 Explicit Time Integration

The explicit time integration is characterized by

$$\boxed{\alpha = 0 \quad \text{and} \quad \mathbf{u}^* = \mathbf{u}^{t+\Delta t}} \quad (29.11)$$

With the time derivative:

$$\dot{\mathbf{u}}^t = \frac{\mathbf{u}^{t+\Delta t} - \mathbf{u}^t}{\Delta t} \quad (29.12)$$

The effective matrix:

$$\mathbf{K}^* = \mathbf{M}^t \quad (29.13)$$

The effective nodal flux vector:

$$\mathbf{F}^* = \Delta t \mathbf{F}^t - \Delta t \mathbf{K}^t \mathbf{u}^t + \mathbf{M}^t \mathbf{u}^t \quad (29.14)$$

29.2 Input of the Finite Element Model(s)

Input of the finite element model is analogous to phased structural analysis [§ 28.1 p. 402].

file.dcf

```
*INPUT
APPEND TABLE 'ELEMEN' 'COORDI'
READ TABLE 'BOUNDA' 'TIMEBO'
```

29.3 Phase Initialization

To initialize a new phase you must invoke Module PHASE.

syntax

```
*PHASE
[ ACTIVE ELEMEN elmnrs sng... ]
```

ACTIVE selects a certain part of the model to be active in the new phase. By default, DIANA activates the complete model, as read previously [§ 28.1]. With the ACTIVE command you may select only a part of the model to be active [§ 28.2].

ELEMEN for selection of active elements where *elmnrs* is a series of element sets, element numbers, and/or or group names. Default: if no selection is specified, all elements available from input are assumed to be active.

file.dcf

```
*PHASE
ACTIVE ELEMEN 1-7
```

29.4 Transient Analysis

29.4.1 Initialization

Each transient analysis part of a phase starts with an initialization of new elements for transient analysis, using specified initial fields if available. During initialization the initial calculation time can be specified to achieve a consistent time offset compared with other phases.

29.4.2 Time Steps

The stored element results from previous phases are used as initial values for active existing elements during the first time step. The calculation and output of time dependent results is performed for active elements.

file.dcf

```

BEGIN OUTPUT
  BEGIN SELECT
    commands
  END SELECT
  commands
END OUTPUT
BEGIN EXECUTE
  commands
  SIZES 100.0 200.0 300.0 600.0 1800. 3600.0(9)
END EXECUTE

```

29.4.3 Aging

In phased analysis there is one global calculation time, but the age of each element is computed separately. The element age is to be considered as the total period of activity. The initial element age always equals zero. Material properties can be time dependent or age dependent, depending on the intended use. Age dependency is specified in table 'MATERI'. See Volume *Material Library* for syntax.

Part IX

Solution Procedures

Chapter 30

Solve System of Equations

The `SOLVE` command is used in various modules (e.g. `LINSTA`, `NONLIN`, `HEATSS`) to solve the system of equilibrium equations of the finite element model. Four different solution methods are available: a parallel direct sparse solver, a Sparse Cholesky method, an out-of-core direct solution method, and an iterative method.

syntax

```
BEGIN SOLVE
[ BEGIN methodw
  PARDIS
  CHOLES
  GENEL
  ITERAT
[ options ]
END methodw ]
  PARDIS
  CHOLES
  GENEL
  ITERAT
[ TOLERA=epsr ]
END SOLVE
```

`PARDIS` applies the Intel PARDISO solution method, i.e. a parallel direct sparse solver (the default) [§ 30.1]. [PARDIS]

`CHOLES` applies a Sparse Cholesky based solution method [§ 30.2].

`GENEL` applies an out-of-core direct solution method [§ 30.3].

`ITERAT` applies an iterative solution method [§ 30.4].

`TOLERA=eps` is the tolerance ϵ for the check on accuracy of the solution process. This [$\epsilon = 10^{-8}$] parameter is not applicable with the `GENEL` method.

For all methods `DIANA` applies useful defaults. You may customize the methods by specifying *options* as described in the referred sections.

30.1 PARDISO – Parallel Direct Sparse Solver

When using the Intel PARDISO parallel direct sparse solver, you can give the following `PARDIS` commands. See § 45.1.3 on page 541 for background theory.

syntax

```

BEGIN SOLVE
  BEGIN PARDIS
    [ FACTOR [OFF] ]
    [ MAXTHR mthread ]
    [ SUBSTR options ]
  END PARDIS
END SOLVE

```

FACTOR OFF suppresses factorization which may be useful if only the right-hand-side (load) vectors changed.

MAXTHR=*mthread* specifies the maximum number of threads to be used.

If the user has not explicitly set the number of threads by MAXTHR then this value can be set by the operating system to the available numbers of processors on the system. It is therefore always recommended to control the parallel execution of the solver by explicitly setting MAXTHR. If less processors are available than specified, the execution may slow down instead of speeding up.

SUBSTR customizes the application of substructuring [§ 30.5].

30.2 Sparse Cholesky Based Solution Method

When using the Sparse Cholesky based solution method, DIANA will automatically choose an optimal solution procedure which is initially based on a Sparse Cholesky method [§ 45.1 p. 539]. If there is not enough memory to carry out the Sparse Cholesky factorization, then an out-of-core direct solution procedure is performed [§ 30.3]. If a factorization from a previous Sparse Cholesky based solution exists, the Sparse Cholesky based solution method uses this as a preconditioner in an iterative process [§ 30.4]. The Sparse Cholesky based solution method does not require any additional SOLVE commands. However, for special cases you may customize the solution procedure via CHOLE commands.

syntax

```

BEGIN SOLVE
  BEGIN CHOLE
    [ FACTOR [OFF] ]
    [ SUBSTR options ]
  END CHOLE
END SOLVE

```

FACTOR OFF suppresses factorization which may be useful if only the right-hand-side (load) vectors changed.

SUBSTR customizes the application of substructuring [§ 30.5].

30.3 Out-of-Core Direct Solution

DIANA offers an out-of-core direct solution method based on Gauss decomposition. To apply this method you must give GENEL commands.

syntax

```

BEGIN SOLVE
  BEGIN GENEL
    [ FACTOR [ OFF ] ]
  END GENEL
END SOLVE

```

FACTOR OFF suppresses factorization which may be useful if only the right-hand-side (load) vectors changed.

30.4 Iterative Solution

To customize the iterative solution procedure for the system of equations you may give ITERAT commands. See § 45.3 on page 546 for background theory.

syntax

```

BEGIN SOLVE
  BEGIN ITERAT
    [ PRECON [ _____ ]          [ MAXITE=mitern ] ]
      ILU
      DIAGON [ OUTCOR ]
    [ MAXTHR mthreadn ]
    [ SUBSTR options ]
  END ITERAT
END SOLVE

```

PRECON specifies the preconditioning process. By default DIANA applies ILU preconditioning. You may customize the preconditioning process via the following options.

ILU for the *Incomplete LU-decomposition* preconditioning, generally known as ILU preconditioning. This method is particularly suited for well-conditioned solid models [§ 45.3.4.1 p. 548]. [*n* = 400]

DIAGON for *diagonal* preconditioning, also called *Jacobi* preconditioning [§ 45.3.4.2 p. 548]. [*n* = 2000]

OUTCOR applies an out-of-core solution algorithm. *The computing time for out-of-core solution is considerably more than for the default in-core solution.*

MAXITE=*miter* is the maximum number of iterations *n*. The default depends on the applied preconditioning.

MAXTHR=*mthread* specifies the number of subdomains that is constructed by the Schwarz domain decomposition method [§ 45.3.3 p. 547] for parallel computations in the iterative solver. If the user has explicitly set the number of subdomains by MAXTHR larger than zero, the domain decomposition solver, a parallel iterative solver, will be used.

The domain decomposition will be performed once. Therefore, multiple specifications of MAXTHR will result in the number of domains that has been specified by the first MAXTHR command. The parallel iterative solver will automatically use the number of decomposition domains that has been set up.

SUBSTR customizes the application of substructuring [§ 30.5].

Default*file.dcf*

SOLVE ITERAT

If you only specify the ITERAT option then DIANA applies the most appropriate method with ILU preconditioning, as if you had given the following commands.

file.dcf

```
BEGIN SOLVE
  BEGIN ITERAT
    PRECON ILU MAXITE=400
    TOLERA=1.E-8
  END ITERAT
END SOLVE
```

30.5 Substructuring

By default DIANA investigates whether substructuring in a structural nonlinear analysis with Module NONLIN could be profitable. If so, then substructuring will be applied automatically except for the Intel PARDISO parallel direct sparse solver, which requires explicit specification of applying substructures. In other types of analysis, for instance linear structural or potential flow, no substructuring is applied by default. To customize the application of substructuring, you may give the SUBSTR command. See § 45.4 on page 549 for background theory.

syntax

```
SUBSTR [ _____ ] [ NUMSUB=nsubn ]
      AUTO
      ON
      OFF
```

[AUTO] AUTO automatically applies substructuring if this seems to be profitable. This is the default option if you only give the SUBSTR command.

ON explicitly applies substructuring.

NUMSUB *nsub* is the number of substructures to be set up. The default for structural nonlinear analysis is one substructure. For all other applications the default is two substructures.

For a structural nonlinear analysis the substructure(s) comprise the elements without nonlinear material properties. For all other applications the substructure(s) comprise all active elements.

OFF explicitly turns off substructuring.

Chapter 31

Eigenvalue Analysis

In order to perform an eigenvalue analysis with DIANA you must take the following actions:

1. Invoke Module FILOS to initialize an analysis database [§ 3.2 p. 48].
2. Invoke Module INPUT to read the finite element model into the database [§ 3.3 p. 50].
3. Invoke Module EIGEN to perform an eigenvalue analysis.
4. You may get a plot of the vibration modes in the postprocessing working environment of iDIANA [Vol. iDIANA].

The primary tasks for Module EIGEN are invoked via the following command sequence.

syntax

```
*EIGEN
[ MODEL ... ]
[ TYPE ... ]
[ EXECUT ... ]
[ RAYLEI ... ]
[ OUTPUT ... ] ...
*END
```

MODEL evaluates and assembles the finite element model [§ 31.1 p. 428].

TYPE to specify the type of the eigenvalue problem [§ 31.2 p. 428].

EXECUT to execute the eigenvalue analysis [§ 31.3 p. 432].

RAYLEI to calculate Rayleigh damping coefficients [§ 31.4 p. 435].

OUTPUT to specify the analysis results for output [§ 31.5 p. 436].

Default. If you only give the *EIGEN command, then DIANA assumes the following default command sequence.

file.dcf

```
*EIGEN
MODEL
BEGIN TYPE
  BEGIN FREEVI
    STIFFN LINEAR
    MASS CONSIG ROTATI ON
  END FREEVI
END TYPE
BEGIN EXECUT
```

```

METHOD ARNOLD
NMODES=1
NTRIAL=21
MAXITE=30
FSHIFT=0.
TOLERA=1.E-6
PARDIS
END EXECUT
*END

```

This yields the first eigenpair (value and mode) of a free vibration eigenvalue problem.

Fluid–structure interaction analysis. In case of a model containing fluid, structural elements and fluid–structure interface elements, automatically a fluid–structure eigenvalue analysis will be performed. DIANA recognizes this element combination and will perform the required preparation to perform a fluid–structure eigenvalue analysis. In a free vibration eigenvalue analysis the added mass effect of the fluid [Eq. (48.100) p. 599] is taken into account. Results will be available for the structural part of the model.

Lumped element matrices may not be used in a fluid–structure eigenvalue analysis.

Free vibration problem. This example illustrates the commands for generalized eigenvalue analysis of a free vibration problem in a concise form. Natural frequencies or eigenfrequencies (in Hertz for SI unit system) and corresponding mode shapes are obtained by means of Module EIGEN with the following commands.

file.dcf

```

*EIGEN
MODEL
TYPE FREEVI
BEGIN EXECUT
  NMODES=20
END EXECUT
OUTPUT DISPLA
*END

```

In the free vibration eigenproblem the system mass matrix has to be used, which is invoked by TYPE FREEVI, and by default assembles the consistent element mass matrices. The EXECUT command block starts the calculation of frequencies and corresponding mode shape vectors, where parameter NMODES=20 asks for twenty frequencies. The number of trial vectors is twice the required number of eigenvalues. DIANA takes reasonable defaults for the maximum number of iterations and the convergence criterion. Output results like natural frequencies and mode shapes are obtained by the OUTPUT command.

Standard eigenproblem. The following example shows how to setup a standard eigenvalue analysis [§ 45.2.1 p. 541] in a more comprehensive form; now the mass matrix **M** must be replaced by the identity matrix **I**. In order to save processing time or improve the accuracy for the iterative solution procedure, a number of parameters may be overruled. Eigenvalues and corresponding eigenmodes are now obtained with the following set of commands.

file.dcf

```

*EIGEN
MODEL
TYPE IDENTI
BEGIN EXECUT
  NMODES=5
  NTRIAL=30
  MAXITE=50

```

```

TOLERA=1.E-9
END EXECUT
OUTPUT DISPLA
*END

```

The use of the identity matrix **I** is specified by **TYPE IDENTI**. In order to overrule the defaults in the solution procedure, optional parameters have been specified: **NTRIAL=30** asks for thirty trial vectors, **MAXITE=50** for a maximum of fifty iterations and **TOLERA=1.E-9** specifies a convergence criterion on eigenvalues of 10^{-9} .

Unsupported system with shift. Consider the following example of an unsupported system which possesses rigid body modes. The stiffness matrix **K** is now positive semidefinite and has zero eigenvalues as much as the number of rigid body modes. For a frequency analysis, the stiffness matrix **K** must be shifted according to Eq. (45.30) on page 545. Before being output, calculated frequencies are corrected according to the performed shift.

file.dcf

```

*EIGEN
MODEL
TYPE FREEVI
BEGIN EXECUT
  NMODES=20
  FSHIFT=1.
END EXECUT
OUTPUT DISPLA
*END

```

The **FSHIFT** parameter performs a frequency shift on the stiffness matrix. For frequency analysis, shifting is done by adding $\mu = (2\pi f_s)^2 \times$ the system mass matrix **M**:

$$(\mathbf{K} + \mu \mathbf{M}) \phi = \omega^2 \mathbf{M} \phi$$

Now a set of twenty shifted frequencies and corresponding mode shape vectors are calculated. Output results like natural frequencies and mode shapes are obtained by the **OUTPUT DISPLA** command. The resulting frequencies have been corrected according to Eq. (45.31) on page 545.

Nonlinear system with shift. This example examines the properties of a tangential stiffness matrix **K_T**, resulting from a static nonlinear analysis. Since the finite element model is already evaluated and assembled in the nonlinear analysis the **MODEL OFF** command is required. The **NONLIN** command specifies that the tangential stiffness matrix **K_T** will be used instead of the linear elastic stiffness matrix **K_{L0}**.

*Note that tangent stiffness matrices are only available when the nonlinear analysis (*NONLIN) involves a Newton-Raphson iteration method (METHOD NEWTON) [§ 13.3.5.1 p. 236]. No tangent stiffness matrices are being constructed when the nonlinear analysis is done with other iteration methods, e.g. Constant Stiffness, Linear stiffness, or secant (Quasi-Newton) method.*

Due to a nonlinear phenomenon like softening behaviour, the stiffness matrix **K_T** may be no longer positive definite and for the negative eigenvalues the corresponding eigenmodes must be determined. By means of a proper shift, sufficiently large to make the smallest eigenvalue positive, the eigenproblem can now be solved by the standard iteration method.

file.dcf

```

*NONLIN
  commands
BEGIN EXECUT

```

```

      commands
      METHOD NEWTON commands
      commands
      END EXECUT
      commands
      *EIGEN
      MODEL OFF
      TYPE IDENTI NONLIN
      BEGIN EXECUT
      NMODES=2
      SHIFT=100.
      MAXITE=50
      TOLERA=1.E-9
      END EXECUT
      OUTPUT DISPLA
      *END

```

The **SHIFT** command performs a shift on the tangential stiffness matrix \mathbf{K}_T , by adding $\mu = 100$ to its diagonal terms. Provided that the shift is sufficiently large, the analysis procedure starts the calculation of two shifted eigenvalues and corresponding eigenmodes. Parameter **MAXITE=50** specifies a maximum of fifty iterations and parameter **TOLERA=1.E-9** a tolerance of 10^{-9} as measure for the convergence condition. Output results are obtained by the **OUTPUT DISPLA** command, where the printed eigenvalues have been corrected according to Eq. (45.31) on page 545.

31.1 Model Evaluation

The **MODEL** commands customize the evaluation of the finite element model prior to the actual eigenvalue analysis.

syntax

```

BEGIN MODEL
[ OFF ]
[ EVALUA ... ]
[ ASSEMB ]
END MODEL

```

EVALUA to check and evaluate geometric and material properties for elements [§ 3.4 p. 51].

ASSEMB to assemble the elements, i.e., creating appropriate system degrees of freedom [§ 3.5 p. 54].

31.2 Type of Eigenvalue Problem

Via the **TYPE** commands you may specify the type of the eigenvalue problem, i.e., the type of matrices to be used in the generalized eigenproblem formulated in Eq. (45.12) on page 542.

syntax

```

BEGIN TYPE
[ OFF ]
...
FREEVI
IDENTI
STABIL
END TYPE

```

OFF switches OFF the type specification which may save computing time if an eigenvalue analysis was performed previously.

FREEVI use mass matrices for free vibration frequency analysis [§ 31.2.1 p. 429]. If you omit the TYPE commands then DIANA assumes this type of analysis by default. [FREEVI]

IDENTI use an identity matrix for the standard eigenproblem [§ 31.2.2 p. 430].

STABIL use element geometric stress-stiffness matrices for linearized buckling analysis [§ 31.2.3 p. 431].

31.2.1 Free Vibration

The FREEVI commands specify the free vibration frequency analysis

$$\mathbf{K}\phi = \omega^2 \mathbf{M}\phi$$

The mass matrix \mathbf{M} may be consistent or lumped, with rotational terms included or excluded. For the stiffness matrix \mathbf{K} , DIANA can apply the linear elastic stiffness with a geometric-stress stiffness matrix added optionally. Alternatively the tangential stiffness from a previously executed nonlinear analysis can be applied for \mathbf{K} . See § 45.2.2 on page 542 for background theory.

Linear elastic stiffness

syntax

```
BEGIN FREEVI
[ BEGIN STIFFN [ LINEAR ]
  [ STRESS [ OFF ] [ _____ ] ]
    INPUT { _____ }
            LOAD=losetn
            FACTOR=facr
    CALCUL { _____ }
            LOAD=losetn
            FACTOR=facr
    PHASE
  END STIFFN ]
[ MASS [ _____ ] [ ROTATI [ _____ ] ]
  CONSIS      ON
  LUMPED      OFF
END FREEVI
```

Tangential stiffness

syntax

```
BEGIN FREEVI
  BEGIN STIFFN NONLIN
  [ STRESS [ OFF ] [ PHASE ] ]
  END STIFFN
[ MASS [ _____ ] [ ROTATI [ _____ ] ]
  CONSIS      ON
  LUMPED      OFF
END FREEVI
```

STIFFN specifies the kind of stiffness matrices to be applied for \mathbf{K} : [LINEAR]

LINEAR applies the linear elastic stiffness matrix \mathbf{K}_{L0} . This solves the eigenproblem

$$\mathbf{K}_{L0}\phi = \omega^2 \mathbf{M}\phi$$

NONLIN applies the tangential stiffness matrix \mathbf{K}_T from a previously executed nonlinear analysis. This solves the eigenproblem

$$\mathbf{K}_T \phi = \omega^2 \mathbf{M} \phi$$

*Note that tangent stiffness matrices are only available when the nonlinear analysis (*NONLIN) involves a Newton-Raphson iteration method (METHOD NEWTON) [§ 13.3.5.1 p. 236]. No tangent stiffness matrices are being constructed when the nonlinear analysis is done with other iteration methods, e.g. Constant Stiffness, Linear stiffness, or secant (Quasi-Newton) method.*

[CALCUL] STRESS adds the geometric stress-stiffness matrix \mathbf{K}_G to the stiffness matrix \mathbf{K} . This solves the eigenproblem

$$(\mathbf{K} + \mathbf{K}_G) \phi = \omega^2 \mathbf{M} \phi$$

A typical example is the calculation of the natural frequencies of a prestressed structure, for instance a guitar string.

INPUT indicates a geometric stress-stiffness matrix, with stresses specified via a prestress load for load set *loset*, in subtable ELEMEN and/or REINFO of table 'LOADS'.

CALCUL indicates a geometric stress-stiffness matrix, with stresses which DIANA will calculate automatically in a linear static analysis for load set *loset* (the default).

PHASE indicates a geometric stress-stiffness matrix based on the current stress state.

To setup the geometric stress-stiffness matrix \mathbf{K}_G from a stress field you may specify a load set number *loset* via the LOAD parameter. This load set number corresponds to a load set in input table 'LOADS' [§ 2.3.8 p. 45]. Default is the lowest available load set number. The optional parameter FACTOR specifies a multiplication factor *fac*.

[*fac*=1.0]

[CONSIS] MASS specifies the kind of mass matrices to be applied for \mathbf{M} :

CONSIS for consistent matrices (the default).

LUMPED for lumped matrices.

[ON] ROTATI indicates whether rotational terms must be included in the mass matrices: ON for 'yes', OFF for 'no'. If you do not specify the ROTATI option explicitly, then DIANA will include rotational terms by default.

31.2.2 Standard Eigenproblem

The IDENTI commands specify the standard eigenproblem with the identity matrix \mathbf{I}

$$\mathbf{K} \phi = \lambda \mathbf{I} \phi$$

For the stiffness matrix \mathbf{K} , DIANA can apply the linear elastic stiffness with a geometric-stress stiffness matrix added optionally. Alternatively the tangential stiffness from a previously executed nonlinear analysis can be applied for \mathbf{K} . See § 45.2.1 on page 541 for background theory.

Linear elastic stiffness

syntax

```
BEGIN IDENTI
[ BEGIN STIFFN [ LINEAR ]
[ STRESS [ OFF ] [ _____ ] ]
      INPUT { _____ }
              LOAD=losetn
              FACTOR=facr
      CALCUL { _____ }
```



```

                                LOAD=losetn
                                FACTOR=facr
                                PHASE
END STIFFN]
END IDENTI

```

Tangential stiffness*syntax*

```

BEGIN IDENTI
BEGIN STIFFN NONLIN
[ STRESS [ OFF ] [ PHASE ] ]
END STIFFN
END IDENTI

```

STIFFN specifies the kind of stiffness matrices to be applied for **K**:

[LINEAR]

LINEAR applies the linear elastic stiffness matrix \mathbf{K}_{L0} . This solves the eigenproblem

$$\mathbf{K}_{L0}\phi = \lambda\mathbf{I}\phi$$

NONLIN applies the tangential stiffness matrix \mathbf{K}_T from a previously executed non-linear analysis. This solves the eigenproblem

$$\mathbf{K}_T\phi = \lambda\mathbf{I}\phi$$

*Note that tangent stiffness matrices are only available when the nonlinear analysis (*NONLIN) involves a Newton-Raphson iteration method (METHOD NEWTON) [§ 13.3.5.1 p. 236]. No tangent stiffness matrices are being constructed when the nonlinear analysis is done with other iteration methods, e.g. Constant Stiffness, Linear stiffness, or secant (Quasi-Newton) method.*

STRESS adds the geometric stress-stiffness matrix \mathbf{K}_G to the stiffness matrix **K**. This solves the eigenproblem

[CALCUL]

$$(\mathbf{K} + \mathbf{K}_G)\phi = \lambda\mathbf{I}\phi$$

INPUT indicates a geometric stress-stiffness matrix, with stresses specified via a prestress load for load set *loset*, in subtable ELEMEN and/or REINFO of table 'LOADS'.

CALCUL indicates a geometric stress-stiffness matrix, with stresses which DIANA will calculate automatically in a linear static analysis for load set *loset* (the default).

PHASE indicates a geometric stress-stiffness matrix, with stresses present from a previous phase.

To setup the geometric stress-stiffness matrix \mathbf{K}_G from a stress field you may specify a load set number *loset* via the LOAD parameter. This load set number corresponds to a load set in input table 'LOADS' [§ 2.3.8 p. 45]. Default is the lowest available load set number. The optional parameter FACTOR specifies a multiplication factor *fac*.

[*fac*=1.0]**31.2.3 Linearized Buckling**

Due to the STABIL commands DIANA applies the geometric stress-stiffness matrix \mathbf{K}_G to solve the linearized buckling eigenproblem

$$\mathbf{K}\phi = \lambda\mathbf{K}_G\phi$$

See Chapter 18 for background theory.

syntax

```

BEGIN STABIL
[ LOAD=losetn ]
END STABIL

```

LOAD=*loset* applies the stresses of the specified load set to setup \mathbf{K}_G . The load set number corresponds to a load set in input table 'LOADS' [§ 2.3.8 p. 45]. Default is the lowest available load set number.

31.3 Execute Eigenvalue Analysis

With EXECUT commands you may customize the actual execution of the eigenvalue analysis.

syntax

```

BEGIN EXECUT
[ OFF ]
[ METHOD [ _____ ] ]
          ARNOLD
          FEAST
[ ... ]   analysis execution commands
END EXECUT

```

OFF switches off execution of the eigenvalue analysis. This may save computing time if you require other or more output of a previously executed analysis.

[METHOD ARNOLD] METHOD *method* specifies the eigenvalue method:

ARNOLD for an Arnoldi method based eigenvalue analysis [§ 31.3.1].

FEAST for a FEAST method based eigenvalue analysis [§ 31.3.2].

31.3.1 Arnoldi Method Based Eigenvalue Analysis

The Arnoldi method based eigenvalue analysis calculates a user-defined number of eigenpairs. At most n eigenpairs can be calculated, where n is the dimension of the system matrix,

Free vibration frequency analysis

syntax

```

BEGIN EXECUT
METHOD ARNOLD
[ soltypw ]
PARDIS
CHOLES
OUTCOR
[ MAXTHR=mthreadn ]
[ NMODES=nmodesn ]
[ FSHIFT=fsr ]
[ NTRIAL=ntrialn ]
[ MAXITE=maxitrn ]
[ TOLERA=epconvr ]
END EXECUT

```

[PARDIS] *soltyp* indicates the solution method to be used for the eigenvalue analysis:

PARDIS applies the Intel PARDISO solution method, i.e. a parallel direct sparse solver.

CHOLES applies a Sparse Cholesky based solution method.

OUTCOR indicates application of an out-of-core solution algorithm.

MAXTHR=*mthread* specifies the maximum number of threads to be used if the Arnoldi method in combination with the parallel direct sparse solver (PARDISO) is selected.

If the user has not explicitly set the number of threads by MAXTHR then this value can be set by the operating system to the available numbers of processors on the system. It is therefore always recommended to control the parallel execution of the solver by explicitly setting MAXTHR. If less processors are available then specified, the execution may slow down instead of speeding up.

NMODES=*nmodes* specifies the number of eigenfrequencies to be calculated. DIANA calculates the eigenfrequencies in ascending order. At most n eigenfrequencies can be calculated, where n is the dimension of the system matrix, which is the number of equations in the problem. If you do not specify the number of eigenfrequencies to be calculated, then DIANA calculates only one pair by default.

($nmodes \leq n$)
[$nmodes = 1$]

FSHIFT=*fs* performs a shift where *fs* is the shift frequency f_s from which the shift factor μ on **K** [§ 45.2.3 p.544] is being calculated:

$$\mu = (2\pi f_s)^2 \quad (31.1)$$

NTRIAL=*ntrial* is the number of trial vectors which may not exceed the dimension n of the mass matrix **M**. If you do not specify the number of trial vectors then DIANA takes twenty more than the number of eigenpairs to be calculated if the number of eigenpairs to be calculated is smaller than twenty, otherwise the number of trial vectors equals twice the number of eigenpairs to be calculated. The number of trial vectors will never exceed the number of degrees of freedom of the model.

($ntrial \leq n$)

MAXITE=*maxitr* is the maximum number of iterations. [MAXITE=30]

TOLERA=*epconv* is a preassigned error tolerance ϵ for the convergence criterion on eigenvalues. [$\epsilon = 10^{-6}$]

Standard eigenproblem and linearized buckling

syntax

BEGIN EXECUT

METHOD ARNOLD

[*soltyp_w*]

PARDIS

CHOLES

OUTCOR

[MAXTHR=*mthread_n*]

[NMODES=*nmodes_n*]

[SHIFT=*mu_r*]

[NTRIAL=*ntrial_n*]

[MAXITE=*maxitr_n*]

[TOLERA=*epconv_r*]

END EXECUT

soltyp indicates the solution method to be used for the eigenvalue analysis: [PARDIS]

PARDIS applies the Intel PARDISO solution method, i.e. a parallel direct sparse solver.

CHOLES applies a Sparse Cholesky based solution method.

OUTCOR indicates application of an out-of-core solution algorithm.

MAXTHR=*mthread* specifies the maximum number of threads to be used if the Arnoldi method in combination with the parallel direct sparse solver (PARDISO) is selected.

If the user has not explicitly set the number of threads by MAXTHR then this value can be set by the operating system to the available numbers of processors on the system. It is therefore always recommended to control the parallel execution of the solver by explicitly setting MAXTHR. If less processors are available than specified, the execution may slow down instead of speeding up.

NMODES=*nmodes* specifies the number of eigenpairs to be calculated. DIANA calculates the eigenvalues in ascending order. At most n eigenpairs can be calculated, where n is the dimension of the system matrix, which is the number of equations in the problem. If you do not specify the number of eigenpairs to be calculated, then DIANA calculates only one pair by default.

SHIFT=*mu* performs a shift where *mu* is the shift factor μ on \mathbf{K} [§ 45.2.3 p. 544].

NTRIAL=*ntrial* is the number of trial vectors which may not exceed the dimension n of the mass matrix \mathbf{M} . If you do not specify the number of trial vectors then DIANA takes twenty more than the number of eigenpairs to be calculated if the number of eigenpairs to be calculated is smaller than twenty, otherwise the number of trial vectors equals twice the number of eigenpairs to be calculated. The number of trial vectors will never exceed the number of degrees of freedom of the model.

MAXITE=*maxitr* is the maximum number of iterations.

TOLERA=*epconv* is a preassigned error tolerance ϵ for the convergence criterion on eigenvalues.

31.3.2 FEAST Method Based Eigenvalue Analysis

The FEAST method [71] based eigenvalue analysis calculates all eigenpairs within a user-defined range using the Intel MKL Extended Eigensolver [91].

Free vibration frequency analysis

syntax

```
BEGIN EXECUT
METHOD FEAST
[ FMIN=fminr ]
[ FMAX=fmaxr ]
[ NMODES=nmodesn ]
[ FSHIFT=fsr ]
[ TOLERA=epconvr ]
END EXECUT
```

[*emin* = 0.] FMIN *emin* is the lower frequency of the range to be searched for eigenfrequencies.

[*emin* = 1.] FMAX *emax* is the upper frequencies of the range to be searched for eigenfrequencies.

NMODES=*nmodes* is the initial guess for the number of eigenfrequencies to be found in the range to be searched for eigenfrequencies specified by the FMIN and FMAX parameters. This number is being used within the FEAST algorithm and therefore influences the performance of the eigenvalue analysis: When *nmodes* is too small, multiple restarts have to be done to find all the eigenfrequencies in the specified range; when *nmodes* is too large, excessive memory is being allocated to find all the eigenfrequencies in the specific range. DIANA calculates the eigenfrequencies in ascending order. At

(*nmodes* ≤ *n*)

most *n* eigenfrequencies can be calculated, where *n* is the dimension of the system matrix, which is the number of equations in the problem. If you do not specify an initial guess for the number of eigenfrequencies to be calculated, then DIANA sets this value to 20 or to *n* if the dimension of the eigenproblem to be solved is smaller than 20.

FSHIFT=*fs* performs a shift where *fs* is the shift frequency *f_s* from which the shift factor *μ* on **K** [§ 45.2.3 p. 544] is being calculated:

$$\mu = (2 * \pi * f_s)^2 \quad (31.2)$$

TOLERA=*epconv* is a preassigned error tolerance *ε* for the convergence criterion on eigenvalues. [*ε* = 10⁻⁶]

Standard eigenproblem and linearized buckling

syntax

```
BEGIN EXECUT
METHOD FEAST
[ EMIN=eminr ]
[ EMAX=emaxr ]
[ NMODES=nmodesn ]
[ SHIFT=mur ]
[ TOLERA=epconvr ]
END EXECUT
```

EMIN *emin* is the lower limit of the range to be searched for eigenvalues. [*emin* = 0.]

EMAX *emax* is the upper limit of the range to be searched for eigenvalues. [*emin* = 1.]

NMODES=*nmodes* is the initial guess for the number of eigenvalues to be found in the range to be searched for eigenvalues specified by the EMIN and EMAX parameters. This number is being used within the FEAST algorithm and therefore influences the performance of the eigenvalue analysis: When *nmodes* is too small, multiple restarts have to be done to find all the eigenvalues in the specified range; when *nmodes* is too large, excessive memory is being allocated to find all the eigenvalues in the specific range. DIANA calculates the eigenvalues in ascending order. At most *n* eigenpairs can be calculated, where *n* is the dimension of the system matrix, which is the number of equations in the problem. If you do not specify an initial guess for the number of eigenpairs to be calculated, then DIANA sets this value to 20 or to *n* if the dimension of the eigenproblem to be solved is smaller than 20. (*nmodes* ≤ *n*)

SHIFT=*mu* performs a shift where *mu* is the shift factor *μ* on **K** [§ 45.2.3 p. 544].

TOLERA=*epconv* is a preassigned error tolerance *ε* for the convergence criterion on eigenvalues. [*ε* = 10⁻⁶]

31.4 Calculate Rayleigh Damping Coefficients

With RAYLEI commands you may calculate the Rayleigh damping coefficients *a* and *b* from two frequencies to set up the damping matrices **C** according to

$$\mathbf{C} = a \mathbf{M} + b \mathbf{K} \quad (31.3)$$

The frequencies are specified explicitly or as eigenfrequencies either by their mode numbers or by the cumulative effective mass contribution in global directions.

syntax

```

BEGIN RAYLEI
[ OFF ]
[ _____ ]
FREQUE EXPLIC  $f1_r$   $f2_r$ 
MODES NUMBER  $m1_n$   $m2_n$ 
EFFMAS PERCEN  $effx_r$   $effy_r$   $effz_r$ 
[ DAMPIN  $c1_r$  [ $c2_r$ ]]
END RAYLEI

```

OFF suppresses the calculation of Rayleigh damping coefficients.

($f2 > f1 > 0$) **FREQUE EXPLIC** specifies the explicit frequencies $f1$ and $f2$ used for calculating the Rayleigh damping coefficients.

($m2 > m1 > 0$) **MODES NUMBER** specifies that the frequencies used for calculating the Rayleigh damping coefficients are the eigenfrequencies of modes $m1$ and $m2$.

($effx \geq 0$) **EFFMAS PERCEN** specifies that the eigenfrequencies used for calculating the Rayleigh damping coefficients, are specified by the cumulative effective mass contribution in the global XYZ directions as displayed for a free vibration eigenvalue analysis [§ 31.5.5]. $effx$, $effy$, $effz$ are the cumulative effective mass percentages in global X , Y , and Z direction respectively. In case the requested cumulative effective mass percentages cannot be reached for the calculated number of eigenfrequencies, the second frequency is the highest calculated eigenfrequency. The first frequency is always the calculated first non-zero eigenfrequency.

[EFFMAS PERCEN 90.
90. 90.]

($c1 \geq 0$) **DAMPIN** specifies the damping coefficients for the two frequencies indicated by either the **FREQUE**, **MODES**, or **EFFMAS** command. Values $c1$ and $c2$ are the damping coefficients in parts of the critical damping factor c_{crit} [Eq. (48.5) p. 586]; a value of 0.05 indicates 5%. If only one factor $c1$ is specified, then the damping is the same for both frequencies of the system.

($c2 \geq 0$)

[$c1=0.05$]

[$c2=0.05$]

Note, that if one of the damping ratios is zero, negative damping occurs for specific frequency ranges, which implies that energy is added to the system.

31.5 Output of Analysis Results

DIANA always writes the following general information resulting from an eigenvalue analysis to the standard output file *file.out*: eigenvalues [§ 31.5.1] and relative errors [§ 31.5.2]. For a free vibration eigenvalue analysis additional information may be written: generalized masses [§ 31.5.3], participation factors [§ 31.5.4], modal damping factors [§ 31.5.9], effective masses [§ 31.5.5], modal masses [§ 31.5.6], and equivalent masses [§ 31.5.7], and transformation factors [§ 31.5.8]. The participation vectors (only available for a free vibration analysis) and eigenmodes may also be output [§ 31.5.10].

eigen.out

```

      4 EIGENVALUES FOUND AFTER      1 ITERATIONS
      EIGEN-FREQUENCIES:
0.39927E-01( 1)  0.10175E+00( 2)  0.15139E+00( 3)  0.21294E+00( 4)
RELATIVE ERROR ||R|| / ||Kx||:
0.25672E-14( 1)  0.50024E-15( 2)  0.12696E-14( 3)  0.21452E-14( 4)

MODE      FREQUENCY      GEN. MASS      PARTICIPATION      MODAL DAMPING
  1      0.39927E-01      0.10000E+01      0.30082E+01      0.34458E-01
  2      0.10175E+00      0.10000E+01      -0.93124E+00      0.27646E-01
  3      0.15139E+00      0.10000E+01      0.28889E+00      0.25010E-01
  4      0.21294E+00      0.10000E+01      -0.17223E-01      0.12886E-01

MODE      FREQUENCY      PART. FAC. TX      PART. FAC. TY      PART. FAC. TZ
  1      0.39927E-01      0.30082E+01      0.00000E+00      0.00000E+00

```

2	0.10175E+00	-0.93124E+00	0.00000E+00	0.00000E+00
3	0.15139E+00	0.28889E+00	0.00000E+00	0.00000E+00
4	0.21294E+00	-0.17223E-01	0.00000E+00	0.00000E+00
MODE	FREQUENCY	EFF.MASS TX	PERCENTAGE	CUM.PERCENT.
1	0.39927E-01	0.90490E+01	0.90490E+02	0.90490E+02
2	0.10175E+00	0.86722E+00	0.86722E+01	0.99162E+02
3	0.15139E+00	0.83457E-01	0.83457E+00	0.99997E+02
4	0.21294E+00	0.29665E-03	0.29665E-02	0.10000E+03
MODE	FREQUENCY	EFF.MASS TY	PERCENTAGE	CUM.PERCENT.
1	0.39927E-01	0.00000E+00	0.00000E+00	0.00000E+00
2	0.10175E+00	0.00000E+00	0.00000E+00	0.00000E+00
3	0.15139E+00	0.00000E+00	0.00000E+00	0.00000E+00
4	0.21294E+00	0.00000E+00	0.00000E+00	0.00000E+00
MODE	FREQUENCY	EFF.MASS TZ	PERCENTAGE	CUM.PERCENT.
1	0.39927E-01	0.00000E+00	0.00000E+00	0.00000E+00
2	0.10175E+00	0.00000E+00	0.00000E+00	0.00000E+00
3	0.15139E+00	0.00000E+00	0.00000E+00	0.00000E+00
4	0.21294E+00	0.00000E+00	0.00000E+00	0.00000E+00
MODE	FREQUENCY	MODAL MASS	PERCENTAGE	
1	0.39927E-01	0.52200E+01	0.52200E+02	
2	0.10175E+00	0.46868E+01	0.46868E+02	
3	0.15139E+00	0.53811E+01	0.53811E+02	
4	0.21294E+00	0.24436E+01	0.24436E+02	
MODE	FREQUENCY	EQ.MASS TX	EQ.MASS TY	EQ.MASS TZ
1	0.39927E-01	0.68728E+01	0.00000E+00	0.00000E+00
2	0.10175E+00	0.20160E+01	0.00000E+00	0.00000E+00
3	0.15139E+00	0.67014E+00	0.00000E+00	0.00000E+00
4	0.21294E+00	0.26924E-01	0.00000E+00	0.00000E+00
MODE	FREQUENCY	TRAN.FAC.TX	TRAN.FAC.TY	TRAN.FAC.TZ
1	0.39927E-01	0.13166E+01	0.00000E+00	0.00000E+00
2	0.10175E+00	0.43016E+00	0.00000E+00	0.00000E+00
3	0.15139E+00	0.12454E+00	0.00000E+00	0.00000E+00
4	0.21294E+00	0.11018E-01	0.00000E+00	0.00000E+00

31.5.1 Eigenvalues

The resulting eigenvalues are output in ascending order of the absolute value, their type depends on the type of eigenvalue analysis performed. See also § 45.2.4 on page 545 for background theory.

- The standard eigenproblem yields the ‘real’ eigenvalues λ from Eq. (45.9) on page 541.
- The free vibration eigenproblem yields the natural frequencies $f = \omega/2\pi$ from Eq. (45.12) on page 542. These frequencies are output as far as they have been found. Note that the unit of the frequencies depends on the units of the input data [§ 1.1 p. 3]. If you supply the input data in ISO units N, m, kg, then the frequencies are in Hz (hertz). However, if you use N, mm, kg, then the unit of the frequencies is $\text{Hz} \times \sqrt{1000}$.
- A linearized buckling analysis yields the buckling values λ from Eq. (45.14) on page 542.

31.5.2 Relative Errors

The error measures ϵ_i for each eigenvalue and eigenmodes approximation is output [Eq. (45.37) p. 545].

31.5.3 Generalized Mass

In a free vibration analysis DIANA will output for each eigenfrequency f_i the corresponding generalized mass m_{ii} [Eq. (45.17) p. 542].

In DIANA all eigenvectors are normalized such that the corresponding generalized mass is equal to one.

31.5.4 Participation Factors

In a free vibration analysis DIANA will output for each eigenfrequency f_i the corresponding participation factor γ_i [Eq. (45.19) p. 542] and the direction dependent participation factors Γ_i for translations and, if available, rotations in global X , Y , and Z direction [§ 45.2.2.4 p. 543].

31.5.5 Effective Mass

In a free vibration analysis DIANA will output for each eigenfrequency f_i the corresponding effective masses $m_{eff,i}$ for translations in global X , Y , and Z direction [§ 45.2.2.5 p. 543]. The effective mass is also printed as percentage of the total mass of the model for each eigenfrequency f_i . Furthermore, the cumulative percentage of effective mass with respect to the total mass is printed.

For fluid–structure analysis the total mass is the structural mass plus the calculated added mass contribution of the fluid domain.

31.5.6 Modal Mass

In a free vibration analysis DIANA will output for each eigenfrequency f_i the corresponding modal masses $m_{mod,ii}$ [§ 45.2.2.6 p. 543]. The modal mass is also printed as percentage of the total mass of the model for each eigenfrequency f_i .

For fluid–structure analysis the total mass is the structural mass plus the calculated added mass contribution of the fluid domain.

31.5.7 Equivalent Mass

In a free vibration analysis DIANA will output for each eigenfrequency f_i the direction dependent equivalent masses $m_{eq,i}$ [§ 45.2.2.7 p. 544].

31.5.8 Transformation Factors

In a free vibration analysis DIANA will output for each eigenfrequency f_i the direction dependent transformation factors $\Gamma_{tr,i}$ [§ 45.2.2.8 p. 544].

31.5.9 Modal Damping Factors

In a free vibration analysis, when you have specified an element damping coefficient h_e based on strain energy, DIANA will output for each eigenfrequency f_i the corresponding modal damping factors h_i [Eq. (45.28) p. 544].

31.5.10 Eigenmodes

For the eigenmodes and participation vectors you may specify an output selection via the OUTPUT command block. For the general, analysis type independent options for output of analysis results see § 3.6 on page 55.

syntax

```

BEGIN OUTPUT [ device_w ] [ outopt_w... ] [ params ]
[ OFF ]
[ BEGIN SELECT
[ ... ]
[ MODES _____ ]
      modes_n...
      ALL
END SELECT ]

```



```
[ LAYOUT ... ]
  itemw ...
DISPLA
END OUTPUT
```

SELECT customizes the batch output selection.

... for model selection see § 3.6.2 on page 59.

MODES selects specific eigenmodes for output: *modes* is a series of mode numbers,
ALL selects all calculated modes. [ALL]

LAYOUT optional commands to customize the layout and style of tabular output [§ 3.6.4.1 p. 65].

item is the name of the analysis result to be output. See § 3.6 on page 55 for complete syntax of this command.

DISPLA for normalized eigenmodes and participation vectors.

syntax

```
DISPLA [ typew ] [ formw ] [ operw ] { compw } { optiw }
      TOTAL      TRANSL      LOCAL
      MATRIX     ROTATI     GLOBAL
      PARTIC
```

DISPLA specifies eigenmodes (displacements) as output item after an eigenvalue analysis.

type specifies the displacement type. [TOTAL]

TOTAL for the total displacements of a structure, i.e., the deformed geometry of the eigenmode. Prior to outputting them, DIANA normalizes the eigenmodes in such a way that the largest translation displacement component has a value of 1.

MATRIX for the total displacement of a structure, i.e., the deformed geometry of the eigenmode. Prior to outputting them, DIANA normalizes the eigenmodes with respect to the matrix used in the eigenvalue analysis, i.e., the mass matrix for a free vibration eigenvalue analysis, the identity matrix for the standard eigenvalue analysis, or the geometric stress–stiffness matrix for a linearized buckling analysis.

DIANA calculates the normalized eigenvectors with respect to the mass matrix according to Eq. (45.18) on page 542. For the other types of eigenvalue analysis the mass matrix will be replaced by the appropriate matrix. See also § 45.2.2 on page 542 for the used matrices.

PARTIC for the participation vectors, i.e., weighted eigenmodes, calculated according to Eq. (45.20) on page 543. This output is only available for a free vibration eigenvalue analysis.

form specifies the displacement formulation [§ 3.6.1 p. 57]. [TRANSL]

oper specifies an operation (transformation) to be performed on the displacements [§ 3.6.1 p. 57]. [GLOBAL]

comp selects displacement components for output. Default is all available components. [all]

opti are additional options [§ 3.6.1 p. 58].

Default

file.dcf

```
*EIGEN
[ commands ]
OUTPUT
*END
```

If you only give a single OUTPUT command, like in the above example, or if you do not give the OUTPUT command at all, then DIANA gives output of the translational displacements in global *XYZ* orientation, as if you had given the following commands.

file.dcf

```
*EIGEN
[ commands ]
BEGIN OUTPUT
  DISPLA TOTAL TRANSL GLOBAL
END OUTPUT
*END
```

<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	<i>comp ...</i>		
				X	Y	Z
DISPLA	TOTAL	TRANSL	LOCAL	Dtx	Dty	Dtz
				u_x	u_y	u_z
DISPLA	TOTAL	TRANSL	GLOBAL	DtX	DtY	DtZ
				u_X	u_Y	u_Z
DISPLA	TOTAL	ROTATI	LOCAL	Drx	Dry	Drz
				ϕ_x	ϕ_y	ϕ_z
DISPLA	TOTAL	ROTATI	GLOBAL	DrX	DrY	DrZ
				ϕ_X	ϕ_Y	ϕ_Z

<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	<i>comp ...</i>		
				X	Y	Z
DISPLA	MATRIX	TRANSL	LOCAL	Dmtx	DmtY	Dmtz
				$u_{m.x}$	$u_{m.y}$	$u_{m.z}$
DISPLA	MATRIX	TRANSL	GLOBAL	DmtX	DmtY	DmtZ
				$u_{m.X}$	$u_{m.Y}$	$u_{m.Z}$
DISPLA	MATRIX	ROTATI	LOCAL	DmrX	DmrY	DmrZ
				$\phi_{m.x}$	$\phi_{m.y}$	$\phi_{m.z}$
DISPLA	MATRIX	ROTATI	GLOBAL	DmrX	DmrY	DmrZ
				$\phi_{m.X}$	$\phi_{m.Y}$	$\phi_{m.Z}$

<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	<i>comp ...</i>		
				X	Y	Z
DISPLA	PARTIC	TRANSL	LOCAL	Dptx	Dpty	Dptz
				$u_{p.x}$	$u_{p.y}$	$u_{p.z}$
DISPLA	PARTIC	TRANSL	GLOBAL	DptX	DptY	DptZ
				$u_{p.X}$	$u_{p.Y}$	$u_{p.Z}$
DISPLA	PARTIC	ROTATI	LOCAL	Dprx	DprY	Dprz
				$\phi_{p.x}$	$\phi_{p.y}$	$\phi_{p.z}$
DISPLA	PARTIC	ROTATI	GLOBAL	DprX	DprY	DprZ
				$\phi_{p.X}$	$\phi_{p.Y}$	$\phi_{p.Z}$

Part X

Reinforcement Grid Design Checking

Glossary of symbols. This glossary is an addition to the general Glossary of Symbols for DIANA on page [xxix](#). It contains the symbols used particularly in reinforcement grid design checking.

Scalars

A	Applied area of reinforcement.
A_{\max}	Maximum area of reinforcement.
A_{\min}	Minimum area of reinforcement.
A_{req}	Required area of reinforcement.
A_{rat}	Ratio of required area of reinforcement over applied area of reinforcement.
co	Coverage.
d	Useful height.
s	Spacing between reinforcement bars.
s_{\max}	Maximum spacing between reinforcement bars.
h_t	Height/thickness.
ϕ	Diameter of reinforcement bar.
ϕ_{\max}	Maximum diameter of bars.
z_r	Relative internal arm.
n'^c	Combined reinforcement forces.

Chapter 32

Introduction to Reinforcement Grid Design Checking

32.1 Background

The DIANA program is mostly used for nonlinear failure analysis of structures. Although the new Eurocode 2 EN 1992-1-1 allows checking of the design of a structure by performing a full nonlinear analysis, the design engineer still has to prove that standard design checks with respect to the amount of steel reinforcements are satisfied. The ***DESIGN** application in DIANA allows the user to perform the most important design checks with respect to reinforcement grids in concrete structures in the same finite element model that can be used for a nonlinear failure analysis of the structure.

Whereas a nonlinear failure analysis evaluates in each integration point the stresses based on the equilibrium condition and the nonlinear material characteristics, the design check analysis checks the cross-section bending moments and forces resulting from a linear analysis against the condition that the failure stresses in the reinforcement and concrete are in equilibrium.

32.2 Results

The basic results in a design check are the linear cross-section forces and bending moments. Although all available element types in DIANA can be applied in a design checking analysis, at the moment the reinforcement grids can only be checked in three types of elements:

- grid reinforcements in curved shell elements
- grid reinforcements in layered shell elements
- grid reinforcements in solids with composed surface elements

In curved and layered shell elements the cross-section forces and bending moments are calculated with reference to the neutral plane of the elements. For solids the reference plane must be defined by composed surface elements (see Volume *Element Library*) which must be defined by the user. For solids the cross-section forces and bending moments are calculated by integration of the stresses in the solid elements in the direction normal to the composed surface elements. Only the reinforcements located in solid elements that contribute to a composed surface element are considered in the design checks.

Chapter 33

Input for Reinforcement Grid Design Checking

The reinforcement grids to be considered in the design check must be identified by the label **DESIGN** in the corresponding geometry table. For design reinforcement grids instead of the usual equivalent thickness parameters, the explicit diameter and spacing of the bars in the grid in both directions must be defined as geometrical properties.

The design checks make use of table 7.2N and 7.3N of Eurocode 2 EN 1992-1-1[66]. This makes the checking unit dependent. That is why the user always has to specify table 'UNITS' [§ 1.1 p. 3].

33.1 Geometrical Properties

				<i>syntax</i>
'GEOMET'				
1	5	6	12 13	80
	DESIGN			
	PHI	<i>phi</i> _{<i>x</i>}	<i>phi</i> _{<i>y</i>}	
	SPACIN	<i>spaci</i> _{<i>x</i>}	<i>spaci</i> _{<i>y</i>}	
	[ZR	<i>zr</i> _{<i>r</i>}		
	[SPREAD	<i>spr</i> _{<i>r</i>}		

DESIGN identifies that the reinforcement grid must be considered in the design check.

PHI *phi*_{*x*} specifies the diameter ϕ_x of the bars in the reinforcement grid in local *x* direction. *phi*_{*y*} specifies the diameter ϕ_y of the bars in the reinforcement grid in local *y* direction. ($\phi_x \geq 0$)
($\phi_y \geq 0$)

If ϕ_x or ϕ_y are defined equal to zero, all results in the respective local grid direction in the respective grid reinforcements are set to zero, except the required area of reinforcement (ASREQ).

SPACIN *spaci*_{*x*} specifies the spacing of the bars in the grid reinforcement in local *x* direction s_x . *spaci*_{*y*} specifies the spacing of the bars in the grid reinforcement in local *y* direction s_y . ($s_x \geq 0$)
($s_y \geq 0$)

ZR *zr* specifies the relative internal arm z_r . When this parameter is not defined, the default value of 0.9 will be used. [$z_r = 0.9$]

SPREAD *spr* defines the length of the line that is required for calculating average results [§ 34.5 p. 454]. The default value for *spr* is two times the effective height of the element or composed element in which the grid is located. [$spr = 2 * z_r * h_t$]

The directions of the bars are defined with the local *x* and *y* axes of the reinforcement grid, which can be set by the item **XAXIS** in the usual way. DIANA will automatically

calculate and apply the equivalent thickness parameter **THICK** for the reinforcement grids and apply this parameter in the model for follow up analyses with other applications.

33.2 Material Properties

33.2.1 Reinforcement Steel

For the design reinforcement grid the Young's modulus and the yield stress must be defined as material properties:

<i>syntax</i>									
'MATERI'									
1	5	6	12	13					
					YOUNG		e_r		
					YLDSTR		$sigy_r$		

($E > 0$) YOUNG e is Young's modulus E .

($\sigma_{yd} > 0$) YLDSTR $sigy$ is the design value of the yield stress σ_{yd} .

Reinforcement grids which are labeled for design checking do neither contribute to the stiffness matrix nor to the cross-section forces and bending moments in a design analysis. However, reinforcements which are not labeled for design checks will contribute to the stiffness matrix and cross-section results in the usual way, e.g. prestressed tendons can be modeled as reinforcement bars and their prestress loading can be applied as one of the load conditions in the design check.

The useful height and coverage of a reinforcement grid are automatically calculated from the position of the grid relative to the reference plane of the grid element and the outer face, respectively.

33.2.2 Concrete

For the concrete the compressive and tensile strength must be defined as material properties. These properties can be either defined by direct input or by using a predefined material from the Eurocode 2 EN 1992-1-1 library (see Volume *Material Library*).

<i>syntax</i>									
Direct input									
'MATERI'									
1	5	6	12	13					
					FCTM		$fctm_r$		
					FCK		fck_r		
					[ENVIRO		$enviro_w$]		

($f_{ctm} > 0$) FCTM $fctm$ specifies the average tensile strength of the concrete f_{ctm} .

($f_{ck} > 0$) FCK fck specifies the characteristic compressive strength of the concrete f_{ck} .

ENVIRO $enviro$ specifies the environmental class given in Eurocode 2 EN 1992-1-1: X0, XC1, XC2, XC3, XC4, XD1, XS1, XD2, XS2, XD3, or XS3.

When ENVIRO is defined in the material properties, it overrules the global environmental class defined in the command file [§ 34.1 p. 451].

Eurocode 2 EN 1992-1-1 library					syntax
'MATERI '					
1	5	6	12	13	80
		CONCRE	EN1992		
		CLASS	<i>class_w</i>		
		[...	...]	optional parameters	
		[ENVIRO	<i>enviro_w</i>]		

CONCRE EN1992 indicates the Eurocode 2 EN 1992-1-1 model code regulations.

CLASS *class* specifies the concrete class, LC12/13, C12/15, LC16/18, C16/20, LC20/22, C20/25, etcetera, where the first number denotes the characteristic cylinder-compressive strength in MPa, and the second number denotes the characteristic cube-compressive strength f'_{ck} in MPa. Classes starting with LC refer to lightweight concrete, while classes starting with C refer to normal weight concrete.

For optional parameters of the Eurocode 2 EN 1992-1-1 predefined concrete classes, see Volume Material Library.

ENVIRO *enviro* specifies the environmental class given in Eurocode 2 EN 1992-1-1: X0, XC1, XC2, XC3, XC4, XD1, XS1, XD2, XS2, XD3, or XS3.

When ENVIRO is defined in the material properties, it overrules the global environmental class defined in the command file [§ 34.1 p. 451].

Chapter 34

Reinforcement Grid Design Checking Analysis

The command syntax of a design checking analysis is as follows:

syntax

```
*DESIGN
[ CODE ... ]
[ EXECUT ... ]
[ COMBIN ... ] ...
[ ENVELO ... ] ...
[ OUTPUT ... ] ...
*END
```

CODE defines Eurocode 2 EN 1992-1-1 settings for design checking: The national annex code, the environmental and construction class can be defined [§ 34.1].

EXECUT defines the execution of the linear analysis and the calculation of the distributed bending moments and forces in reference planes [§ 34.2].

COMBIN defines one load case combination [§ 34.3 p. 453]. This command block may be repeated. Each load case combination may contain only a single mobile load case.

ENVELO defines one load envelope [§ 34.4 p. 454]. This command block may be repeated. Envelopes may be defined over load case or load combinations or combinations of both.

OUTPUT defines results to be output to an output device [§ 34.5 p. 454]. This command block may be repeated, e.g. for different output devices. Output can be selected for load cases, load combinations, or load envelopes. The output command executes the definition of load combinations and load envelopes as defined in the command blocks above.

34.1 Eurocode 2 EN 1992-1-1 Design Checking

The **CODE** command block defines settings for the design checking according to Eurocode 2 EN 1992-1-1.

syntax

```
BEGIN CODE
[ ANNEX annexw ]
      DUTCH
      NONE
```

[ENVIRO *enviro_w*]
 [CONSTR *constr_w*]
 END CODE

ANNEX *annex* defines which national annex will be used in combination with Eurocode 2 EN 1992-1-1. Table 34.1 presents the parameters according to the chosen annex.

[NONE]

ENVIRO *enviro* specifies the environmental class according to Eurocode 2 EN 1992-1-1:
 [XC4] X0, XC1, XC2, XC3, XC4, XD1, XS1, XD2, XS2, XD3, or XS3.

CONSTR *constr* defines the construction class according to Eurocode 2 EN 1992-1-1: S1,
 [S4] S2, S3, S4, S5, or S6.

Table 34.1: NATIONAL ANNEX PARAMETERS

Parameter	Symbol	NONE	DUTCH
Partial factor concrete	γ_c	1.5	1.5
Partial factor steel	γ_s	1.15	1.15
Min. reinforcement area	$A_{s,min}$	$0.26 f_{ctm} d / (f_{yk} d)$ $0.0013 d$ $1.2 A_{S_{req}}$	$0.26 f_{ctm} d / (f_{yk} d)$ $0.0013 d$ $1.25 A_{S_{req}}$
Max. reinforcement area	$A_{s,max}$	$0.04 h$	$0.04 h$
Additional safety margin	$\Delta c_{dur,g}$	0 mm	0 mm
Red. min. coverage stainless steel	$\Delta c_{dur,st}$	0 mm	0 mm
Red. min. coverage add. protection	$\Delta c_{dur,add}$	0 mm	0 mm
Performance tolerance	Δc_{dev}	10 mm	5 mm
Long term effects comp. strength	α_{cc}	1.0	1.0
Coefficient	$C_{Rd,c}$	$0.18 / \gamma_c$	$0.18 / \gamma_c$
Coefficient	v_{min}	$0.035 k^{3/2} \cdot f_{ck}^{1/2}$	$0.035 k^{3/2} \cdot f_{ck}^{1/2}$
Coefficient	k_1	0.15	0.15

34.2 Linear Analysis

The EXECUT command block is defined as follows:

syntax

```
BEGIN EXECUT
[ OFF ]
[ MODEL ... ]
[ SOLVE ... ]
END EXECUT
```

OFF suppresses the execution of the linear analysis.

MODEL evaluates and assembles the finite element model [§ 34.2.1].

SOLVE allows to define non-default solver settings. See Chapter 30 for customization of the solution procedure.

34.2.1 Model Evaluation

The MODEL commands customize the evaluation of the finite element model prior to the actual analysis.

syntax

```

BEGIN MODEL
[ OFF ]
[ EVALUA [ OFF ] ... ]
[ ASSEMB [ OFF ] ... ]
[ MATRIX [ OFF ] ]
[ LOADS [ OFF ] ]
END MODEL

```

EVALUA to check and evaluate geometric and material properties for elements [§ 3.4 p. 51].

ASSEMB to assemble the elements, i.e., creating appropriate system degrees of freedom [§ 3.5 p. 54].

MATRIX to setup the element stiffness matrices.

LOADS to setup the load vectors.

Provided that the FILOS file of a previous analysis is still available, some tasks may be skipped which may save a considerable amount of computing time, especially for large finite element models.

34.3 Load Case Combination

The COMBIN command block is defined as follows:

syntax

```

BEGIN COMBIN
[ OFF ]
[ NAME names ]
[ TYPE types ]
      ULS
      SLS
CASES casesn...
FACTOR facr...
END COMBIN

```

NAME defines the name of the load combination. These names can be referred to in load envelope definitions or in output selection. *name* must be a string of maximum 18 characters.

TYPE defines whether the load case is a Ultimate Limit State (ULS) (*type* = ULS) or a Serviceability Limit State (SLS) (*type* = SLS). [ULS]

CASES defines the load case numbers or load combination numbers as defined in table 'LOADS'.

FACTOR defines the factors for the load cases defined in the **CASES** record. The number of integer items following **CASES** must be equal to the number of reals following **FACTOR**.

34.4 Load Envelope

An envelope load is defined as a new combination for which the basic results (displacements, distributed forces, and distributed moments) are chosen for a set of selected load cases and load combinations, such that for every element node the load case combination with the minimum or maximum selected result component is searched and for the node all basic results from this load case combination are placed in the load envelope. The ENVELO command block is defined as follows:

syntax

```

BEGIN ENVELO
[ NAME names ]
[ TYPE typew ]
      MIN
      MAX
BEGIN RESULT
      DTX
      DTY
      DTZ
      NXX
      NYY
      NXY
      QYZ
      QXZ
      MXX
      MYY
      MXY
END RESULT
BEGIN SELECT
      CASE casen...
      COMBIN combins...
      TYPE typew
            ULS
            SLS
END SELECT
END ENVELO

```

NAME defines the name of the load envelope. These names can be referred to in in output selection. *name* must be a string of maximum 18 characters.

- [MAX] TYPE defines whether the envelope scan is done for minimum (*type* = MIN) or for maximum results (*type* = MAX).
- [MXX] RESULT defines the result type over which the envelope scan is performed: DTX, DTY, and DTZ for displacement components referring to global coordinate system; NXX, NYY, NXY, QYZ, and QXZ for distributed force components referring to the local element coordinate system; MXX, MYY, and MXY for distributed moment components referring to the local element coordinate system.
- [ULS] SELECT defines the load cases numbers, load combinations or types, i.e. Ultimate Limit State (*type* = ULS) or a Serviceability Limit State (*type* = SLS), over which the envelope scan shall be performed.

34.5 Output of Analysis Results

There are four types of output in the *DESIGN application:

- Output related to the reinforcement area, names started with the letters **AS**. This output is related to the Ultimate Limit State. There are also two unity check output, **UCMIN** and **UCMAX**, related to the reinforcement area. These unity checks are defined as ratios, so the values of these unity checks should be less or equal to 1.0 to comply with the Eurocode.
- Unity check output related to the spacing (**UCSPA**) and diameter (**UCPHI**) of the reinforcement. These two output items are related to the Serviceability Limit State. The unity check related to the shearforce (**UCSHR**) is related to the Ultimate Limit State. The unity check related to the coverage (**UCCOV**) checks the model input and is independent of the loading. The names of all these unity check output items start with **UC** and should be less or equal to 1.0 to comply with the Eurocode 2 EN 1992-1-1.
- Cross-section forces and bending moments in the shell elements or in the composed elements.
- Combined reinforcement forces in the upper and lower reinforcement grid of a cross-section.

The OUTPUT command block is defined as follows:

syntax

```

BEGIN OUTPUT [devicew] [outoptw...] [params]
[BEGIN SELECT
CASE casen...
COMBIN combins...
TYPE typew
      ULS
      SLS
ENVELO envelos
END SELECT ]
[ASREQ _____]
      REINFO
      ELEMEN
      INTPNT
      NODES
      AVERAG
[ASAPL ]
[ASRAT _____]
      REINFO
      ELEMEN
      INTPNT
      NODES
      AVERAG
[ASRXY _____]
      REINFO
      ELEMEN
      INTPNT
      NODES
[UCMIN _____]
      REINFO
      ELEMEN
      INTPNT
      NODES
      AVERAG
[UCMAX ]

```

```

[UCPHI _____]
    REINFO
    ELEMEN
    INTPNT
    NODES
    AVERAG
[UCSPA _____]
    REINFO
    ELEMEN
    INTPNT
    NODES
    AVERAG
[UCCOV _____]
    REINFO
    ELEMEN
    INTPNT
    NODES
[UCSHR _____]
    REINFO
    ELEMEN
    INTPNT
    NODES
[STRESS DISFOR]
[STRESS DISMOM]
[STRESS COMBIN _____]
    REINFO
    ELEMEN
    INTPNT
    NODES

```

END OUTPUT

device specifies the output device: **TABULA** for tabular output, **FXPLUS** for interactive postprocessing of analysis results with **FX⁺**, **FEMVIE** for output to a database for the **¡DIANA Results** environment. Note that the layout of tabular output can not be specified in this application.

SELECT defines the load cases numbers, load combinations or load envelopes or load envelope types for which the results shall be output.

[INTPNT] **ASREQ** specifies the required area of the reinforcement A_{req} in local directions as output item.

Note that reinforcement will only be required when positive values for ASREQ are found. Therefore, calculated negative values for ASREQ will be reset to zero.

ASAPL specifies the applied area of the reinforcement A in local grid directions as output item. These results are always constant over the reinforcement.

[INTPNT] **ASRAT** specifies the ratio of the required area A_{req} over the applied reinforcement area A in local grid directions as output item.

[INTPNT] **ASRXY** specifies the ratio of the reinforcement area in local x direction over the reinforcement area in local y direction. This ratio is specified for both the required area A_{req} (result label **ASRQxy**) as for the applied reinforcement area A (result label **ASAPxy**).

[INTPNT] **UCMIN** specifies the unity check related to the minimum reinforcement area $A_{\text{s,min}}$ in local grid directions as output item. This unity check equals the ratio of the minimum reinforcement area according to the Eurocode over the applied area of reinforcement.

- UCMAX** specifies the unity check related to the maximum reinforcement area $A_{s,max}$ in local grid directions as output item. This unity check equals the ratio of the maximum reinforcement area according to the Eurocode over the applied area of reinforcement. These results are always constant over the reinforcement.
- UCPHI** specifies the unity check related to the maximum diameter of the reinforcement bar ϕ_s^* for crack control in local grid directions as output item. This unity check equals the ratio of the applied diameter of the reinforcement bar over the maximum diameter of the reinforcement bar ϕ_s^* according to the Eurocode. [INTPNT]
- UCSPA** specifies the unity check related to the maximum spacing between the reinforcement bars for crack limitation/restriction in local grid directions as output item. This unity check equals the ratio of the applied spacing of the reinforcement bars over the maximum spacing of the reinforcement bars according to the Eurocode. [INTPNT]
- UCCOV** specifies the unity check related to the coverage of the reinforcement grid. This unity check equals the ratio of the nominal coverage over the applied coverage according to the Eurocode. [INTPNT]
- UCSHR** specifies the unity check related to the shearforce. This unity check equals the ratio of the shearforce over the design concrete shear ($V_{rd,c}$) according to the Eurocode. [INTPNT]
- STRESS DISFOR** specifies the cross-section distributed forces in the local coordinate system of the shell or composed element.
- STRESS DISMOM** specifies the cross-section distributed moments in the local coordinate system of the shell or composed element.
- STRESS COMBIN** specifies the combined reinforcement forces as defined by Eqs. (35.1-35.3) in the local coordinate system of the upper and lower reinforcement grids. [NODES]

If **ELEMEN** or **REINFO** is specified the maximum value per element or grid is calculated based on integration point results and output as a constant value for all integration points of the element or reinforcement grid, respectively. If **INTPNT** is specified the result values in each integration point of the grid will be output individually. If **NODES** is specified the result values in each node of the grid reinforcement is calculated and output individually. If **AVERAG** is specified, for each integration point in a grid and for each result component in the direction in the grid normal to the result component direction, a line with length as defined with parameter **SPREAD** in table 'GEOMET' [§ 33.1 p. 447] is defined. The average results over the reinforcement particles that are intersected by this line are calculated and displayed in the integration points.

Example.

file.dcf

```
*DESIGN
EXECUT
BEGIN COMBIN
  NAME ULS1
  CASES 1 2 3 /
  FACTOR 1. 2. 3. /
  TYPE ULS
END COMBIN
BEGIN COMBIN
  NAME ULS2
  CASES 1 2 3 /
  FACTOR 3. 1. 1. /
  TYPE ULS
END COMBIN
BEGIN ENVELO
  NAME MAX-ULS
```

```

TYPE MAX
RESULT MOMENT COMP XX
SELECT COMBIN ULS1 ULS2
END ENVELO
BEGIN OUTPUT FEMVIE
  SELECT ENVELO MAX-ULS
  ASRAT ELEMEN
END OUTPUT
BEGIN OUTPUT
  BEGIN SELECT
    CASE 1 2
    ENVELO MAX-ULS
  END SELECT
  ASREQ ELEMEN
  ASAPL
  UCMIN REINFO
END OUTPUT
*END

```

In this example two load combinations are defined with names ULS1 and ULS2. One load envelope is defined with the name MAX-ULS as a scan over the maximum of the distributed moment xx component for both load combinations. The required cross-section over applied cross-section ratio is output as `iDIANA` result. The required and applied cross-section and minimum reinforcement for load case 1, load case 2, and for the load envelope are output as tabular result.

Chapter 35

Result Definitions

The result definition of design checking according to the Eurocode[66] is as follows:

Reinforcement bar diameter: ϕ

Spacing between bars: s

Coverage: co

Thickness: h_t

Bar cross-section: $A = \frac{1}{4}\pi\phi^2$

Tensile strength: f_{ctm}

Useful height: $d = h_t - co - \frac{1}{2}\phi$

Absolute internal beam arm: $z_d = z_r * d$

Design yield strength of steel: f_{yd}

Cross-section forces: $n_{xx}, n_{yy}, n_{xy}, q_{yz}, q_{xz}$

Cross-section bending moments: m_{xx}, m_{yy}, m_{xy}

35.1 Reinforcement Area Checks

The design checks on the reinforcement area are related to the Ultimate Limit State (ULS). These checks are strongly related to the calculation of the reinforcement forces and moments [§ 47.2.6 p. 576].

The combined reinforcement forces, n'_{xx} , n'_{yy} , n'_{xy} for reinforcement grids above the neutral plane are defined as:

$$n'_{xx} = \frac{n_{xx}}{2} + \frac{m_{xx}}{z_d} \quad (35.1)$$

$$n'_{yy} = \frac{n_{yy}}{2} + \frac{m_{yy}}{z_d} \quad (35.2)$$

$$n'_{xy} = \left| \frac{n_{xy}}{2} + \frac{m_{xy}}{z_d} \right| \quad (35.3)$$

For reinforcement grids below the neutral plane the combined reinforcement are defined as:

$$n'_{xx} = \frac{n_{xx}}{2} - \frac{m_{xx}}{z_d} \quad (35.4)$$

$$n'_{yy} = \frac{n_{yy}}{2} - \frac{m_{yy}}{z_d} \quad (35.5)$$

$$n'_{xy} = \left| \frac{n_{xy}}{2} - \frac{m_{xy}}{z_d} \right| \quad (35.6)$$

The required area of the reinforcement in local directions is defined as:

$$A_{\text{req},x} = (n'_{xx} + n'_{xy})/f_{yd} \quad (35.7)$$

$$A_{\text{req},y} = (n'_{yy} + n'_{xy})/f_{yd} \quad (35.8)$$

The applied area of the reinforcement in local directions is defined as:

$$A_x = \left(\frac{1}{4} \pi \phi_x^2 \right) / s_x \quad (35.9)$$

$$A_y = \left(\frac{1}{4} \pi \phi_y^2 \right) / s_y \quad (35.10)$$

The ratio of the required area over the applied area in local directions is defined as:

$$A_{\text{rat},x} = \frac{A_{\text{req},x}}{A_x} \quad (35.11)$$

$$A_{\text{rat},y} = \frac{A_{\text{req},y}}{A_y} \quad (35.12)$$

The ratio of the required area in local x direction over the required area in local y direction is defined as:

$$A_{\text{req},xy} = \min(A_{\text{req},x}/A_{\text{req},y}, A_{\text{req},y}/A_{\text{req},x}) \quad (35.13)$$

For the applied area holds:

$$A_{xy} = \min(A_x/A_y, A_y/A_x) \quad (35.14)$$

The unity check on the minimum reinforcement area as defined in §9.2.1.1 of the Eurocode[66] is defined as:

$$A_{s,\min} = \min(A_{\min1}, A_{\min2}) \quad (35.15)$$

$$A_{\min1} = \max(0.0013 * d, 0.26 * f_{ctm} * d / f_{yk}) \quad (35.16)$$

Where f_{yk} is the characteristic value for the yield stress:

$$f_{yk} = \gamma_s * f_{yd} \quad (35.17)$$

$$A_{\min2} = 1.2 * A_{\text{req}} \quad (35.18)$$

$$\text{UCMIN}_x = A_{s,\min} / A_x \quad (35.19)$$

$$\text{UCMIN}_y = A_{s,\min} / A_y \quad (35.20)$$

The unity check on the maximum reinforcement area as defined in §9.2.1.1 of the Eurocode[66] is defined as:

$$A_{\max} = 0.04 * A_c \quad (35.21)$$

Where A_c represents the concrete area of the cross-section.

$$\text{UCMAX}_x = \frac{A_x}{A_{\max}} \quad (35.22)$$

$$\text{UCMAX}_y = \frac{A_y}{A_{\max}} \quad (35.23)$$

35.2 Diameter and Spacing Checks

The design checks on spacing and diameter of the reinforcements are related to the Serviceability Limit State (SLS). They are based on §7.3.3 of the Eurocode[66].

The effective stress in the reinforcement bars is calculated as:

$$\sigma = f_{yd} * \frac{A_{req}}{A} \quad (35.24)$$

From Table 7.2N of the Eurocode[66] and the calculated effective stress σ , the maximum required bar diameter ϕ_s^* is calculated. This maximum bar diameter will be adapted using Equation 7.7N of the Eurocode:

$$\phi_s = \phi_s^* (F_{ctm}/2.9) * \frac{1}{2} * h_t / (2 * (h - d)) \quad (35.25)$$

The adapted maximum bar diameter is used to calculate the unity check related to the diameter of the reinforcement bar in local directions:

$$UCPHI_x = \frac{\phi_x}{\phi_s} \quad (35.26)$$

$$UCPHI_y = \frac{\phi_y}{\phi_s} \quad (35.27)$$

From Table 7.3N of the Eurocode and the calculated effective stress σ the unity check on the maximum spacing in local directions can be defined:

$$UCSPA_x = \frac{s_x}{s_{max}} \quad (35.28)$$

$$UCSPA_y = \frac{s_y}{s_{max}} \quad (35.29)$$

35.3 Coverage Check

The design check on the coverage is a model check independent of the loading. The nominal coverage c_{nom} equals Equation 4.1 of the Eurocode:

$$c_{nom} = c_{min} + \Delta c_{dev} \quad (35.30)$$

The minimum coverage c_{min} can be calculated with Equation 4.2 of the Eurocode:

$$c_{min} = \max(c_{min,b}, c_{min,dur} + \Delta c_{dur,g} - \Delta c_{dur,st} - \Delta c_{dur,add}, 10 \text{ mm}) \quad (35.31)$$

$c_{min,b}$ is equal to the bar diameter. The parameters $c_{min,dur}$, $\Delta c_{dur,g}$, $\Delta c_{dur,st}$, $\Delta c_{dur,add}$, and Δc_{dev} are dependent on the chosen annex [§ 34.1 p. 451]. $c_{min,dur}$ is based on table 4.4N of the Eurocode and depends on the environmental and construction class [§ 34.1]. The unity checks on coverage can be defined as:

$$UCCOV_x = \frac{c_{nom,x}}{c o_x} \quad (35.32)$$

$$UCCOV_y = \frac{c_{nom,y}}{c o_y} \quad (35.33)$$

35.4 Shearforce Check

The design checks on shearforce are related to the Ultimate Limit State. The design concrete shear $V_{Rd,c}$ can be calculated using Equation 6.2a of the Eurocode:

$$V_{Rd,c} = \left(C_{rRd,c} \cdot k \cdot (100 \cdot \rho_1 \cdot f_{ck})^{1/3} + k_1 \cdot \sigma_{cp} \right) \cdot d \quad (35.34)$$

The minimum value for $V_{\text{Rd,c}}$ is given in Equation 6.2b of the Eurocode:

$$V_{\text{Rd,c}} = (v_{\text{min}} + k_1 \cdot \sigma_{\text{cp}}) \cdot d \quad (35.35)$$

With

$$k = 1 + \sqrt{(200/d)} \leq 2.0 \quad (35.36)$$

$$\rho_1 = A_{\text{sl}}/(b_w \cdot d) \leq 0.02 \quad (35.37)$$

$$A_{\text{sl},y} = \frac{1}{4} \pi \phi_y^2 1000 / s_y \quad (35.38)$$

$$A_{\text{sl},x} = \frac{1}{4} \pi \phi_x^2 1000 / s_x, \quad (35.39)$$

$$\sigma_{\text{cp}} = n_{xx}/d < 0.2 f_{\text{cd}} \quad (35.40)$$

Where f_{ck} and f_{cd} in MPa and d in mm.

The parameters $C_{\text{Rd,c}}$, v_{min} and k_1 depend on the chosen annex [§ 34.1 p. 451]. The unity checks on the shearforce can be defined as:

$$\text{UCSHR}_x = q_{xz}/V_{\text{Rd,c},x} \quad (35.41)$$

$$\text{UCSHR}_y = q_{yz}/V_{\text{Rd,c},y} \quad (35.42)$$

Part XI

Stiffness Adaptation Analysis

Chapter 36

Introduction to Stiffness Adaptation Analysis

36.1 Background

As alternative for a full nonlinear analysis with application NONLIN for calculating load distributions, deformations, crack patterns, and crack openings in constructions with nonlinear materials, such as reinforced concrete, a stiffness adaptation analysis with application STADAP may be performed.

A stiffness adaptation analysis performs a sequence of linear static analyses, where in a subsequent iteration the elastic stiffness will be reduced in those integration points in which the stresses in a previous iteration were beyond a user-specified uniaxial stress-strain curve. In such case the isotropic elastic stiffness model is changed into an orthotropic elastic stiffness model with a reduced stiffness in the direction of the maximum stress, such that, with the same strain in the integration point, the maximum stress will be mapped on the stress-strain curve.

The stiffness adaptation method in DIANA is originally based on the concept as proposed in the thesis of De Boer[9]. Among other differences between both methods, in DIANA the stiffness is evaluated and modified in every integration point and in every principal strain direction, whereas De Boer reduced stiffness of the complete element and only checked the highest principal tensile strain.

36.2 Stiffness Reduction Method

In stiffness adaptation analysis both standard linear elastic material as well as nonlinear material behaviour can be defined. Nonlinear materials can be defined through a uniaxial stress-strain curve, both in the tensile and in the compressive regime. Nonlinear stress-strain curves may also be defined for bar and grid reinforcements.

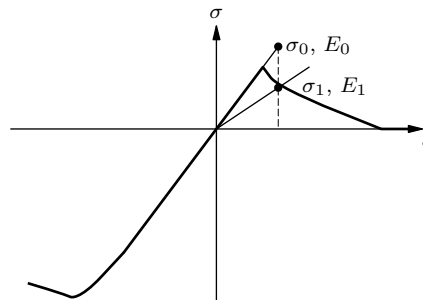


Figure 36.1: Uniaxial stress-strain curve with stress and stiffness reduction

Figure 36.1 displays that a maximum stress σ_0 in an integration point is beyond the stress-strain curve. As a consequence, in the respective integration point and stress

direction the stiffness E_0 is reduced in the next iteration such that the related stress σ_1 and corresponding stiffness E_1 in this direction exactly is located on the stress-strain curve. Because of the local stiffness change the deformation of the model will change in the next iteration. In general a number of iterations will be required before for all the integration points in the model the stress-strain combinations are on or below the user-defined stress-strain curve for the user-defined load level. DIANA applies a tolerance of 1% in the stress as criterion whether stiffness needs to be adapted or not.

In every integration point, two stiffness values are used: one for the tensile regime in the direction of the highest principal stress, and one for the compressive regime in the direction of the lowest principal stress. In three-dimensional stress conditions such as in solids and shell elements for the stiffness in the direction of the second principal stress, the highest stiffness from the other two directions is applied. Initially the user defined linear elastic stiffness is applied in all directions. The Poisson's ratio and shear stiffness are reduced with the same ratio at the maximum stiffness reduction in the respective integration point. The compressive strength is automatically reduced according to the relation as defined by Vecchio and Collins as applied to the Total Strain crack model in DIANA, see Volume *Material Library*. The stiffness reduction is limited to 0.001 times the original Young's modulus.

36.3 Load Increments

In a stiffness adaptation analysis different loads can be applied subsequently to simulate the loading history of the construction. Every loading can be applied in one step or in several steps with constant or varying step sizes. The user defines the maximum number of iterations per load step. If at a certain load level no further stiffness adaptations are required, because in all integration points the maximum stresses are within the 1% tolerance on or below the stress-strain curve, DIANA will stop the iteration process for that load step and continue to the next load step [Fig. 36.2]. In a stiffness adaptation analysis the material status parameters never can be corrected in later iterations, in contrary to full nonlinear analysis where the update of material status parameters is first confirmed at the end of a load step. Therefore, it is important to select the load increments carefully. If too large load increments are chosen when cracks develop in the model, the number of elements with stresses beyond the stress-strain curve will be large and the number of elements for which stiffnesses needs to be adapted is also large. In such situation large load increments may lead to widely spread areas with stiffness reduction, whereas, when smaller load increments are defined, the damage or cracks will be much more localized, resulting in a single line of elements with reduced stiffness.

In many cases it is sufficient to limit the maximum number of iterations per load step to 10 or 25, also when after this number of iterations still some stresses are beyond the stress-strain curve, because these errors are usually small. In order to describe the load and damage history more accurate it is recommended to apply more and smaller load steps with a lower maximum number of iterations per load step then using fewer, but larger steps with a higher number of iterations per load step.

36.4 Comparison with Full Nonlinear Analysis

Stiffness adaptation analysis can be used efficiently for calculating load distributions in nonlinear structures, for deformations at different load levels, and for crack patterns and crack openings as well as for plastic deformations in reinforcements. In comparison with full nonlinear analysis (*NONLIN) the analysis times of the stiffness adaptation method will be shorter, and no advanced analysis procedures and settings are required. Stiffness adaptation analysis can effectively be used to predict cracks and crack openings in serviceability loading conditions. For ultimate limit state analysis and analysis with ambient influences the use of a full nonlinear analysis with application *NONLIN is recommended.

In the present implementation of stiffness adaptation analysis combination with other modules, such as phased analysis is not allowed.

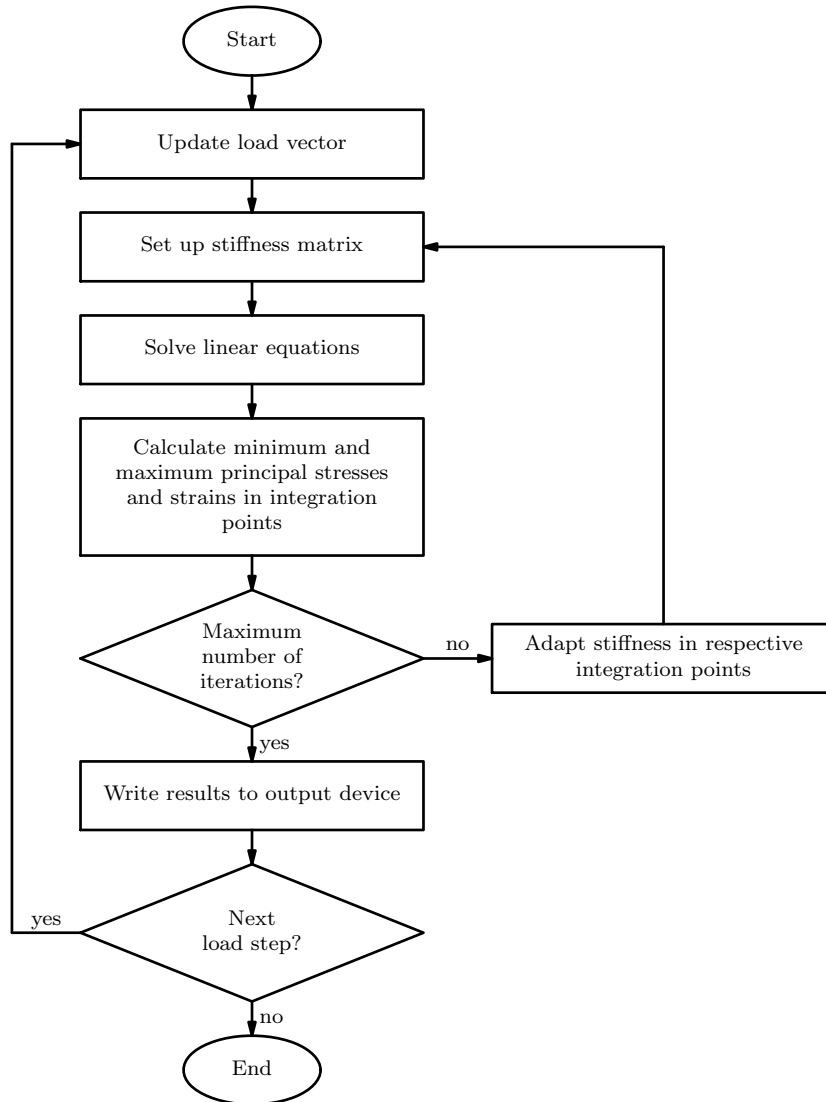


Figure 36.2: Flowchart of analysis sequence for stiffness adaptation analysis

36.5 Element Types

In stiffness adaption analysis nonlinear stress-strain curves can be assigned to the following element types:

- Two-dimensional membrane elements
- Plane strain and axisymmetric elements, excluding axisymmetric shells and infinite shells
- Class-II and Class-III beam elements
- Truss elements
- Curved shell elements, excluding layered shell elements
- Solid elements
- Interface elements
- Bar and grid reinforcements

Other element types can be applied in a stiffness adaptation analysis, but only with linear elastic material properties.

Chapter 37

Input for Stiffness Adaptation Analysis

The material definition for nonlinear behaviour in stiffness adaptation analysis is similar to the Total Strain crack models. The direction of the reduced stiffness is coupled to the principal stress which can vary every iteration. The user has to specify linear elastic properties and a uniaxial stress-strain curve for tensile and for compressive behaviour. There is no need to define a shear retention factor. Additionally, Von Mises plasticity parameters can be defined for steel. Alternatively, predefined materials from concrete and steel model codes may be used [Vol. *Material Library*].

Direct input

syntax

'MATERI '					
1	5	6	12	13	80
		YOUNG		e_r	
		POISON		nu_r	
		[TENCrv		$tenmod_w$]	
		...			
		[COMCrV		$commod_w$]	
		...			
		[YIELD		VMISES	
		YLDSTR		sy_r	
		KAPSIG		$k1_r$ $sy1_r$ [$k2_r$ $sy2_r$...] kn_r syn_r]	

YOUNG e is the Young's modulus E .

POISON nu is the Poisson's ratio ν .

TENCrv specifies the nonlinear tensile stress-strain curve of type $tenmod$ followed by specific parameters as specified in [§ 37.1 p. 470].

COMCrV specifies the nonlinear compressive stress-strain curve of type $commod$ followed by specific parameters as specified in [§ 37.2 p. 474].

YIELD VMISES specifies the Von Mises plasticity model with either ideal plasticity or a hardening diagram.

YLDSTR sy is the yield stress σ_y for ideal plasticity.

KAPSIG specifies a hardening diagram for the Von Mises plasticity model: $k1$ to kn the equivalent plastic strains κ and $sy1$ to syn are the corresponding yield stresses σ_y .

($n \leq 100$)

Concrete model code libraries

syntax

'MATERI'							
1	5	6	12	13			
		CONCRE	<i>codnam_s</i>				
				<i>code specific parameters</i>	
		[PLASTN]					
		[CRACKN]					

CONCRE *codnam* indicates which concrete model code to use. For available model codes and code specific parameters, see Volume *Material Library*.

PLASTN indicates that the plasticity functions, i.e. nonlinear compressive stress-strain curve, according to the selected model code will be used.

CRACKN indicates that the cracking functions, i.e. the nonlinear tensile stress-strain curve, according to the selected model code will be used.

Steel model code libraries

syntax

'MATERI'							
1	5	6	12	13			
		STEEL	<i>codnam_s</i>				
				<i>code specific parameters</i>	

STEEL *codnam* indicates which steel model code to use. For available model codes and code specific parameters, see Volume *Material Library*.

37.1 Tensile Stress-Strain Curves

For a stiffness adaptation analysis model you can choose a predefined tension softening function by specification of the curve name and appropriate parameters.

syntax

'MATERI'							
1	5	6	12	13			
		[TENCrv	<i>curve_w</i>				
			ELASTI				
			CONSTA				
			BRITTL				
			LINEPS				
			LINEAR				
			EXPONE				
			HORDYK				
			MULTLN				
			JSCESO				
			JSCETS				
			MC1990				
			MC2010				
			FRCCON				
			DUTSIG				
		[...	...			<i>tensile parameters</i>	

TENCrv *curve* specifies a predefined tension softening function [Fig. 37.1]. Beyond the tensile strength f_t the shape of these curves is like the tension softening curves for the multi-directional fixed crack models. See Volume *Material Library* for background theory.

Tensile parameters. You must specify the tensile parameters, depending on the softening function, as outlined in the following.

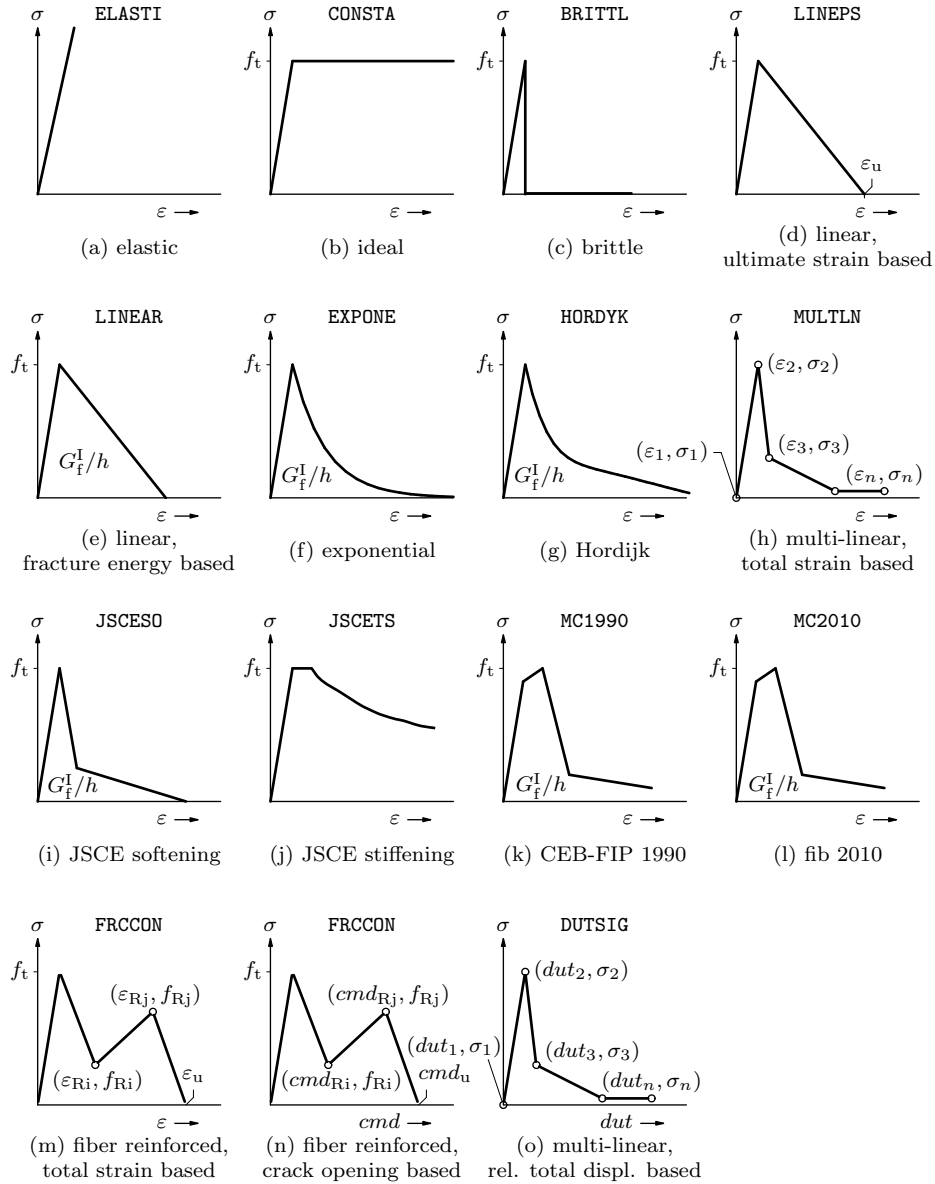


Figure 37.1: Predefined tension softening for stiffness adaptation analysis

Elastic*syntax*

'MATERI'

1	5	6	12	13	80
TENCrv			ELASTI		

ELASTI for elastic behaviour in tension, i.e., no stiffness adaptation [Fig. 37.1a].

Ideal and brittle*syntax*

'MATERI'

1	5	6	12	13	80
TENCrv			<i>curve_w</i>		
			CONSTA		
			BRITTL		
TENSTR			<i>ft_r</i>		

CONSTA for ideal behaviour [Fig. 37.1b].

BRITTL for brittle behaviour [Fig. 37.1c].

TENSTR ft is the tensile strength f_t .

Linear tension softening – based on ultimate strain

syntax

'MATERI'

1	5	6	12	13	80
		TENCRV		LINEPS	
		TENSTR		ft_r	
		EPSULT		eu_r	

LINEPS for ultimate strain based linear softening [Fig. 37.1d].

TENSTR ft is the tensile strength f_t .

EPSULT eu is the Mode-I ultimate tensile strain ε_u as depicted in Figure 37.1d.

Tension softening curves – based on fracture energy

syntax

'MATERI'

1	5	6	12	13	80
		TENCRV		$curve_w$	
				LINEAR	
				EXPONE	
				HORDYK	
				JSCE50	
		TENSTR		ft_r	
		GF1		$gf1_r$	

LINEAR for fracture energy based linear softening [Fig. 37.1e].

EXPONE for exponential softening [Fig. 37.1f].

HORDYK for softening according to Hordijk et al. [Fig. 37.1g].

JSCE50 for softening according to JSCE [48] [Fig. 37.1i].

TENSTR ft is the tensile strength f_t .

GF1 $gf1$ is the Mode-I fracture energy G_f^I .

Tension softening curves based on fracture energy is not available for the stiffness reduction model with constant strain increments.

Multi-linear – based on total strain

syntax

'MATERI'

1	5	6	12	13	80
		TENCRV		MULTLN	
		EPSIGT		$e1_r \ s1_r \ [\ e2_r \ s2_r \ \dots] \ en_r \ sn_r$	

A multi-linear diagram with indicator MULTLN fully describes the strain–stress relation in the tensile region, therefore input of the tensile strength f_t is not necessary.

MULTLN for a multi-linear diagram [Fig. 37.1h]. With a multi-linear diagram the cracking of the concrete is initiated at a strain where the tensile stress starts to decrease for the first time.

EPSIGT are the points of the multi-linear diagram: n pairs of values (ε, σ) ; $e1 \dots en$ are the total strains ε , $s1 \dots sn$ are the corresponding tensile stresses σ . In general the curve should start with a linear elastic slope from the origin to the tensile strength f_t as in Figure 37.1h.

($n \leq 100$)

JSCE tension stiffening*syntax*

'MATERI '

1	5	6	12	13	80
		TENCRV		JSCETS	
		TENSTR		ft_r	
		EPSTU		etu_r	
		CPOWER		c_r	

JSCETS for JSCE tension stiffening [48] [Fig. 37.1j]:

$$\sigma = f_t(\varepsilon_{tu}/\varepsilon)^c \quad (37.1)$$

TENSTR ft is the tensile strength f_t .EPSTU etu is the end of plateau strain ε_{tu} .[$\varepsilon_{tu} = 0.0002$]CPower c is the power c .[$c = 0.4$]**CEB-FIP Model Code 1990***syntax*

'MATERI '

1	5	6	12	13	80
		TENCRV		MC1990	
		TENSTR		ft_r	
		GF1		$gf1_r$	
		[DMAX		$dmax_i$]	

MC1990 for tension softening according to Paragraph 2.1.4.4.2 of the European CEB-FIP Model Code 1990 [16] [Fig. 37.1k]:

TENSTR ft is the tensile strength f_t .GF1 $gf1$ is the Mode-I fracture energy G_f^I .DMAX $dmax$ is the maximum aggregate size of concrete in mm. Possible maximum aggregate sizes are 8, 16, or 32 mm.[$dmax = 16$ mm]**fib Model Code for Concrete Structures 2010***syntax*

'MATERI '

1	5	6	12	13	80
		TENCRV		MC2010	
		TENSTR		ft_r	
		GF1		$gf1_r$	

MC2010 for tension softening according to Paragraph 5.1.8.2 of the fib Model Code for Concrete Structures 2010 [31] [Fig. 37.1l]:

TENSTR ft is the tensile strength f_t .GF1 $gf1$ is the Mode-I fracture energy G_f^I .

Fiber reinforced concrete model (fib)*syntax*

'MATERI'

1	5	6	12	13	80
		TENCRV		FRCCON	
		<i>curve_w</i>			
		FRCEPS	f_{t_r}	f_{ri_r}	$epsri_r$
			frj_r	$epsrj_r$	$epsu_r$
		FRCCMD	f_{t_r}	f_{ri_r}	$cmodri_r$
			frj_r	$cmodrj_r$	$cmodu_r$

FRCCON for the fiber reinforced concrete model as defined by the *fédération internationale du béton*/International Federation for Structural Concrete (fib) working groups. The model can be either specified as function of the total strain [Fig. 37.1m] or as function of the crack opening [Fig. 37.1n]. With the fiber reinforced concrete curve the cracking of concrete is initiated at the strain where the tensile strength f_t is reached.

FRCEPS f_t is the tensile strength f_t , f_{ri} and $epsri$ represent reference point i in the stress-strain curve (ε_{Ri} , f_{Ri}), frj and $epsrj$ represent reference point j in the stress-strain curve (ε_{Rj} , f_{Rj}), and $epsu$ is the ultimate strain ε_u [Fig. 37.1m].

FRCCMD f_t is the tensile strength f_t , f_{ri} and $cmodri$ represent reference point i in the stress-crack opening curve (cmd_{Ri} , f_{Ri}), frj and $cmodrj$ represent reference point j in the stress-crack opening curve (cmd_{Rj} , f_{Rj}), and $cmodu$ is the ultimate crack opening cmd_u [Fig. 37.1n]. The crack strain is obtained by dividing the crack opening by the crack bandwidth h .

Multi-linear – based on relative total displacement*syntax*

'MATERI'

1	5	6	12	13	80
		TENCRV		DUTSIG	
		DUTSIG	$dut1_r$	$s1_r$	[$dut2_r$ $s2_r$...] $dutn_r$ sn_r

A multi-linear diagram with indicator DUTSIG fully describes the relation between relative total displacement and stress in the tensile region, therefore input of the tensile strength f_t is not necessary. The relative total displacement is defined as the product of the total strain and the crack bandwidth h . The total strain ε_{tot} relates to the crack strain ε_{crk} as follows:

$$\varepsilon_{crk} = \varepsilon_{tot} - f_t/E \quad (37.2)$$

where f_t is the stress at which the first decay of stress occurs and E is the initial Young's modulus. When E is relatively high, the crack strain approaches the total strain and the total relative displacement will be close to the crack mouth opening.

TENCRV DUTSIG for a multi-linear diagram [Fig. 37.1o]. With a multi-linear diagram the cracking of the concrete is initiated at a relative total displacement defined where the tensile stress starts to decrease for the first time.

DUTSIG are the points of the multi-linear diagram: n pairs of values (dut , σ); $dut1$... $dutn$ are the relative total displacements dut , $s1$... sn are the tensile stresses σ . In general the curve should start with a linear elastic slope from the origin to the tensile strength f_t as in Figure 37.1p.

(n ≤ 100)

37.2 Compressive Stress-Strain Curves

For a stiffness adaptation analysis model you can choose a predefined compressive softening function by specification of the curve name and appropriate parameters.

syntax

'MATERI'									
1	5	6	12	13					
		[COMCRV		<i>curve_w</i>					
				CONSTA					
				THOREN					
				LINHAR					
				MULTLN					
				SATURA					
				PARABO					
				EN1992					
				MC1990					
				MC2010					
				EC2					
		[...		...	<i>compression parameters</i>				

COMCRV *curve* is the name of the compression function which models the crushing behaviour of concrete [Fig. 37.2].

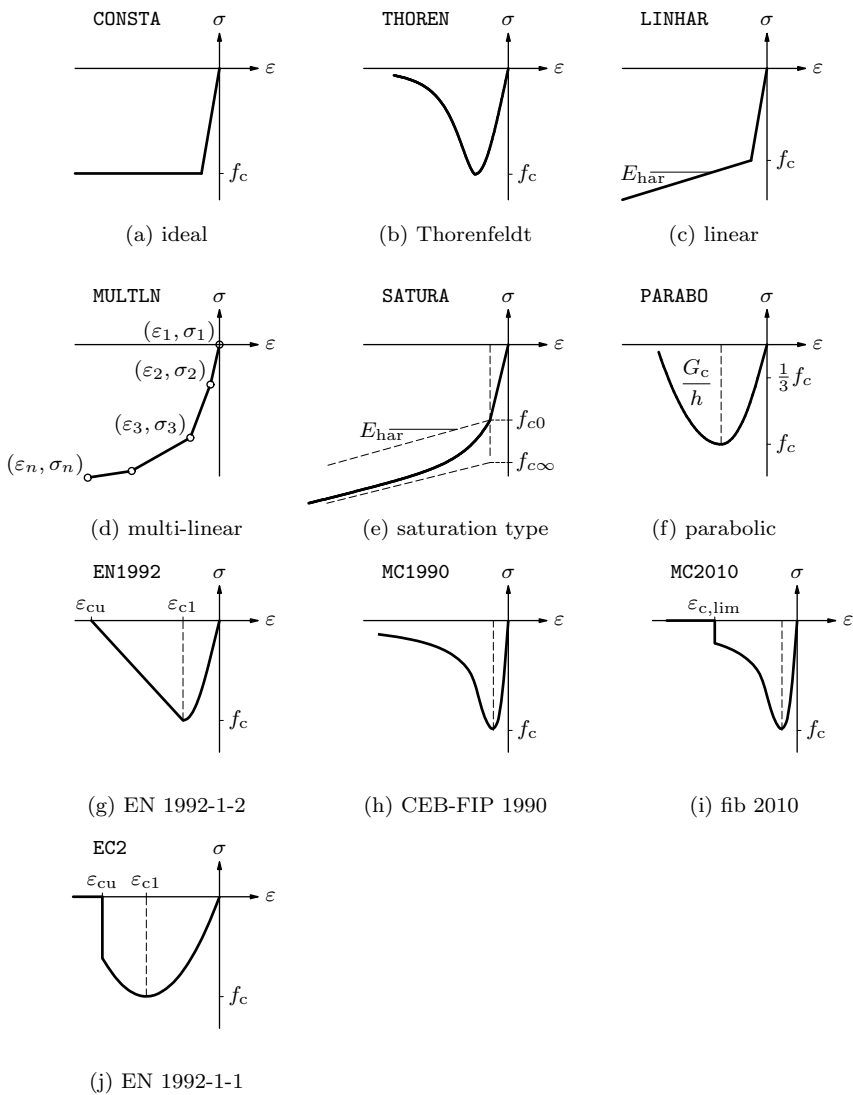


Figure 37.2: Predefined compression behaviour for stiffness adaptation analysis

Compression parameters. You must specify the compression parameters, depending on the compression function, as outlined in the following.

Ideal syntax

'MATERI'							
1	5	6	12	13			
		COMCRV		CONSTA			80
		COMSTR		f_c			

CONSTA for a constant diagram [Fig. 37.2a].

COMSTR f_c is the compressive strength f_c .

Thorenfeldt syntax

'MATERI'							
1	5	6	12	13			
		COMCRV		THOREN			80
		COMSTR		f_c			
		NTHORE		n			
		KTHORE		k			

THOREN for the function of Thorenfeldt at al. [Fig. 37.2b]. The parameters of the Thorenfeldt curve, see Volume *Material Library*, are unit-dependent. To calculate these parameters, DIANA assumes by default that the input data is in SI units.

If you describe the finite element model in units other than SI, then you must explicitly specify the units that you used in input table 'UNITS' [§ 1.1 p. 3].

COMSTR f_c is the compressive strength f_c .

NTHORE n is the parameter n of the Thorenfeldt curve. This n overrules the default computation of n as described in Volume *Material Library*.

KTHORE k is the parameter k of the Thorenfeldt curve. This k overrules the default computation of k as described in Volume *Material Library* for $\alpha \leq \alpha_p$.

Linear hardening syntax

'MATERI'							
1	5	6	12	13			
		COMCRV		LINHAR			80
		COMSTR		f_c			
		EHAR		e_{har}			

LINHAR for a linear hardening diagram [Fig. 37.2c].

COMSTR f_c is the compressive strength f_c .

EHAR e_{har} is the hardening modulus E_{har} .

Multi-linear*syntax*

'MATERI '									
1	5	6	12	13					
		COMCRV		MULTLN					
		EPSIGC		$e1_r \ s1_r \ [\ e2_r \ s2_r \ \dots] \ en_r \ sn_r$					

A multi-linear diagram fully describes the relation between the total strain and the compressive stress, therefore input of the compressive strength f_c is not necessary.

MULTLN for a multi-linear diagram [Fig. 37.2d].

EPSIGC are the points of the multi-linear diagram: n pairs of values (ε, σ) ; $e1$ to en are the total strains ε . $s1$ to sn are the corresponding compression stresses σ , Note that you should enter stresses rather than strengths. Following the standard sign convention of DIANA, compressive stresses and compressive strains should be input as negative. ($n \leq 30$)

file.dat

'MATERI '									
1	YOUNG	3.0E+10							
	TOTCRK	ROTATE							
	TENCRV	HORDYK							
	TENSTR	3.0E+06							
	GF1	600.0							
	COMCRV	MULTLN							
	EPSIGC	0.0E+00	0.0E+00						
		-1.0E-03	-30.0E+06						
		-1.0E+00	-60.0E+06						

Saturation type*syntax*

'MATERI '									
1	5	6	12	13					
		COMCRV		SATURA					
		COMSTR		$fc0_r$					
		COMSTO		$fcinf_r$					
		EHAR		$ehar_r$					
		GAMMA		gam_r					

SATURA for saturation type hardening [Fig. 37.2e].

COMSTR $fc0$ is the initial compressive strength f_{c0} .

COMSTO $fcinf$ is the ultimate compressive strength $f_{c\infty}$ at infinite strain.

EHAR which defines the constant hardening modulus E_{har} .

GAMMA gam is the decay factor γ .

Parabolic*syntax*

'MATERI '									
1	5	6	12	13					
		COMCRV		PARABO					
		COMSTR		fc_r					
		GC		gc_r					
		[CRACKB		h_r]					

PARABO for a parabolic diagram [Fig. 37.2f]. The parabolic curve is based on fracture energy by the definition of the crack bandwidth of the element, for which DIANA assumes a value h related to the square root of the area of the element. In special cases, it may be useful to specify the crack bandwidth explicitly via the CRACKB input data item, see Volume *Material Library*.

COMSTR fc is the compressive strength f_c .

GC gc is the compressive fracture energy G_c .

Eurocode 2 1992-1-2

syntax

'MATERI'

1	5	6	12	13	80
		COMCRV		EN1992	
		COMSTR		fc_r	
		EPSC1		$ec1_r$	
		EPSCU		ecu_r	

EN1992 for a compressive curve for fire load according to Figure 3.1 of Eurocode 2 EN 1992-1-2 [67] [Fig. 37.2g].

COMSTR fc is the compressive strength f_c .

EPSC1 $ec1$ is the total strain ε_{c1} when the maximum compressive strength f_c occurs.

EPSCU ecu is the ultimate compressive strain ε_{cu} .

Note that DIANA will adapt the user specified Young's modulus when it is not consistent with the tangent of the curve in the origin. Consistency requires that $E = 1.5f_c/\varepsilon_{c1}$.

CEB-FIP Model Code 1990

syntax

'MATERI'

1	5	6	12	13	80
		COMCRV		MC1990	
		COMSTR		fc_r	

MC1990 for a compression curve according to Paragraph 2.1.4.4.1 of the European CEB-FIP Model Code 1990 [16] [Fig. 37.2h].

COMSTR fc is the compressive strength f_c under uniaxial stress situations.

fib Model Code for Concrete Structures 2010

syntax

'MATERI'

1	5	6	12	13	80
		COMCRV		MC2010	
		COMSTR		fc_r	

MC2010 for a compression curve according to Paragraph 5.1.8.1 of the fib Model Code for Concrete Structures 2010 [31] [Fig. 37.2i].

COMSTR fc is the compressive strength f_c under uniaxial stress situations.

Eurocode 2 EN 1992-1-1

syntax

'MATERI '

1	5	6	12	13	80
		COMCRV		EC2	
		COMSTR		f_{c_r}	
		EPSC1		$ec1_r$	
		EPSCU		ecu_r	
		YOUNCM		ecm_r	

EC2 for a compressive curve for nonlinear constructive computations according to Formula 3.14 of Eurocode 2 EN 1992-1-1 [66] [Fig. 37.2j]. See Volume *Material Library* for background theory.

COMSTR f_c is the compressive strength f_c under uniaxial stress situations.

EPSC1 $ec1$ is the total strain ε_{c1} when the maximum compressive strength f_c occurs.

EPSCU ecu is the ultimate compressive strain ε_{cu} .

YOUNCM ecm is the Young's modulus approximated at $0.4f_c$.

37.3 No-tension Function for Interface Elements

For no-tension behaviour of interface elements, a function for reduction of the normal stiffness when the relative displacement in normal direction is larger than a critical value may be defined. Functions for reduction of the shear stiffnesses when the relative displacement in normal direction is larger than a critical value can also be defined.

syntax

'MATERI '

1	5	6	12	13	80
		[NOTENS		$[un_r [rfac_r]]]$	
		[NOSHTE		$[un_r [rfac_r]]]$	

NOTENS specifies no-tension behaviour in normal direction of the interface, where un [un=0] defines the limit value of the relative displacement in normal direction. Below the limit value the normal stiffness (DSNX, DSNY and/or DSNZ) will be applied and above the limit value the normal stiffness will be reduced by the factor $rfac$. [rfac=0]

NOSHTE specifies no-tension behaviour in shear directions of the interface, where un [un=0] defines the limit value of the relative displacement in normal direction. Below the limit value the shear stiffness (DSSX, DSSY and/or DSSZ) will be applied and above the limit value the shear stiffness will be reduced by the factor $rfac$. [rfac=0]

file.dat

'MATERI '

1	DSNZ	1000.		
	DSSX	500.		
	DSSY	300.		
	NOTENS	0.	0.000001	
	NOSHTE	0.01	0.0001	

This example is stiff in compression in normal and both shear directions. When the relative displacement in normal direction is larger than 0 the stiffness in normal direction will be reduced by a factor 0.000001. When the relative displacement in normal direction is larger than 0.01 the stiffness in the shear directions will be reduced by a factor 0.0001.

Chapter 38

Stiffness Adaptation Analysis

After the input file has been read, you can perform a stiffness adaptation analysis by defining the following commands:

syntax

```
*STADAP
[ MODEL ... ]
[ SOLVE ... ]
[ EXECUT ... ] ...
[ OUTPUT ... ] ...
*END
```

MODEL evaluates the finite element model [§ 38.1].

SOLVE customizes the settings for the solution method as described for regular linear static analysis in Chapter 30.

EXECUT executes steps [§ 38.2 p. 482].

OUTPUT selects analysis results for output [§ 38.3 p. 482].

38.1 Model Evaluation

The MODEL commands customize the evaluation of the finite element model prior to the actual analysis.

syntax

```
BEGIN MODEL
[ OFF ]
[ EVALUA [ OFF ] ... ]
[ ASSEMB [ OFF ] ... ]
[ MATRIX [ OFF ] ]
[ LOADS [ OFF ] ]
END MODEL
```

EVALUA to check and evaluate geometric and material properties for elements and reinforcements as described in detail in [§ 3.4 p. 51].

ASSEMB to assemble the elements, i.e., creating appropriate system degrees of freedom as described in detail in [§ 3.5 p. 54].

MATRIX to setup the element stiffness matrices.

LOADS to setup the load vectors.

38.2 Step Execution

With the EXECUT commands you ask DIANA to execute load steps. A command file may contain one or more EXECUT blocks to reconstruct the loading history and the execution of the stiffness adaptation analysis for every loadcase.

syntax

```
BEGIN EXECUT
LOAD=losetn
STEPS EXPLIC SIZES sizesr...
ITERAT MAXITE=min
END EXECUT
```

LOAD *loset* indicates the load set number to be incremented to the external applied load.

STEPS EXPLIC SIZES *sizes* are explicitly specified load increment sizes.

[MAXITE=10] ITERAT MAXITE *mi* is the maximum number of iterations for each load increment step.

38.3 Output of Analysis Results

You can obtain output of stiffness adaptation analysis results via an obligatory OUTPUT command block which selects the analysis results to be output. For the general, analysis type independent options for output of analysis results see [§ 3.6.1 p. 58].

syntax

```
BEGIN OUTPUT [ devicew ] [ outoptw... ] [ params ]
[ OFF ]
[ SELECT ... ]
[ LAYOUT ... ]
itemw ...
DISPLA
STRAIN
STRESS
FORCE
NODFOR
ELMFOR
PARAM
END OUTPUT
```

SELECT optional commands to customize the batch output: model selection [§ 3.6.2 p. 59], selection of load sets [§ 4.2.1 p. 75] and stress- and strain transformation [§ 3.6.2.3 p. 60].

LAYOUT optional commands to customize the layout and style of tabular output [§ 3.6.4.1 p. 65].

item is the name of the analysis result to be output. See § 3.6.1 on page 56 for complete syntax of this command.

DISPLA for displacements as described for Regular Linear Static Analysis [§ 4.2.2 p. 76].

STRAIN for strains as described for Regular Linear Static Analysis [§ 4.2.3 p. 78]. In addition crack width results can be output [§ 38.3.1 p. 483].

STRESS for stresses as described for Regular Linear Static Analysis [§ 4.2.4 p. 81].

FORCE for nodal forces as described for Regular Linear Static Analysis [§ 4.2.6 p. 88].

NODFOR for element nodal forces as described for Regular Linear Static Analysis [§ 4.2.7 p. 89].

ELMFOR for internal element forces as described for Regular Linear Static Analysis [§ 4.2.8 p. 91].

PARAME for stiffness and model parameters of the stiffness adaptation analysis [§ 38.3.2 p. 483].

38.3.1 Crack Width

In the stiffness adaptation analysis of DIANA crack width can be output with the following commands.

syntax

```
STRAIN CRKWDT GREEN [ operw ] { compw } { locaw } { optiw }
                        LOCAL          INTPNT
                        GLOBAL          NODES
                        PRINCI          CENTER
                        VONMIS
```

CRKWDT for crack width, which is defined as the product of the crack strain and the crack bandwidth h_{cr} . In stiffness adaptation analysis the crack strain is defined as the total strain - stress / E_0 , with E_0 being the original Young's modulus.

oper specifies an operation to be performed on the primary strains [§ 3.6.1 p. 57] . [GLOBAL]

comp selects crack width components for output. Default is all available components.

loca specifies the location for the crack width to be output [§ 3.6.1 p. 58]. [NODES]

opti are additional options [§ 3.6.1 p. 58].

Crack width				<i>comp</i> ...								
<i>item</i>	<i>type</i>	<i>form</i>	<i>oper</i>	XX	YY	ZZ	XY	YZ	ZX	1	2	3
STRAIN	CRKWDT	GREEN	LOCAL	Ecwx	Ecwy	Ecwz	Gcwx	Gcwy	Gcwz			
				ε_{xx}^{cw}	ε_{yy}^{cw}	ε_{zz}^{cw}	γ_{xy}^{cw}	γ_{yz}^{cw}	γ_{zx}^{cw}			
STRAIN	CRKWDT	GREEN	GLOBAL	EcwXX	EcwYY	EcwZZ	GcwXY	GcwYZ	GcwZX			
				ε_{XX}^{cw}	ε_{YY}^{cw}	ε_{ZZ}^{cw}	γ_{XY}^{cw}	γ_{YZ}^{cw}	γ_{ZX}^{cw}			
STRAIN	CRKWDT	GREEN	PRINCI							Ecw1	Ecw2	Ecw3
										ε_1^{cw}	ε_2^{cw}	ε_3^{cw}
STRAIN	CRKWDT	GREEN	VONMIS	Ecweq								
				ε_{eq}^{cw}								

38.3.2 Stiffness and Model Parameters

In a stiffness adaptation analysis some stiffness and model parameters can be output. The stiffness parameters are the reduction factor in the tensile and compressive directions in integration points of elements can be output. If the reduction factor is equal to 1.0, the stiffness is equal to the original Young's modulus. The lowest stiffness reduction factor is 0.001. For some tension softening curves DIANA will calculate the area under the stress-strain curve which is equal to the crack energy G_f divided by the crack bandwidth h_{cr} . Sometimes, these parameters may be useful in postprocessing. Via the OUTPUT block you can ask DIANA to output these stiffness and model parameters.

syntax

```
PARAME [ typew ] INTPNT { compw } { optiw }
      STADAP
      SEAREA
```

PARAME

STADAP specifies the stiffness reduction factor in tensile and compressive directions in integration points of elements. If the reduction factor is equal to 1.0, the stiffness is equal to the original Young's modulus. The lowest stiffness reduction factor is 0.001.

SEAREA gives the area under the stress-strain curve, which is equal to the crack energy G_f divided by the crack bandwidth h_{cr} . This output item is only available for the linear tension softening curve based on crack energy (**LINEAR**), softening according to Hordijk et al. (**HORDYK**), exponential tension softening (**EXPONE**), softening according to Japan Society of Civil Engineers (**JSCESO**), CEB-FIP Model Code 1990 tension softening (**MC1990**), fib Model Code for Concrete Structures 2010 tension softening (**MC2010**), and the model for tensile failure of fiber reinforced concrete as defined by the *fédération internationale du béton*/International Federation for Structural Concrete (fib) working groups (**FRCCON**) [§ 37.1 p. 470].

comp selects components for parameter output. Default is all available components.

opti are additional options [§ 3.6.1 p. 58].

Crack width		<i>comp</i> ...	
<i>item</i>	<i>type</i>	PMST	PMSC
PARAME	STADAP	PMST	PMSC
		tension red. factor	compr. red. factor
PARAME	SEAREA	G_f/h_{cr}	
		G_f/h_{cr}	

Part XII

Parameter Estimation

Glossary of symbols. This glossary is an addition to the general Glossary of Symbols for DIANA as presented on page [xxix](#). It contains the symbols used particularly in parameter estimation analysis.

Scalars

- k Set number.
- m Number of observables per set.
- n Number of parameters.
- N Number of sets.
- S Least squares function.

Vectors

- \mathbf{h} Finite element model for observables.
- \mathbf{v} Observation errors.
- \mathbf{x} Unknown parameters.
- \mathbf{y} Observables, usually measured displacements.

Matrices

- \mathbf{H} Observation matrix.
- \mathbf{I} Unity matrix.
- \mathbf{K} Gain matrix.
- \mathbf{P} Weighting matrix for parameters.
- \mathbf{Q} Additional weighting matrix for parameters.
- \mathbf{R} Weighting matrix for observables.

Chapter 39

Introduction to Parameter Estimation

Module PAREST¹ may be used to determine non-shape parameters by minimizing the differences between calculated and target displacements. The confrontation of target displacement field data with calculated field data leads to a quantitative determination of the unknown parameters. The parameters may comprise material properties, geometric properties (like the thickness of a plate), and load factors within combined load cases. Section 40.1 presents the input syntax of the target displacement field data. Successively, it describes the input of coordinates of materials points, the target displacements of these points and (statistical) information representing the accuracy of the targets.

Apart from the targets data PAREST also allows the use of prior information, for instance results of laboratory experiments, on the parameters in the identification process. Section 40.2 deals with syntax of the estimation input. Successively, it describes the specification of the unknown parameters and (statistical) information representing the accuracy of the initial guesses for these unknown parameters.

Module PAREST uses these data in an integrated manner and is based on a sequential minimum variance estimator. It is an iterative technique: complete finite element analyses are performed repeatedly; each analysis is expected to yield better values for the parameters.

Figure 39.1 shows the general idea. Within PAREST so-called non-shape parameters can be identified. The targets may comprise displacement components (both translations

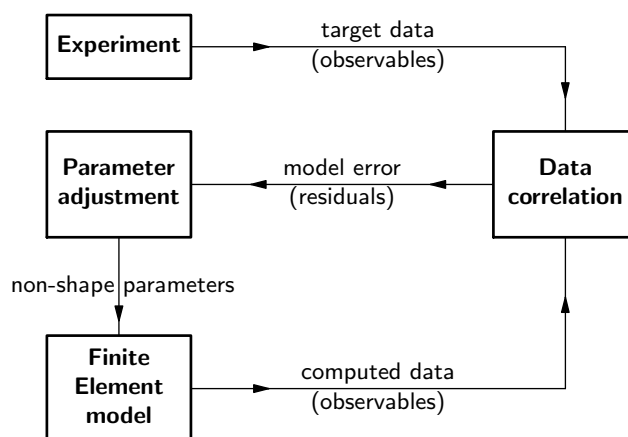


Figure 39.1: Diagram of identification approach

and rotations), forces, and pressures in true coupled analysis of porous filled media.

¹PAREST is short for PARAMETER ESTimation. In the realm of geomechanics the term *Backward Analysis* is more common.

Other modules of DIANA perform the actual finite element analyses. The features of DIANA for linear static analysis and nonlinear analysis to make and analyse a finite element model may be used to estimate the parameters. Crucial in the process is that the target data contain sufficient information about the unknown parameters!

39.1 Some Background Theory

In this section the identification theory will be quoted shortly. We will present a parameter estimation approach, or Backward Analysis, based on the combination of two elements:²

- Finite element modelling.
- A technique to adapt the non-shape parameters in the finite element model by means of a comparison between the target data and the outcomes of the finite element model.

The method is based on the *Sequential minimum variance* approach and resembles *Kalman filtering* techniques. Alternatively the method can be termed as a *Weighted least squares* approach.

The target data are assumed to consist of a set of columns with data $\mathbf{y}_k, k = 1, \dots, N$, where k indicates a load set or a discrete time parameter. Each column contains target values of material points of a loaded object. These targets are modeled with help of a finite element model. The modeled targets are considered to be a nonlinear function of a set of unknown parameters:

$$\mathbf{y}_k = \mathbf{h}_k(\mathbf{x}) + \mathbf{v}_k \quad (39.1)$$

where \mathbf{x} is a column with unknown parameters, \mathbf{h}_k is a finite element model for the targets \mathbf{y}_k and \mathbf{v}_k is a column of residuals.

First assume that only one column of targets is available. The basic estimation problem is the use of the targets \mathbf{y}_1 to estimate parameter column \mathbf{x} . The estimator can be specified from the model Eq. (39.1), an uncertainty model for \mathbf{v}_1 and a priori knowledge of \mathbf{x} . The optimal parameter column based on \mathbf{y}_1 minimizes the following quadratic expression.

$$S_1 = (\mathbf{y}_1 - \mathbf{h}_1(\mathbf{x}))^T \mathbf{R}_1^{-1} (\mathbf{y}_1 - \mathbf{h}_1(\mathbf{x})) + (\hat{\mathbf{x}}_0 - \mathbf{x})^T \mathbf{Q}_1^{-1} (\hat{\mathbf{x}}_0 - \mathbf{x}) \quad (39.2)$$

where $\hat{\mathbf{x}}_0$ is an initial guess for the parameter column \mathbf{x} . In weighted least squares estimation the matrices \mathbf{R}_1 and \mathbf{Q}_1 are chosen on the basis of engineering judgment. Matrix \mathbf{R}_1 is a nonnegative symmetric matrix. Matrix \mathbf{Q}_1 is a positive symmetric matrix. It is obvious that the introduction of \mathbf{Q}_1 makes it possible to put more weight to the a priori estimate $\hat{\mathbf{x}}_0$ (and less weight to the displacements \mathbf{y}_1). The least squares estimate does not make use of the statistics of the observation errors, in any way. In many applications it is not uncommon to know the mean and variance of the expected residuals. Minimum variance estimates utilize this extra information, which results in specific choices for \mathbf{R}_1 and \mathbf{Q}_1 . In minimum variance estimation \mathbf{R}_1 represents the covariance matrix of the residuals \mathbf{v}_1 . Matrix \mathbf{Q}_1 represents the covariance matrix of the estimation error in $\hat{\mathbf{x}}_0$. Generally: the larger \mathbf{P}_1 , the smaller the influence of $\hat{\mathbf{x}}_0$.

Solving the nonlinear inverse problem, defined by Equations Eqs. (39.1) and (39.2), leads to an iterative scheme which results in an estimation $\hat{\mathbf{x}}_1$ for \mathbf{x} :

$$\begin{aligned} \hat{\mathbf{x}}^{(i+1)} = \hat{\mathbf{x}}^{(i)} &+ \left(\mathbf{H}^{(i)T} \mathbf{R}^{-1} \mathbf{H}^{(i)} + \mathbf{Q}^{-1} \right)^{-1} \\ &\times \left(\mathbf{H}^{(i)T} \mathbf{R}^{-1} (\mathbf{y} - \mathbf{h}(\hat{\mathbf{x}}^{(i)})) + \mathbf{Q}^{-1} (\hat{\mathbf{x}}^{(0)} - \hat{\mathbf{x}}^{(i)}) \right) \end{aligned} \quad (39.3)$$

where the superscripts refer to the iteration number and where the subscripts are temporarily dropped. In each iteration, Module PAREST executes $n+1$ complete finite element calculations, where n is the number of parameters. The n calculations are carried out to

²See also Hendriks' Ph.D. thesis [40]

determine a matrix $\mathbf{H}_1^{(i)}$ numerically, as a linearization of \mathbf{h}_1 with respect to the most recent estimation $\hat{\mathbf{x}}_1^{(i-1)}$. Within Module BALANCE this is not necessary, as function \mathbf{h}_1 is a linear function of the load multiplication factors.

The sequential property of the estimator [Eq. (39.3)] is clear when a column \mathbf{y}_2 with additional target data becomes available. This can be data from another load set or from another point in time. These data can be used together with the initial conditions $\hat{\mathbf{x}}_1$, \mathbf{R}_2 and \mathbf{Q}_2 resulting in an improved estimation $\hat{\mathbf{x}}_2$.

Summary. The input for the estimator is summarized as follows:

- Unknown parameters \mathbf{x} , specified in subtable PARAME [§ 40.2.1 p. 499].
- $n \times n$ symmetric weighting matrices \mathbf{Q}_k , $k = 1, \dots, N$, specified in N subtables QMATRI [§ 40.2.3 p. 501].
- Targets \mathbf{y}_k , $k = 1, \dots, N$, specified in N subtables OBSERV [§ 40.1.2 p. 495].
- $m \times m$ symmetric weighting matrices \mathbf{R}_k , $k = 1, \dots, N$, where m indicates the number of components of \mathbf{y} , specified in N subtables RMATRI [§ 40.1.3 p. 496].

39.2 How to Use Module PAREST

To perform a parameter estimation analysis, you should sequentially take the following actions:

1. *Prepare linear elastic input.* Prepare an input file for regular linear static analysis of a finite element model as described in Chapter 4. Alternatively, you may use the interactive preprocessing environment of iDIANA as described in Volume iDIANA.
2. *Add nonlinear input (optional).* Add the data that describes the nonlinear material behaviour or temperature data or both to the input file, as described in Volume Material Library.
3. *Perform (non)linear analysis.* The analysis may include also phased analysis [Ch. VIII]. You are encouraged to check the forward (non)linear analysis carefully for correctness and efficiency. Module PAREST will repeatedly restart the above analysis with different values for the specified parameters.
4. *Add parameter estimation input data.* Add the two input tables as described in Chapter 40.
5. *Perform parameter estimation.* Use commands as described in § 41.2 on page 505. It is good practice to check the parameter estimation scheme with simulated experimental data generated with known parameters before using the actual experimental data.

Load optimization. Module BALANCE is a dedicated interface to PAREST to find optimal load combinations for linear static analysis. See Chapter 42 for description of input and commands for this module and Examples bridge in Volume Analysis Examples for an instructive example.

Chapter 40

Input for Parameter Estimation

This chapter describes the syntax of input tables appropriate for parameter estimation analysis. The general concept of input tables is described in Chapter 1.

40.1 Target Data

Target data is input via table 'TARGET' in three subtables in obligatory sequence. Successively, the subtables input coordinates of materials points, the observed data and (statistical) information representing the accuracy of the observables.

			<i>syntax</i>
'TARGET'			
[POINTS			
1	5	6	80
<i>material points</i>]			
OBSERV [k_n]			
1	5	6	80
<i>observables</i>			
[RMATRI [k_n]			
1			80
<i>weighting matrix</i>]			

'TARGET' is the table heading for target data input.

POINTS is the subtable heading for input of coordinates of material points [§ 40.1.1]. If this subtable is not specified then the observables must be defined in the nodes of the finite element model via subtable OBSERV.

OBSERV is the subtable heading for input of observables, referring to material points or nodes. Usually the observables comprise target displacements [§ 40.1.2].

RMATRI is the subtable heading for the weighting matrix \mathbf{R} [§ 40.1.3].

For some estimation schemes several sets of observables may be distinguished, for instance measured displacements of material points at different load levels or points in time. In this case several subtables OBSERV and matching RMATRI subtables will be input, where value k then indicates these sets. See Chapter 41 how to specify the estimation scheme. Value $k = 1$ for estimation schemes with a single set of observables.

Observables in material points

file.dat

```

'TARGET'
POINTS
  1  0.35  0.17
  2  0.79  0.17
101  0.70  0.34
  5  0.35  0.34
OBSERVABLES 1
  1  POINT  1  DISPX  0.08
  2  POINT  1  DISPY  0.09
  3  POINT  2  DISPY  0.01
  4  POINT  5  DISPX  0.01
  5  POINT 101  DISPX  0.02
  6  POINT 101  DISPY  0.01
RMATRIX 1
  1.E-4  0.  0.  0.  0.  0.
  0.  1.E-4  0.  0.  0.  0.
  0.  0.  1.E-4  0.  0.  0.
  0.  0.  0.  1.E-4  0.  0.
  0.  0.  0.  0.  1.E-4  0.
  0.  0.  0.  0.  0.  1.E-4

```

This example defines four material points. Point 101 has model coordinates $X = 0.70$, $Y = 0.34$ and $Z = 0.00$. The six observables all refer to these points. Observable 6 is defined as the displacement of point 101 in model Y direction. Its value is 0.01. The load at which this value is measured is defined in the command file [Ch. 41]. The weighting matrix is input as a full 6×6 matrix with off-diagonal terms equal to zero and identical diagonal terms. This means that the six observables have been measured with the same accuracy and that there is no correlation between the measurement errors.

Observables in the nodes

file.dat

```

'TARGET'
OBSERV 1
  1  NODE  3  PDISPX -0.1044E+02
  2  NODE 13  PDISPX -0.4564E+01
  3  NODE 13  PDISPY -0.2760E+00
  4  NODE 23  PDISPX -0.5031E-01
  5  NODE 23  PDISPY -0.4955E+00
  6  NODE 22  PDISPX -0.4865E-01
  7  NODE 22  PDISPY -0.8406E+01
  8  NODE 21  PDISPY -0.8989E+01
RMATRIX 1
  1.0  0.01  0.01  0.0001  0.01  0.01  1.0  1.0

```

Contrary to the previous example, the observables now refer to nodal data. The 8×8 weighting matrix is now input as a diagonal matrix by specifying the diagonal terms only.

Multiple sets of observed data

file.dat

```

'TARGET'
POINTS
  1  0.5  1.0  0.0
OBSERV 1
  1  POINT 1  DISPY  0.4668D-01
OBSERV 2
  1  POINT 1  DISPY  0.8805D-01
OBSERV 3
  1  POINT 1  DISPY  0.4668D-01
RMATRIX 1

```



```

1.00E-6
RMATRIX 2
1.00E-6
RMATRIX 3
1.00E-6

```

In this example three sets of observed data are specified. Each set comprises a single observable, viz. the displacement of point 1 in model Y direction. In the command file the corresponding load levels will be defined referring to the subtable indicators $k = 1$, $k = 2$ and $k = 3$.

40.1.1 Material Points

Material points must be specified in subtable **POINTS** of table '**TARGET**'. This subtable is only necessary if the observables are defined in arbitrary material points rather than in the nodes of the model [§ 40.1.2].

syntax

```

'TARGET'
POINTS
1 5 6 80
pointn xr [yr [zr]]

```

point is the material point number. Points may be input in arbitrary order, numbers may be skipped. Values x y z are the model XYZ coordinates, any y or z value that you do not specify is assumed to be zero.

[y =0]

[z =0]

The first-off action of Module PAREST is to embed the specified points in the element mesh.

PAREST can only embed points in four- or eight-node quadrilaterals, i.e., numerically integrated elements for plane stress, plane strain, axisymmetry, plate bending and flat shells.

In other cases the observables should be specified as nodal data [§ 40.1.2].

40.1.2 Observables

Observables must be specified in subtable **OBSERV** of table '**TARGET**'.

syntax

```

'TARGET'
OBSERV [kn]
1 5 6 80
obsn locaw locnumn typew datumr
      NODE          DISPX
      POINT         DISPY
                      DISPZ
                      RESIDX
                      RESIDY
                      RESIDZ
                      ROTATX
                      ROTATY
                      ROTATZ
                      PDISPX
                      PDISPY
                      PDISPZ
                      PRESS

```

k is the set number. If you omit it, DIANA assumes set number one by default. [$k = 1$]

obs is the observable number. Observables may be input in arbitrary order, but no number may be skipped(!).

loca specifies the type of location of the observable: **NODE** for nodal data, **POINT** for data in an embedded point [§ 40.1.1].

locnum is the node or point number.

type specifies the type of the data. **DISPX**, **DISPY** or **DISPZ** for displacement in model X , Y or Z direction respectively. **RESIDX**, **RESIDY** or **RESIDZ** for residual forces in model X , Y or Z direction respectively. **ROTATX**, **ROTATY** or **ROTATZ** for rotation in model X , Y or Z direction respectively. **PDISPX**, **PDISPY** or **PDISPZ** the same displacements but for the current phase in a phased analysis.. **PRESS** for pressure in true coupled analysis of porous filled media, see Chapter 25.

datum is the actual value of the observed data, either a displacement component or a pressure.

40.1.3 Weighting the Observables

The weighting matrix [Eq. (39.2) p. 490] must be specified in subtable **RMATRI** of table 'TARGET'.

syntax

'TARGET'	
RMATRI [k_n]	
1	80
<i>wfac_{r...}</i>	

[$k = 1$] k is the set number. If you omit it, DIANA assumes set number one by default.

wfac are one or more weighting factors specifying the terms of the weighting matrix. This matrix may be input in one of three forms: as *full matrix* of $m \times m$ terms where m is the number of observables specified in subtable **OBSERV** [§ 40.1.2], as *diagonal matrix* by specifying m terms, as diagonal matrix with *identical diagonal elements* by specifying one single term.

[*wfac* = 1] If you do not specify an **R** matrix then DIANA assumes an identity matrix.

The input matrices must be symmetric and positive definite.

40.2 Parameter Estimation

Table 'ESTIMA' contains the input data related to the parameters to be estimated. This table consists of a number of subtables to be specified in the sequence as indicated below.

syntax

'ESTIMA'	
PARAME	
1 5 6 12 13	80
<i>unknown parameters</i>	
[SUPERP]	
1 5 6 12 13	80
<i>superparameters</i>	
[QMATRI [k_n]]	
1	80
<i>weighting matrix</i>	

PARAME is the subtable heading for definition of unknown parameters [§ 40.2.1].

SUPERP is the subtable heading for definition of superparameters [§ 40.2.2].

QMATRI is the subtable heading for input of the weighting matrix **Q** [§ 40.2.3].

For some estimation schemes several sets of observables may be distinguished, for instance measured displacements of material points at different load levels or points in time. In this case several subtables QMATRI will be input, where value *k* then indicates these sets. Default is set number one.

[*k* = 1]

See Chapter 41 how to specify the estimation scheme. For estimation schemes with a single set of observables, *k* is not necessary.

Estimating parameters referring to 'MATERI' and 'GEOMET'

file.dat

```

'MATERIALS'
  1  YOUNG  20.0
    POISON 0.30
  2  YOUNG  15.0  30.0
    POISON 0.30  10.0
'GEOMETRY'
  1  THICK  0.1
    XAXIS  1.0 1.0 0.0
'ESTIMA'
PARAMETERS
  1  NAME      POISON
    MATERI    1
    BOUNDS    0.10 0.49
  2  NAME      YOUNG
    MATERI    1
    BOUNDS    0.10 1E10
  3  NAME      YOUNG
    MATERI    2
    ITEM       1
    BOUNDS    0.10 1E10
  4  NAME      XAXIS
    GEOMET     1
    ITEM       2
    BOUNDS    0.5 1.5
QMATRIX 1
  1E-6(4)

```

Subtable PARAME specifies the unknown parameters. Four parameters are indicated which are to be identified by Module PAREST. The third parameter is defined by pointing to the first data item of the Young's modulus of material number 2. During the estimation the possible values for this parameter are limited by the values 0.10 and 10^{10} . The initial guesses for the parameters are the corresponding values of the tables 'MATERI' and 'GEOMET', which is 15.0 for parameter 3. Subtable QMATRI contains the parameter weighting matrix.

Estimating parameters referring to table 'LOADS'

file.dat

```

'ESTIMA'
PARAMETERS
  1  NAME      FACTOR
    LOAD       1
    ITEM       1
  2  NAME      FACTOR
    LOAD       1
    ITEM       3
'LOADS'
CASE  1

```

```

ELEMEN
  1  PRESTR  1.0  0.   0.  0.
CASE   2
ELEMEN
  2  PRESTR  1.0  0.   0.  0.
CASE   3
ELEMEN
  3  PRESTR  1.0  0.   0.  0.
COMBIN
  1  1 1.  2 1.  3 1.
'END'

```

In this example subtable **PARAME** specifies two unknown parameters. They both point to the load factors of subtable **COMBIN**. The first unknown parameter is the first load factor. The second unknown parameter is the third load factor. For both parameters the initial guess is 1.

Using superparameter SINPHI

file.dat

```

'MATERI'
  1  DENSIT 1.0
      YIELD MOHRCO
      COHESI 1.0
      PHI   0.315193
      PSI   0.315193
      KO    0.5
      YOUNG 1.0
      POISON 0.2
  2  DENSIT 1.0
      YOUNG 1000.0
      POISON 0.3
'ESTIMA'
PARAME
  1  NAME  SINPHI
      SUPERP 1
      BOUNDS 0.01 100.0
SUPERP
  1  SINPHI 0.31
QMATRIX 1
      1.OE-2

```

In this example only one unknown parameter will be determined. It is a so-called superparameter named **SINPHI**. Parameter **SINPHI** is defined in an additional subtable **SUPERP**. **SINPHI** is internally connected to the the friction angle ϕ (**PHI**) and dilatancy angle ψ (**PSI**) of material number 1. The initial guess for **SINPHI** is also specified in subtable **SUPERP**.

Multiple sets of observed data

file.dat

```

'MATERIALS'
  1  YOUNG  1.1
      POISON 0.3
'ESTIMA'
PARAME
  1  NAME  YOUNG
      MATERI 1
      BOUNDS 0.1 1000.0
QMATRIX 1
      1.OE-2
QMATRIX 2
      1.OE-2
QMATRIX 3
      1.OE-2

```

Also in this example only one unknown parameter will be determined. Now it is assumed that in table 'TARGET' three sets of observed data are specified, like in the last example of § 40.1 on page 495.

40.2.1 Parameter Specification

Subtable PARAME specifies which of the parameters will be estimated. The parameters in this subtable refer to the properties of tables 'MATERI' and 'GEOMET' or to subtable SUPERP.

syntax

'ESTIMA'

PARAME

1	5	6	12	13	80
$parnr_n$		NAME		$parnam_w$	
		MATERI		$matnr_n$ [$frnr_n$]	
		GEOMET		$geonr_n$	
		LOAD		$loset_n$	
		SUPERP		$supnr_n$	
		[ITEM		$index_n$]	
		[BOUNDS		$bndlow_n$ $bndup_n$]	
		[USRBOU			

$parnr$ is the number of the parameter. Parameters may be input in arbitrary order but no numbers may be skipped(!).

NAME $parnam$ is the name of the material or geometry property. For a load set referring to subtable COMBIN of table 'LOADS', this name is FACTOR.

MATERI $matnr$ refers to a material number of table 'MATERI'. A fraction number $frnr$ may be specified for the fraction model or for the Maxwell Chain model in viscoelasticity [Vol. *Material Library*].

GEOMET $geonr$ refers to a geometry number of table 'GEOMET'.

LOAD $loset$ refers to a load set number in subtable COMBIN of table 'LOADS'. You must specify this load set in subtable COMBIN.

SUPERP $supnr$ refers to a superparameter in subtable SUPERP [§ 40.2.2].

ITEM $index$ is the index in the property data record. If this data record contains more than one value, for instance YOUNG 1.2E6 2.5E6, the $index$ indicates which one of the values must be estimated. If you do not specify an index then DIANA takes the first value by default. [$index = 1$]

For load optimization, the index indicates the load factor to be estimated. For instance index 2 points to the second load factor in subtable COMBIN.

BOUNDS $bndlow$ and $bndup$ respectively are the lower and upper bounds of the parameter. ($bndlow < bndup$)
If, during iteration, the parameter gets a value outside the interval then DIANA replaces it by the value of the exceeded boundary. By default no bounds are active.

USRBOU indicates that DIANA must determine the boundary values via a user-supplied subroutine USRBOU [§ 40.3.2].

40.2.2 Superparameters

Usually parameters of subtable `PARAME` refer directly to data entries in the tables `'MATERI'`, `'GEOMET'` or `'LOADS'`. For some applications there is a need for more freedom in the definition of the unknown parameters. The optional subtable `SUPERP` offers the possibility to define *superparameters* additional to the parameters of tables `'MATERI'` and `'GEOMET'`.

syntax

<code>'ESTIMA'</code>		
<code>SUPERP</code>		
1	5 6	12 13
<hr/>		
<code>matnr_n</code>	<code>SINPHI</code>	<code>iguess_r</code>
0	<code>CIRCUM</code>	<code>cx_r cy_r</code>
0	<code>BILIN</code>	<code>b0_r bx_r by_r</code>
<code>supnr_n</code>	<code>USRPAR</code>	<code>iguess_{r...}</code>

You may specify one of the following superparameters.

`SINPHI` specifies a superparameter for associated Mohr–Coulomb and Drucker–Prager plasticity [Vol. *Material Library*]. This superparameter is internally connected to the `PHI` and `PSI` data entries of material `matnr` from table `'MATERI'`. It considers associative plasticity by regarding the friction angle ϕ (`PHI`) and the dilatancy angle ψ (`PSI`) as one single parameter. Value `iguess` denotes the initial guess for $\sin \phi$ and $\sin \psi$.

`CIRCUM` specifies a set of two superparameters for circumferential orthotropy. The superparameters are internally connected to all(!) `XAXIS` data entries of table `'GEOMET'`. Parameter `CIRCUM` is useful for curvilinear orthotropic materials, i.e., materials in which the orientation of the orthotropic symmetry differs from point to point. The local orthotropic orientations are tangent to concentric circles, where (cx, cy) denotes the initial guess for the model XY coordinates of the center of the circles. It is recommended to use one geometry number for each element, each one with its own `XAXIS` [Vol. *Element Library*].

`BILIN` specifies a set of three superparameters for bilinear orthotropy. The superparameters are internally connected to all(!) `XAXIS` data entries of table `'GEOMET'`. Parameter `BILIN` is useful for curvilinear orthotropic materials, i.e., materials in which the orientation of the orthotropic symmetry differs from point to point. The local orthotropic orientations are a bilinear function of the model XY coordinates: $\alpha = b_0 + b_x X + b_y Y$, where α denotes the positive rotation of the orthotropic orientation. Values `b0`, `bx` and `by` are the initial guesses for b_0 , b_x and b_y respectively. It is recommended to use one geometry number for each element, each one with its own `XAXIS` [Vol. *Element Library*].

`USRPAR` specifies a user-defined superparameter via a user-supplied subroutine [§ 40.3.1]. DIANA passes the number `supnr` to the argument list of user-supplied subroutine `USRPAR`. There you may use it to refer to a material number. Values `iguess` denote the initial guesses.

Circumferential orthotropy

file.dat

```

'ESTIMA'
PARAME
  1  SUPERP  0
      NAME   CIRCUM
      BOUNDS 0.00  1.00
  2  SUPERP  0
      NAME   CIRCUM
      ITEM   2
      BOUNDS 0.00  1.00

```

SUPERP

0 CIRCUM 0.5 0.5

In this example circumferential orthotropy is used. Both parameters cx and cy are regarded as unknown parameters with the same initial guess 0.5.

40.2.3 Weighting the Parameters

Whereas the observables are weighted with matrix \mathbf{R} [§ 40.1.3], the parameters are weighted with matrix \mathbf{Q} of Eq. (39.2) on page 490. Subtable QMATRI contains the weighting matrix. You must define the parameters themselves in subtable PARAME and their start values in table 'MATERI', 'GEOMET' or 'LOADS, or in subtable SUPERP [§ 40.2.2].

syntax

'ESTIMA'

QMATRI [k_n]

1

80

$wfac_{r\dots}$

QMATRI is the subtable heading for the additional weighting matrices \mathbf{Q} . For some estimation schemes several sets of observables may be distinguished, for instance measured displacements of material points at different load levels or points in time. In this case several subtables QMATRI will be input, by analogy with subtables OBSERV and RMATRI of table 'TARGET' [§ 40.1], where value k indicates these sets. See Chapter 41 how to specify the estimation scheme.

If you do not specify the \mathbf{Q} matrix, then DIANA assumes a default such that \mathbf{Q} inverse is a zero matrix. This means that the initial guesses for the parameters vanish in the least squares expression. Thus, the prior information about the parameter values which they may have contained is not used. [$wfac = \infty$]

$wfac$ are one or more weighting factors specifying the terms of the weighting matrix. This matrix may be input in one of three forms: as *full matrix* of $n \times n$ terms where n is the number of parameters specified in subtable PARAME [§ 40.2.1], as *diagonal matrix* by specifying n terms, as diagonal matrix with *identical diagonal elements* by specifying one single term.

The input matrices must be symmetric and positive definite.

40.3 User-supplied Subroutines

This section contains a description of the predefined user-supplied subroutines for parameter estimation analysis with Module PAREST. See also Chapter A for a general description of DIANA's user-supplied subroutine option.

40.3.1 Superparameter USRPAR

If you put USRPAR in subtable SUPERP of table 'ESTIMA' [§ 40.2.2], then you must define the superparameter via subroutine USRPAR. This routine is useful if the standard parameters of the tables 'MATERI' and 'GEOMET' are impracticable and if the predefined superparameters of subtable SUPERP do not apply. Routine USRPAR must specify how the standard input parameters of the tables 'MATERI' and 'GEOMET' depend on the user-supplied parameter.

Fortran

SUBROUTINE USRPAR(parrcd, npar, parnr, supnr)

in	DBL	parrcd (npar)	User parameters.
in	INT	npar	Number of user parameters.
in	INT	parnr	Parameter number, subtable PARAME .
in	INT	supnr	Parameter number, subtable SUPERP .

Example. A simple example of the user-supplied superparameter **USRP** is the identification of the bulk and shear modulus rather than **DIANA**'s standard isotropic parameters, the Young's modulus and the Poisson's ratio. In this example the bulk and shear modulus are implemented as user-supplied parameter number 1. The dependent Young's modulus and Poisson's ratio are calculated and stored on the **FILOS** file via routine **PTXL**.

file.f

```

SUBROUTINE USRP( PARRCD, NPAR, PARNR, SUPNR )
DOUBLE PRECISION PARRCD(*)
INTEGER          NPAR, PARNR, SUPNR
C
DOUBLE PRECISION K, G, E, NU
C
IF ( SUPNR .EQ. 1 ) THEN
  IF ( NPAR .EQ. 2 ) THEN
    K = PARRCD(1)
    G = PARRCD(2)
    E = ( 9.DO*K*G ) / ( 3.DO*K + G )
    NU = ( 1.5DO*K - G ) / ( 3.DO*K + G )
    CALL PTXL( '/MATERI/YOUNG', 5, E, 1 )
    CALL PTXL( '/MATERI/POISON', 5, NU, 1 )
  ELSE
    PRINT *, 'WRONG LENGTH: ', NPAR
    CALL PRGERR( 'USRP', 1 )
  END IF
ELSE
  PRINT *, 'WRONG SUPNR: ', SUPNR
  CALL PRGERR( 'USRP', 2 )
END IF
END

```

For description of the error handling routine **PRGERR** see Chapter [A](#). The following could be the data file for this example.

file.dat

```

'MATERIALS'
  5  YOUNG  2.0
    POISON 0.3
'ESTIMA'
PARAME
  1 NAME  USRP
    SUPERP 1
    BOUNDS 0.1 1000.0
  2 NAME  USRP
    ITEM  2
    SUPERP 1
    BOUNDS 0.1 1000.0
SUPERP
  1 USRP 2.0 1.0

```

The start values for the bulk modulus K and the shear modulus G are 2.0 and 1.0 respectively.

40.3.2 Bounds USRBOU

If you put USRBOU in subtable PARAME of table 'ESTIMA' [§ 40.2.1], then you must define the bounds on the set of parameters via subroutine USRBOU. This routine is useful for cases where a simple upper and lower bound for each individual parameter does not suffice.

Fortran

```

      SUBROUTINE USRBOU( npar, xcol )

in   INT  npar           Number of parameters to be established.
in   DBL  xcol(npar)     Set of parameter values.
out  DBL  xcol(npar)     Set of parameter values.

```

file.f

```

      SUBROUTINE USRBOU( NPAR, XCOL )
      DOUBLE PRECISION XCOL(*)
      INTEGER          NPAR
C
      IF ( NPAR .GE. 2 ) THEN
        IF ( XCOL(2) .GT. XCOL(1) ) XCOL(2) = XCOL(1)
      ELSE
        PRINT *, 'SOMETHING WRONG WITH NPAR: ', NPAR
        CALL PRGERR( 'USRBOU', 1 )
      END IF
      END

```

In this simple example the maximum tolerated value for the second parameter depends on the value of the first value.

Chapter 41

Parameter Estimation Analysis

This chapter describes how to perform a parameter estimation analysis. Such analysis may be performed in one single run which includes the structural analysis itself. However,

It is good practice first to test the normal analysis commands for correctness and then to add the parameter estimation commands to the command file.

We will subsequently present how to perform a preliminary analysis [§ 41.1], the formal command syntax for the actual parameter estimation analysis [§ 41.2], the user-supplied subroutines [§ 40.3], and the analysis results [§ 41.3].

41.1 Preliminary Analysis

Linear static. For syntax description of preliminary linear static analysis commands see Chapter 4. Below is a brief example of such commands.

file.dcf

```
*FILOS
INITIA
*INPUT
*LINSTA
...
*END
```

Module FILOS is used to maintain the FILOS file, i.e., the central database for each analysis with DIANA. The INITIA command initializes the FILOS file. Module INPUT reads the complete input data file. This file must contain data for a complete linear and/or nonlinear analysis. The two special input tables for parameter estimation, 'TARGET' and 'ESTIMA', must be present as well [Ch. 40].

Module LINSTA performs a linear static analysis of the finite element model. For postprocessing of analysis results see also Volume *iDIANA*.

41.2 Analysis Commands

After the input file has been read and the finite element model analysed, you can perform the actual parameter estimation run with the usual commands for linear and nonlinear analysis (optionally phased) followed by commands for the parameter estimation.

Linear static

file.dcf

```
*LINSTA
...          analysis commands
*PAREST
...
*END
```

Nonlinear*file.dcf*

```
*NONLIN
...      analysis commands
*PAREST
...
*END
```

Due to the *PAREST commands, Module PAREST will repeatedly execute the previous *analysis commands*.

syntax

```
*PAREST
[ OUTPUT TABULA PAREST ]
[ BEGIN EXECUT
  [ OBSERV= $k_n$  ]
  [ analysisw ... ]
  LINSTA
  NONLIN
  [ ITERAT ... ]
  END EXECUT ] ...
```

OUTPUT asks for tabular output of the estimation report.

EXECUT specifies the parameter estimation step. You may give multiple EXECUT blocks.

OBSERV k is the set number which corresponds with the number of the subtables OBSERV, RMATRI, and QMATRI [Ch. 40].

analysis specifies the type of analysis results to be used: LINSTA results of linear static analysis [§ 41.2.1], or NONLIN results of nonlinear analysis [§ 41.2.2].

ITERAT specifies the iteration process in the parameter estimation analysis [§ 41.2.3].

Linear analysis*file.dcf*

```
*LINSTA
...
*PAREST
BEGIN EXECUT
  OBSERV=1
  BEGIN LINSTA
    INDEX=1
    LOAD=1
  END LINSTA
  BEGIN ITERAT
    MAXITE=10
    METHOD MODIFI
    BEGIN CONVER
      RESIDU TOLCON=1.E-5
      PARAME TOLCON=1.E-6
    END CONVER
  END ITERAT
END EXECUT
*END
```

Nonlinear analysis*file.dcf*

```

*NONLIN
...
*PAREST
BEGIN EXECUT
  OBSERV=1
  BEGIN NONLIN
    INDEX=1
    STEP=1
  END NONLIN
  BEGIN ITERAT
    MAXITE=10
    BEGIN CONVER
      PARAME TOLCON=1.E-6
    END CONVER
  END ITERAT
END EXECUT
*END

```

Nonlinear phased analysis*file.dcf*

```

*PHASE
ACTIVE ELEMEN 1
*NONLIN
...
*PHASE
ACTIVE ELEMEN 1
*NONLIN
...
*PAREST
BEGIN EXECUT
  OBSERV=1
  BEGIN NONLIN
    INDEX=2
    STEP=5
  END NONLIN
  BEGIN ITERAT
    MAXITE=10
    BEGIN CONVER
      RESIDU TOLCON=1.E-5
    END CONVER
  END ITERAT
END EXECUT
*END

```

In this example each forward analysis comprises two phases. The observables refer to the results of the second phase.

True coupled analysis of porous filled media*file.dcf*

```

*NONLIN
...
*PAREST
BEGIN EXECUT
  OBSERV=1
  BEGIN NONLIN
    INDEX=1
    STEP=1
  END NONLIN
  BEGIN ITERAT
    MAXITE=10
    METHOD FULL

```

```

      BEGIN CONVER
      PARAME TOLCON=1.E-6
      END CONVER
      END ITERAT
      END EXECUT
      BEGIN EXECUT
      OBSERV=2
      BEGIN NONLIN
      INDEX=1
      STEP=2
      END NONLIN
      BEGIN ITERAT
      MAXITE=10
      METHOD MODIFI
      BEGIN CONVER
      RESIDU TOLCON=1.E-5
      END CONVER
      END ITERAT
      END EXECUT
      *END

```

file.dat

```

      'TIMELO'
      LOAD 1
      TIMES 0.0 0.1 0.2 /
      FACTOR 0.0 1.0 1.0 /

```

This example shows the specification of an estimation scheme with multiple sets of observables. Two subtables `OBSERV 1` and `OBSERV 2` should be input.

41.2.1 Linear Static Analysis Results

You may customize a parameter estimation analysis with results from a linear static analysis via `LINSTA` commands.

syntax

```

      BEGIN LINSTA
      [ INDEX=indn ]
      [ LOAD=losetn ]
      END LINSTA

```

[*ind*=1] `INDEX ind` specifies the number of analysis block: *ind*=1 for the first block, *ind*=2 for the second block, etc.

[*loset*=1] `LOAD loset` specifies the load set number in the `LINSTA` block.

41.2.2 Nonlinear Analysis Results

You may customize a parameter estimation analysis with results from a nonlinear analysis via `NONLIN` commands.

syntax

```

      BEGIN NONLIN
      [ INDEX=indn ]
      [ STEP=stepn ]
      END NONLIN

```

[*ind*=1] INDEX *ind* specifies the number of analysis block.

[*step*=1] STEP *step* specifies the step number in the NONLIN block.

41.2.3 Iteration

You may customize the iteration process in parameter estimation analysis via ITERAT commands.

syntax

```

BEGIN ITERAT
[ MAXITE=mitern ]
[ METHOD methodw ]
    FULL
    MODIFI
[ BEGIN CONVER
    criterw [ TOLCON=epsr ] ...
    RESIDU
    PARAME
    END CONVER ]
END ITERAT

```

MAXITE *miter* is the maximum number of iterations.

[*miter*=1]

METHOD *method* specifies when the sensitivity matrix must be calculated. This matrix comprises the derivatives of the observables to the parameters. The calculation of these derivatives is time consuming and therefore you may customize the calculation via this option.

FULL indicates that the sensitivity matrix must be calculated at the beginning of each iteration (the default).

[FULL]

MODIFI indicates that the sensitivity matrix must be calculated only at the beginning of the first iteration of every parameter estimation step.

CONVER specifies convergence criteria for a parameter estimation step.

RESIDU applies the residual convergence criterion. The iteration for the current step is terminated when the Euclidian norm of the differences between experimental and calculated displacements becomes less than ϵ .

PARAME applies the parameter convergence criterion. The iteration for the current step is terminated when the Euclidian norm of the parameter corrections becomes less than ϵ .

TOLCON *eps* is the value ϵ of the convergence criterion.

[$\epsilon = 1 \times 10^{10}$]

41.3 Output of Analysis Results

The output of a parameter estimation analysis comprises log information of the analysis process [§ 41.3.1] and a report of the estimation [§ 41.3.2].

41.3.1 Job Logging

For a general description of job logging see Chapter 3. This section gives some specific information on the job logging of a PAREST job which monitors the progress of the parameter estimation. All the repeated forward analyses are logged as usual to the standard output file. Amongst this information PAREST will produce additional log information like:

file.out

```

PARAMETER ESTIMATION STEP      1
PARAMETER ESTIMATION ITERATION 1
INITIALIZING A FORWARD ANALYSIS

```

These lines indicate that the forward analyses for the first parameter iteration have been prepared, forward analyses will follow. The forward analyses within an iteration to calculate the derivatives of the observables to the parameters are preceded by log lines like:

file.out

```

INITIALIZING A FORWARD ANALYSIS; PERTURBING PARAMETER      2

```

41.3.2 Estimation Report

Module PAREST writes an estimation report on the tabular output file. Typically an estimation report at the end of each estimation step comprises four blocks: estimation of the parameters, statistics of residuals, and estimation of observables.

41.3.2.1 Estimation of Parameters

The first block summarizes the unknown parameters, including the initial guesses as specified by the user.

file.tb

```

ESTIMATION OUTPUT: PARAMETERS
=====

```

PARAMETER NUMBER	:	1	2	3	4
PARAMETER NAME	:	POISON	YOUNG	YOUNG	XAXIS
TABLE NAME	:	MATERI	MATERI	MATERI	GEOMET
TABLE NUMBER	:	1	1	2	1
ITEM	:	1	1	1	2
STEP 0; ITERAT 0	:	0.3000E+00	0.2000E+02	0.1500E+02	0.1000E+01
STEP 1; ITERAT 1	:	0.2728E+00	0.1999E+02	0.1500E+02	0.1028E+01
...					

In this example the estimated parameter values of the first iteration of the first estimation step are output.

41.3.2.2 Statistics of Residuals

Each iteration starts with the calculation of the residuals, i.e., the difference between calculated and measured observables. Four statistical indicators with respect to these residuals are then calculated and output.

file.tb

```

ESTIMATION OUTPUT: STATISTICS OF RESIDUALS
=====

```

		NORM	MAXIMUM	STAND.DEV	MEAN
STEP 1; ITERAT 0	:	0.2935E+01	-0.7794E+00	0.2306E+00	-0.1969E+00
STEP 1; ITERAT 1	:	0.2522E+01	0.4069E+00	0.1982E+00	0.1646E+00
...					

The indicators are defined as the Euclidian norm of all residual components (**NORM**), the maximum component of the residuals in absolute sense (**MAXIMUM**), the standard deviation of the components (**STAND.DEV**) and the mean (**MEAN**).

41.3.2.3 Estimation of Observables

Finally the observables at the beginning of an iteration are output.

file.tb

ESTIMATION OUTPUT: OBSERVABLES
=====

STEP 1; ITERAT 1 :
1 POINT 1 DISPX -0.6102E-03
2 POINT 1 DISPY 0.8328E-03
3 POINT 2 DISPY 0.8328E-03
4 POINT 5 DISPX -0.6561E-03
5 POINT 101 DISPX -0.1006E-02
6 POINT 101 DISPY 0.1666E-02
...

Chapter 42

Loads Optimization

Module `BALANCE` offers a more compact interface to `PAREST`, dedicated to loads optimization in linear static analysis. As a side effect, `BALANCE` is faster than the general interface via `PAREST`.

42.1 Input Data

This section describes the syntax of input tables appropriate for loads optimization. The general concept of input tables is described in Chapter 1. See also Volume *Getting Started*.

42.1.1 Target Data

Target data for loads optimization, typically observed displacements, is input via subtable `OBSERV` of table '`TARGET`'.

				<i>syntax</i>
'TARGET'				
OBSERV				
1	5	6		80
<i>obs_n</i>	NODE <i>nodnr_n</i>	<i>type_w</i>	<i>datum_r</i>	
		DISPX		
		DISPY		
		DISPZ		
		RESIDX		
		RESIDY		
		RESIDZ		
		ROTATX		
		ROTATY		
		ROTATZ		

obs is the observable number. Observables may be input in arbitrary order, but no number may be skipped(!).

NODE *nodnr* is the node number for the observable.

type specifies the type of the data. `DISPX`, `DISPY` or `DISPZ` for displacement in model *X*, *Y* or *Z* direction respectively. `RESIDX`, `RESIDY` or `RESIDZ` for residual forces in model *X*, *Y* or *Z* direction respectively. `ROTATX`, `ROTATY` or `ROTATZ` for rotation in model *X*, *Y* or *Z* direction respectively.

datum is the target (displacement, rotation, or residual force).

42.2 Analysis Commands

The `*BALANC` command invokes module BALANC to optimize a combination of specified load sets by estimating multiplication factors.

syntax

```
*BALANC
[ MODEL ... ]
[ EXECUT ... ]
*END
```

MODEL evaluates the finite element model [§ 42.2.1].

EXECUT executes the loads optimization analysis [§ 42.2.2].

42.2.1 Model Evaluation

The MODEL commands customize the evaluation of the finite element model prior to the actual loads optimization analysis. For full syntax description see Chapter 3.

syntax

```
BEGIN MODEL
[ OFF ]
[ EVALUA [OFF] ... ]
[ ASSEMB [OFF] ... ]
[ MATRIX [OFF] ]
[ LOADS [OFF] ]
END MODEL
```

EVALUA to check and evaluate geometric and material properties for elements and reinforcements [§ 3.4 p. 51].

ASSEMB to assemble the elements, i.e., creating appropriate system degrees of freedom [§ 3.5 p. 54].

MATRIX to setup the element stiffness matrices.

LOADS to setup the load vectors.

42.2.2 Analysis Execution

The EXECUT commands perform the actual loads optimization analysis.

syntax

```
BEGIN EXECUT
[ SOLVE ... ]
[ BEGIN CALCUL
  [ BEGIN LOADS
    [ ESTIMA _____ ]
      lodvarn...
      ALL
    [ FIXED _____ ]
      lodfixn...
      NONE
  END LOADS ]
[ BALANC LOAD=lodnewn ]
```

```
END CALCUL ]
END EXECUT
```

SOLVE specifies the solution method. See Chapter 30 for full syntax description.

CALCUL specifies parameters for the loads optimization analysis.

LOADS specifies the appropriate loads for the optimization analysis.

ESTIMA *lodvar* specifies the variable load sets, i.e., for which multiplication factors must be determined. By default DIANA will consider all load sets to be variable. The ALL option conforms to this default behaviour. [ALL]

FIXED specifies fixed loads where *lodfix* are load set numbers. Fixed loads will be considered to act invariably on the model, i.e., with a multiplication factor fixed to 1. By default DIANA will not apply fixed load sets. The NONE option conforms to this default behaviour. [NONE]

BALANC LOAD=*lodnew* is a number for a new load set which will combine the variable load sets multiplied by their determined factors. By default DIANA will increment the value of the current highest load set to get the number for the new load set.

file.dcf

```
*BALANC
BEGIN EXECUT
  BEGIN CALCUL
    BEGIN LOADS
      FIXED 1
      ESTIMA 2-11
    END LOADS
    BALANC LOAD=15
  END CALCUL
END EXECUT
```

Through these example commands DIANA will assemble the finite element model and perform a linear elastic analysis by default. Next a load optimization analysis will be executed with load set 1 as fixed (typically the dead weight) and load sets 2 to 11 to be varied. After the analysis the load sets 2 to 11 will be combined in a new load set 15, with the resulting multiplication factors applied.

42.3 Output of Analysis results

Module BALANC does not deliver regular analysis results like displacements, strains, and stresses. Conversely, the output comprises an estimation report and a new load set with combines the variable loads with their determined multiplication factor. See also the example bridge in Volume *Analysis Examples*.

42.3.1 Estimation Report

Module BALANCE produces an optimization report on the standard output file. This report contains the optimal load factor for each variable load case. It also contains the residual which is the Euclidian norm of all residual components. The following is an example of an estimation report, produced by Module BALANC.

file.out

```
BALANCE OUTPUT:
=====
LOADSET FACTOR
  1  0.1000E+01 (FIXED)
  2  0.5168E+00
```

```

3  0.1330E+00
4  0.9441E-01
5  0.9858E-01
6  0.9453E-01
7  0.1076E+00
8  0.1173E+00
9  0.1441E+00
10 0.1491E+00
11 0.2128E+00
RESIDUAL:  0.6048E-04
/DIANA/DC/END      18:28:37      0.88-CPU      0.30-IO      259-FA      STOP

```

In this example load cases 2–11 are variable. Load case 1 is fixed, typically the dead weight load. Note that fixed load cases are reported with a factor 1.

42.3.2 Optimized Load

Module BALANCE creates a new load set which combines the variable load sets with the determined optimal multiplication factors. The new load set is left on the FILOS file; you may apply it to check the correctness of the optimization analysis via a subsequent linear static analysis. To see the actual load set you may give the **REMAKE** command [§ 3.3 p. 50].

file.dcf

```

*BALANC
...
*INPUT
REMAKE TABLE LOADS
*LINSTA
...

```

Chapter 43

Orthotropic Elastic Membrane

Name: memore
Path: /Examples/ParEst/memore
Keywords: ANALYS: linear parest static.
CLASS: hidden.
CONSTR: suppor.
ELEMEN: pstres q8mem.
LOAD: force node.
MATERI: elasti orthot.
OPTION: direct.
POST: binary femvie tabula.
RESULT: cauchy displa stress total.

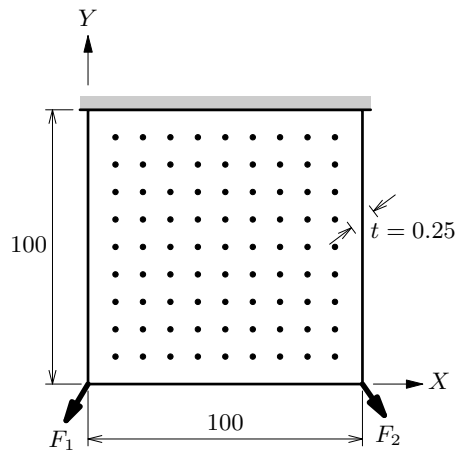


Figure 43.1: Identification experiment

This example illustrates the use of Module PAREST for a parameter estimation problem with an orthotropic elastic membrane. First the experimental setup will be explained briefly, followed by the creation of a linear finite element model of the membrane. Then, the special PAREST input data will be added to the data file and finally the PAREST commands used to determine the five unknown parameters of the membrane will be explained.

43.1 Experimental Setup

In the experimental setup [Fig. 43.1], a membrane of $100 \times 100 \times 0.25$ mm is tested. The membrane was clamped along one edge and was free to deform on the other sides. It was loaded with two forces, $F_1 = 0.1$ kN and $F_2 = 0.05$ kN, in the plane of the membrane. This resulted in strains up to a maximum of 0.03. With this load, wrinkling of the membrane

was avoided. The dots in Figure 43.1 represent material points of the membrane. The displacements of these points were measured in the experiment.

The material used in the experiment is a woven textile with orthotropic behaviour. Moreover, the textile can be considered as a membrane under plane stress conditions. Poisson's ratio ν_{xy} , Young's moduli E_x and E_y and the shear modulus G_{xy} of the material are unknown. Also the material orientation is unknown. These parameters will be estimated with Module PAREST using the displacements of the material points.

43.2 Finite Element Model

The finite element model consists of 100 Q8MEM plane stress elements [Fig. 43.2].

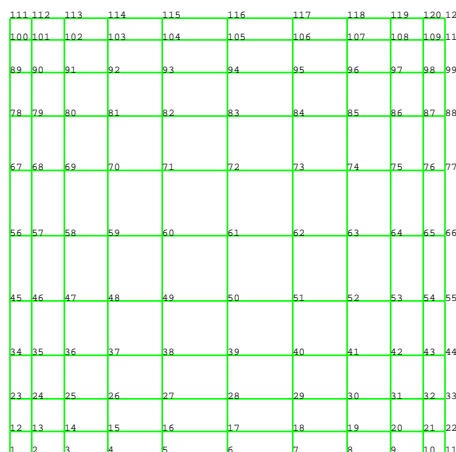


Figure 43.2: Finite element model

43.2.1 Element Mesh

The finite element mesh is given on an input data file in DIANA batch format. Below we show some fragments of this file.

```

membra.dat
-----
ORTHOTROPIC ELASTIC MEMBRANE
UNITS: mm, kN
'COORDINATES'
  1  0.00000E+00  0.00000E+00  0.00000E+00
  2  5.00000E+00  0.00000E+00  0.00000E+00
  3  1.25000E+01  0.00000E+00  0.00000E+00
...
 120  9.50000E+01  1.00000E+02  0.00000E+00
 121  1.00000E+02  1.00000E+02  0.00000E+00
'ELEMENTS'
SET "Element set 0"
CONNECT
  1 Q8MEM  1 2 13 12
  2 Q8MEM  2 3 14 13
...
 99 Q8MEM 108 109 120 119
100 Q8MEM 109 110 121 120
MATERIAL 1
GEOMETRY 1
-----
```

lines skipped

lines skipped

43.2.2 Material Data, Supports and Loads

The definition of the material data, supports and loads is given on the same input data file.

```

membra.dat

'MATERI'
  1 YOUNG    5.00000E-01    5.00000E-01
    POISON   2.50000E-01
    SHRMOD   1.50000E-01
'GEOMET'
  1 THICK    2.00000E-01
    XAXIS    1.00000E+00    1.10000E+00    0.00000E+00
'DIRECTIONS'
  1  1.00000E+00    0.00000E+00    0.00000E+00
  2  0.00000E+00    1.00000E+00    0.00000E+00
  3  0.00000E+00    0.00000E+00    1.00000E+00
  4  1.00000E+00   -1.42800E+00    0.00000E+00
  5 -1.00000E+00   -1.66200E+00    0.00000E+00
'SUPPOR'
NAME SET_1
/ 111-121 / TR 1
/ 111-121 / TR 2
'LOADS'
CASE 1
NAME "Load case 1"
NODAL
11  FORCE  4    5.00000E-02
1  FORCE  5    1.00000E-01

```

Note that orthotropic behaviour is modeled by providing two data items behind **YOUNG** and two items behind **POISON**. The material orientation is specified by means of the **XAXIS** keyword. Note that these denote initial estimations for the stiffness parameters and the orthotropic orientation.

43.3 Preliminary Analysis

It is strongly recommended to check the model and to perform a standard linear analysis, before using **PAREST** for the actual parameter estimation. This can be done with the following analysis commands.

```

prelim.dcf

*FILOS
  INITIA
*INPUT
*LINSTA
BEGIN OUTPUT FEMVIE
  DISPLA
  STRESS INTPNT
END OUTPUT
*END

```

The **DISPLA** command stores the displacements of the nodes. The **STRESS INTPNT** command stores the stresses in the integration points of the elements. Now we run **DIANA** with this command file and the additional input file.

```
diana prelim membra.dat
```

This job creates a data base for postprocessing of model **PRELIM**.

43.3.1 Deformation

We now enter the *i*DIANA Postprocessing environment and first display the deformed mesh.

```
prelim.fvc
```

```
FEMVIEW PRELIM
VIEW MESH
RESULTS LOADCASE LC1
RESULTS NODAL DTX....G RESDTX
PRESENT SHAPE
```

The FEMVIEW command opens the database for postprocessing of model PRELIM. The VIEW MESH command displays the undeformed mesh in green. Then the RESULTS NODAL command with the DT attribute selects the nodal displacements as result item. The PRESENT SHAPE command displays the deformed mesh in red [Fig. 43.3].

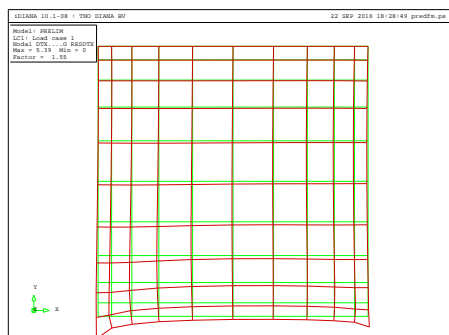


Figure 43.3: Deformed mesh

43.3.2 Stresses

With the following RESULTS and PRESENT commands we display the stress distribution.

```
prelim.fvc
```

```
RESULTS GAUSSIAN EL.SXX.G SXX
RESULTS CALCULATE P-STRESS ALL
PRESENT VECTORS
RESULTS CALCULATE VONMISES
PRESENT CONTOUR LEVELS
```

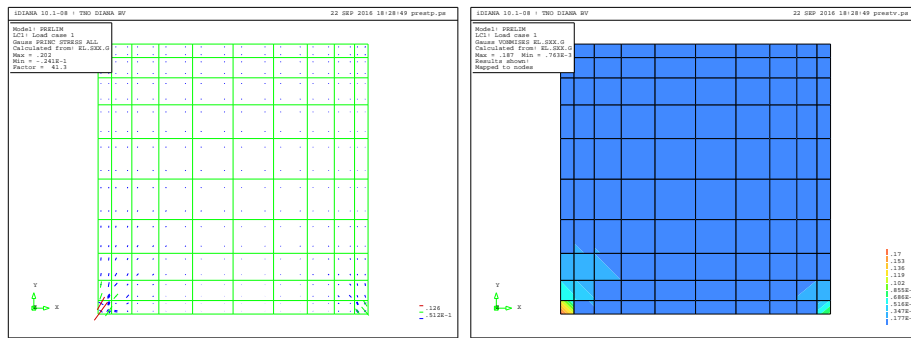
The GAUSSIAN option with the EL.S attribute selects the stresses in the integration points of the elements. The CALCULATE P-STRESS option asks *i*DIANA to calculate the principal stresses. The VECTORS option displays these stresses in vector style [Fig. 43.4a]. The CALCULATE VONMISES option yields the equivalent Von Mises stresses and the CONTOUR LEVELS option displays these in filled contour style [Fig. 43.4b].

43.4 Parameter Estimation Analysis

Parameter Estimation requires some data additionally to that for the preliminary linear analysis.

43.4.1 Additional Data

To determine the unknown parameters DIANA needs two extra tables in addition to the input data shown previously. The first table is 'TARGET' which contains information about the experiment [§ 40.1]. The second table is 'ESTIMA' and specifies the unknown parameters [§ 40.2]. These extra tables for this example are given below.



(a) principal stress vectors

(b) Von Mises stress contours

Figure 43.4: Stresses

add2.dat

```

: Diana Datafile written by Diana Dev (Latest update: 2014-06-26)
'DIRECTIONS'
  1  1.00000E+00  0.00000E+00  0.00000E+00
  2  0.00000E+00  1.00000E+00  0.00000E+00
  3  0.00000E+00  0.00000E+00  1.00000E+00
'MODEL'
GRAVDI 3
GRAVAC -9.81000E+00
'TARGET'
POINTS
  1  1.00000E+01  9.00000E+01  0.00000E+00
  2  2.00000E+01  9.00000E+01  0.00000E+00
  3  3.00000E+01  9.00000E+01  0.00000E+00
  4  4.00000E+01  9.00000E+01  0.00000E+00
  5  5.00000E+01  9.00000E+01  0.00000E+00
  6  6.00000E+01  9.00000E+01  0.00000E+00
  7  7.00000E+01  9.00000E+01  0.00000E+00
  8  8.00000E+01  9.00000E+01  0.00000E+00
  9  9.00000E+01  9.00000E+01  0.00000E+00
 10  1.00000E+01  8.00000E+01  0.00000E+00
 11  2.00000E+01  8.00000E+01  0.00000E+00
 12  3.00000E+01  8.00000E+01  0.00000E+00
 13  4.00000E+01  8.00000E+01  0.00000E+00
 14  5.00000E+01  8.00000E+01  0.00000E+00
 15  6.00000E+01  8.00000E+01  0.00000E+00
 16  7.00000E+01  8.00000E+01  0.00000E+00
 17  8.00000E+01  8.00000E+01  0.00000E+00
 18  9.00000E+01  8.00000E+01  0.00000E+00
 19  1.00000E+01  7.00000E+01  0.00000E+00
 20  2.00000E+01  7.00000E+01  0.00000E+00
 21  3.00000E+01  7.00000E+01  0.00000E+00
 22  4.00000E+01  7.00000E+01  0.00000E+00
 23  5.00000E+01  7.00000E+01  0.00000E+00
 24  6.00000E+01  7.00000E+01  0.00000E+00
 25  7.00000E+01  7.00000E+01  0.00000E+00
 26  8.00000E+01  7.00000E+01  0.00000E+00
 27  9.00000E+01  7.00000E+01  0.00000E+00
 28  1.00000E+01  6.00000E+01  0.00000E+00
 29  2.00000E+01  6.00000E+01  0.00000E+00
 30  3.00000E+01  6.00000E+01  0.00000E+00
 31  4.00000E+01  6.00000E+01  0.00000E+00
 32  5.00000E+01  6.00000E+01  0.00000E+00
 33  6.00000E+01  6.00000E+01  0.00000E+00
 34  7.00000E+01  6.00000E+01  0.00000E+00

```

35	8.00000E+01	6.00000E+01	0.00000E+00
36	9.00000E+01	6.00000E+01	0.00000E+00
37	1.00000E+01	5.00000E+01	0.00000E+00
38	2.00000E+01	5.00000E+01	0.00000E+00
39	3.00000E+01	5.00000E+01	0.00000E+00
40	4.00000E+01	5.00000E+01	0.00000E+00
41	5.00000E+01	5.00000E+01	0.00000E+00
42	6.00000E+01	5.00000E+01	0.00000E+00
43	7.00000E+01	5.00000E+01	0.00000E+00
44	8.00000E+01	5.00000E+01	0.00000E+00
45	9.00000E+01	5.00000E+01	0.00000E+00
46	1.00000E+01	4.00000E+01	0.00000E+00
47	2.00000E+01	4.00000E+01	0.00000E+00
48	3.00000E+01	4.00000E+01	0.00000E+00
49	4.00000E+01	4.00000E+01	0.00000E+00
50	5.00000E+01	4.00000E+01	0.00000E+00
51	6.00000E+01	4.00000E+01	0.00000E+00
52	7.00000E+01	4.00000E+01	0.00000E+00
53	8.00000E+01	4.00000E+01	0.00000E+00
54	9.00000E+01	4.00000E+01	0.00000E+00
55	1.00000E+01	3.00000E+01	0.00000E+00
56	2.00000E+01	3.00000E+01	0.00000E+00
57	3.00000E+01	3.00000E+01	0.00000E+00
58	4.00000E+01	3.00000E+01	0.00000E+00
59	5.00000E+01	3.00000E+01	0.00000E+00
60	6.00000E+01	3.00000E+01	0.00000E+00
61	7.00000E+01	3.00000E+01	0.00000E+00
62	8.00000E+01	3.00000E+01	0.00000E+00
63	9.00000E+01	3.00000E+01	0.00000E+00
64	1.00000E+01	2.00000E+01	0.00000E+00
65	2.00000E+01	2.00000E+01	0.00000E+00
66	3.00000E+01	2.00000E+01	0.00000E+00
67	4.00000E+01	2.00000E+01	0.00000E+00
68	5.00000E+01	2.00000E+01	0.00000E+00
69	6.00000E+01	2.00000E+01	0.00000E+00
70	7.00000E+01	2.00000E+01	0.00000E+00
71	8.00000E+01	2.00000E+01	0.00000E+00
72	9.00000E+01	2.00000E+01	0.00000E+00
73	1.00000E+01	1.00000E+01	0.00000E+00
74	2.00000E+01	1.00000E+01	0.00000E+00
75	3.00000E+01	1.00000E+01	0.00000E+00
76	4.00000E+01	1.00000E+01	0.00000E+00
77	5.00000E+01	1.00000E+01	0.00000E+00
78	6.00000E+01	1.00000E+01	0.00000E+00
79	7.00000E+01	1.00000E+01	0.00000E+00
80	8.00000E+01	1.00000E+01	0.00000E+00
81	9.00000E+01	1.00000E+01	0.00000E+00
OBSERV 1			
1	POINT 1 DISPX	2.86400E-02	
2	POINT 1 DISPY	-1.76200E-01	
3	POINT 2 DISPX	-2.49800E-02	
4	POINT 2 DISPY	-1.75200E-01	
5	POINT 3 DISPX	-6.24300E-02	
6	POINT 3 DISPY	-1.79700E-01	
7	POINT 4 DISPX	-9.12700E-02	
8	POINT 4 DISPY	-1.81200E-01	
9	POINT 5 DISPX	-1.15000E-01	
10	POINT 5 DISPY	-1.79200E-01	
11	POINT 6 DISPX	-1.31100E-01	
12	POINT 6 DISPY	-1.71100E-01	
13	POINT 7 DISPX	-1.45900E-01	
14	POINT 7 DISPY	-1.63800E-01	
15	POINT 8 DISPX	-1.61500E-01	

```
16 POINT 8 DISPY -1.61700E-01
17 POINT 9 DISPX -1.90800E-01
18 POINT 9 DISPY -1.84300E-01
19 POINT 10 DISPX 1.41100E-02
20 POINT 10 DISPY -3.84300E-01
21 POINT 11 DISPX -7.04400E-02
22 POINT 11 DISPY -3.86800E-01
23 POINT 12 DISPX -1.34500E-01
24 POINT 12 DISPY -3.89000E-01
25 POINT 13 DISPX -1.82400E-01
26 POINT 13 DISPY -3.85000E-01
27 POINT 14 DISPX -2.19800E-01
28 POINT 14 DISPY -3.75200E-01
29 POINT 15 DISPX -2.45700E-01
30 POINT 15 DISPY -3.55500E-01
31 POINT 16 DISPX -2.72800E-01
32 POINT 16 DISPY -3.38700E-01
33 POINT 17 DISPX -3.05000E-01
34 POINT 17 DISPY -3.32000E-01
35 POINT 18 DISPX -3.53100E-01
36 POINT 18 DISPY -3.57700E-01
37 POINT 19 DISPX -5.69000E-02
38 POINT 19 DISPY -6.11500E-01
39 POINT 20 DISPX -1.53100E-01
40 POINT 20 DISPY -6.22600E-01
41 POINT 21 DISPX -2.25600E-01
42 POINT 21 DISPY -6.18500E-01
43 POINT 22 DISPX -2.76000E-01
44 POINT 22 DISPY -6.02300E-01
45 POINT 23 DISPX -3.11300E-01
46 POINT 23 DISPY -5.77800E-01
47 POINT 24 DISPX -3.34700E-01
48 POINT 24 DISPY -5.44200E-01
49 POINT 25 DISPX -3.63000E-01
50 POINT 25 DISPY -5.14900E-01
51 POINT 26 DISPX -4.00200E-01
52 POINT 26 DISPY -4.96300E-01
53 POINT 27 DISPX -4.50700E-01
54 POINT 27 DISPY -5.03100E-01
55 POINT 28 DISPX -1.67900E-01
56 POINT 28 DISPY -8.64200E-01
57 POINT 29 DISPX -2.69500E-01
58 POINT 29 DISPY -8.74600E-01
59 POINT 30 DISPX -3.35900E-01
60 POINT 30 DISPY -8.53100E-01
61 POINT 31 DISPX -3.71300E-01
62 POINT 31 DISPY -8.14300E-01
63 POINT 32 DISPX -3.85700E-01
64 POINT 32 DISPY -7.66200E-01
65 POINT 33 DISPX -3.91300E-01
66 POINT 33 DISPY -7.19800E-01
67 POINT 34 DISPX -4.08100E-01
68 POINT 34 DISPY -6.79500E-01
69 POINT 35 DISPX -4.41200E-01
70 POINT 35 DISPY -6.49000E-01
71 POINT 36 DISPX -4.90200E-01
72 POINT 36 DISPY -6.32500E-01
73 POINT 37 DISPX -3.04100E-01
74 POINT 37 DISPY -1.13700E+00
75 POINT 38 DISPX -4.09300E-01
76 POINT 38 DISPY -1.13700E+00
77 POINT 39 DISPX -4.59900E-01
78 POINT 39 DISPY -1.08800E+00
```

```
79 POINT 40 DISPX -4.67300E-01
80 POINT 40 DISPY -1.01800E+00
81 POINT 41 DISPX -4.46500E-01
82 POINT 41 DISPY -9.39500E-01
83 POINT 42 DISPX -4.22900E-01
84 POINT 42 DISPY -8.81900E-01
85 POINT 43 DISPX -4.18900E-01
86 POINT 43 DISPY -8.32800E-01
87 POINT 44 DISPX -4.42100E-01
88 POINT 44 DISPY -7.92700E-01
89 POINT 45 DISPX -4.89500E-01
90 POINT 45 DISPY -7.53600E-01
91 POINT 46 DISPX -4.76300E-01
92 POINT 46 DISPY -1.49000E+00
93 POINT 47 DISPX -5.75100E-01
94 POINT 47 DISPY -1.42600E+00
95 POINT 48 DISPX -5.81100E-01
96 POINT 48 DISPY -1.29400E+00
97 POINT 49 DISPX -5.34600E-01
98 POINT 49 DISPY -1.15700E+00
99 POINT 50 DISPX -4.60200E-01
100 POINT 50 DISPY -1.03000E+00
101 POINT 51 DISPX -3.98000E-01
102 POINT 51 DISPY -9.76900E-01
103 POINT 52 DISPX -3.62100E-01
104 POINT 52 DISPY -9.42700E-01
105 POINT 53 DISPX -3.65600E-01
106 POINT 53 DISPY -9.23700E-01
107 POINT 54 DISPX -4.14100E-01
108 POINT 54 DISPY -8.86000E-01
109 POINT 55 DISPX -6.54600E-01
110 POINT 55 DISPY -1.90200E+00
111 POINT 56 DISPX -7.18500E-01
112 POINT 56 DISPY -1.70900E+00
113 POINT 57 DISPX -6.56500E-01
114 POINT 57 DISPY -1.45800E+00
115 POINT 58 DISPX -5.56000E-01
116 POINT 58 DISPY -1.24500E+00
117 POINT 59 DISPX -4.46700E-01
118 POINT 59 DISPY -1.07900E+00
119 POINT 60 DISPX -3.61900E-01
120 POINT 60 DISPY -1.03300E+00
121 POINT 61 DISPX -2.97900E-01
122 POINT 61 DISPY -1.02200E+00
123 POINT 62 DISPX -2.72100E-01
124 POINT 62 DISPY -1.04500E+00
125 POINT 63 DISPX -3.11700E-01
126 POINT 63 DISPY -1.04000E+00
127 POINT 64 DISPX -8.21900E-01
128 POINT 64 DISPY -2.39600E+00
129 POINT 65 DISPX -7.83300E-01
130 POINT 65 DISPY -1.95100E+00
131 POINT 66 DISPX -6.37900E-01
132 POINT 66 DISPY -1.55000E+00
133 POINT 67 DISPX -5.04000E-01
134 POINT 67 DISPY -1.26900E+00
135 POINT 68 DISPX -3.96200E-01
136 POINT 68 DISPY -1.08000E+00
137 POINT 69 DISPX -3.17300E-01
138 POINT 69 DISPY -1.04000E+00
139 POINT 70 DISPX -2.40500E-01
140 POINT 70 DISPY -1.05700E+00
141 POINT 71 DISPX -1.78100E-01
```

```

142 POINT 71 DISPY -1.14000E+00
143 POINT 72 DISPX -1.80300E-01
144 POINT 72 DISPY -1.22600E+00
145 POINT 73 DISPX -8.21000E-01
146 POINT 73 DISPY -2.91100E+00
147 POINT 74 DISPX -5.86900E-01
148 POINT 74 DISPY -2.04700E+00
149 POINT 75 DISPX -4.35200E-01
150 POINT 75 DISPY -1.54700E+00
151 POINT 76 DISPX -3.54600E-01
152 POINT 76 DISPY -1.25100E+00
153 POINT 77 DISPX -3.10500E-01
154 POINT 77 DISPY -1.06400E+00
155 POINT 78 DISPX -2.81900E-01
156 POINT 78 DISPY -1.02300E+00
157 POINT 79 DISPX -2.39600E-01
158 POINT 79 DISPY -1.04500E+00
159 POINT 80 DISPX -1.71500E-01
160 POINT 80 DISPY -1.16600E+00
161 POINT 81 DISPX -8.35300E-02
162 POINT 81 DISPY -1.41800E+00
'ESTIMA'
PARAME
  1 NAME  YOUNG
    MATERI 1
    ITEM  1
    BOUNDS 1.00000E-03 1.00000E+02
  2 NAME  YOUNG
    MATERI 1
    ITEM  2
    BOUNDS 1.00000E-03 1.00000E+02
  3 NAME  POISON
    MATERI 1
    ITEM  1
    BOUNDS 1.00000E-03 4.99000E-01
  4 NAME  SHRMOD
    MATERI 1
    ITEM  1
    BOUNDS 1.00000E-03 1.00000E+02
  5 NAME  XAXIS
    GEOMET 1
    ITEM  2
    BOUNDS 1.00000E-01 2.00000E+00
'END'
```

The subtables POINTS and OBSERV contain the original positions and displacements of the material points measured in the experiment. In subtable PARAME five parameters are specified. The first refers to the first entry behind the YOUNG keyword of material 1 in table 'MATERI'.

43.4.2 Analysis

To perform the parameter estimation we apply the following commands.

estima.dcf

```

*INPUT
*LINSTA
  OUTPUT FEMVIE
*PAREST
OUTPUT TABULA PAREST
BEGIN EXECUT
OBSERV=1
```

```

BEGIN LINSTA
  INDEX=1
  LOAD=1
END LINSTA
BEGIN ITERAT
  MAXITE=10
  CONVER PARAMETER TOLCON=1.E-4
END ITERAT
END EXECUT
*END

```

We run DIANA with this command file and the additional input file.

```
diana estima add2.dat
```

43.4.3 Estimation Report

The following is a selection of the results of the parameter estimation.

estima.tb

```

ESTIMATION OUTPUT: PARAMETERS
=====

```

PARAMETER NUMBER :	1	2	3	4
PARAMETER NAME :	YOUNG	YOUNG	POISON	SHRMOD
TABLE NAME :	MATERI	MATERI	MATERI	MATERI
TABLE NUMBER :	1	1	1	1
ITEM :	1	2	1	1
STEP 0; ITERAT 0 :	0.5000E+00	0.5000E+00	0.2500E+00	0.1500E+00
STEP 1; ITERAT 1 :	0.3702E+00	0.6029E+00	0.1235E+00	0.8477E-01
STEP 1; ITERAT 2 :	0.3890E+00	0.5771E+00	0.1703E+00	0.9986E-01
STEP 1; ITERAT 3 :	0.3988E+00	0.5967E+00	0.1531E+00	0.1004E+00
STEP 1; ITERAT 4 :	0.4000E+00	0.5999E+00	0.1501E+00	0.1000E+00
STEP 1; ITERAT 5 :	0.4000E+00	0.5999E+00	0.1501E+00	0.1000E+00

PARAMETER NUMBER :	5
PARAMETER NAME :	XAXIS
TABLE NAME :	GEOMET
TABLE NUMBER :	1
ITEM :	2
STEP 0; ITERAT 0 :	0.1100E+01
STEP 1; ITERAT 1 :	0.7638E+00
STEP 1; ITERAT 2 :	0.9209E+00
STEP 1; ITERAT 3 :	0.1001E+01
STEP 1; ITERAT 4 :	0.1000E+01
STEP 1; ITERAT 5 :	0.1000E+01

```

ESTIMATION OUTPUT: STATISTICS OF RESIDUALS
=====

```

		NORM	MAXIMUM	STAND.DEV	MEAN
STEP 1; ITERAT 0 :		0.2935E+01	-0.7794E+00	0.2306E+00	-0.1969E+00
STEP 1; ITERAT 1 :		0.2522E+01	0.4069E+00	0.1982E+00	0.1646E+00
STEP 1; ITERAT 2 :		0.5659E+00	0.9229E-01	0.4447E-01	0.2840E-01
STEP 1; ITERAT 3 :		0.2729E-01	-0.5772E-02	0.2144E-02	-0.2930E-03
STEP 1; ITERAT 4 :		0.1839E-02	-0.5239E-03	0.1445E-03	0.1445E-04

```

ESTIMATION OUTPUT: OBSERVABLES
=====

```

STEP 1; ITERAT 4 :			
1	POINT	1	DISPX 0.2866E-01
2	POINT	1	DISPY -0.1762E+00
3	POINT	2	DISPX -0.2496E-01
4	POINT	2	DISPY -0.1752E+00
159	POINT	80	DISPX -0.1715E+00
160	POINT	80	DISPY -0.1166E+01
161	POINT	81	DISPX -0.8352E-01
162	POINT	81	DISPY -0.1418E+01

lines skipped

We observe that the five parameters converge while the residuals decrease.

Part XIII

Background Theory

The purpose of Part XIII is to provide the DIANA user the theory on which the DIANA code has been based. It is not intended to be a general reference for the Finite Element Method. For that, readers are referred to the well-known handbooks like for instance Bathe [5]¹, Zienkiewicz [93] or Hughes [47].

The theory description here is subdivided in two major parts: the general concept of the Finite Element Method and theoretical aspects of the derivation of the element stiffness matrices. Chapter 44 presents a summary of the general concepts of the Finite Element Method. This chapter intends to give an understanding of the basic principles of finite element theory and how these have been implemented in DIANA. Chapters 45 and 46 present the methods of solution of the system of equations. This should provide the user insight in the command structure of DIANA. In short, these chapters include the composition of the element matrices, the composition of the load vector, the methods of processing boundary conditions and tyings, solving the displacement vectors, and iterative solution of nonlinear systems.

The remaining chapters give some background theory of particular types of analysis: structural dynamics, stability, potential flow, and soil-pore fluid.

¹Notation conventions in this manual follow more or less Bathe's book.

Chapter 44

General Concepts of FEM

For linear elastic problems the system of equations to be solved is

$$\mathbf{K}\mathbf{u} = \mathbf{f} \quad (44.1)$$

where \mathbf{K} is the *system stiffness matrix*, \mathbf{u} is a vector of the unknown nodal *degrees of freedom* such as displacements and rotations and \mathbf{f} is the vector of the *nodal forces* corresponding with the degrees of freedom \mathbf{u} . This chapter describes the composition of the system stiffness matrix, the solution of the degrees of freedom and the calculation of the element results as strains and stresses.

44.1 Global Formulation

When considering a general three-dimensional body, denoted by V , the problem is identified by unknown displacements \mathbf{u} and known body forces per unit volume \mathbf{g} . External forces in the form of concentrated forces and known tractions \mathbf{t} are applied to the part S_t of the boundary and are called the *natural boundary conditions*. The displacements \mathbf{u} are specified as known values $\bar{\mathbf{u}}$ on the part S_u of the boundary and are called the *essential boundary conditions*. In the Finite Element Method the body V will be approximated as an assemblage of finite elements, which are connected by nodal points on the element boundaries.

44.1.1 Displacements

In order to solve the problem the displacements \mathbf{u} has to satisfy a continuity and differentiability to the necessary degree. On the boundary S_u the displacements must satisfy the essential boundary condition

$$\mathbf{u} = \bar{\mathbf{u}} \quad \text{on} \quad S_u \quad (44.2)$$

The displacements of a particular point (x, y, z) are assumed to be continuous functions expressed in terms of discretized variables at the nodal points and are approximated as

$$\mathbf{u}_c(x, y, z) \approx \tilde{\mathbf{u}}(x, y, z) = \mathbf{N}(x, y, z)\mathbf{u} \quad (44.3)$$

where \mathbf{N} is the displacement interpolation matrix and \mathbf{u} is a vector of nodal point variables such as components of displacements and rotations, and is denoted as the vector of *degrees of freedom*. The interpolation matrix \mathbf{N} comprises *interpolation* or *shape functions* described in terms of independent variables, such as coordinates and are locally defined for the individual elements.

44.1.2 Strains and Stresses

The strains at any point in the structure can be determined by

$$\boldsymbol{\varepsilon} = \mathbf{L}\mathbf{u} \quad (44.4)$$

where \mathbf{L} is a differential operator defining a compatible strain field. Now the strain field can be written as the derivative of the vector \mathbf{u} as

$$\boldsymbol{\varepsilon} = \mathbf{L}\tilde{\mathbf{u}} = \mathbf{L}\mathbf{N}\mathbf{u} = \mathbf{B}\mathbf{u} \quad (44.5)$$

where the matrix \mathbf{B} defines the *strain-displacement relation* for a particular point and is called the *differential matrix*. Assuming linear elastic behaviour, the relation between stresses and strains in a particular point can be written in the form

$$\boldsymbol{\sigma} = \mathbf{D}(\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_0) + \boldsymbol{\sigma}_0 \quad (44.6)$$

where the matrix \mathbf{D} is the *stress-strain relation* and is a function of material properties like Young's modulus E and Poisson's ratio ν . The vector $\boldsymbol{\varepsilon}_0$ denotes the initial strains changes etc. and the vector $\boldsymbol{\sigma}_0$ contains the initial residual stresses.

44.1.3 Equilibrium

In a structural problem the governing equilibrium equations can be written as

$$\begin{aligned} \mathbf{L}^T \boldsymbol{\sigma} + \mathbf{g} &= 0 & \text{on } V \\ \mathbf{L}_n^T \boldsymbol{\sigma} &= \mathbf{t} & \text{on } S_t \end{aligned} \quad (44.7)$$

where \mathbf{g} is the vector of the known body forces per unit volume, with V as the total volume or domain of the model. Vector \mathbf{t} represents the known traction forces on the boundary S_t such as surface, edge and point loads. For the derivation of the equilibrium equations the stationarity condition of the total potential energy can be used.

44.1.4 Principle of Virtual Displacements

A simpler way of introducing the equilibrium relations of Eq. (44.7) can be done by invoking the *principle of virtual displacements*. This principle states that an elastic structure is in equilibrium under a given loading system if, for any virtual displacement from a compatible state of deformation, the virtual work is equal to the virtual strain energy. The virtual work equation can be written as

$$\int_V \delta \boldsymbol{\varepsilon}^T \boldsymbol{\sigma} dV = \int_V \delta \mathbf{u}^T \mathbf{g} dV + \int_{S_t} \delta \mathbf{u}^T \mathbf{t} dS_t \quad (44.8)$$

where $\delta \boldsymbol{\varepsilon}$ are the virtual strains which correspond to the virtual displacements $\delta \mathbf{u}$. Substituting Eqs. (44.3) and (44.5) into Eq. (44.8) gives

$$\delta \mathbf{u}^T \int_V \mathbf{B}^T \boldsymbol{\sigma} dV = \delta \mathbf{u}^T \left(\int_V \mathbf{N}^T \mathbf{g} dV + \int_{S_t} \mathbf{N}^T \mathbf{t} dS_t \right) = \delta \mathbf{u}^T \mathbf{r} \quad (44.9)$$

where \mathbf{r} is the vector of the *internal forces* corresponding to the vector of the nodal degrees of freedom \mathbf{u} . The principle of the virtual work states that Eq. (44.9) should be satisfied for any \mathbf{u} so that

$$\int_V \mathbf{B}^T \boldsymbol{\sigma} dV = \mathbf{r} \quad (44.10)$$

These equations do not ensure that the equilibrium is satisfied at *any* point, but only guarantee that the stresses satisfy equilibrium in a weighted average sense. Substituting Eqs. (44.6) and (44.5), the left hand side of Eq. (44.10) can be written as

$$\int_V \mathbf{B}^T \boldsymbol{\sigma} dV = \left(\int_V \mathbf{B}^T \mathbf{D} \mathbf{B} dV \right) \mathbf{u} - \int_V \mathbf{B}^T \mathbf{D} \boldsymbol{\varepsilon}_0 dV + \int_V \mathbf{B}^T \boldsymbol{\sigma}_0 dV = \mathbf{r} \quad (44.11)$$

Combining the expression for \mathbf{r} in Eqs. (44.9) and (44.11) we obtain

$$\mathbf{K}\mathbf{u} = \mathbf{f} \quad (44.12)$$

where

$$\mathbf{K} = \int_V \mathbf{B}^T \mathbf{D} \mathbf{B} dV \quad (44.13)$$

is the system stiffness matrix, and \mathbf{f} is the right hand side vector defined by

$$\mathbf{f} = \mathbf{f}_g + \mathbf{f}_t + \mathbf{f}_{\varepsilon_0} - \mathbf{f}_{\sigma_0} + \mathbf{f}_c \quad (44.14)$$

with

$$\begin{aligned} \mathbf{f}_g &= \int_V \mathbf{N}^T \mathbf{g} dV && \text{the contribution of the body forces.} \\ \mathbf{f}_t &= \int_{S_t} \mathbf{N}^T \mathbf{t} dS_t && \text{the contribution of the surface tractions.} \\ \mathbf{f}_{\varepsilon_0} &= \int_V \mathbf{B}^T \mathbf{D} \varepsilon_0 dV && \text{the effect of the initial strains.} \\ \mathbf{f}_{\sigma_0} &= \int_V \mathbf{B}^T \sigma_0 dV && \text{the effect of the initial stresses.} \\ \mathbf{f}_c &&& \text{the contribution of the concentrated nodal forces.} \end{aligned}$$

This provides a set of linear simultaneous equations which can be solved in a direct or indirect way:

$$\mathbf{u} = \mathbf{K}^{-1} \mathbf{f} \quad (44.15)$$

The method of solution is outlined in Chapter 45.

44.2 Discretization to Elements

In the Finite Element Method the solution domain V is divided into a finite number of elements V_e , which are connected by nodal points at the inter-element boundaries. In this way the solution domain is discretized and represented as a patch of elements. The unknown displacements in each element are now approximated by continuous functions expressed in terms of nodal variables; the functions over each finite element are called *interpolation* or *shape functions*. See Volume *Element Library* for more background theory.

44.2.1 Displacements

In each element the displacements of an arbitrary point (x, y, z) can be measured in a convenient local *Cartesian* coordinate system and are approximated by shape functions and nodal variables

$$\mathbf{u}_c(x, y, z) = \mathbf{N} \mathbf{u}_e \quad (44.16)$$

where \mathbf{N} is the interpolation matrix with shape functions $N(x, y, z)$ and \mathbf{u}_e the element nodal displacement vector,¹ expressed in local xyz axes. This element vector can be composed from the nodal variables of the system degrees of freedom vector \mathbf{u} by

$$\mathbf{u}_e = \mathbf{T}_e \mathbf{u} \quad (44.17)$$

where \mathbf{T}_e is the *element transformation matrix* which transforms the corresponding system degrees of freedom to the local element degrees of freedom, oriented in the xyz coordinate system.

The rest is identical to §44.1.1.

¹From now on, the distinction between element and structural matrices, vectors and scalars has been done by adding a subscript $_e$ to the element quantities whenever this distinction cannot be obviously recognized.

44.2.2 Strains and Stresses

Using the strain–displacement law for compatibility and assuming that the shape functions \mathbf{N} are known, the discrete form of the strain–displacement relation can be written as

$$\boldsymbol{\varepsilon} = \mathbf{B}\mathbf{u}_e \quad (44.18)$$

Likewise Eq. (44.6) for the entire domain, the relation between strains and stresses, including initial strains and initial stresses, can be written for an element as

$$\boldsymbol{\sigma} = \mathbf{D}(\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_0) + \boldsymbol{\sigma}_0 \quad (44.19)$$

where \mathbf{D} is the *rigidity matrix* representing the stress–strain law, usually derived from Hooke's law, varying from element to element. Often the matrix \mathbf{D} is only defined in a local element Cartesian (x_l, y_l, z_l) coordinate system. In order to obtain the strains in this system, it is necessary to apply a strain transformation

$$\boldsymbol{\varepsilon}_l = \mathbf{T}_\varepsilon \boldsymbol{\varepsilon} \quad (44.20)$$

where \mathbf{T}_ε is the strain transformation matrix. With Eq. (44.18) the local strain vector $\boldsymbol{\varepsilon}_l$ can now be related directly to the local element degrees of freedom vector \mathbf{u}_e by

$$\boldsymbol{\varepsilon}_l = \mathbf{T}_\varepsilon \mathbf{B}\mathbf{u}_e = \mathbf{B}_l \mathbf{u}_e \quad (44.21)$$

44.2.3 Element Assembly

In DIANA, the assembling of elements is performed due to the **ASSEMB** command [§ 3.5 p. 54]. During this process, also the constraints and the linear constraints (tyings) are handled.

44.2.3.1 Linear Constraints

The general tying equation for eccentric connection of three translations and three rotations is

$$\begin{Bmatrix} u_{x_i} \\ u_{y_i} \\ u_{z_i} \\ \phi_{x_i} \\ \phi_{y_i} \\ \phi_{z_i} \end{Bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & -\Delta_z & \Delta_y \\ 0 & 1 & 0 & \Delta_z & 0 & -\Delta_x \\ 0 & 0 & 1 & -\Delta_y & \Delta_x & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} u_{x_j} \\ u_{y_j} \\ u_{z_j} \\ \phi_{x_j} \\ \phi_{y_j} \\ \phi_{z_j} \end{Bmatrix} \quad (44.22)$$

Where u_{x_i} denotes the x translation of the slave node, ϕ_{y_j} the y rotation of the master node, Δ_x the eccentricity in x direction, etc. etc.

44.2.4 Virtual Strain Energy

Now the structure has been idealized as an assemblage of elements, the integral form of the virtual work Eq. (44.8) can be rewritten as a summation of the virtual work done by the individual elements having volumes V_e and boundary surfaces S_e like

$$\sum_{e=1}^{n_e} \int_{V_e} \delta \boldsymbol{\varepsilon}^T \boldsymbol{\sigma} dV = \sum_{e=1}^{n_e} \int_{V_e} \mathbf{u}^T \mathbf{g}_e dV + \sum_{e=1}^{n_e} \int_{S_e} \mathbf{u}^T \mathbf{t}_e dS \quad (44.23)$$

where n_e is the total number of elements, \mathbf{g}_e is the element body force per unit volume and \mathbf{t}_e are the *element tractions*² per unit area acting along the element boundary S_e . For each element, its surface boundary S_e can be separated in an exterior part and an interior part having imaginary interfaces with adjacent elements.

²Here the term 'tractions' is not limited to an actual boundary surface, but is also used for the interior surfaces.

Eq. (44.23) is of fundamental importance for the displacement based Finite Element Method and imposes some restrictions on the displacement functions. In the ‘Principle of Virtual Displacements’ finite element approximation, we will attempt to ensure equilibrium, which for an element looks like

$$\int_{V_e} \delta \mathbf{u}^T (\mathbf{L}^T \boldsymbol{\sigma} + \mathbf{g}_e) dV - \int_{S_e} (\mathbf{L}_n^T \boldsymbol{\sigma} - \mathbf{t}_e) dS = 0 \quad (44.24)$$

It can be proved that this theorem is only valid provided that all derivatives of \mathbf{u} and $\boldsymbol{\sigma}$ are *finite* through V . In general the stresses do not achieve continuity across the element interfaces. However, if the shape functions are chosen such that the displacements match at the nodes and the adjacent elements (i and j) have identical displacements at their interface, then a continuity condition on the stresses *in the mean* is met in the form of

$$\int_{S_{e_i+j}} \delta \mathbf{u}^T (\mathbf{L}_n^T \boldsymbol{\sigma}_i - \mathbf{L}_n^T \boldsymbol{\sigma}_j - \bar{\mathbf{t}}_e) dS = 0 \quad (44.25)$$

where $\bar{\mathbf{t}}_e$ is the contribution of applied external loads. This expression is another approximation of satisfying equilibrium and therefore the equilibrium equation Eq. (44.24) is true *within* a single element and *up to* its surface boundary S_e . Assuming that the displacement functions satisfy the conditions of Eq. (44.24), the integrations may now be performed over the element volumes and surfaces. Substituting for the element displacements and strains respectively Eqs. (44.16) and (44.18), the virtual work equation for an individual element can now be written as

$$\delta \mathbf{u}_e^T \int_{V_e} \mathbf{B}^T \boldsymbol{\sigma} dV = \delta \mathbf{u}_e^T \int_{V_e} \mathbf{N}^T \mathbf{g}_e dV + \delta \mathbf{u}_e^T \int_{S_e} \mathbf{N}^T \mathbf{t}_e dS \quad (44.26)$$

The integral form of the element boundary tractions $\int \mathbf{N}^T \mathbf{t}_e dS$ can be replaced by a kinematically equivalent nodal force vector \mathbf{r}_e corresponding with the element degrees of freedom vector \mathbf{u}_e . Reordering and substituting \mathbf{r}_e for the boundary tractions, the virtual work equation can now be expressed in a form

$$\delta \mathbf{u}_e^T \left(\int_{V_e} \mathbf{B}^T \boldsymbol{\sigma} dV - \int_{V_e} \mathbf{N}^T \mathbf{g}_e dV \right) = \delta \mathbf{u}_e^T \mathbf{r}_e \quad (44.27)$$

As this relation is valid for any virtual displacement $\delta \mathbf{u}_e$, the equilibrium equation for an element can be written as

$$\int_{V_e} \mathbf{B}^T \boldsymbol{\sigma} dV - \int_{V_e} \mathbf{N}^T \mathbf{g}_e dV = \mathbf{r}_e \quad (44.28)$$

44.2.5 Element Stiffness Matrix

Eq. (44.28) is valid for any stress-strain relation and in case of a linear elastic behaviour, substituting Eq. (44.19) for the stresses yields

$$\mathbf{K}_e \mathbf{u}_e + \mathbf{f}_e = \mathbf{r}_e \quad (44.29)$$

where

$$\mathbf{K}_e = \int_{V_e} \mathbf{B}^T \mathbf{D} \mathbf{B} dV \quad (44.30)$$

is the element stiffness matrix and

$$\mathbf{f}_e = \int_{V_e} \mathbf{N}^T \mathbf{g}_e dV - \int_{V_e} \mathbf{B}^T \mathbf{D} \boldsymbol{\varepsilon}_0 dV + \int_{V_e} \mathbf{B}^T \boldsymbol{\sigma}_0 dV \quad (44.31)$$

is the element contribution to the right hand side vector \mathbf{f} .

Going back to Eq. (44.23) and using the piecewise approximation for the displacements Eq. (44.16) and the discrete strain-displacement relation Eq. (44.18), the virtual work equation is now obtained by

$$\sum_{e=1}^{n_e} \delta \mathbf{u}_e^T \int_{V_e} \mathbf{B}^T \boldsymbol{\sigma} dV = \sum_{e=1}^{n_e} \delta \mathbf{u}_e^T \int_{V_e} \mathbf{N}^T \mathbf{g}_e dV + \sum_{e=1}^{n_e} \delta \mathbf{u}_e^T \int_{S_e} \mathbf{N}^T \mathbf{t}_e dS \quad (44.32)$$

Substitution of the stress-strain relation Eq. (44.19) in case of linear elastic behaviour yields

$$\begin{aligned} \sum_{e=1}^{n_e} \delta \mathbf{u}_e^T \left(\int_{V_e} \mathbf{B}^T \mathbf{D} \mathbf{B} dV \right) \mathbf{u}_e = \\ \sum_{e=1}^{n_e} \delta \mathbf{u}_e^T \int_{V_e} \mathbf{N}^T \mathbf{g}_e dV + \sum_{e=1}^{n_e} \delta \mathbf{u}_e^T \int_{S_{t_e}} \mathbf{N}^T \mathbf{t}_e dS + \\ \sum_{e=1}^{n_e} \delta \mathbf{u}_e^T \left(\int_{V_e} \mathbf{B}^T \mathbf{D} \boldsymbol{\varepsilon}_0 dV - \int_{V_e} \mathbf{B}^T \boldsymbol{\sigma}_0 dV \right) \end{aligned} \quad (44.33)$$

44.2.5.1 Transformation

With \mathbf{T} for the transformation matrix Eq. (44.30) is written as

$$\mathbf{K}_e = \mathbf{T}^T \left(\int_{V_e} \mathbf{B}^T \mathbf{D} \mathbf{B} dV \right) \mathbf{T} \quad (44.34)$$

The pre- and post-multiplication with \mathbf{T} transforms the element stiffness from local to global coordinates. The creation of element stiffness matrices \mathbf{K}_e according to Eq. (44.34) is performed in DIANA's Module LINSTA due to the **MATRIX** command [§ 4.1 p. 72].

44.3 Assembling the Load Vector

The load vector \mathbf{f} is composed of the *external nodal forces* as specified in the input file and of the assembly of the *element loads*. These element loads can be subdivided into the following components

1. Equivalent nodal forces due to thermal effects, effects resulting from difference in concentration and initial strains. Summing these effects results in an equivalent initial strain, which can be transformed to nodal loads.
2. Equivalent nodal forces resulting from initial stresses.
3. Equivalent nodal forces resulting from loads on element boundaries.
4. Equivalent nodal forces resulting from acceleration effects (dead weight).

Due to the **LOADS** command [Ch. 4.1 p. 72], DIANA's Module LINSTA calculates the above contributions per element and after that, for each degree of freedom the contribution of the connected elements are superposed and added to the external nodal point loads thus forming the load vector \mathbf{f} .

44.4 Equilibrium

Invoking the theorem of the virtual displacements, the equilibrium equations of the element assemblage are

$$\mathbf{K} \mathbf{u} = \mathbf{f} \quad (44.35)$$

where the matrix \mathbf{K} is the stiffness matrix of the element assemblage

$$\mathbf{K} = \sum_{e=1}^{n_e} \mathbf{T}_e^T \mathbf{K}_e \mathbf{T}_e \quad (44.36)$$

and the vector \mathbf{f} is the right hand side vector

$$\mathbf{f} = \mathbf{f}_g + \mathbf{f}_t + \mathbf{f}_{\varepsilon_0} - \mathbf{f}_{\sigma_0} + \mathbf{f}_c \quad (44.37)$$

with

$\mathbf{f}_g = \sum_{e=1}^{n_e} \mathbf{T}_e^T \int_{V_e} \mathbf{N}^T \mathbf{g}_e dV$	the contribution of the element body forces.
$\mathbf{f}_t = \sum_{e=1}^{n_e} \mathbf{T}_e^T \int_{S_e} \mathbf{N}^T \mathbf{t}_e dS$	the contribution of the element surface tractions.
$\mathbf{f}_{\varepsilon_0} = \sum_{e=1}^{n_e} \mathbf{T}_e^T \int_{V_e} \mathbf{B}^T \mathbf{D} \varepsilon_0 dV$	the effect of the element initial strains.
$\mathbf{f}_{\sigma_0} = \sum_{e=1}^{n_e} \mathbf{T}_e^T \int_{V_e} \mathbf{B}^T \sigma_0 dV$	the effect of the element initial stresses.
\mathbf{f}_c	the contribution of the concentrated nodal forces.

Chapter 45

Solution Procedures of Sparse Linear Systems

Once the equations have been determined for each element, they are assembled by addition of the terms defining the stiffness relation between each pair of nodal freedoms. In this way the stiffness matrix \mathbf{K} is composed, in which the term at row i and column j represents the total stiffness from freedom i to j . Only if freedom i and j are immediate neighbors, this term is non-zero.

Next the constraint degrees of freedom are to be substituted in the system matrix. This might be expressed as

$$\begin{bmatrix} \mathbf{K}_{uu} & \mathbf{K}_{uc}^T \\ \mathbf{K}_{uc} & \mathbf{K}_{cc} \end{bmatrix} \begin{Bmatrix} \mathbf{u}_u \\ \mathbf{u}_c \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}_u \\ \mathbf{f}_c \end{Bmatrix} \quad (45.1)$$

Here subscript $_u$ is associated with the *unconstrained* degrees of freedom and subscript $_c$ with the *constrained* degrees of freedom. Eq. (45.1) can be reduced to

$$\mathbf{K}_{uu}\mathbf{u}_u = \mathbf{f}_u - \mathbf{K}_{uc}\mathbf{u}_c \quad \text{or} \quad \hat{\mathbf{K}}\hat{\mathbf{u}} = \hat{\mathbf{f}} \quad (45.2)$$

Now all boundary conditions, tyings and loads have been treated, the remaining system of equations can be solved. For convenience we write this equation in a more simple form

$$\boxed{\mathbf{K}\mathbf{u} = \mathbf{f}} \quad (45.3)$$

The solution of \mathbf{u} from this system of equations is often the most computation intensive part of a large-scale Finite Element Analysis. Generally matrix \mathbf{K} is a known sparse $n \times n$ matrix¹ with a symmetric structure, \mathbf{f} is a known right-hand-side vector and \mathbf{u} the unknown solution vector to be computed.

The solution of the system of equations Eq. (45.3) is done via the **SOLVE** commands [Ch. 30]. Basically there are two methods: *direct solution* and indirect or *iterative solution*. The sequel of this chapter first gives an overview of sparse matrix terminology, followed by a description of the available solution methods.

45.1 Direct Solution Methods

A general approach to solve the linear system of Eq. (45.3) is first to *factorize* \mathbf{K} :

$$\mathbf{K} \Rightarrow \mathbf{L}\mathbf{U} \quad (45.4)$$

$$\mathbf{K} \Rightarrow \mathbf{L}\mathbf{D}\mathbf{U} \quad (45.5)$$

where \mathbf{L} , \mathbf{U} and \mathbf{D} are lower, upper and diagonal matrices respectively. The process of Eq. (45.5) is known as *LDU decomposition*. After the factorization, the solution vector \mathbf{u}

¹ *sparse* = ‘with many zero-terms’.

can be computed by successive forward and backward substitution to solve the triangular system of equations Eq. (45.3)

$$\mathbf{L} \mathbf{w} = \mathbf{f} \quad \mathbf{D} \mathbf{U} \mathbf{u} = \mathbf{w} \quad (45.6)$$

Under the assumption that \mathbf{K} is a positive definite symmetric matrix the decomposition becomes $\mathbf{L}\mathbf{L}^T$ or $\mathbf{L}\mathbf{D}\mathbf{L}^T$.

45.1.1 Factorization and Fill-in

If \mathbf{K} is a sparse matrix then generally fill-in occurs during the factorization and the factor \mathbf{L} will be more dense than the original matrix \mathbf{K} . Fill-in is caused in the Gaussian elimination if in the operation

$$k_{ij} \leftarrow k_{ij} - k_{ik} \frac{k_{kj}}{k_{kk}} \quad (45.7)$$

the entry in location (i, j) of the original matrix was zero. It has been proved, see George & Liu [33], that the number of operations to compute the triangular factor \mathbf{L} of a symmetric matrix \mathbf{K} is equal to

$$\frac{1}{2} \sum_{i=1}^{N-1} (\eta(L(*, i)) - 1) \times (\eta(L(*, i)) + 2) \quad (45.8)$$

where $L(*, i)$ denotes the i -th column in the triangular factor \mathbf{L} and $\eta(L(*, i))$ is the number of non-zero terms, including the fill-ins, in column i .

However, to take advantage of the sparseness of the coefficient matrix \mathbf{K} , the equations must be arranged in a special order. To find a good ordering for a sparse symmetric matrix \mathbf{K} we have to determine a permutation \mathbf{P} that minimizes the fill-ins in the factor \mathbf{L} of \mathbf{PKP}^T . DIANA uses the ordering Metis ordering algorithm proposed by Karypis & Kumar [50]. If \mathbf{K} is a symmetric positive definite matrix, pivoting is not required during factorization to maintain stability. Therefore, once the order is known, the non-zero structure of the factor \mathbf{L} can be predicted in advance.

The solution of Eq. (45.3) involves the following steps.

1. *Ordering.* Find a suitable ordering \mathbf{P} for \mathbf{K} , i.e., determine a permutation matrix \mathbf{P} such that the fill of the factor \mathbf{L} of \mathbf{PKP}^T is minimized.
2. *Symbolic factorization.* Determine the non-zero terms in \mathbf{L} of \mathbf{PKP}^T and setup the corresponding data structure to store the lower half of \mathbf{PKP}^T and the factor \mathbf{L} .
3. *Numerical factorization.* Assemble the terms of \mathbf{K} into the data structure and compute the factor \mathbf{L} of \mathbf{PKP}^T .
4. *Triangular solution.* Solve $\mathbf{L}\mathbf{y} = \mathbf{P}\mathbf{f}$ and $\mathbf{D}\mathbf{L}^T\mathbf{x} = \mathbf{y}$ and then set $\mathbf{u} = \mathbf{P}^T\mathbf{x}$.

The first two steps are symbolic operations and have to be performed only once. Newton–Raphson solution methods for nonlinear systems of equations [§ 46.1.1.1 p. 553] are usually related to the following iteration scheme:

```
iterate:  for  $k := 0, 1, 2, \dots$  do
solve:     $\mathbf{K}(\mathbf{u}_k) \delta \mathbf{u}_{k+1} = \delta \mathbf{f}_k$ 
set:       $\mathbf{u}_{k+1} = \mathbf{u}_k + \delta \mathbf{u}_{k+1}$ 
          end do
```

Only step three and four have to be performed for each iteration.

45.1.2 Sparse Cholesky Method

The Sparse Cholesky direct solver in DIANA basically is an implementation of the inner product formulation of the Cholesky factorization, see George & Liu [33]. This algorithm fully uses the nonzero pattern of the system matrix at the expense of introducing indirectness of the inner loops. Prior to the factorization, the Sparse Cholesky solver reorders the system matrix according to the Metis algorithm, see Karypis & Kumar [50]. This reordering minimizes the fill-in of the matrix during factorization. Furthermore, to minimize the indirect access to arrays, the implementation applies so-called i-nodes, i.e., zones in the matrix with identical sparsity patterns.

The code of the Sparse Cholesky solver is optimized for cache based memory access and achieves a performance which in most cases is superior to the Generalized Element method solver. However, the Sparse Cholesky solver is an in-core implementation and generally needs more primary memory than the out-of-core Generalized Element solver.

45.1.3 PARDISO – Parallel Direct Sparse Solver

Intel Math Kernel Library (Intel MKL) provides a direct sparse solver PARDISO which can be used for solving real symmetric and structurally symmetric sparse linear systems of equations.

The PARDISO solver shows both a high performance and memory efficient usage for solving large sparse symmetric and unsymmetric linear systems of equations by shared multiprocessors. The solver uses a combination of left- and right-looking supernode techniques, see Schenk [79, 80]. For sufficiently large problems, the scalability of the parallel algorithm is nearly independent of the shared-memory multiprocessing architecture and a speedup of up to five using eight processors has been observed.

45.2 Eigenvalue Procedures

This section briefly summarizes the eigenproblems that can be solved and does not describe the actual solution procedure. We will emphasize on the *standard eigenproblem*, the *generalized eigenproblem* and on *shifting* of eigenvalues. Some attention will be paid to the analysis results as eigenvalues and -vectors, their accuracy and the generalized mass, modal damping factors, participation factors, and participation vectors.

45.2.1 Standard Eigenproblem

The simplest problem is the standard eigenproblem

$$\mathbf{K}\phi = \lambda\phi \quad (45.9)$$

where $\mathbf{K}_{n \times n}$ is the symmetric stiffness matrix of the finite element model.² There are n eigenvalues and corresponding eigenvectors satisfying Eq. (45.9). The i -th eigenpair is marked as (λ_i, ϕ_i) if the eigenvalues are ordered increasingly:

$$\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_{n-1} \leq \lambda_n \quad (45.10)$$

The solution for p eigenpairs can be written as

$$\mathbf{K}\Phi = \Phi\Lambda \quad (45.11)$$

where $\Phi_{n \times p}$ is the matrix with eigenvectors and $\Lambda_{p \times p}$ is a diagonal matrix with the corresponding eigenvalues.

²The dimension of the stiffness matrix \mathbf{K} is defined by the number of independent equations n and is denoted as order n .

45.2.2 Generalized Eigenproblem

A very frequently occurring eigenproblem is the *free vibration* equation to be solved in the mode superposition method (see also Chapter 48) written as

$$\mathbf{K}\phi = \omega^2 \mathbf{M}\phi \quad (45.12)$$

where \mathbf{K} is the symmetric stiffness matrix and \mathbf{M} is the mass matrix of the finite element model, ω is the circular natural frequency in radians per second. Eigenvalue λ is equal to ω^2 . The eigenvector is the corresponding mode shape vector ϕ . The mass matrix \mathbf{M} can be obtained in a *consistent mass* analysis or in a *lumped mass* analysis. Analogous to Eq. (45.11) the solution for p natural frequencies squared and the corresponding mode shape vectors of Eq. (45.12) can be written as

$$\mathbf{K}\Phi = \mathbf{M}\Phi\Omega^2 \quad (45.13)$$

Two other possible generalized eigenproblems can be encountered in *stability* analysis and *heat flow* analysis. In stability analysis the problem is solved by linearized buckling analysis with

$$\mathbf{K}\phi = \lambda \mathbf{K}_G \phi \quad (45.14)$$

where \mathbf{K} is the stiffness matrix and \mathbf{K}_G the geometric stress-stiffness matrix [Ch. 54]. In heat flow analysis the equation is

$$\mathbf{K}\phi = \lambda \mathbf{C}\phi \quad (45.15)$$

where \mathbf{K} is the heat conductivity matrix and \mathbf{C} the heat capacity matrix [Ch. 55]. From now on, the eigenproblem will be discussed as

$$\mathbf{K}\phi = \omega^2 \mathbf{M}\phi \quad (45.16)$$

from Eq. (45.12). However, the analysis description is also applicable to the solution of the other problems. For example, to solve the standard eigenproblem in Eq. (45.9), the mass matrix must be replaced by an identity matrix \mathbf{I} and in case of a linearized buckling analysis Eq. (45.14) the mass matrix \mathbf{M} must be replaced by the geometric stress-stiffness matrix \mathbf{K}_G .

45.2.2.1 Generalized Mass

For each calculated frequency f_i , DIANA determines the corresponding generalized mass m_{ii} by

$$m_{ii} = \phi_i^T \mathbf{M} \phi_i \quad (45.17)$$

With the eigenvectors ϕ_i normalized such that

$$m_{ii} = 1 \quad (45.18)$$

45.2.2.2 Participation Factor

For each calculated frequency f_i , DIANA determines the corresponding participation factor by

$$\gamma_i = \frac{\phi_i^T \mathbf{M} \mathbf{i}}{m_{ii}} \quad (45.19)$$

Where \mathbf{i} is the unity vector, i.e., a vector with a unity displacement for each degree of freedom and the eigenvectors ϕ normalized according to Eq. (45.18).

45.2.2.3 Participation Vector

For each calculated frequency f_i , DIANA determines the corresponding participation vector by

$$\phi_{p.i} = \gamma_i \phi_i \quad (45.20)$$

The sum of all possible participation vectors of a structure will give the unity vector, i.e., a unity displacement for each degree of freedom:

$$\sum_{i=1}^n \gamma_i \phi_i = \mathbf{i} \quad (45.21)$$

45.2.2.4 Direction Dependent Participation Factor

For each calculated frequency f_i DIANA determines the corresponding direction dependent participation factors Γ_i for the translational and rotational degrees of freedom in global X , Y , and Z direction.

$$\begin{aligned} \Gamma_{t_{Xi}} &= \frac{l_{t_{Xi}}}{m_{ii}} \\ \Gamma_{t_{Yi}} &= \frac{l_{t_{Yi}}}{m_{ii}} \\ \Gamma_{t_{Zi}} &= \frac{l_{t_{Zi}}}{m_{ii}} \\ \Gamma_{r_{Xi}} &= \frac{l_{r_{Xi}}}{m_{ii}} \\ \Gamma_{r_{Yi}} &= \frac{l_{r_{Yi}}}{m_{ii}} \\ \Gamma_{r_{Zi}} &= \frac{l_{r_{Zi}}}{m_{ii}} \end{aligned} \quad (45.22)$$

Where \mathbf{l}_t are the coefficient vectors for each translational degree of freedom, \mathbf{l}_r are the coefficient vectors for each rotational degree of freedom according to

$$\mathbf{l}_i = \phi_i^T \mathbf{M} \mathbf{r} \quad (45.23)$$

Where \mathbf{r} is the influence vector which represents the displacements resulting from a static unit ground displacement in the direction of the corresponding translational or rotational degree of freedom. The eigenvectors ϕ are normalized according to Eq. (45.18).

45.2.2.5 Effective Mass

For each calculated frequency f_i DIANA determines the corresponding effective masses $m_{eff.i}$ for the translational degrees of freedom in global X , Y , and Z direction.

$$\begin{aligned} m_{eff.t_{X.i}} &= \frac{l_{t_{Xi}}^2}{m_{ii}} \\ m_{eff.t_{Y.i}} &= \frac{l_{t_{Yi}}^2}{m_{ii}} \\ m_{eff.t_{Z.i}} &= \frac{l_{t_{Zi}}^2}{m_{ii}} \end{aligned} \quad (45.24)$$

Where \mathbf{l}_t are the coefficient vectors according to Eq. (45.23).

45.2.2.6 Modal Mass

For each calculated frequency f_i , DIANA determines the corresponding modal mass $m_{mod.ii}$ by

$$m_{mod.ii} = \phi_i^T \mathbf{M} \phi_i \quad (45.25)$$

With the eigenvectors ϕ_i normalized such that the largest translation displacement component has a value of 1. Note that if no translation displacement component exists, the modal mass will be set zero for that frequency.

45.2.2.7 Equivalent Mass

For each calculated frequency f_i DIANA determines the corresponding direction dependent equivalent masses $m_{eq,i}$ for the translational degrees of freedom in global X , Y , and Z direction.

$$\begin{aligned} m_{eq_X i} &= \phi_i^T \mathbf{M} \mathbf{r}_X \\ m_{eq_Y i} &= \phi_i^T \mathbf{M} \mathbf{r}_Y \\ m_{eq_Z i} &= \phi_i^T \mathbf{M} \mathbf{r}_Z \end{aligned} \quad (45.26)$$

Where \mathbf{r} is the influence vector which represents the displacements resulting from a static unit ground displacement in the direction of the corresponding translational or rotational degree of freedom. With the eigenvectors ϕ_i normalized such that the largest translation displacement component has a value of 1. Note that if no translation displacement component exists, the equivalent mass will be set zero for that frequency.

45.2.2.8 Transformation Factors

For each calculated frequency f_i , DIANA determines the corresponding direction dependent transformation factors $\Gamma_{tr,i}$ by

$$\begin{aligned} \Gamma_{tr_X i} &= \frac{m_{eq_X i}}{m_{mod,ii}} \\ \Gamma_{tr_Y i} &= \frac{m_{eq_Y i}}{m_{mod,ii}} \\ \Gamma_{tr_Z i} &= \frac{m_{eq_Z i}}{m_{mod,ii}} \end{aligned} \quad (45.27)$$

Where $m_{eq,i}$ is the equivalent mass and $m_{mod,ii}$ is the modal mass. Note that if no translation displacement component exists, the transformation factor will be set zero for that frequency.

45.2.2.9 Modal Damping Factor

When you have specified an element damping factor h_e based on strain energy, then DIANA will determine the modal damping factors h_i for each calculated frequency f_i according to

$$h_i = \frac{\sum_{j=1}^n h_{e,j} \phi_i^T \mathbf{K}_j \phi_i}{\sum_{j=1}^n \phi_i^T \mathbf{K}_j \phi_i} \quad (45.28)$$

The element damping factor h_e indicates the proportion of the element strain energy that has to be used as damping. The modal damping factors h_i are useful to determine the two modes that should be employed to calculate the Rayleigh damping coefficients.

45.2.3 Shifting

In an eigenvalue problem, a user-specified shift factor μ [§ 31.3 p. 432] on the stiffness matrix \mathbf{K} causes Module EIGEN to provide the eigenvalues and -modes close to $-\mu$ as first. There can be several reasons to apply a shift to an eigenvalue problem:

- *Zero/negative eigenvalues.* Some solution methods are not designed explicitly to calculate zero or negative eigenvalues. Shifting the stiffness matrix \mathbf{K} may be the solution in these situations.

- *Softening.* In case of softening material behaviour, only the first negative eigenvalue is of interest [Ch. 31 p. 427]. Shifting the stiffness matrix \mathbf{K} may help you to find this eigenvalue and -mode.
- *Perturbation analysis.* In case of perturbation analysis [§ 54.3 p. 621], where the nonlinear interaction of eigenmodes is considered, the accuracy of the interacting eigenvalues and -modes can be improved by a shift of the stiffness matrix \mathbf{K} .

Application of a shift μ reformulates the eigenvalue problem. For positive μ the eigenproblem formulation becomes

$$\hat{\mathbf{K}}\phi = \hat{\omega}^2 \mathbf{M}\phi \quad (45.29)$$

$$\hat{\mathbf{K}} = \mathbf{K} + \mu \mathbf{M} \quad (45.30)$$

$$\hat{\omega}^2 = \omega^2 + \mu \quad (45.31)$$

in which $\hat{\mathbf{K}}$ is the modified stiffness matrix, the eigenvalue $\lambda \equiv \omega^2$ is related by Eq. (45.31) and the eigenvector ϕ is unchanged.

45.2.4 Eigenvalues and Frequencies

The calculated natural circular frequencies ω of the generalized eigenproblem of Eq. (45.12) are expressed in radians per second. The corresponding period T may be computed with

$$\omega T = 2\pi \quad \therefore \quad T = \frac{2\pi}{\omega} \quad (45.32)$$

The cyclic frequency f , which is usually referred to as the *frequency of motion*, is the reciprocal of the period T and is given by

$$f = \frac{1}{T} = \frac{\omega}{2\pi} \quad (45.33)$$

and expressed in hertz, whereby one hertz is one cycle per second (CPS). For the standard eigenproblem as given in Eq. (45.9), the eigenvalue λ can be obtained by taking the frequency ω squared

$$\lambda \equiv \omega^2 = 4\pi^2 f^2 \quad (45.34)$$

The quantities ω and λ can be determined from relation Eq. (45.33) and Eq. (45.34) respectively.

Shifted system. If the eigenproblem has been shifted by a factor μ according to Eq. (45.30), DIANA corrects the actual frequencies according to Eq. (45.31) with

$$\omega = (\hat{\omega}^2 - \mu)^{\frac{1}{2}} = \left(4\pi^2 \hat{f}^2 - \mu\right)^{\frac{1}{2}} \quad (45.35)$$

and the eigenvalues with

$$\lambda = (\hat{\omega}^2 - \mu) = \left(4\pi^2 \hat{f}^2 - \mu\right) \quad (45.36)$$

where $\hat{\omega}$ and \hat{f} are the shifted frequencies and ω and f are the output frequencies.

45.2.5 Accuracy

The error measure for the calculated eigenvalue and eigenvector approximation $\tilde{\lambda}$ and $\tilde{\phi}$ is determined as

$$\epsilon_i = \frac{\left\| \mathbf{K}\tilde{\phi}_i - \tilde{\lambda}_i \mathbf{M}\phi_i \right\|_2}{\left\| \mathbf{K}\tilde{\phi}_i \right\|_2} \quad (45.37)$$

with ϵ_i as relative error for the i th eigenpair. This quantity should be small if $\tilde{\lambda}_i$ and $\tilde{\phi}_i$ are an accurate solution of an eigenpair.

45.2.6 Solution Techniques

DIANA applies an implicitly restarted Arnoldi method to solve the eigenvalue problem, see Lehoucq et al. [55].

45.3 Iterative Solution Methods

The direct solution method has some drawbacks. The most important one is that the background storage requirements can be extremely high for large three-dimensional problems. Another disadvantage is that the Gauss decomposition without pivoting (i.e., interchanging rows and columns) is not numerically stable if the stiffness matrix is not positive definite. For these reasons two iterative methods are available in DIANA as alternatives for the direct solution method.

The common idea of all iterative methods for solving the linear system of equations $\mathbf{Ku} = \mathbf{f}$ is to generate a sequence of approximations \mathbf{u}_i to the solution vector \mathbf{u} via the recursion

$$\mathbf{u}_{i+1} = \mathbf{u}_i + \gamma_i \mathbf{P}^{-1}(\mathbf{f} - \mathbf{Ku}_i) \quad (45.38)$$

in which \mathbf{P}^{-1} is the preconditioning matrix or the preconditioner. In some way, \mathbf{P}^{-1} should resemble the inverse of the stiffness matrix \mathbf{K} and it should be easy to solve the linear system $\mathbf{Px} = \mathbf{y}$.

The performance of iterative solution methods is mainly determined by its convergence speed. For symmetric positive definite matrices, the convergence significantly depends on the so-called condition number of the (preconditioned) system of equations, which is for symmetric positive definite matrices equal to

$$\kappa = \frac{\lambda_{\max}}{\lambda_{\min}} \quad (45.39)$$

For nonsymmetric or indefinite matrices, the convergence depends on the eigenvalues *and* the eigenvectors of the stiffness matrix. If the stiffness matrix is symmetrical dominantly, then the convergence mainly depends on the condition number as well.

The condition number is always larger or equal to one and a small condition number is favorable. In practice, badly shaped elements, large stiffness jumps in the model, and mixture or shell elements can result in a large condition number. A popular technique to improve the condition number is to apply preconditioning [§ 45.3.4 p. 548].

Many algorithms exist that compute the iteration parameters [76]. The so-called Krylov subspace methods have proved to be efficient and the idea is to build an orthogonal subspace of \mathbf{K} . Two strategies are popular: orthogonalizing the residuals and minimizing the residuals over an increasing subspace. The orthogonalization procedure can be performed by the Lanczos method for symmetric matrices and the Arnoldi method for nonsymmetric matrices. Figure 45.1 presents four Krylov subspace methods, classified on symmetry and strategy. The most popular choices are Conjugate Gradient (CG) and Generalized Minimal Residual (GMRES). DIANA uses CG for symmetric matrices and GMRES for nonsymmetric matrices.

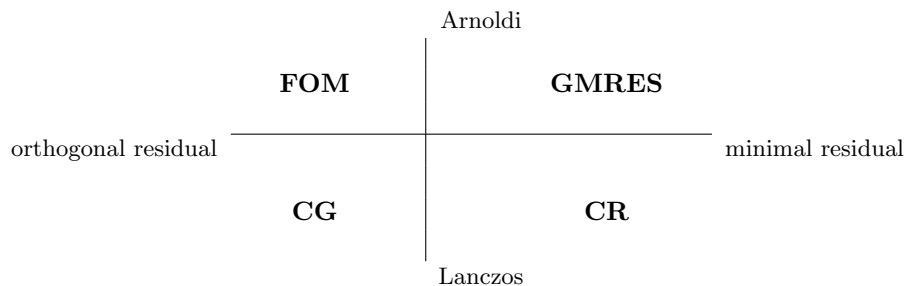


Figure 45.1: Krylov-based iterative solution methods.

45.3.1 Conjugate Gradient

The Conjugate Gradient method, see Hestenes & Stiefel [42], is currently the most popular and probably the best iterative method for systems with a symmetric positive definite stiffness matrix, for example all linear elastic problems. The CG algorithm is a Krylov subspace based method that generates γ_i such that all residuals \mathbf{r}_i are perpendicular. By making clever use of the symmetry of \mathbf{K} it is possible to orthogonalize the residual \mathbf{r}_i against all previous residuals by making use of *only* the residuals of the two previous iterations.

If the stiffness matrix is nonsymmetric or indefinite, i.e., has negative eigenvalues, the Conjugate Gradient method need not converge.

The convergence of Conjugate Gradient depends on the condition number κ of the stiffness matrix as in Eq. (45.39). The following bound on the error $\mathbf{u} - \mathbf{u}_i$ at iteration i of Conjugate Gradient is well-known [76]:

$$\|\mathbf{u} - \mathbf{u}_i\|_{\mathbf{K}} \leq 2 \left[\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right]^i \|\mathbf{u} - \mathbf{u}_0\|_{\mathbf{K}} \quad (45.40)$$

It follows that the ratio between the largest and smallest eigenvalue should be small for CG to converge fast.

45.3.2 Generalized Minimal Residual

The GMRES method, see Saad & Schultz [77], is a Krylov subspace based method that converges even if the stiffness matrix is not positive definite, for instance if the stiffness matrix is nonsymmetric. The iteration parameters are computed by minimizing the residual over an increasing subspace of \mathbf{K} . In every iteration a Krylov vector is orthogonalized with respect to *all* previous Krylov vectors. Moreover, the number of computations per iteration increases since the orthogonalization process becomes more expensive every iteration. Therefore the iteration is restarted after a fixed number of Krylov vectors has been added to the basis.

The convergence of the GMRES method depends on the eigenvalues *and* the eigenvectors of the stiffness matrix. If the matrix is symmetrical dominantly, then the convergence mainly depends on the condition number of \mathbf{K} .

45.3.3 Schwarz Domain Decomposition

The purpose of Schwarz domain decomposition is to divide the domain into a number of subdomains for parallel processing. The DIANA user can specify the number of available threads when using the iterative solver. This number is equal to the number of subdomains the parallel iterative solver will use. The subdomains are partitioned using METIS software.

The Schwarz domain decomposition mainly aims at shared memory computers. All parallel tasks execute the same sequential problem, which classifies this implementation as ‘Single Program, Multiple Data’ (SPMD). The software library is customized and is based on OpenMP.

The boolean left and right restriction operators of the i -th subdomain will be defined [57]: The left restriction operator \mathbf{L}_i correspond to the internal degrees of freedom of the i -th subdomain, while the right restriction operator \mathbf{R}_i corresponds to internal and interface degrees of freedom of the i -th subdomain. The non-zero columns of $(\mathbf{R}_i - \mathbf{L}_i)$ indicate the interface degrees of freedom of the i -th subdomain. The stiffness matrix \mathbf{K} can be expressed in terms of the subdomain matrices \mathbf{K}_i (which may be overlapping) by

$$\mathbf{K} = \sum_{i=1}^n \mathbf{L}_i^T \mathbf{K}_i \mathbf{R}_i \quad (45.41)$$

The right restriction operator can be used to extract subdomain matrices and vectors from the global matrices and vectors:

$$\begin{aligned}\mathbf{K}_i &= \mathbf{R}_i \mathbf{K} \mathbf{R}_i^T \\ \mathbf{x}_i &= \mathbf{R}_i \mathbf{x}\end{aligned}\tag{45.42}$$

The left operator can be used to assemble the local vectors into a global vector, which is useful for inner products:

$$\mathbf{x} = \sum_{i=1}^n \mathbf{L}_i^T \mathbf{x}_i\tag{45.43}$$

The parallel computations in Schwarz domain decomposition involve different types of data exchange, which involve multiplications with the left and right restrictors [57, 78].

45.3.4 Preconditioning

Preconditioning is a technique to improve the convergence of iterative solution methods. Implicitly, the system of equations $\mathbf{K}\mathbf{u} = \mathbf{f}$ is multiplied by a preconditioner \mathbf{P}^{-1} . If the multiplication is from the left, then the iterative solution method needs to solve the following preconditioned system of equations:

$$\mathbf{P}^{-1} \mathbf{K} \mathbf{u} = \mathbf{P}^{-1} \mathbf{f}\tag{45.44}$$

The idea of preconditioning is that this newly acquired system of equations has better convergence properties, such as a smaller condition number. A preconditioner can be applied centrally, from the left as in Eq. (45.44), and from the right.

45.3.4.1 Incomplete LU-decomposition

By default DIANA applies *Incomplete LU-decomposition preconditioning*, generally known as ILU preconditioning, see Meijerink & Van Der Vorst [58]. The idea of ILU preconditioning is to approximate the system matrix \mathbf{K} by the product of a lower diagonal matrix \mathbf{L} and an upper matrix \mathbf{U}

$$\mathbf{P} = \mathbf{L} \cdot \mathbf{U} \approx \mathbf{K}\tag{45.45}$$

If the factorization is carried out exactly, we get a direct solution method. The disadvantage of the exact factorization is that fill-in occurs: the matrices \mathbf{L} and \mathbf{U} contain far more non-zero entries than the original matrix \mathbf{K} . In the ILU approximation we try to restrict the fill-in of \mathbf{L} and \mathbf{U} . First we limit the fill to the sparsity pattern of \mathbf{K} , i.e., $\mathbf{L}_{ij} \neq 0$ and $\mathbf{U}_{ij} \neq 0$ only if $\mathbf{K}_{ij} \neq 0$. The ILU decomposition is uniquely defined by

$$\mathbf{P}_{ij} = \sum_k \mathbf{L}_{ik} \cdot \mathbf{U}_{kj} \quad \text{if } \mathbf{K}_{ij} \neq 0\tag{45.46}$$

If the setup of \mathbf{P} fails, or if the subsequent iteration does not converge, we improve the preconditioner by allowing more fill-in. Therefore we use a drop-tolerance strategy: non-zero elements are only included in the incomplete factors if they are larger than a given threshold parameter (ILUT, see Saad [75]). This threshold parameter is determined adaptively: we decrease it until the iteration has converged. We notice that we obtain the exact factorization if the drop tolerance is small enough.

45.3.4.2 Jacobi Preconditioning

The simplest preconditioning technique is to scale the stiffness matrix with a diagonal matrix. For problems with a diagonally dominant stiffness matrix, the preconditioner \mathbf{P} is chosen equal to $\text{diag}(\mathbf{K})$. This preconditioner is known as *Jacobi preconditioning* or *diagonal scaling*. It is cheap to compute and use in the iteration process, but on the other hand can be ineffective for ill-conditioned systems of equations.

45.3.5 Termination Criterion

The natural choice for the termination criterion of the iterative process is based on the reduction of the residual \mathbf{r}_i ,

$$\mathbf{r}_i = \mathbf{f} - \mathbf{K} \cdot \mathbf{u}_i \quad (45.47)$$

with \mathbf{u}_i the approximation of the solution \mathbf{u} after i iterations. The iteration is stopped if

$$\|\mathbf{r}_i\| < \epsilon \|\mathbf{f}\| \quad (45.48)$$

with ϵ a user-defined tolerance.

45.3.6 Some Remarks on Practical Use

The advantage of a direct solution method is that for a given system of equations it is possible to compute how many floating points operations are necessary to determine the solution vector. Unfortunately this is not true for iterative methods. The number of floating point operations needed depends upon the rate of convergence of the iteration process.

45.3.6.1 Reordering

For direct methods it is of utmost importance that the equations are reordered such that the bandwidth becomes minimal. The savings in both computer time and memory consumption can be huge, especially for three-dimensional problems. This is due to the fact that the stiffness matrix is decomposed.

The situation is different for iterative solvers. The diagonal preconditioning does not depend on the ordering of the equations. For the ILU preconditioning the efficiency does depend on the ordering, but the improvements are less consistent than in the case of a direct method. Extensive experimentation has indicated that ILU performs best with Metis ordering.

45.3.6.2 Convergence

The rate of convergence of the iterative method is hard to predict for a particular problem. However, there are some rules of thumb to give. The iterative solution method is best suited for the solution of the large sized linear systems that arise in three-dimensional simulations. Especially large potential flow problems can often be solved very efficiently. If the stiffness matrix is ill-conditioned, the rate of convergence will be very low. This is often the case if structural elements like beams, plates or shells are used in the model.

Another notorious example is if the model involves elements based on a penalty formulation. Ill-conditioning of the stiffness matrix may also be caused by improper or few boundary conditions. An ill-conditioned linear system is susceptible to rounding errors, these may even ruin the solution. This is also the case for the direct solution method, but more hidden and therefore even more dangerous.

45.4 Substructuring

Substructuring is a standard technique in Finite Element Analysis (see Bathe [4, § 8.2.4]). Substructuring is one of the two common domain decomposition methods [76, 18] and is also called the Schur Complement domain decomposition method. The basic idea is to treat a group of elements as a single substructure (superelement). The use of substructures is attractive in various cases. For example, if many elements in a nonlinear model behave linearly, these elements can be put in a substructure. The internal degrees of freedom in the substructure are then removed by *static condensation*.

Suppose that we have two substructures that only contain linear elements. The degrees of freedom in the substructures can then be divided in internal degrees of freedom and

interface degrees of freedom. After a proper reordering, the stiffness matrix of this system can be written as

$$\begin{bmatrix} \mathbf{A}_1 & \mathbf{0} & \mathbf{B}_1 \\ \mathbf{0} & \mathbf{A}_2 & \mathbf{B}_2 \\ \mathbf{B}_1^T & \mathbf{B}_2^T & \mathbf{C} \end{bmatrix} \quad (45.49)$$

where \mathbf{A}_1 and \mathbf{A}_2 are the submatrices representing the connectivity of the internal degrees of freedom of the substructure. The rectangular matrices \mathbf{B}_1 and \mathbf{B}_2 are the connectivity of the internal degrees of freedom and the interface degrees of freedom of the substructures. We explicitly assume that the stiffness matrices of the elements in a substructure are symmetric. Finally, \mathbf{C} represents the connectivity of the interface degrees of freedom and the degrees of freedom of elements not in any substructure. By construction, there is no connection between the internal degrees of freedom of the two substructures.

The matrix Eq. (45.49) can be factorized as follows

$$\begin{bmatrix} \mathbf{A}_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_2 & \mathbf{0} \\ \mathbf{B}_1^T & \mathbf{B}_2^T & \mathbf{C}^* \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{A}_1^{-1}\mathbf{B}_1 \\ \mathbf{0} & \mathbf{I} & \mathbf{A}_2^{-1}\mathbf{B}_2 \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \quad (45.50)$$

where \mathbf{I} is the identity matrix and \mathbf{C}^* the so-called Schur complement

$$\mathbf{C}^* = \mathbf{C} - \mathbf{B}_1^T \mathbf{A}_1^{-1} \mathbf{B}_1 - \mathbf{B}_2^T \mathbf{A}_2^{-1} \mathbf{B}_2 \quad (45.51)$$

The terms $\mathbf{B}^T \mathbf{A}^{-1} \mathbf{B}$ are nothing but the substructures after static condensation of the internal degrees of freedom. DIANA assumes that the matrix \mathbf{A} is symmetric positive definite, and first constructs its Cholesky factorization \mathbf{LL}^T . Then $\mathbf{B}^T \mathbf{A}^{-1} \mathbf{B}$ is efficiently computed as follows:

$$\mathbf{B}^T \mathbf{A}^{-1} \mathbf{B} = \mathbf{B}^T (\mathbf{LL}^T)^{-1} \mathbf{B} = (\mathbf{L}^{-1} \mathbf{B})^T \mathbf{L}^{-1} \mathbf{B} \quad (45.52)$$

After all substructures have been assembled, it remains to solve a system of the form

$$\mathbf{C}^* \mathbf{u} = \mathbf{f} \quad (45.53)$$

In DIANA the solution of this system can be obtained by factorization of \mathbf{C}^* , or by using an iterative solver.

A final word about the effectiveness of substructuring. Even if we assume that the cost of assembling the substructures is negligible (they can be reused as is), it is not always true that the factorization of \mathbf{C}^* is cheaper than the factorization of the full matrix. The matrix \mathbf{C}^* is not only smaller but also but also denser than the full matrix. Especially if the substructures are characterized by a high ratio of interface degrees of freedom to internal degrees of freedom, the use of substructures may be detrimental to the performance. In this case DIANA will give a warning message and turn off substructuring.

Chapter 46

Solution Procedures for Nonlinear Systems

46.1 Incremental-Iterative Solution

In nonlinear Finite Element Analysis the relation between a force vector and displacement vector is no longer linear. For several reasons, discussed in Volume *Material Library* and §46.2, the relation becomes nonlinear and the displacements often depend on the displacements at earlier stages, e.g. in case of plastic material behaviour. Just as with a linear analysis, we want to calculate a displacement vector that equilibrates the internal and external forces. In the linear case, the solution vector could be calculated right away but in the nonlinear case it cannot. To determine the state of equilibrium we not only make the problems discrete in space (with finite elements) but also in time (with increments). To achieve equilibrium at the end of the increment, we can use an *iterative* solution algorithm. The combination of both is called an *incremental-iterative* solution procedure.

In this chapter we will consider a vector of displacement increments that must yield an equilibrium between internal and external forces, and a stiffness matrix relating internal forces to incremental displacements. In reality the physical meaning of items in the ‘displacement’ vector can also be e.g. a velocity or a Lagrange multiplier. In this chapter the physical meaning of what we call the displacement and force vector and the stiffness matrix is irrelevant. Most often it represents a continuous system that is approximated using the Principle of Virtual Work, Galerkin discretization or another method.

A good starting point is to strive to an equilibrium state in which the internal force vector equals the external force vector, satisfying boundary conditions.

$$\mathbf{f}_{\text{int}} = \mathbf{f}_{\text{ext}} \quad (46.1)$$

$$\mathbf{u}_i = \mathbf{u}_i^0 \quad (i \text{ prescribed}) \quad (46.2)$$

In nonlinear analysis the internal force vector usually depends nonlinearly on the displacements (e.g. nonlinear elasticity). It can also depend on the displacements in the history. This is e.g. the case if the material is ‘path dependent’ such as in plasticity and if large displacements facilitate multiple equilibrium solutions.

The external force vector can also be displacement dependent. This is the case in geometrically nonlinear analysis, if the magnitude or the direction of the loading depends on the displacements such as with pressure on a wall. We can now write

$$\mathbf{f}_{\text{int}}(\mathbf{u}, \text{history}) = \mathbf{f}_{\text{ext}}(\mathbf{u}) \quad (46.3)$$

The system described above is already discretized in space. To enable a numerical solution, a time discretization is performed as well. Here ‘time’ can have a real physical meaning e.g. in a creep analysis or it can be a pseudo-time, only to describe a sequence of situations. Starting at time t with an approximated solution ${}^t\mathbf{u}$, a solution ${}^{t+\Delta t}\mathbf{u}$ is

searched for which Eq. (46.3) holds. Within the time-increment, only the displacements at start and end are known. The internal force vector, which may be path dependent, is calculated from the situation at time t , the time increment Δt and the displacement increment $\Delta \mathbf{u}$. The external forces only depend on the current geometry. If we consider only one increment, the time increment and the situation at the start of the increment (history) are fixed. The equilibrium equation within the increment then only depends on $\Delta \mathbf{u}$. We can write the nonlinear problem as: find $\Delta \mathbf{u}$ such that

$${}^{t+\Delta t}\mathbf{u} = {}^t\mathbf{u} + \Delta \mathbf{u} \quad (46.4)$$

and, with \mathbf{g} as the *out-of-balance* force vector (the residual forces).

$$\mathbf{g}(\Delta \mathbf{u}) = \mathbf{f}_{\text{ext}}(\Delta \mathbf{u}) - \mathbf{f}_{\text{int}}(\Delta \mathbf{u}) = \mathbf{0} \quad (46.5)$$

Acceleration forces (dynamic analysis) are considered in Chapter 48 and are ignored in this chapter. Starting the analysis at time t_{begin} we can increment the time with a number of increments, until the desired end value t_{end} is reached.

46.1.1 Iterative Procedures

A purely incremental method usually leads to inaccurate solutions in nonlinear analysis, unless very small step sizes are used. In an iterative process the errors that occur can be reduced successively. This in fact realizes an *implicit procedure*. The allowable step size is usually higher than in case of a process without iterations (e.g. an explicit process). The general procedure is the same for all iteration processes [Fig. 46.1]. In all procedures, the

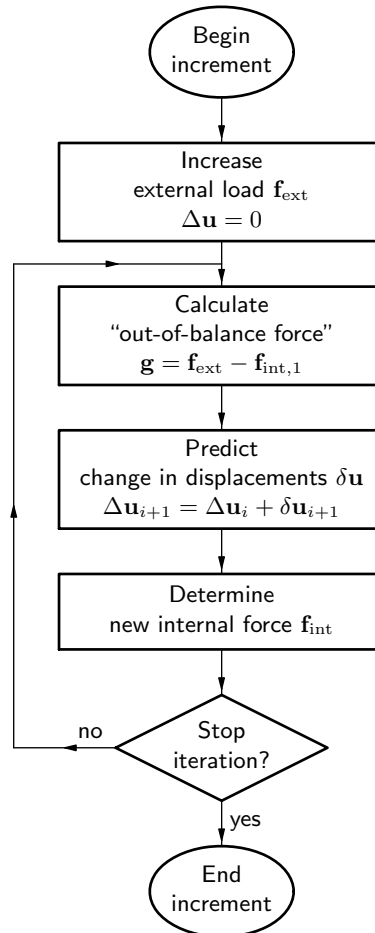


Figure 46.1: Iteration process

total displacement increment $\Delta \mathbf{u}$ is adapted iteratively by iterative increments $\delta \mathbf{u}$ until

equilibrium is reached, up to a prescribed tolerance. Indicating the iteration number with a right subscript, the incremental displacements at iteration $i + 1$ are calculated from

$$\Delta \mathbf{u}_{i+1} = \Delta \mathbf{u}_i + \delta \mathbf{u}_{i+1} \quad (46.6)$$

The difference between several procedures is the way in which $\delta \mathbf{u}$ is determined. The iterative increments are calculated by use of a ‘stiffness matrix’ \mathbf{K} that represents some kind of linearized form of the relation between the force vector and displacement vector. The used stiffness matrix can change every iteration, the matrix that is used in iteration i is called \mathbf{K}_i . A direct approach is to determine the iterative increments by

$$\delta \mathbf{u}_i = \mathbf{K}_i^{-1} \mathbf{g}_i \quad (46.7)$$

where \mathbf{g}_i is the out-of-balance force vector at the start of iteration i . In this case a linear set of equations is solved at every iteration.

Next sections describe the methods that are available in DIANA. First three pure iterative procedures are presented: the *Newton–Raphson* method, the *Quasi-Newton* method and the *Constant Stiffness* method. Next, two variations that can be used in combination with these procedures are considered: the *Continuation* method and the *Line Search* method. Finally, several criteria to stop the iteration loop will be discussed.

Another variation of the iteration algorithm is the Arc-length method [§ 46.1.5.2 p. 561]. This method adapts the increment size.

46.1.1.1 Newton–Raphson

Within the class of Newton–Raphson methods, generally two subclasses are distinguished: the *Regular* and the *Modified* Newton–Raphson method. Both methods use Eq. (46.7) to determine the iterative increment of the displacement vector. In a Newton–Raphson method, the stiffness matrix \mathbf{K}_i represents the tangential stiffness of the structure:

$$\mathbf{K}_i = \frac{\partial \mathbf{g}}{\partial \Delta \mathbf{u}} \quad (46.8)$$

The difference between the Regular and the Modified Newton–Raphson method is the point at which the stiffness matrix is evaluated.

Regular Newton–Raphson. In the Regular Newton–Raphson iteration the stiffness relation Eq. (46.8) is evaluated every iteration [Fig. 46.2]. This means that the prediction of Eq. (46.7) is based on the last known or predicted situation, even if this is not an equilibrium state. The Regular Newton–Raphson method yields a quadratic convergence

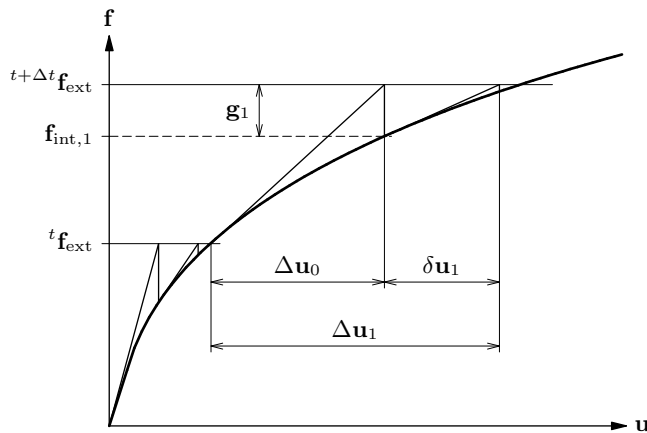


Figure 46.2: Regular Newton–Raphson iteration

characteristic, which means that the method converges to the final solution within only a few iterations.

A disadvantage of the method is that the stiffness matrix has to be set up at every iteration and, if a direct solver is used to solve the linear set of equations, the time consuming decomposition of the matrix has to be performed every iteration as well. Moreover, the quadratic convergence is only guaranteed if a correct stiffness matrix is used and if the prediction is already in the neighborhood of the final solution. If the initial prediction is far from the final solution, the method easily fails because of divergence. In short:

The Regular Newton–Raphson method usually needs only a few iterations, but every iteration is relatively time consuming.

Modified Newton–Raphson. The Modified Newton–Raphson method only evaluates the stiffness relation Eq. (46.8) at the start of the increment [Fig. 46.3]. This means that the prediction is always based on a converged equilibrium state. Usually, Modified

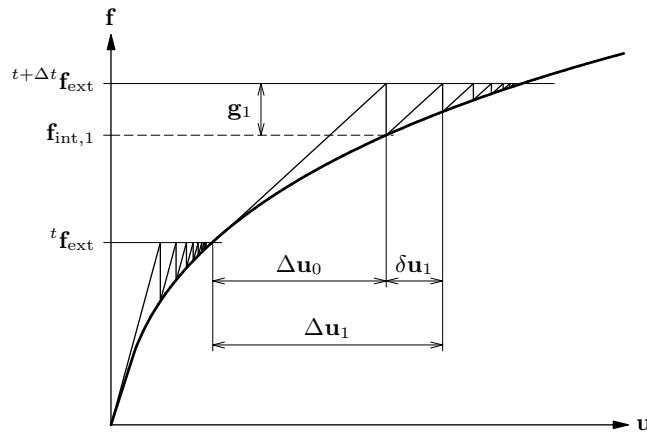


Figure 46.3: Modified Newton–Raphson iteration

Newton–Raphson converges slower to equilibrium than Regular Newton–Raphson. However, for every iteration only the prediction of the iterative incremental displacements and the internal force vector has to be calculated, it is not necessary to set up a new stiffness matrix. If a direct solver for the linear set of equations is used, it is not necessary to perform the decomposition again, only the relatively fast substitution part will do. In short:

The Modified Newton–Raphson method usually needs more iterations, but every iteration is faster than in Regular Newton–Raphson.

In situations where Regular Newton–Raphson does not converge anymore, the Modified Newton–Raphson process can sometimes still converge. Small variations of both processes are possible by using the linear or previous stiffness for the first prediction and by setting up the current stiffness matrix after the first prediction. If unloading occurs, it can be advantageous to return to the linear stiffness, e.g. in a plasticity analysis.

46.1.1.2 Quasi-Newton

The Quasi-Newton method (also called ‘Secant method’) essentially uses the information of previous solution vectors and out-of-balance force vectors during the increment to achieve a better approximation [Fig. 46.4]. Unlike Regular Newton–Raphson, the Quasi-Newton method does not set up a completely new stiffness matrix every iteration. In this case the stiffness of the structure is determined from the known positions at the equilibrium path. If the iterative displacement increment is called $\delta \mathbf{u}_i$ and the change in out-of-balance force vector related to this increment $\delta \mathbf{g}_i = \mathbf{g}_{i+1} - \mathbf{g}_i$, the Quasi-Newton relation is

$$\mathbf{K}_{i+1} \delta \mathbf{u}_i = \delta \mathbf{g}_i \quad (46.9)$$

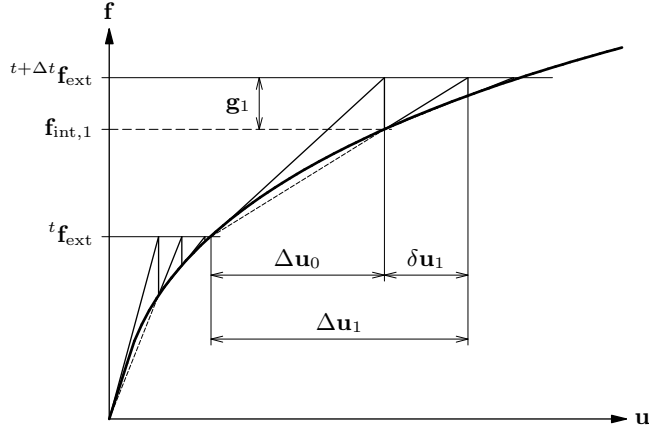


Figure 46.4: Quasi-Newton iteration

With a matrix \mathbf{K}_i that fulfills Eq. (46.9), the next iterative increment is calculated from Eq. (46.7). For a system with more than one degree of freedom, the secant stiffness matrix \mathbf{K} is not unique. The methods implemented in DIANA are known as the *Broyden*, the *Broyden–Fletcher–Goldfarb–Shanno* (BFGS) and the *Crisfield* methods. By substitution it can be seen that the following two matrices fulfill the Quasi-Newton relation Eq. (46.9).

$$\mathbf{K}_{i+1} = \mathbf{K}_i + \frac{(\delta \mathbf{g}_i - \mathbf{K}_i \delta \mathbf{u}_i) \mathbf{c}^T}{\mathbf{c}^T \delta \mathbf{u}_i} \quad (46.10)$$

$$\begin{aligned} \mathbf{K}_{i+1} = \mathbf{K}_i + & \frac{(\delta \mathbf{g}_i - \mathbf{K}_i \delta \mathbf{u}_i) \mathbf{c}^T + \mathbf{c}(\delta \mathbf{g}_i - \mathbf{K}_i \delta \mathbf{u}_i)^T}{\mathbf{c}^T \delta \mathbf{u}_i} \\ & - \frac{(\delta \mathbf{g}_i - \mathbf{K}_i \delta \mathbf{u}_i)^T \delta \mathbf{u}_i \mathbf{c} \mathbf{c}^T}{(\mathbf{c}^T \delta \mathbf{u}_i)^2} \end{aligned} \quad (46.11)$$

In Eq. (46.10) and Eq. (46.11) the vector \mathbf{c} can be chosen freely. The Quasi-Newton methods can be used efficiently because the inverse of the new stiffness matrix can be derived directly from the previous secant stiffness and the update vectors by using the Sherman–Morrison formula.

Broyden. If in Eq. (46.10) \mathbf{c} is substituted by $\delta \mathbf{u}$ and \mathbf{K}_{i+1} is inverted, the Broyden method results:

$$\mathbf{K}_{i+1}^{-1} = \mathbf{K}_i^{-1} + \frac{(\delta \mathbf{u}_i - \mathbf{K}_i^{-1} \delta \mathbf{g}_i) \delta \mathbf{u}_i^T \mathbf{K}_i^{-1}}{\delta \mathbf{u}_i^T \mathbf{K}_i^{-1} \delta \mathbf{g}_i} \quad (46.12)$$

BFGS. More elaborative, Eq. (46.11) can yield the relation attributed to Broyden, Fletcher, Goldfarb and Shanno, and therefore known as the BFGS method:

$$\mathbf{K}_{i+1}^{-1} = \left(\mathbf{I} + \frac{\delta \mathbf{u}_i \delta \mathbf{g}_i^T}{\delta \mathbf{u}_i^T \delta \mathbf{g}_i} \right) \mathbf{K}_i^{-1} \left(\mathbf{I} - \frac{\delta \mathbf{g}_i \delta \mathbf{u}_i^T}{\delta \mathbf{u}_i^T \delta \mathbf{g}_i} \right) + \frac{\delta \mathbf{u}_i \delta \mathbf{u}_i^T}{\delta \mathbf{u}_i^T \delta \mathbf{g}_i} \quad (46.13)$$

The inverse secant stiffness matrices are not calculated explicitly, but the iterative displacements $\delta \mathbf{u}$ are calculated directly by substitution of Eq. (46.12) or Eq. (46.13) in Eq. (46.7). By successive application of Eq. (46.12) or Eq. (46.13), the correct secant stiffness can be calculated from the stiffness \mathbf{K}_0 that was used at the start of the increment and an update vector for every iteration. For every intermediate iteration one additional update vector is to be stored with size ‘number of degrees of freedom’. The higher the iteration number, the more additional storage is needed and the more additional vector calculations are to be performed.

Crisfield. To avoid the increasing storage and computation time requirements for the Broyden and BFGS methods, Crisfield [22, §9.8] suggested to use only the most recent correction vector. For a one-dimensional situation this method still behaves as in Figure 46.4, but the Quasi-Newton relation Eq. (46.9) is not matched.

All three Quasi-Newton methods can be used irrespectively of the stiffness matrix \mathbf{K}_0 used for the first prediction. This could be a tangential stiffness matrix, as used in Figure 46.4, as well as a linear elastic stiffness matrix. These methods usually have a convergence rate between that of the Regular Newton–Raphson and the Modified Newton–Raphson schemes. This holds also for the time consumption. For large systems, especially when using a direct solver, the time used per iteration will be more closely to the Modified Newton–Raphson than to the Regular Newton–Raphson scheme. For the Broyden and the BFGS schemes the memory and time consumption will increase with the number of iterations.

46.1.1.3 Linear and Constant Stiffness

The Linear and Constant Stiffness iteration methods can be used if the other methods become unstable, or if it is desirable to keep certain characteristics.

Linear Stiffness. The Linear Stiffness iteration method uses the linear stiffness matrix all the time [Fig. 46.5]. This method potentially has the slowest convergence, but it

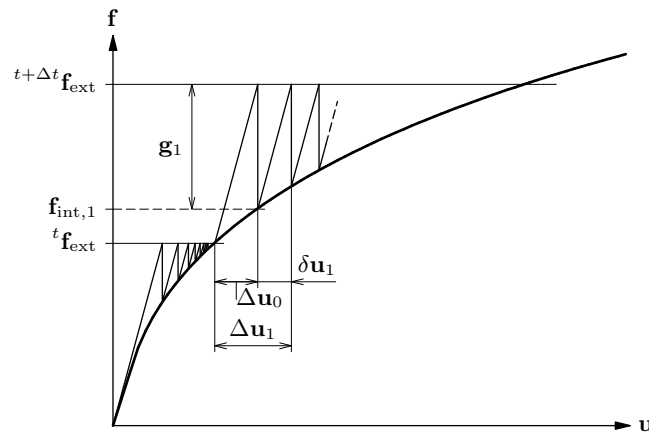


Figure 46.5: Linear Stiffness iteration

costs the least time per iteration since the stiffness matrix needs to be set up only once. Moreover, in case of a direct linear solver, the costly decomposition has to be performed only once. The Linear Stiffness method can also be advantageous if it is desirable that the stiffness matrix remains symmetric where the tangential stiffness matrix would become non-symmetric.

The Linear Stiffness method is usually very robust, but it is very well possible that it follows unstable equilibrium paths after bifurcations.

Constant Stiffness. The Constant Stiffness method uses the stiffness matrix left behind by the previous increment. This means that if Newton–Raphson iterations are used during the first phase of an analysis and Constant Stiffness iterations in a second phase, the stiffness in the latter will be equal to the last calculated stiffness in the first. If the Constant Stiffness iteration is used since the first increment, this method equals the Linear Stiffness method.

The Constant Stiffness method can be used if Newton–Raphson or Quasi-Newton methods fail after a number of successful increments.

We can try to find the solution to $s(\eta) = 0$ by calculating s at various values of η . The first two values are readily derived from the original iteration process. Once the search direction is calculated, the values $s(0)$ and $s(1)$ are calculated by the inner product of $\delta \mathbf{u}$ with respectively the out-of-balance force at the start and at the end of the iteration. Usually the Line Search method in DIANA is only used to ‘help’ the ordinary iteration processes, therefore the line searches do not really continue until a value $s = 0$ is found, but the line search is terminated if the absolute value of $s(\eta)$ has a value that is less than Ψ times the value $s(0)$ [Fig. 46.7]. In most cases a value $\Psi = 0.8$ is sufficient to

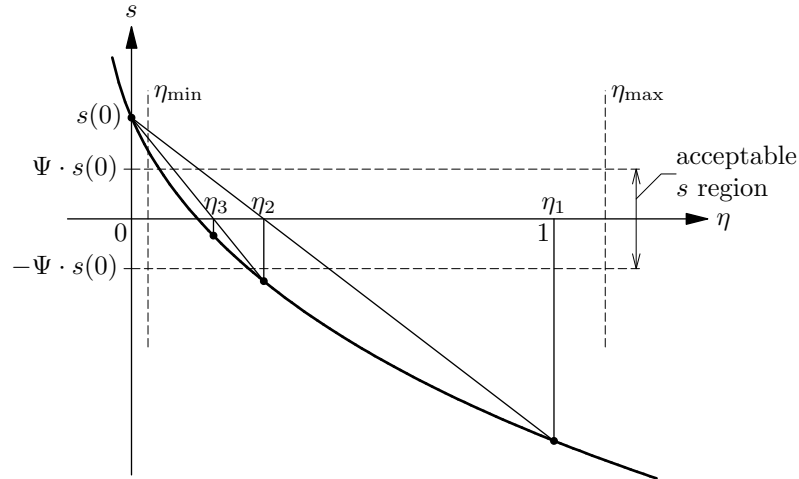


Figure 46.7: Line Search iteration

stabilize the global iteration process. If the values of s have an opposite sign, a Regula-Falsi method is used to predict the value where $s(\eta) = 0$. If both values have the same sign, extrapolation is used to determine a new value for η . To avoid unrealistic values, DIANA bounds η between an upper and lower bound, η_{\min} and η_{\max} . DIANA stops the Line Search process if the change in η per iteration is less than the selected tolerance.

46.1.4 Convergence Criteria

The iteration process must be stopped if the results are satisfactory [Fig. 46.1 p. 552]. For this purpose, DIANA offers several convergence norms. Besides stopping the iteration in case of convergence, the iteration process is also stopped if a specified maximum number of iterations has been reached or if the iteration obviously leads to divergence. The detection of divergence is based on the same norms as the detection of convergence. Figure 46.8 specifies the items used to set up the various norms.

46.1.4.1 Force Norm

The force norm is the Euclidian norm of the out-of-balance force vector \mathbf{g} . To check convergence, the force norm after the current iteration is checked against the norm of the initial unbalance \mathbf{g}_0

$$\text{Force norm ratio} = \frac{\sqrt{\mathbf{g}_i^T \mathbf{g}_i}}{\sqrt{\mathbf{g}_0^T \mathbf{g}_0}} \quad (46.16)$$

Because the reference force norm is known before the first prediction of displacements, the force norm ratio can be calculated directly after the first prediction, $i = 1$ in Eq. (46.16). This means that if the first prediction is correct (nearly linear behaviour) the force norm can detect convergence right away and no unnecessary iterations have to be performed.

46.1.4.2 Displacement Norm

The displacement norm

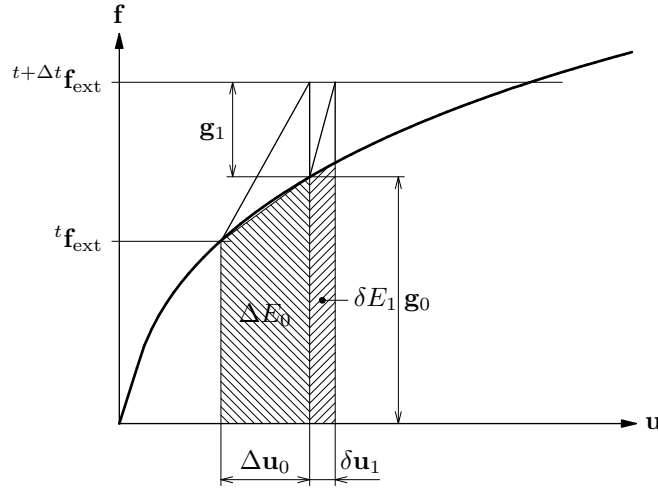


Figure 46.8: Norm items

is the Euclidian norm of the iterative displacement increment. To check convergence, the displacement norm is checked against the norm of the displacement increments in the first prediction of the increment.

$$\text{Displacement norm ratio} = \frac{\sqrt{\delta \mathbf{u}_i^T \delta \mathbf{u}_i}}{\sqrt{\Delta \mathbf{u}_0^T \Delta \mathbf{u}_0}} \quad (46.17)$$

From Eq. (46.17) it is clear that the ratio of the displacement norm after the first prediction (iteration 0) equals 1 by definition. To check convergence, always one additional iteration is necessary.

46.1.4.3 Energy Norm

A third way to check convergence is the energy norm. This norm is composed of internal forces and relative displacements as indicated in Figure 46.8 with ΔE_0 and δE_1 . To determine convergence, the energy ratio is calculated as

$$\text{Energy norm ratio} = \left| \frac{\delta \mathbf{u}_i^T (\mathbf{f}_{\text{int},i+1} + \mathbf{f}_{\text{int},i})}{\Delta \mathbf{u}_0^T (\mathbf{f}_{\text{int},1} + \mathbf{f}_{\text{int},0})} \right| \quad (46.18)$$

Note that here the internal force is used and not the out-of-balance force. Use of the out-of-balance force would be improper, for a Line Search procedure could then minimize the norm, see Eq. (46.15), before the solution really converges to equilibrium. As with the displacement norm, the energy norm also requires an additional iteration to detect convergence.

The choice of the proper norm and its convergence criterion depends on the type of analysis. Using a lot of prescribed displacements generally makes the displacement norm less useful. On the other hand, a structure that can expand freely will hardly build up any internal forces and the force norm may be less useful. Always be sure that the reference norm (the denominator in the ratios) has a reasonable value i.e., not close to zero.

Experience shows that the convergence criterion for softening type behaviour should be more strict than the criterion that can be used in a hardening type analysis. If there is any doubt about the criterion to be used, it is advisable to perform the analysis with two distinct criteria and check the differences in results. If large differences occur, at least the less strict norm was to large.

46.1.4.4 Residual Norm

The residual norm is also a Euclidian norm of the out-of-balance force vector \mathbf{g} . Contrary to the force norm, the residual norm also takes the values in constrained degrees of freedom

(supports and tyings) into account. To check convergence, DIANA compares the change in the residual norm during the current iteration with the change in the residual norm during the first prediction of displacements in the current step.

$$\text{Residual norm ratio} = \frac{|\sqrt{\mathbf{g}_i^T \mathbf{g}_i} - \sqrt{\mathbf{g}_{i-1}^T \mathbf{g}_{i-1}}|}{|\sqrt{\mathbf{g}_0^T \mathbf{g}_0} - \sqrt{\mathbf{g}_n^T \mathbf{g}_n}|} \quad (46.19)$$

Where \mathbf{g}_n denotes the out-of-balance force vector in the last iteration of the previous step. In the first step DIANA takes its value as zero.

46.1.5 Incremental Procedures

The incremental-iterative solution procedure consist of two parts: the *increment* part and the *iteration* part. The iteration part was discussed in §46.1.1, in this section the increment part is treated. We first describe the most simple types: *load control* and *displacement control*. Then the *Arc-length method* is discussed, a method that can adapt the step size depending on the results in the current step. The initial choice of the step size for every increment is an important factor in the incremental-iterative process. Therefore, two methods are presented to determine step sizes and two methods to choose between loading and unloading depending on the previous analysis results.

Finally, we present the Automatic Incremental Loading method [§ 46.1.5.4]. This method requires only the final loading to be specified. From this, DIANA automatically determines the intermediate step size.

46.1.5.1 Load and Displacement Control

In §46.1.1 we have presented iteration processes where the external load was increased at the start of the increment, by directly increasing the external force vector \mathbf{f}_{ext} . This is usually called ‘load control’ [Fig. 46.9a]. Another way to put an external load on a structure is to prescribe certain displacements \mathbf{u}^c . This is called ‘displacement control’ [Fig. 46.9b]. In case of displacement control the external force vector is not increased di-

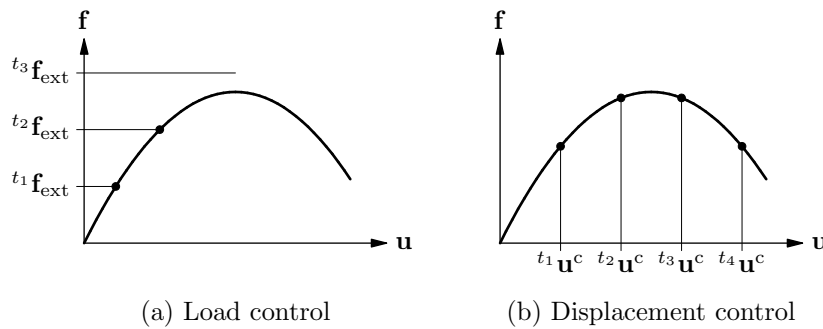


Figure 46.9: Load and displacement control

rectly. To get a proper first prediction of the displacements, the prescribed displacements must be incorporated in the external force vector. This effective force can be calculated by rewriting Eq. (46.7) and splitting the displacement increment vector in two parts: one referring to the unconstrained and an other referring to the constrained displacements, respectively $\Delta \mathbf{u}^u$ and $\Delta \mathbf{u}^c$. The stiffness matrix and force vector are split likewise:

$$\begin{bmatrix} \mathbf{K}^{uu} & \mathbf{K}^{uc} \\ \mathbf{K}^{cu} & \mathbf{K}^{cc} \end{bmatrix}_0 \begin{Bmatrix} \Delta \mathbf{u}^u \\ \Delta \mathbf{u}^c \end{Bmatrix}_0 = \begin{Bmatrix} \mathbf{g}^u \\ \mathbf{g}^c \end{Bmatrix}_0 \quad (46.20)$$

The unknown displacement increments $\Delta \mathbf{u}^u$ can be calculated from the first row in Eq. (46.20) as

$$\Delta \mathbf{u}_0^u = (\mathbf{K}^{uu})^{-1} \{-\mathbf{K}_0^{uc} \Delta \mathbf{u}^c + \mathbf{g}_0^u\} \quad (46.21)$$

Comparing Eq. (46.7) and Eq. (46.21) indicates that $-\mathbf{K}_0^{\text{uc}} \Delta \mathbf{u}^c$ can be regarded as the *effective force vector*, equivalent with the prescribed displacements. In subsequent iterations, the iterative increments of the prescribed displacements are zero and hence the effective force vector vanishes.

A similar effective force vector can be generated in case of influence of time on the analysis e.g. prescribed temperature increments or viscoelastic material behaviour. In this case, the effective force vector contains the effect on the internal force vector during the time increment if the displacements remain constant. The addition of this effective force vector in the first prediction (zero iteration) will improve the convergence of the iteration process significantly. In subsequent iterations, the time does not change anymore and also this effective load vector will vanish.

In real-life analysis, the loading does not have to be restricted to load control, displacement control or time increments, but they can be combined in any way. In that case the ‘real’ external load and the effective force vectors from prescribed displacement increments and time influences must be used together.

46.1.5.2 Arc-length Control

In an ordinary iteration process the predictions for the displacement increments can become very large. This is the case especially if the load–displacement curve is almost horizontal. If a fixed load increment is prescribed, this results in very large predictions for the displacements. The problem can be overcome with the use of an *Arc-length* method. Using the Arc-length method the *snap-through* behaviour of Figure 46.10a¹ can

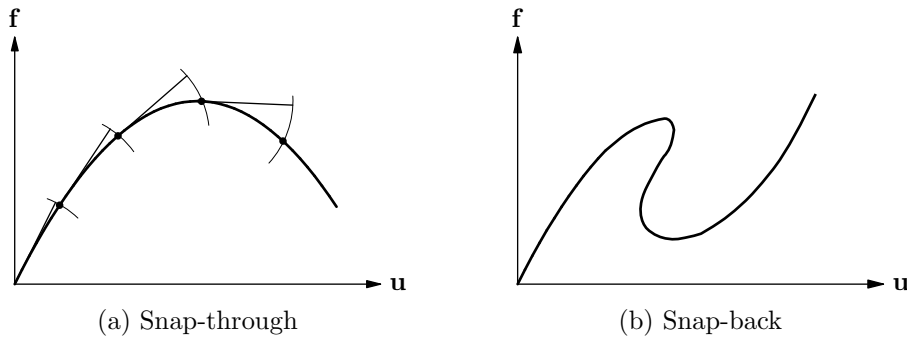


Figure 46.10: Arc-length control

be analysed, just as displacement control could. Here, however, it is possible to define a system of loads that could not be substituted by prescribed displacements. Moreover, the Arc-length method is also capable of passing *snap-back* behaviour [Fig. 46.10b], where displacement control fails.

The Arc-length method constrains the norm of the incremental displacements to a prescribed value. This is done by simultaneously adapting the size of the increment. Note that the size is adapted within the iteration process and is not fixed at the moment the increment starts. For this purpose we define the external force vector at the start of the increment as ${}^t\mathbf{f}_{\text{ext}}$ and the increment of the external force vector as $\Delta\lambda_i \hat{\mathbf{f}}$. The load factor $\Delta\lambda_i$ multiplies a unit load $\hat{\mathbf{f}}$ and can change every iteration. Substitution in Eq. (46.7) results in

$$\delta \mathbf{u}_i = \mathbf{K}_i^{-1} \left(\Delta\lambda_i \hat{\mathbf{f}} + {}^t\mathbf{f}_{\text{int}} - \mathbf{f}_{\text{int},i} \right) \quad (46.22)$$

The solution $\delta \mathbf{u}_i$ is now split in two parts:

$$\delta \mathbf{u}_i^{\text{I}} = \mathbf{K}_i^{-1} ({}^t\mathbf{f}_{\text{int}} - \mathbf{f}_{\text{int},i}) \quad \text{and} \quad \delta \mathbf{u}_i^{\text{II}} = \mathbf{K}_i^{-1} \Delta\lambda_i \hat{\mathbf{f}} \quad (46.23)$$

The total iterative increment is then derived from

$$\delta \mathbf{u}_i = \delta \mathbf{u}_i^{\text{I}} + \Delta\lambda_i \delta \mathbf{u}_i^{\text{II}} \quad (46.24)$$

¹In this figure, the load factor is included in the definition of the arc length. This is not the case in DIANA, but the results in case of more than one degree of freedom are similar as indicated in the figures for one degree of freedom.

The load factor $\Delta\lambda_i$ is still undefined and can now be used to constrain the incremental displacement vector. DIANA offers a quadratic and a linearized constraint, leading to the *Spherical Path* Arc-length method and the *Updated Normal Plane* method, see Crisfield [22, §9.3].

Spherical Path. In the spherical constraint, the constraint equation is

$$\Delta\mathbf{u}_i^T \Delta\mathbf{u}_i = \Delta l^2 \quad (46.25)$$

where Δl is the required arc length. Substitution of Eq. (46.6) and Eq. (46.24) in Eq. (46.25) gives the value for $\Delta\lambda$

$$\Delta\lambda_i = \frac{-a_2 \pm \sqrt{a_2^2 - 4a_1a_3}}{2a_1} \quad (46.26)$$

with

$$\begin{aligned} a_1 &= (\delta\mathbf{u}_i^{\text{II}})^T \delta\mathbf{u}_i^{\text{II}} \\ a_2 &= 2(\delta\mathbf{u}_i^{\text{I}})^T \delta\mathbf{u}_i^{\text{II}} + 2(\Delta\mathbf{u})^T \delta\mathbf{u}_i^{\text{II}} \\ a_3 &= 2(\Delta\mathbf{u})^T \delta\mathbf{u}_i^{\text{I}} + (\delta\mathbf{u}_i^{\text{I}})^T \delta\mathbf{u}_i^{\text{I}} + (\Delta\mathbf{u})^T \Delta\mathbf{u} - \Delta l^2 \end{aligned}$$

Normally, two solutions for $\Delta\lambda$ fulfill Eq. (46.25) but if the discriminant $a_2^2 - 4a_1a_3 < 0$ then DIANA uses a linearized equivalent of the Spherical Path method as described by Forde and Stierner [32]. To determine which of the two regular solutions should be used, the angle θ between the displacement increment vector of the previous iteration and the current iteration is calculated for both solutions

$$\cos \theta = \frac{(\Delta\mathbf{u}_{i-1})^T \delta\mathbf{u}_i}{\|\Delta\mathbf{u}_{i-1}\| \|\delta\mathbf{u}_i\|} \quad (46.28)$$

If one of the solutions yields a negative cosine and the other a positive, DIANA chooses the solution with the positive cosine (acute angle). If both solutions yield acute angles, the solution closest to the linear solution $\Delta\lambda = -a_3/a_2$ is used.

Updated Normal Plane. The second constraint is a linearized constraint. If Eq. (46.25) is matched for $\Delta\mathbf{u}_{i-1}$, then the constraint equation for $\Delta\mathbf{u}_i = \Delta\mathbf{u}_{i-1} + \delta\mathbf{u}_i$ can be written as

$$(\Delta\mathbf{u}_{i-1})^T \delta\mathbf{u}_i = 0 \quad (46.29)$$

where the quadratic term in $\delta\mathbf{u}_i$ is ignored. Substituting Eq. (46.24) into Eq. (46.29) leads to the expression for $\Delta\lambda_i$

$$\Delta\lambda_i = -\frac{(\Delta\mathbf{u}_{i-1})^T \delta\mathbf{u}_i^{\text{I}}}{(\Delta\mathbf{u}_{i-1})^T \delta\mathbf{u}_i^{\text{II}}} \quad (46.30)$$

Geometrically this constraint means that the iterative increment must be perpendicular to the total increment at the previous iteration. The solution is projected on the plane, normal to the previous solution, hence the method is referred to as the Updated Normal Plane method.

Indirect Displacement Control. In the previous description of the constraint equations all displacements were gathered together. For global nonlinear behaviour this is adequate, but for local collapse mechanisms the method can perform better if only a part of the displacements is considered. The constraint equations can remain the same as in Eq. (46.25) and Eq. (46.29), but instead of using the vectors $\delta\mathbf{u}$ and $\Delta\mathbf{u}$ vectors $\delta\mathbf{v}$ and $\Delta\mathbf{v}$ are considered, defined by

$$\mathbf{v} = \begin{Bmatrix} \alpha_1 u_1 \\ \alpha_2 u_2 \\ \alpha_3 u_3 \\ \dots \\ \alpha_n u_n \end{Bmatrix} \quad (46.31)$$

In the extreme case that only one item in \mathbf{v} is non-zero, the arc length is defined as the displacement of the corresponding degree of freedom. A constant arc length during the analysis will result in this case in equal displacement increments for this degree of freedom. Because the loading is defined as an external force, this type of control is called *Indirect Displacement* control. A variant of Indirect Displacement control is *Crack Mouth Opening Displacement* control, usually called CMOD. This can be used, just as in experiments, to control the increase in crack width et cetera. In case of CMOD control, a vector is formed with new ‘degrees of freedom’ that can e.g. represent the difference in displacements on opposite nodes on a crack plane.

$$\mathbf{v} = \begin{Bmatrix} \dots \\ \dots \\ \alpha_1 u_p + \alpha_2 u_q \\ \dots \end{Bmatrix} \quad (46.32)$$

This vector is used in the constraint equations Eq. (46.25) or Eq. (46.29).

As long as the displacement increments per step remain relatively small, the difference between the Spherical Path method and the Updated Normal Plane method are small. More important than the choice between these two methods is the choice of the value for the arc length l . The available Arc-length methods are particularly useful if they are combined with *adaptive load incrementation*, as described in the next section.

46.1.5.3 Adaptive Loading and Time Increments

Up to here we have used the initial load, displacement or time increment as a fixed value. In an analysis, we could e.g. reach a load level of 100 N, by defining ten increments of 10 N. In combination with Arc-length control, the size of the increment can change inside an increment, but the start value was still fixed.

The size of the increments is limited by the physical behaviour in case of path dependency and by the convergence characteristics of the selected iteration process. Especially in the latter case, the allowable step size depends on the amount of nonlinearity in the increment. This is usually not known a priori, i.e., at the moment that the analysis starts, and therefore the optimum increment sizes cannot be fixed beforehand. To allow for result dependent increment sizes ‘adaptive loading’ can be used. An even more important question that can usually not be answered before the analysis is started is whether at a certain load level the load must increase further or must decrease, e.g. in case of a snap-through.

Three adaptive loading methods are implemented in DIANA, an *iteration based* method for all types of loading, an *energy based* method that can only be used in combination with Arc-length control [§ 46.1.5.2 p. 561], and the *cutback based automatic incremental loading* method [§ 46.1.5.4]. For the first two methods, two algorithms are available to decide whether the next step must be an increment or a decrement. The cutback based automatic incremental loading method does not allow for loading–unloading switching but offers the possibility for error-controlled time increments if the physical behaviour explicitly depends on the rate-of-change of the solution.

Iteration Based. In the iteration based method, the experience is used that for many analyses, the iteration process converges faster if the increment size is smaller. Based on a ‘desired number of iterations’ the new increment size can be made larger than the previous if the actual number of iterations in the previous increment was smaller than the desired number and vice versa. The size of the new load increment ${}^{t+\Delta t}\Delta\lambda_0$ is calculated by

$${}^{t+\Delta t}\Delta\lambda_0 = \frac{{}^t\Delta l}{\sqrt{{}^\delta\mathbf{u}_0^\top {}^\delta\mathbf{u}_0}} \left(\frac{N^d}{{}^tN} \right)^\gamma \quad (46.33)$$

Here ${}^t\Delta l$ is the length of the predictor displacements of the previous step. Further N^d is the desired number of iterations and tN is the actual number of iterations in the previous increment. The power γ is usually set equal to 0.5 .

The iteration based method is also very useful to pass sharp snap-throughs or softening behaviour in crack propagation analyses. When parameter γ is set equal to zero, a constant arc length is applied throughout the whole analysis. Experience shows that this method is stable in case of softening behaviour. Since

$${}^{t+\Delta t}\Delta\lambda_0 = \frac{{}^t\Delta l}{\sqrt{\delta\mathbf{u}_0^T \delta\mathbf{u}_0}} \quad \text{and} \quad {}^{t+\Delta t}\Delta l = {}^{t+\Delta t}\Delta\lambda_0 \sqrt{\delta\mathbf{u}_0^T \delta\mathbf{u}_0} \quad (46.34)$$

it follows that the length of the incremental displacement vector remains constant.

Energy Based. The energy based method can only be used in combination with the Arc-length method. This method calculates a load increment such that the vector product of the load increment and the displacement increment in the first prediction equals the vector product of the final load increment and displacement increment of the previous step [Fig. 46.11]. Here ${}^t\mathbf{W}$ indicates a kind of final energy increment of step 1 and ${}^{t+\Delta t}\tilde{\mathbf{W}}$ a first prediction thereof in step 2.

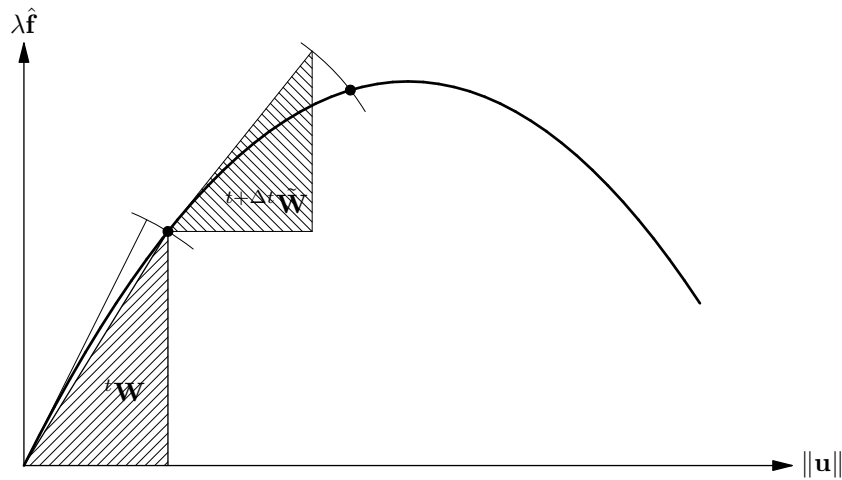


Figure 46.11: Work increment

The new loading factor is derived from

$${}^{t+\Delta t}\Delta\lambda_0 = \sqrt{\frac{|{}^t\Delta\lambda_n ({}^t\Delta\mathbf{u}_n)^T \hat{\mathbf{f}}|}{|\delta\mathbf{u}_0^T \hat{\mathbf{f}}|}} \quad (46.35)$$

where index n indicates the last iteration (of the previous increment). In order to get a proper initial value for $\Delta\lambda$ the vectors $\delta\mathbf{u}_0^I$ and $\delta\mathbf{u}_0^{II}$ must be calculated with a tangential stiffness matrix.

Loading–Unloading. A simple way to choose between increments or decrements is based on the appearance of *negative pivots* in the global system of equations. Often a negative pivot indicates unstable structural behaviour that is related with some type of snap-through. If this occurs the sign of the load increment must be changed from loading to unloading.

Another method, proposed by Crisfield, can only be used in combination with the Arc-length methods and is similar to the method used in the Spherical Path Arc-length method. The angle between the new prediction and the previous increment should be acute, so

$${}^{t+\Delta t}\Delta\lambda = \begin{cases} +|{}^{t+\Delta t}\Delta\lambda| & \text{if } {}^t\Delta\mathbf{u}_n^T \delta\mathbf{u}_0^{II} \geq 0 \\ -|{}^{t+\Delta t}\Delta\lambda| & \text{if } {}^t\Delta\mathbf{u}_n^T \delta\mathbf{u}_0^{II} < 0 \end{cases} \quad (46.36)$$

In many occasions, both loading and unloading criteria will yield the same result. In case of multiple equilibrium paths (bifurcations) the methods may differ and one should be aware that ‘an’ equilibrium path is followed and not necessarily a stable equilibrium path.

46.1.5.4 Cutback Based Automatic Incremental Loading

Cutback based automatic load stepping is a simple tool for adaptive load increments. Given a final loading, the automatic load step controller tries to take as few load steps as possible and at the same time tries to limit the number of steps in the iterative procedure. The main advantage over the iterations based load controller is that the automatic load controller recovers from non-convergence in the iterative solver.

An outline of the algorithm is as follows.

$${}^{t+\Delta t}\lambda_0 \times \text{minsiz} \leq {}^{t+\Delta t}\lambda \leq {}^{t+\Delta t}\lambda_0 \times \text{maxsiz} \quad (46.37)$$

First the full loading is applied in a single step. If the iterative procedure fails to converge, the load step is decreased by a factor *cutb* [§ 13.3.2.4 p. 227] and the calculation is restarted. If, after successive failures, the load step becomes smaller than a user-specified part of the full loading, (the MINSIZ parameter [§ 13.3.2.4 p. 227]) the automatic load controller gives up and issues an error message. On the other hand, if the iterative procedure converges fast with respect to the maximum number of iterations (user-specified parameter MAXITE [§ 13.3.5 p. 235]), the step size is increased. Optionally, the maximum load step size can be limited to a user specified part of the full loading (the MAXSIZ parameter [§ 13.3.2.4 p. 227]), for example if some intermediate results are needed for output purposes.

Adaptive error controlled time increments. If the physical behaviour of the model explicitly depends on the rate-of-change of the displacements or an internal variable (for instance dynamics, viscous behaviour), then the computed solution at a certain time depends on the time step sizes used. If very small time increments are taken, this influence is negligible. However, a priori it is not clear how small the step size should be. Moreover, it is often unattractive to use a fixed time step size. Clearly, one would like to use small time steps if there are rapid changes in the system, and to increase the step size if the system slowly relaxes to an equilibrium state.

To explain the adaptive time step strategy in DIANA we consider the initial value problem

$$\frac{d\mathbf{u}}{dt} = \mathbf{f}(\mathbf{u}) \quad (46.38)$$

Suppose that we use a time stepping method to advance the solution from t_n to t_{n+1} with a step size $\Delta t = t_{n+1} - t_n$. By doing so we introduce an error due to the finite time step size Δt . Let $\mathbf{u}(t)$ be the solution to Eq. (46.38) with initial condition $\mathbf{u}(t_n) = \mathbf{u}_n$. The time step error is then

$$\mathbf{u}(t_{n+1}) - \mathbf{u}_{n+1} = \mathcal{O}(\Delta t^{m+1}) \quad (46.39)$$

with m the order of consistency of the method. Suppose that we also have a less accurate method, with order of consistency \hat{m} (with $\hat{m} < m$). In DIANA a second order Runge–Kutta method is available, i.e., the SDIRK2 method [§ 48.4.5 p. 594], that has an embedded first order method. So we have a second order accurate solution \mathbf{u}_{n+1} , and a first order solution $\hat{\mathbf{u}}_{n+1}$. We want the error to be less than a prescribed error tolerance

$$\|\mathbf{u}_{n+1} - \mathbf{u}(t_{n+1})\| \leq \|\hat{\mathbf{u}}_{n+1} - \mathbf{u}_{n+1}\| \leq \epsilon_{\text{rel}} \times \|\mathbf{u}_{n+1}\| + \epsilon_{\text{abs}} \quad (46.40)$$

This criterion is both used to reject time steps and estimate the next time step:²

$$\Delta t_{n+1} = \Delta t_n \times \frac{\|\hat{\mathbf{u}}_{n+1} - \mathbf{u}_{n+1}\|^{\frac{1}{2}}}{\epsilon_{\text{rel}} \times \|\mathbf{u}_n\| + \epsilon_{\text{abs}}} \quad (46.41)$$

Some extra logic is introduced to avoid unnecessary step rejections, and to keep the step size constant to save LU-factorizations. Moreover, care is taken not to overstep changes in the loading.

Automatic time stepping without the SDIRK2 time stepping method is very much like automatic load stepping [§ 46.1.5.4 p. 565]. Conversely, if SDIRK2 is used while there

²See Hairer and Warner [38, § IV.8].

is no explicit rate dependence in the model, then a time accurate solution is produced. However, if you are not interested in intermediate results it is better, for efficiency reasons, not to use SDIRK2.

46.2 Geometric Nonlinearity

By default, DIANA assumes that in a nonlinear analysis the model behaves geometrically linear. In this case, the equilibrium equations are based on the undeformed geometry and the strains are linear functions of the nodal displacements. This limits the applicability of the analysis to small displacements, small rotations and small strains. Figure 46.12 shows two examples of structures for which the small displacement and rotation assumptions do no longer hold.

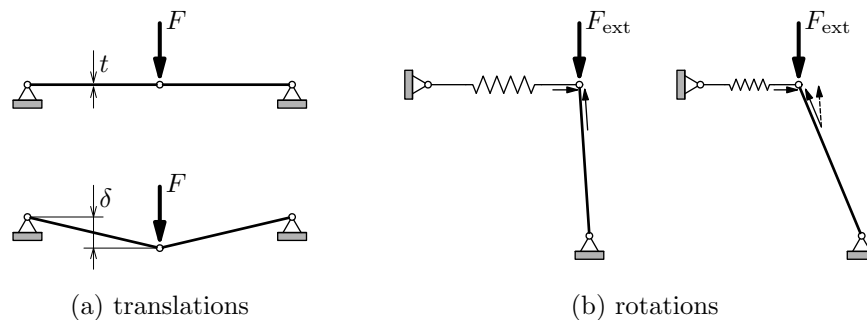


Figure 46.12: Large displacements

Figure 46.12a shows a flexible beam structure. The displacements will be considered ‘large’ when the vertical displacement δ is slightly larger than the beam thickness t . In this particular case, the initial stiffness of the structure is zero and the force F can be counter acted only by the development of tensile stresses in the beams that act under a small angle. This geometrically nonlinear phenomenon is known as *stress stiffening*.

The structure of Figure 46.12b shows large rotations. If the dimensions of the vertical bar are relatively short, large displacements are not necessarily encountered. Since the degrees of freedom are set up at the start of an analysis, the algorithm assumes for a geometrically linear case that the element contributions to the nodal forces are determined by the undeformed geometry. In this case the direction of the compressive force in the bar would be incorrect. In a geometrically nonlinear analysis, these large rotations are accounted for.

It is clear that certain types of loading, e.g. pressures on a wall, are heavily influenced by large displacements (change of area) as well as by large rotations (change of direction). In DIANA these types of loading are known as *nonconservative loading*.

Large strains should not be calculated as linear functions of the displacements, as this would lead to, for example, large calculated strains in the case of a rigid body rotation or non-unique strain measures. Even without large displacements large strains can show up, for instance in the vicinity of a crack tip, where the local strains can be very large. On the other hand, large displacements are possible without large strains e.g. in thin-walled flexible structures.

DIANA offers two types of geometrically nonlinear analysis: a *Total Lagrange* and an *Updated Lagrange* description, see for instance Crisfield [22, Ch. 5] or Bathe [5, § 6.4]. A Total Lagrange description is useful if rotations and displacements are large and strains are small, and is even obligatory for large strain hyperelastic (rubber-like) material behaviour. An Updated Lagrange description can be used advantageously in case of large plastic deformations. The choice of the geometrically nonlinear description determines the stress and strain measures that will be used. These can be other than ‘force over area’ for stress or ‘displacement over initial length’ for strain. In a combination of physically and geometrically nonlinear behaviour, a stress-strain relation must be defined compatible with the stress and strain measures used in the geometrically nonlinear description.

46.2.1 Definitions

In the following sections, a leading superscript ($^t \dots$) indicates the state of a quantity, and a leading subscript (${}_t \dots$) indicates the reference coordinate frame for derivatives.

A coordinate frame is attached to the material. The ‘material axes’ can both rotate and stretch during the transition from one coordinate frame to another [Fig. 46.13]. The

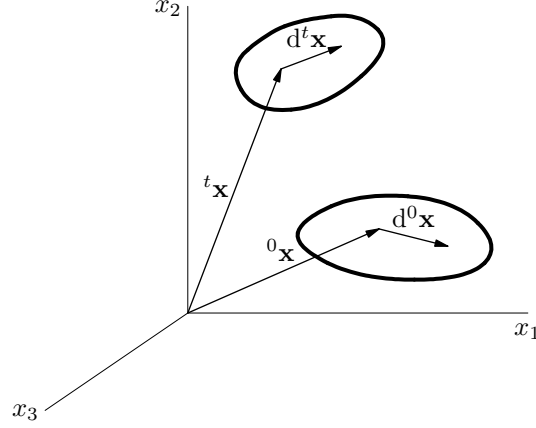


Figure 46.13: Coordinate Frames

material point position \mathbf{x} and material axis $d\mathbf{x}$ are shown for the coordinate frame at state 0 and state t . The displacement \mathbf{u} is denoted by

$${}^t\mathbf{u} = {}^t\mathbf{x} - {}^0\mathbf{x} \quad (46.42)$$

The deformation gradient \mathbf{F} expresses the rotation and stretch of $d\mathbf{x}$. The deformation gradient is

$${}^t_0\mathbf{F} = {}^t\mathbf{x}_0 \overleftarrow{\nabla} = \frac{\partial {}^t\mathbf{x}}{\partial {}^0\mathbf{x}} \quad (46.43)$$

with $d {}^t\mathbf{x} = {}^t_0\mathbf{F} d {}^0\mathbf{x}$. In case of an arbitrary deformation, the rotation matrix \mathbf{R} is defined by the polar decomposition of the deformation gradient

$$\mathbf{F} = \mathbf{R} \cdot \mathbf{U} = \mathbf{V} \cdot \mathbf{R} \quad (46.44)$$

Here \mathbf{U} and \mathbf{V} are symmetrical right and left stretch matrices and \mathbf{R} is an orthogonal matrix. The determinant J of the deformation gradient gives a volume change

$${}^tV = J {}^0V \quad ; \quad J = \det {}^t_0\mathbf{F} \quad (46.45)$$

The velocity of a point is denoted as

$$\dot{\mathbf{u}} = {}^t\dot{\mathbf{x}} \quad (46.46)$$

The velocity gradient is

$$\mathbf{L} = \dot{\mathbf{u}} \overleftarrow{\nabla} = \frac{\partial {}^t\dot{\mathbf{u}}}{\partial {}^t\mathbf{x}} \quad (46.47)$$

The spin $\mathbf{\Omega}$ and rate of deformation \mathbf{D} are defined in terms of the local velocity gradient

$$\mathbf{D} = \frac{1}{2} (\mathbf{L} + \mathbf{L}^T) \quad \text{and} \quad \mathbf{\Omega} = \frac{1}{2} (\mathbf{L} - \mathbf{L}^T) \quad (46.48)$$

The Green–Lagrange strain is defined as

$${}^t_0\mathbf{E} = \frac{1}{2} ({}^t_0\mathbf{F}^T \cdot {}^t_0\mathbf{F} - \mathbf{I}) \quad (46.49)$$

In small deformation problems, the stress is usually defined as ‘force over area’. For geometrically nonlinear analyses, this definition is not unique because the area may change in magnitude and/or direction during deformation. An important stress measure is the

Cauchy stress $\boldsymbol{\sigma}$. This stress is defined as the ‘force over area’ in the deformed configuration

$$\boldsymbol{\sigma} \mathbf{n} \, dA = d\mathbf{f} \quad (46.50)$$

where $d\mathbf{f}$ is the force acting on an area dA with a unit normal vector \mathbf{n} . Since the Finite Element Method is based on energy principles, the choice for a stress measure determines the strain measure and vice versa. The energy conjugate of the Cauchy stress is the linearized strain. The energy variation can be calculated from

$$\delta W = \int_V \boldsymbol{\sigma} \delta \boldsymbol{\varepsilon} \, dV \quad (46.51)$$

In a Total Lagrange analysis, the stress must be related to the undeformed configuration and must be energy conjugated to the Green–Lagrange strain. This stress measure is the 2nd Piola–Kirchhoff stress \mathbf{S} and is related to the Cauchy stress by

$${}^t_0\mathbf{S} = \det {}^t_0\mathbf{F} \cdot {}^t_0\mathbf{F}^{-1} \cdot {}^t\boldsymbol{\sigma} \cdot {}^t_0\mathbf{F}^{-T} \quad (46.52)$$

DIANA interpretes all input parameters which indicate a stress, such as a yield stress, as 2nd Piola–Kirchhoff stresses.

The Jaumann rate is an objective rate, which means that it transforms properly as a tensor under rigid body motions. The Jaumann derivative of the Cauchy stress is defined as

$$\overset{\nabla}{\boldsymbol{\sigma}} = \dot{\boldsymbol{\sigma}} + \boldsymbol{\sigma} \cdot \boldsymbol{\Omega} - \boldsymbol{\Omega} \cdot \boldsymbol{\sigma} \quad (46.53)$$

Here $\dot{\boldsymbol{\sigma}}$ is the time derivative of the Cauchy stress. The Jaumann stress rate can be considered as the stress rate in the coordinate system that rotates with the material. Stress–strain relations in an Updated Lagrange analysis are defined in this co-rotating coordinate system. The Jaumann stress rate is useful in the case of rate formulated material laws such as plasticity and viscoplasticity.

46.2.2 Total Lagrange

In a Total Lagrange description, strain and stress measures are defined with reference to the undeformed geometry.

46.2.3 Updated Lagrange

The Updated Lagrange analysis, as opposed to the Total Lagrange description, uses an updated reference geometry. Theoretically, many intermediate configurations could serve as a reference frame. However, in Finite Element Analysis it is usual to take the last known equilibrium state, i.e., the situation at the end of the previous step. The incremental equations of motion will be presented here in terms of the Updated Lagrangian formulation. It should be noted that although large displacements, rotations and strain are described correctly, still a constitutive relation appropriate for large strain behaviour has to be used.

46.2.3.1 Weak Equation

Using the principles of virtual work, virtual power or using the Galerkin weighted residual method, the following weak form of stress equilibrium can be derived as follows

$$\delta W = \delta W_1 + \delta W_2 + \delta W_3 = 0 \quad \forall \delta \mathbf{v} \quad (46.54)$$

$$\delta W_1 = \int_V \vec{\nabla} \delta \mathbf{v} : \boldsymbol{\sigma} \, dV \quad (46.55)$$

$$\delta W_2 = \int_V \rho \ddot{\mathbf{u}} \cdot \delta \mathbf{v} \, dV \quad (46.56)$$

$$\delta W_3 = \int_S \mathbf{t} \cdot \delta \mathbf{v} \, dS + \mathbf{f} \cdot \delta \mathbf{v} \, dV \quad (46.57)$$

with a test function $\delta \mathbf{v}$, Cauchy stress $\boldsymbol{\sigma}$, acceleration $\ddot{\mathbf{u}}$, body force \mathbf{f} and surface traction \mathbf{t} . If we consider the test function $\delta \mathbf{v}$ to be a virtual displacement then δW_1 is the virtual internal force, δW_2 is the virtual inertial force, and δW_3 is the virtual external force.

The Jaumann stress rate is related to the deformation rate as

$$\overset{\nabla}{\boldsymbol{\sigma}} = \dot{\boldsymbol{\sigma}} + \boldsymbol{\sigma} \cdot \boldsymbol{\Omega} - \boldsymbol{\Omega} \cdot \boldsymbol{\sigma} = \frac{\dot{\rho}}{\rho} \boldsymbol{\sigma} + \mathbb{L} : \mathbf{D} \quad (46.58)$$

with fourth order constitutive tensor \mathbb{L} and deformation rate \mathbf{D} . Using this rate type constitutive model we can derive δW_1 from

$$\delta W_1 = \int_0^t \delta \dot{W}_1 dt \quad (46.59)$$

After taking the time derivative of Eq. (46.55) and using Eq. (46.58) the required expression for $\delta \dot{W}_1$ reads, see Van den Boogaard [11] and Huétink [46]:

$$\delta \dot{W}_1 = \int_V \left(\delta \mathbf{D} : (\mathbb{L} - \mathbb{I} \cdot \boldsymbol{\sigma} - \boldsymbol{\sigma} \cdot \mathbb{I}) : \mathbf{D} + (\vec{\nabla} \delta \mathbf{v} \cdot \dot{\mathbf{u}} \overleftarrow{\nabla}) : \boldsymbol{\sigma} \right) dV \quad \forall \delta \mathbf{v} \quad (46.60)$$

with \mathbb{I} the fourth order unit tensor with components $\mathbb{I}_{ijkl} = \delta_{ik} \delta_{jl}$.

46.2.3.2 Finite Element Equations

By using displacement and test functions that are interpolated from discrete values in the nodes of elements, in combination with numerical integration in space and time, the weak form can be transformed to a system of finite element equations that can be solved numerically.

Interpolation. The displacement field is discretized in space by means of interpolation functions Ψ

$$\mathbf{u} = \sum_n \Psi_n \mathbf{u}_n \quad (46.61)$$

with \mathbf{u}_n the displacement in a node. The velocity, acceleration and test function are interpolated analogously. The deformation rate \mathbf{D} is interpolated with

$$\mathbf{D} = \sum_n \mathbf{B}_n \dot{\mathbf{u}}_n \quad \mathbf{B}_n = \frac{1}{2} \left((\mathbb{I} + \mathbb{T}) : (\vec{\nabla} \Psi_n \mathbf{I}) \right) \quad (46.62)$$

where \mathbb{T} is the transpose operator with components $\mathbb{T}_{ijkl} = \delta_{jk} \delta_{il}$, and where \mathbf{v} is the velocity.

Incremental iterative solution. In the following, the subscript n in \mathbf{u}_n is dropped, which means that \mathbf{u} is redefined to the vector containing the nodal displacements. The system of governing equations for a transient dynamic problem at time t after space discretization is written as

$$\mathbf{M}^t \ddot{\mathbf{u}} + \mathbf{C}^t \dot{\mathbf{u}} + {}^t \mathbf{f}_{\text{int}} = {}^t \mathbf{f}_{\text{ext}} \quad (46.63)$$

where \mathbf{M} is the mass matrix, derived from Eq. (46.56); \mathbf{C} is the damping matrix; \mathbf{f}_{ext} is the external force vector, derived from Eq. (46.57). Vectors $\ddot{\mathbf{u}}$, $\dot{\mathbf{u}}$, and \mathbf{u} are the resulting acceleration, velocity, and displacement vectors. Vector \mathbf{f}_{int} is the internal set of forces opposing the displacements, derived from Eq. (46.55).

During direct time integration, displacement increments $\Delta \mathbf{u} = {}^{t+\Delta t} \mathbf{u} - {}^t \mathbf{u}$ are determined at discrete time points in an iterative fashion, with subscript (i) denoting the iteration number, starting with 1.

$$(a\mathbf{M} + b\mathbf{C} + {}^{t+\Delta t} \mathbf{K}) \cdot (\Delta \mathbf{u}_{(i)} - \Delta \mathbf{u}_{(i-1)}) = {}^{t+\Delta t} \mathbf{f}_{\text{ext}} - {}^{t+\Delta t} \mathbf{f}_{\text{int.}(i-1)} \quad (46.64)$$

Internal force vector. The internal force vector ${}^{t+\Delta t}\mathbf{f}_{\text{int.}(i-1)}$ is determined at state $t + \Delta t$, with derivatives to the reference coordinate frame at $t + \Delta t$.

$${}^{t+\Delta t}\mathbf{f}_{\text{int.}(i-1)} = \sum_m {}^{t+\Delta t}{}_{t+\Delta t} \int_V \mathbf{B}_m^* : \boldsymbol{\sigma} \, dV \quad (46.65)$$

This means that besides the stress also the strain interpolation \mathbf{B}_m^* and volume contributions dV must be updated in each iteration. \mathbf{B}_m^* is the transposed matrix of \mathbf{B}_n .

$$\mathbf{B}_m^* = \frac{1}{2} \left((\mathbf{I} \Psi_m \overleftarrow{\nabla}) : (\mathbb{I} + \mathbb{T}) \right) \quad (46.66)$$

The stress update is performed via integration of the Jaumann rate, assuming a constant velocity inside the step:

$$\dot{\mathbf{u}} = \frac{\Delta \mathbf{u}}{\Delta t} \quad (46.67)$$

The stress at t based on the reference frame at t is first rotated to the reference frame at $t + \alpha \Delta t$. Then the stress increment at $t + \alpha \Delta t$ is determined and superposed. Finally the stress at state $t + \Delta t$ in reference frame $t + \alpha \Delta t$ is rotated to the end of the increment. This reads:

$${}^{t+\Delta t}{}_{t+\Delta t} \boldsymbol{\sigma} = {}_t \mathbf{R} \cdot \left({}_t \mathbf{R} \cdot {}^t \boldsymbol{\sigma} \cdot {}_t \mathbf{R}^T + \int_t^{t+\Delta t} \dot{\boldsymbol{\sigma}} \, dt \right) \cdot {}_t \mathbf{R}^T \quad (46.68)$$

$$\int_t^{t+\Delta t} \dot{\boldsymbol{\sigma}} \, dt = \mathbb{L} : {}_{t+\alpha \Delta t} \mathbf{D} \Delta t \quad (46.69)$$

The transformation tensor

$${}_t \mathbf{R} = {}^{t+\alpha \Delta t}(\mathbf{F} \cdot \mathbf{U}^{-1}) \quad (46.70)$$

represents the rotation of the reference coordinate frame at t to the reference coordinate frame at $t + \alpha \Delta t$.

Stiffness matrix. The stiffness matrix consists of different contributions:

$$\mathbf{K} = \sum_n \sum_m (\mathbf{K}_{mn} + \mathbf{K}_{mn_S}) \quad (46.71)$$

with \mathbf{K}_{mn} from Eq. (46.60):

$$\mathbf{K}_{mn} = \int_V \mathbf{B}_m^* : (\mathbb{L} - \mathbb{I} \cdot \boldsymbol{\sigma} - \boldsymbol{\sigma} \cdot \mathbb{I}) : \mathbf{B}_n + \mathbf{I} \left(\overrightarrow{\nabla} \Psi_m \cdot \boldsymbol{\sigma} \cdot \overrightarrow{\nabla} \Psi_n \right) dV \quad (46.72)$$

where \mathbf{B}_n comes from Eq. (46.62) and \mathbf{B}_m^* from Eq. (46.66). The contribution from a nonconservative traction load comes from Eq. (46.57):

$$\mathbf{K}_{mn_S} = - \int_S \Psi_m \mathbf{t} \overrightarrow{\nabla}_S \Psi_n \, dS \quad (46.73)$$

46.2.4 Nonconservative Loading

In Finite Element Analysis it usually is sufficient to calculate the force vector at the start of the analysis, or at most to combine some basic force vectors at the start of an increment. This is an efficient way of analysis if the loading does not depend on the displacements. If the loading (magnitude or direction) does depend on the displacements of the model, the load vector must be set up based on the current displacements. An obvious example of this is water pressure on a containment wall in combination with large displacements.

In DIANA these types of loading are described as *nonconservative*. The part of the load that is defined as nonconservative is calculated in every iteration. Equilibrium is now based on the internal and external force vector which both are calculated after the prediction of the displacements. Since the forces now depend on the displacements, there formally is a contribution of the nonconservative loading to the tangential stiffness matrix. However, this leads to an asymmetric matrix and DIANA disregards it.

46.2.5 Contact Analysis

When performing a contact analysis, DIANA automatically invokes the Constrained Minimization Solver, to solve the contact problem. This solver can *not* be used in combination with Line Search algorithms [§ 46.1.3 p. 557]. Therefore, contact analysis and Line Search do not go together.

Contact analysis is only possible with a symmetric tangent stiffness matrix. Therefore, contact analysis cannot be performed when the tangent stiffness matrix becomes nonsymmetric, for instance due to friction or dilatancy in plasticity analysis.

Chapter 47

Element Results and Nodal Forces

In Chapter 45 the solution of the displacement vector \mathbf{u} has been described. This solution is the basis for determination of the element strains and stresses and nodal reaction forces. Below a description is given how these results are derived.

47.1 Element Strains

To determine the strains of an element, the nodal results for this element are transformed to the local Cartesian coordinate system of the element and placed in the vector \mathbf{u}_e . As described in § 44.1.2 on page 531, the strain-displacement relation is defined by the matrix \mathbf{B} as follows

$$\boldsymbol{\varepsilon} = \mathbf{B}\mathbf{u}_e \quad (47.1)$$

This relation is valid at any point within the element. But as the Finite Element Method minimizes the error at the integration points, it is obvious that the strains will be determined at these points. So for each integration point the expression

$$\boldsymbol{\varepsilon}(\xi, \eta, \zeta) = \mathbf{B}(\xi, \eta, \zeta)\mathbf{u}_e \quad (47.2)$$

will be evaluated at the standard values for ξ , η and ζ .

47.1.1 Equivalent Von Mises Strain

DIANA calculates the equivalent Von Mises strain according to

$$\varepsilon_{\text{eq}} = \frac{2}{3} \sqrt{\frac{3(e_{xx}^2 + e_{yy}^2 + e_{zz}^2)}{2} + \frac{3(\gamma_{xy}^2 + \gamma_{yz}^2 + \gamma_{zx}^2)}{4}} \quad (47.3)$$

With the deviatoric strains:

$$\begin{aligned} e_{xx} &= +\frac{2}{3}\varepsilon_{xx} - \frac{1}{3}\varepsilon_{yy} - \frac{1}{3}\varepsilon_{zz} \\ e_{yy} &= -\frac{1}{3}\varepsilon_{xx} + \frac{2}{3}\varepsilon_{yy} - \frac{1}{3}\varepsilon_{zz} \\ e_{zz} &= -\frac{1}{3}\varepsilon_{xx} - \frac{1}{3}\varepsilon_{yy} + \frac{2}{3}\varepsilon_{zz} \end{aligned} \quad (47.4)$$

The engineering strains γ are defined as:

$$\gamma_{ij} = 2 \times \varepsilon_{ij} \quad (47.5)$$

For some calculations the strains are placed in a strain matrix \mathbf{E} which for the general three-dimensional strain situation is given by

$$\mathbf{E} = \begin{bmatrix} \varepsilon_{xx} & \varepsilon_{xy} & \varepsilon_{xz} \\ \varepsilon_{xy} & \varepsilon_{yy} & \varepsilon_{yz} \\ \varepsilon_{xz} & \varepsilon_{yz} & \varepsilon_{zz} \end{bmatrix} \quad (47.6)$$

47.1.2 Principal Strains

DIANA calculates the principal strains $\varepsilon_{1,2,3}$ as the roots of Eq. (47.7) ordered such that $\varepsilon_1 \geq \varepsilon_2 \geq \varepsilon_3$.

$$\begin{bmatrix} \varepsilon_{xx} - \lambda & \varepsilon_{xy} & \varepsilon_{xz} \\ \varepsilon_{xy} & \varepsilon_{yy} - \lambda & \varepsilon_{yz} \\ \varepsilon_{xz} & \varepsilon_{yz} & \varepsilon_{zz} - \lambda \end{bmatrix} = 0 \quad (47.7)$$

For plane strain and axisymmetric elements, however, the third principal strain ε_3 is always the out-of-plane strain, while the first and second principal strains are the inplane principal strains ordered such that $\varepsilon_1 \geq \varepsilon_2$.

47.1.3 Volumetric Strain

DIANA calculates the volumetric strain ε_{vol} by summation of the principal strains:

$$\varepsilon_{\text{vol}} = \varepsilon_1 + \varepsilon_2 + \varepsilon_3 \quad (47.8)$$

47.1.4 Distributed Seismic Moment

Plastic fault slip in the subsurface can lead to seismic events. The seismic moment M_0 is a quantitative measure of the seismic energy released during the rupture of a fault. Aki [1] defines M_0 as the product of the amount of relative shear displacement r of the fault, the area of the fault rupture, and the shear modulus G of the adjacent rock formation through which the fault takes place. M_0 can be calculated as integral over the slipping fault of the distributed seismic moment \mathcal{P}_S which DIANA can calculate and output via the STRAIN DISSEI option. Such integration can be performed via the function RESULT CALCULATE INTEGRATE in the iDIANA Results environment [Vol. iDIANA].

The distributed seismic moment is defined as the product of shear stiffness of the fault, i.e., the second stiffness coefficient D_{22} of the plane interface element, and the relative shear displacement r of the fault:

$$\mathcal{P}_S = D_{22} \times r \quad (47.9)$$

For this purpose it is strongly recommended to define D_{22} as representative for the shear modulus of the adjacent rock formations according to

$$D_{22} = \frac{E}{2(1+\nu)h} \quad (47.10)$$

where E is the Young's modulus of the adjacent rock, ν is the Poisson's ratio of the adjacent rock, and h is the thickness of the fault for which a minimum value of 1 meter should be applied. See also the material parameters DSSX, DSSY and/or DSSZ in Volume *Material Library*. It is the user's responsibility to make sure that the input of the fault shear stiffness in the DIANA calculations is correct and based on Eq. (47.10) for a proper determination of the seismic moment.

DIANA derives the relative shear displacement r from the traction vector $\Delta \mathbf{u}$ which, for plane structural interface elements, has three components

$$\Delta \mathbf{u} = \begin{Bmatrix} \Delta u_x \\ \Delta u_y \\ \Delta u_z \end{Bmatrix} \quad (47.11)$$

where x denotes the normal direction and y and z the first and second tangential direction [Vol. *Element Library*]. You can specify the second tangential direction via the XAXIS input item in table 'GEOMET'. The relative shear displacement r is derived as follows.

$$r = \begin{cases} -\frac{\Delta u_z}{\|\Delta u_z\|} \sqrt{\Delta u_z^2 + \Delta u_y^2} & \text{if } \Delta u_z \neq 0 \\ -\|\Delta u_y\| & \text{if } \Delta u_z = 0 \end{cases} \quad (47.12)$$

Positive values of r represent normal fault slip, i.e., fault slip whereby the hanging wall moves downwards relative to the footwall. Negative values of r represent reverse fault slip, i.e., fault slip whereby the hanging wall moves upwards relative to the footwall. The sign convention for the distributed seismic moment is equal: positive values denote normal fault slip, whereas negative values denote reverse fault slip.

When both types of fault slip occur simultaneously on the same fault plane, a distinction has to be made between negative and positive values during integration in order to avoid an underestimation of the seismic moment. Therefore, the result items \mathcal{P}_{S-} and \mathcal{P}_{S+} for respective negative and positive values of \mathcal{P}_S are output. The command **RESULTS CALCULATE INTEGRATE SURFACE** in the jDIANA Results environment integrates then solely over positive | negative values, resulting in a positive | negative seismic moment representative for normal | reverse fault slip.

47.2 Element Stresses

In order to determine the stresses, the stress-strain relation \mathbf{D} will be used.

$$\boldsymbol{\sigma} = \mathbf{D} (\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_0) + \boldsymbol{\sigma}_0 \quad (47.13)$$

Here $\boldsymbol{\varepsilon}_0$ and $\boldsymbol{\sigma}_0$ are input entities, whereas $\boldsymbol{\varepsilon}$ is determined according to Eq. (47.1). Above expression is used to determine the stresses in the integration points only. Therefore the same accuracy argument holds as for the strains.

For some calculations the stresses are placed in a stress matrix \mathbf{S} which for the general three-dimensional stress situation is given by

$$\mathbf{S} = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{xy} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{yz} & \sigma_{zz} \end{bmatrix} \quad (47.14)$$

47.2.1 Equivalent Von Mises Stress

DIANA calculates the equivalent Von Mises stress according to

$$\sigma_{eq} = \sqrt{\frac{3(s_{xx}^2 + s_{yy}^2 + s_{zz}^2)}{2} + 3(\sigma_{xy}^2 + \sigma_{yz}^2 + \sigma_{zx}^2)} \quad (47.15)$$

With the deviatoric stresses:

$$\begin{aligned} s_{xx} &= +\frac{2}{3}\sigma_{xx} - \frac{1}{3}\sigma_{yy} - \frac{1}{3}\sigma_{zz} \\ s_{yy} &= -\frac{1}{3}\sigma_{xx} + \frac{2}{3}\sigma_{yy} - \frac{1}{3}\sigma_{zz} \\ s_{zz} &= -\frac{1}{3}\sigma_{xx} - \frac{1}{3}\sigma_{yy} + \frac{2}{3}\sigma_{zz} \end{aligned} \quad (47.16)$$

47.2.2 Principal Stresses

DIANA calculates the principal stresses $\sigma_{1,2,3}$ as the roots of Eq. (47.17) ordered such that $\sigma_1 \geq \sigma_2 \geq \sigma_3$.

$$\begin{bmatrix} \sigma_{xx} - \lambda & \sigma_{xy} & \sigma_{xz} \\ \sigma_{xy} & \sigma_{yy} - \lambda & \sigma_{yz} \\ \sigma_{zx} & \sigma_{yz} & \sigma_{zz} - \lambda \end{bmatrix} = 0 \quad (47.17)$$

For plane strain and axisymmetric elements, however, the third principal stress σ_3 is always the out-of-plane stress, while the first and second principal stresses are the inplane principal stresses ordered such that $\sigma_1 \geq \sigma_2$.

47.2.3 Plastic Yield

DIANA calculates the plastic yield Y_p from the deviatoric and shear stresses according to

$$Y_p = \frac{s_{xx}^2 + s_{yy}^2 + s_{zz}^2 + 2(\sigma_{xy}^2 + \sigma_{yz}^2 + \sigma_{zx}^2) - 2\sigma_y}{3} \quad (47.18)$$

Where σ_y is the yield strength.

47.2.4 Pressure

DIANA can calculate and output a ‘pressure’ p from three stress components according to

$$p = -\frac{1}{3} (\sigma_{xx} + \sigma_{yy} + \sigma_{zz}) \quad (47.19)$$

47.2.5 Stress Invariants

DIANA calculates the stress invariants p' , q , and θ defined as

$$\begin{aligned} p' &= -\frac{1}{3} (\sigma_1 + \sigma_2 + \sigma_3) \\ q &= \sqrt{\frac{1}{2} ((\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2)} \\ \theta &= \frac{1}{3} \arcsin \left(-\frac{27}{2} \frac{(\sigma_1 + p')(\sigma_2 + p')(\sigma_3 + p')}{q^3} \right) \end{aligned} \quad (47.20)$$

In this form, p' is the isotropic stress or mean stress, q is the equivalent shear stress, and θ is referred to as Lode angle. The equivalent shear stress is a composition of deviatoric stress components. Under triaxial stress conditions where $\sigma_2 = \sigma_3$, q is just the principal stress difference: $|\sigma_1 - \sigma_3|$.

47.2.6 Reinforcement Moments and Forces

The civil engineer often is interested in the amount and direction of reinforcement needed in concrete structures (e.g. plates), to prevent failure of the structure. DIANA allows for a relatively easy determination of the forces and moments which are to be supported by the reinforcement. The engineer may use this data in designing the reinforcement. The theoretical considerations are completely based on research by Merks [59].

Although the behaviour of reinforced concrete is essentially nonlinear, a fair approximation is gained by calculating reinforcement moments and forces from a linear elastic analysis. Doing so, the following assumptions are used:

- Only the reinforcement steel supports the tension forces.
- The concrete only supports compressive forces.
- Stresses from a linear elastic calculation can be used (thus, redistribution of stresses due to concrete damage, etc. is neglected).
- Reinforcement in two directions (not necessarily perpendicular).
- Straight reinforcement.
- As few reinforcement as possible.

By means of these assumptions, the moments and/or forces which need to be supported by the reinforcement can be determined. In this derivation of reinforcement moments/forces, several approximations are used.

47.2.6.1 Transformation Rules for Stresses, Forces and Moments

Stresses, etc. are firstly determined with respect to the element axes. For determination of stresses, forces and moments acting on the reinforcement, the component of these stress quantities are needed in the frame of the reinforcement however. For this purpose, transformation rules are needed. These rules are presented in this section.

Consider the Cauchy stresses following from a linear elastic analysis:

$$\sigma_{xx}, \sigma_{yy}, \sigma_{yx} = \sigma_{xy} \quad (47.21)$$

In the following, we ever use the fact that stress components are symmetric: $\sigma_{yx} = \sigma_{xy}$. The stresses can be stored in a matrix:

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} \\ \sigma_{xy} & \sigma_{yy} \end{bmatrix} \quad (47.22)$$

This matrix contains the components of the Cauchy stress tensor with respect to the element x and y axes. Stress components in the frame dictated by the reinforcement, x' and y' , are required however [Fig. 47.1]. Note that α is the angle between x and x' and β

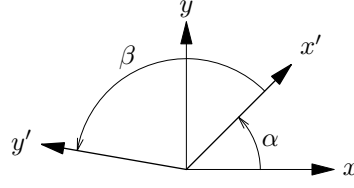


Figure 47.1: Reinforcement directions

is the angle between x' and y' . The outer product of x' and y' always coincides with the element z -axis. The stress components in the reinforcement frame can be shown to be:

$$\sigma' = \frac{1}{\sin \beta^2} \mathbf{R}_\beta \mathbf{R}_\alpha \sigma \mathbf{R}_\alpha^T \mathbf{R}_\beta^T \quad (47.23)$$

where \mathbf{R}_α and \mathbf{R}_β are matrices defined by

$$\mathbf{R}_\alpha = \begin{bmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{bmatrix} \quad \mathbf{R}_\beta = \begin{bmatrix} \sin \beta & -\cos \beta \\ 0 & 1 \end{bmatrix} \quad (47.24)$$

It can be shown that the same transformation rule holds for moments (m_{xx} , m_{yy} , m_{xy}):

$$\mathbf{M}' = \frac{1}{\sin \beta^2} \mathbf{R}_\beta \mathbf{R}_\alpha \mathbf{M} \mathbf{R}_\alpha^T \mathbf{R}_\beta^T \quad (47.25)$$

where \mathbf{M}' is a matrix containing the moments m'_{xx} , m'_{yy} and $m'_{xy} = m'_{yx}$, and that it also holds for forces (n_{xx} , n_{yy} , n_{xy}):

$$\mathbf{N}' = \frac{1}{\sin \beta^2} \mathbf{R}_\beta \mathbf{R}_\alpha \mathbf{N} \mathbf{R}_\alpha^T \mathbf{R}_\beta^T \quad (47.26)$$

where \mathbf{N}' is a matrix containing the forces n'_{xx} , n'_{yy} and $n'_{xy} = n'_{yx}$. Definitions for primary forces and moments for elements are given in Volume *Element Library*. The quantities \mathbf{M}' and \mathbf{N}' represent moments and forces in the reinforcement frame per unit length perpendicular to the reinforcement axes, e.g. n'_{xx} is a force in reinforcement direction x' per unit length perpendicular to x' .

47.2.6.2 Reinforcement Moments Only

If membrane behaviour can be neglected, the reinforcement only needs to support moments. These moments follow from the condition that as few reinforcement as possible is to be used. Following this condition, the moments with respect to the x' y' frame read

$$m'_1 = m'_{xx} \pm m'_{xy} \quad m'_2 = m'_{yy} \pm m'_{xy} \quad (47.27)$$

These formulas represent reinforcement moments per unit distance perpendicular to the reinforcement axes x' and y' [Fig. 47.2]. And thus, the following moments need to be

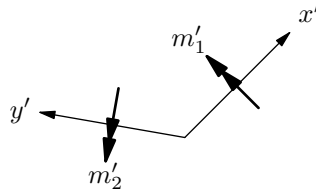


Figure 47.2: Reinforcement moments \perp reinforcement

supported by the reinforcement in the upper and lower plane of the plate.

$$m'_{1,\text{up}} = m'_{xx} + |m'_{xy}| \quad m'_{2,\text{up}} = m'_{yy} + |m'_{xy}| \quad (47.28)$$

$$m'_{1,\text{lo}} = m'_{xx} - |m'_{xy}| \quad m'_{2,\text{lo}} = m'_{yy} - |m'_{xy}| \quad (47.29)$$

In $m'_{1,lo}$ the character m stands for moment, the superscript $'$ indicates a reinforcement variable, subscript $_1$ states that the moment is in x' direction (per unit length $\perp x'$) and subscript $_{lo}$ states that it is the reinforcement moment for the lower plane. The upper and lower planes are defined in positive and negative element z -direction respectively. Additionally, the transverse shear forces q_x and q_y are used to dimension transverse shear reinforcement by means of the quantity

$$q' = \sqrt{q_x^2 + q_y^2} \quad (47.30)$$

The latter five stress quantities are used to dimension reinforcement in plates. Positive moments $m'_{1,up}$ or $m'_{2,up}$ in the upper plane require reinforcement respectively in the x' and y' direction. In the lower plane however, due to the definition of moments, negative moments $m'_{1,lo}$ or $m'_{2,lo}$ require reinforcement respectively in the x' and y' direction.

47.2.6.3 Reinforcement Forces Only

If membrane forces are present, it is no longer possible to express the reinforcement loading by moments. Reinforcement forces, expressing the forces which must be supported by the reinforcement, are calculated instead. These reinforcement forces per unit length perpendicular to the reinforcement directions x' and y' are given by

$$n'_{1,up} = \frac{n'_{xx}}{2} + \left| \frac{n'_{xy}}{2} \right| \quad n'_{2,up} = \frac{n'_{yy}}{2} + \left| \frac{n'_{xy}}{2} \right| \quad (47.31)$$

$$n'_{1,lo} = \frac{n'_{xx}}{2} + \left| \frac{n'_{xy}}{2} \right| \quad n'_{2,lo} = \frac{n'_{yy}}{2} + \left| \frac{n'_{xy}}{2} \right| \quad (47.32)$$

These formulas represent forces in reinforcement directions per unit distance perpendicular to the reinforcement directions [Fig. 47.3]. Note that the forces in upper and lower

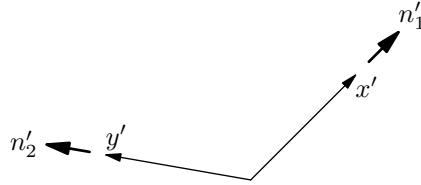


Figure 47.3: Forces in reinforcement directions

plane are the same, in the absence of bending moments. The latter four stress quantities are used to dimension reinforcement in membranes. Reinforcement is required for positive forces in the corresponding direction; e.g. a positive force $n'_{1,up}$ denotes a tension force and hence reinforcement in x' direction is necessary.

47.2.6.4 Combined Moments and Forces

If both membrane forces and bending moments are present, the reinforcement loading is expressed by forces equivalent with the combined results of these membrane forces and bending moments. These 'combined' forces can also be determined if only bending moments are present (plate elements). This section outlines that the required reinforcement is not just the addition of reinforcement needed for separate membrane and bending behaviour. As the membrane and torsional shear stresses may act in opposite direction, these stresses should be taken into account with care [Fig. 47.4].

The combined reinforcement forces read, in x' -direction

$$n'^c_{xx,up} = \frac{n'_{xx}}{2} + \frac{m'_{xy}}{z_d} \quad n'^c_{xy,up} = \left| \frac{n'_{xy}}{2} + \frac{m'_{xy}}{z_d} \right| \quad (47.33)$$

$$n'^c_{xx,lo} = \frac{n'_{xx}}{2} - \frac{m'_{xy}}{z_d} \quad n'^c_{xy,lo} = \left| \frac{n'_{xy}}{2} - \frac{m'_{xy}}{z_d} \right| \quad (47.34)$$

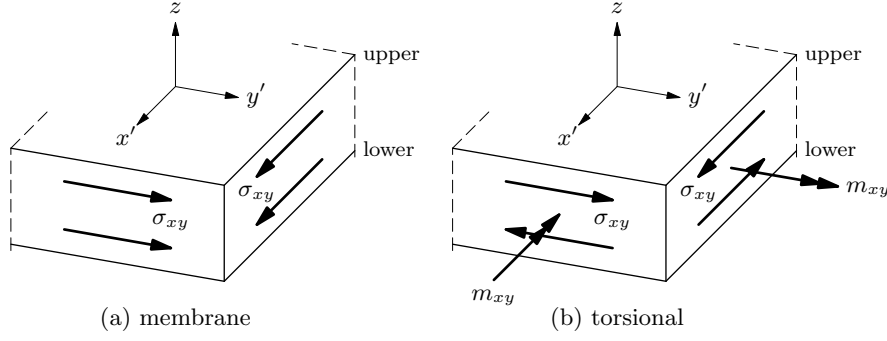


Figure 47.4: Shear stresses

and in y' -direction:

$$n_{yy,up}^{lc} = \frac{n'_{yy}}{2} + \frac{m'_{yy}}{z_d} \quad n_{xy,up}^{lc} = \left| \frac{n'_{xy}}{2} + \frac{m'_{xy}}{z_d} \right| \quad (47.35)$$

$$n_{yy,lo}^{lc} = \frac{n'_{yy}}{2} - \frac{m'_{yy}}{z_d} \quad n_{xy,lo}^{lc} = \left| \frac{n'_{xy}}{2} - \frac{m'_{xy}}{z_d} \right| \quad (47.36)$$

where z_d denotes the ‘absolute internal beam arm’ for the moment. This beam arm follows from

$$z_d = (h_t - co) \times z_r \quad (47.37)$$

with h_t the local thickness, co the coverage (distance from reinforcement center to outer plane) and z_r the ‘relative internal beam arm’. The relative internal beam arm z_r in the case of mainly bending is about 80 to 90 percent of the effective height ($h_t - co$). If normal forces are present then z_r can be much less. In general, z_r depends on the ratio of bending moments and membrane forces. Several examples for determination of z_r are given further.

Finally, the reinforcement forces per unit length perpendicular to the reinforcement directions x' and y' are given by

$$n_{1,up}^{lc} = n_{xx,up}^{lc} + n_{xy,up}^{lc} \quad n_{2,up}^{lc} = n_{yy,up}^{lc} + n_{xy,up}^{lc} \quad (47.38)$$

$$n_{1,lo}^{lc} = n_{xx,lo}^{lc} + n_{xy,lo}^{lc} \quad n_{2,lo}^{lc} = n_{yy,lo}^{lc} + n_{xy,lo}^{lc} \quad (47.39)$$

Additionally, the transverse shear forces q_x and q_y are used to dimension transverse shear reinforcement by means of the quantity:

$$q' = \sqrt{q_x^2 + q_y^2} \quad (47.40)$$

The latter five stress quantities are used to dimension reinforcement for combined bending–membrane behaviour. Reinforcement is required for positive forces in the corresponding direction, e.g. a positive force $n_{1,up}^{lc}$ denotes a tension force and hence upper plane reinforcement in x' direction is necessary.

47.2.6.5 Examples for the Internal Beam Arm

The absolute internal beam arm z_d follows from user-specified values for coverage co and relative internal beam arm z_r . The value of z_r to be specified strongly depends on the ratio of moment and normal force. This is illustrated in the following examples.

Bending only $n_a = n_b$ and $n_a \times z_d = m$ [Fig. 47.5].

Bending plus compressive force In the case of a compressive normal force, an additional eccentricity should be taken into account¹ [Fig. 47.6]: $n_b - n_a = n_e$ and $n_a \times z_d = m$.

¹ See for instance Dutch regulations VB74-84, art. E304.3.3, NEN 3880.

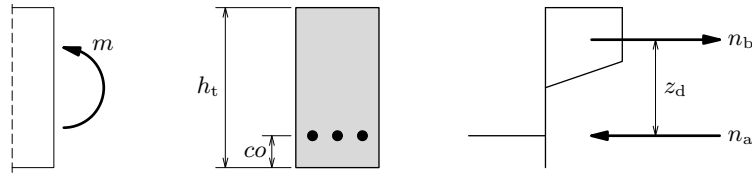


Figure 47.5: Internal beam arm for bending only

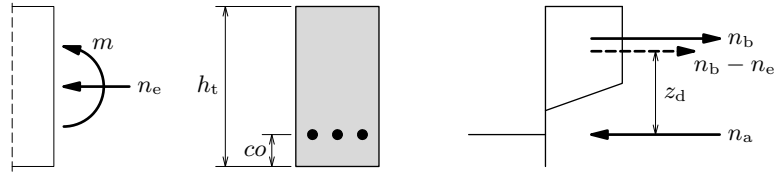


Figure 47.6: Internal beam arm for bending plus compressive force

Bending plus large tension force $n_b + n_a = n_e$ and $(n_a - \frac{1}{2}n_e) \times z_d = m$ [Fig. 47.7].

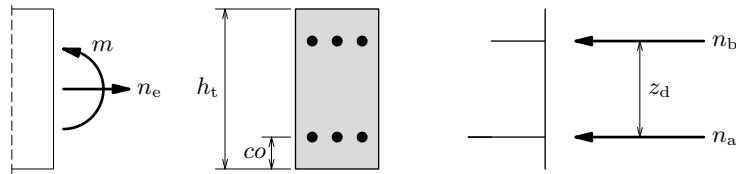


Figure 47.7: Internal beam arm for bending plus large tension force

47.2.6.6 Numerical Example

Consider the following stress situation with bending moments

$$m_{xx} = 500 \text{ kNm/m} \quad m_{yy} = 100 \text{ kNm/m} \quad m_{xy} = 100 \text{ kNm/m}$$

and with membrane forces

$$n_{xx} = -250 \text{ N/mm} \quad n_{yy} = -50 \text{ N/mm} \quad n_{xy} = 20 \text{ N/mm}$$

with plate thickness $t = 550 \text{ mm}$, coverage $co = 50 \text{ mm}$ and relative beam arm $z_r = 0.8$. Suppose these stresses need to be supported by orthogonal reinforced concrete (with the first reinforcement direction coinciding with x). If there are only bending moments, the required reinforcement should be determined from:

$$\begin{aligned} m'_{1,\text{up}} &= 500 + 100 = 600 \text{ kN} & m'_{2,\text{up}} &= 100 + 100 = 200 \text{ kN} \\ m'_{1,\text{lo}} &= 500 - 100 = 400 \text{ kN} & m'_{2,\text{lo}} &= 100 - 100 = 0 \text{ kN} \end{aligned}$$

If there are only membrane forces, the required reinforcement should be determined from

$$\begin{aligned} n'_{1,\text{up}} &= \frac{-250}{2} + \left| \frac{20}{2} \right| = -115 \text{ N/mm} & n'_{2,\text{up}} &= \frac{-50}{2} + \left| \frac{20}{2} \right| = -15 \text{ N/mm} \\ n'_{1,\text{lo}} &= \frac{-250}{2} + \left| \frac{20}{2} \right| = -115 \text{ N/mm} & n'_{2,\text{lo}} &= \frac{-50}{2} + \left| \frac{20}{2} \right| = -15 \text{ N/mm} \end{aligned}$$

However, as both bending moments and membrane forces are present, the reinforcement must support the combined action. Thus, it must support the following combined forces

(with relative distance $z_d = (550 - 50) \times 0.8 = 400$ mm):

$$n_{xx,\text{up}}^{\prime c} = \frac{-250}{2} + \frac{500000}{400} = 1125 \text{ N/mm}$$

$$n_{xy,\text{up}}^{\prime c} = \left| \frac{20}{2} + \frac{100000}{400} \right| = 260 \text{ N/mm}$$

$$n_{yy,\text{up}}^{\prime c} = \frac{-50}{2} + \frac{100000}{400} = 225 \text{ N/mm}$$

and in the lower plane:

$$n_{xx,\text{lo}}^{\prime c} = \frac{-250}{2} - \frac{500000}{400} = -1375 \text{ N/mm}$$

$$n_{xy,\text{lo}}^{\prime c} = \left| \frac{20}{2} - \frac{100000}{400} \right| = 240 \text{ N/mm}$$

$$n_{yy,\text{lo}}^{\prime c} = \frac{-50}{2} - \frac{100000}{400} = -275 \text{ N/mm}$$

Hence, the following (extreme) values for the reinforcement forces are obtained, in the upper plane:

$$n_{1,\text{up}}^{\prime c} = 1125 + 260 = 1385 \text{ N/mm} \quad n_{2,\text{up}}^{\prime c} = 225 + 260 = 485 \text{ N/mm}$$

and in the lower plane:

$$n_{1,\text{lo}}^{\prime c} = -1375 + 240 = -1135 \text{ N/mm} \quad n_{2,\text{lo}}^{\prime c} = -275 + 240 = -35 \text{ N/mm}$$

Note that the combined action leads to different results than simply adding membrane and bending action. It can be seen that only the upper plane needs reinforcement.

47.2.7 Stress Gradient in Reinforcement Bar

For reinforcement bars DIANA can calculate the gradient of the axial stress σ'_{xx} . This is the derivative of the stress with respect to the local x axis of the bar:

$$\sigma'_{xx} = \frac{d\sigma_{xx}}{dx} \quad (47.41)$$

47.2.8 Shear Stress in Reinforcement Mother Element Connection

The shear stress in the connection between reinforcement bar and mother element is defined as the stress gradient [§ 47.2.7] multiplied by the cross-section of the reinforcement and divided by the perimeter of the reinforcement. If the perimeter is not defined by the user, then DIANA assumes a circular cross-section and calculates the perimeter from the user-defined cross-section.

47.2.9 Shear Capacity and Hydrostatic Pressure Capacity

DIANA can calculate the shear capacity and hydrostatic pressure capacity of stress against the Mohr–Coulomb failure criterion [§ 47.2.9.1], and against the Hoek–Brown failure criterion [§ 47.2.9.2]. For structural interface elements, DIANA can calculate the shear capacity and hydrostatic pressure capacity of stress against the Coulomb friction failure criterion [§ 47.2.9.3].

47.2.9.1 Mohr–Coulomb Failure Criterion

DIANA calculates the shear-stress capacity ψ and the hydrostatic pressure capacity χ of stress against the Mohr–Coulomb failure criterion, which are defined as:

$$\psi = \frac{q}{q_{mc}}, \quad (47.42)$$

and

$$\chi = \frac{3 - \sin \phi}{6 \sin \phi} (q_{mc} - q), \quad (47.43)$$

with q the equivalent shear stress [§ 47.2.5 p. 576] and

$$q_{mc} = \frac{6 \sin \phi}{3 - \sin \phi} p + \frac{6c \cos \phi}{3 - \sin \phi}. \quad (47.44)$$

Where p is the hydrostatic pressure [§ 47.2.5 p. 576], and c and ϕ are the cohesion and the friction angle for the Mohr–Coulomb material, respectively.

47.2.9.2 Hoek–Brown Failure Criterion

DIANA calculates the shear-stress capacity ψ and the hydrostatic pressure capacity χ of stress against the Hoek–Brown failure criterion [89], which are defined as:

$$\psi = \frac{q}{q_{hb}}, \quad (47.45)$$

and

$$\chi = p_{hb} - p \quad (47.46)$$

with q the equivalent shear stress, p the hydrostatic pressure, and θ the Lode angle of the current stress state [§ 47.2.5 p. 576] and q_{hb} the solution of the Hoek–Brown yield surface:

$$F(\sigma) = q_{hb}^2 g^2(\theta) + \sigma_c^* q_{hb} g(\theta) + 3\sigma_c^* p - s\sigma_{ci}^2 = 0 \quad (47.47)$$

and p_{hb} the solution of the Hoek–Brown yield surface:

$$F(\sigma) = q^2 g^2(\theta) + \sigma_c^* q g(\theta) + 3\sigma_c^* p_{hb} - s\sigma_{ci}^2 = 0 \quad (47.48)$$

with

$$\sigma_c^* = m_b \frac{\sigma_{ci}}{3} \quad (47.49)$$

where σ_{ci} is the unconfined compressive strength of the (intact) rock sample, m_b and s are constants of the Hoek–Brown rock plasticity material model [Vol. *Material Library*]. An elliptical function $g(\theta)$ is used to describe the variation of the trace of the surface in the deviatoric planes for a sextant of the stress space, i.e. $-\pi/6 \leq \theta \leq \pi/6$ to eliminate the irregularities of the Hoek–Brown yield surface:

$$g(\theta) = \frac{4(1 - e^2) \cos^2(\pi/6 + \theta) + (1 - 2e)^2}{2(1 - e^2) \cos(\pi/6 + \theta) + (2e - 1)D} \quad (47.50)$$

with

$$D = \sqrt{4(1 - e^2) \cos^2(\pi/6 + \theta) + 5e^2 - 4e} \quad (47.51)$$

where e an eccentricity parameter, which is defined by the ratio of the deviatoric stress q in the tension branch to the one in the compression branch. In DIANA the eccentricity parameter e equals 0.65.

47.2.9.3 Coulomb Friction Failure Criterion

For interface elements, DIANA calculates the shear-stress capacity ψ and the hydrostatic pressure capacity χ of stress against Coulomb friction failure criterion, which are defined as: which are defined as:

$$\psi = \frac{q}{q_{mc}}, \quad (47.52)$$

and

$$\chi = (q_{mc} - q) \cdot \tan \phi \quad (47.53)$$

with

$$q_{mc} = c + p \cdot \tan \phi \quad (47.54)$$

where p is the normal traction and q is defined as the absolute sum of the combined shear tractions.

47.3 Nodal Forces

47.3.1 Internal Nodal Forces

The *internal nodal forces* for an element can be defined as the forces (moments) the element exhibits at its environment through its nodes, which are caused by the internal element stresses only.² As these forces actually are related to the elastic deformation we write

$$\mathbf{f}_{e,in} = \mathbf{K}_e \mathbf{u}_e^{no} - \int_{V_e} \mathbf{B}^T \mathbf{C} \boldsymbol{\varepsilon}_0 dV + \int_{V_e} \mathbf{B}^T \boldsymbol{\sigma}_0 dV \quad (47.55)$$

which can be rewritten as

$$\mathbf{f}_{e,in} = \int_{V_e} \mathbf{B} \boldsymbol{\sigma} dV \quad (47.56)$$

This integral can be solved numerically, as the terms in the integrand are all known at the integration points. Summing the internal nodal forces for all elements yields the vector of internal nodal forces \mathbf{f}_{in} . For nodes that are both unconstrained and non-loaded these are zero or very small.

With internal nodal element forces DIANA offers the possibility to perform the summation only for a selection of elements. With such a selection of elements you define a ‘section’ of the model as an assembly of elements. In that case the internal nodal element forces on the edges of this ‘section’ will in general be non-zero. These internal nodal forces might be interpreted as the internal forces that act on that ‘section’. For internal nodal element forces DIANA also offers the possibility to perform the summation only for the embedded reinforcements or only for the mother elements containing the embedded reinforcements.

47.3.2 External Nodal Loads

The external nodal loads are defined as the vector of applied loads, as the nodal point loads, pressure loads and gravity effects. For one element, the element loads are expressed by the vector $\mathbf{f}_{e,ex}$ as

$$\mathbf{f}_{e,ex}^{el} = \int_{V_e} \mathbf{N} \mathbf{p}_{V_e} dV + \int_{A_e} \mathbf{N} \mathbf{p}_{A_e} dA \quad (47.57)$$

All terms of this equation can be determined by numerical integration per element. Summing the external loads over all elements gives the total external load vector \mathbf{f}_{ex}^{el} . The vector \mathbf{f}_{ex}^{pt} contains the nodal point loads, so the total external nodal loads can be defined as

$$\mathbf{f}_{ex} = \mathbf{f}_{ex}^{el} + \mathbf{f}_{ex}^{pt} \quad (47.58)$$

² This definition has been made for pragmatic reasons only and has not much meaning in linear elastic analysis.

47.3.3 Reaction Forces and Residual Forces

The *residual forces* \mathbf{f}_r are defined as the difference between the external forces \mathbf{f}_{ex} and the internal forces \mathbf{f}_{in} :

$$\mathbf{f}_r = \mathbf{f}_{ex} - \mathbf{f}_{in} \quad (47.59)$$

In case of static equilibrium the internal forces are equal to the external forces for unconstrained degrees of freedom:

$$\mathbf{f}_{in} = \mathbf{f}_{ex} \quad (47.60)$$

Therefore, the residual forces for unconstrained degrees of freedom are zero:

$$\mathbf{f}_r = \mathbf{f}_{ex} - \mathbf{f}_{in} = 0 \quad (47.61)$$

To check the quality of the numerical solution the residual forces can be output. The calculated residual forces should be zero or small compared to the external and internal forces in case of equilibrium for unconstrained degrees of freedom.

For constrained degrees of freedom (supports) the residual force will have a finite value equal to the force that the element model exhibits on the constrained degree of freedom. The *reaction forces* \mathbf{f}_R are defined as the forces that constrained degrees of freedom (supports) exhibits on the element model. Hence, the reaction forces \mathbf{f}_R are opposite to the residual forces in the constrained degrees of freedom:

$$\mathbf{f}_R = -\mathbf{f}_r \quad (47.62)$$

Chapter 48

Dynamic Analysis

The governing equation of motion for a linear dynamic finite element system reads

$$\boxed{\mathbf{M}\ddot{\mathbf{u}} + \mathbf{C}\dot{\mathbf{u}} + \mathbf{K}\mathbf{u} = \mathbf{f}(t)} \quad (48.1)$$

Where \mathbf{M} is the mass matrix, \mathbf{C} the damping matrix and \mathbf{K} the stiffness matrix of the finite element model, $\mathbf{f}(t)$ is the right-hand-side vector of forcing functions and $\ddot{\mathbf{u}}$, $\dot{\mathbf{u}}$ and \mathbf{u} are the resulting acceleration, velocity and displacement vectors. The complete solution \mathbf{u} consists of two contributions

$$\mathbf{u} = \mathbf{u}_{tr} + \mathbf{u}_{st} \quad (48.2)$$

The response \mathbf{u}_{tr} depends on the conditions with which the response is initiated and is induced by the initial conditions

$${}^0\mathbf{u} = \mathbf{u}(t_0) \quad \text{and} \quad {}^0\dot{\mathbf{u}} = \dot{\mathbf{u}}(t_0) \quad (48.3)$$

By contrast \mathbf{u}_{st} is the particular solution which satisfies the forcing function of $\mathbf{f}(t)$ of Eq. (48.1), but does not take account of the initial conditions. This particular solution \mathbf{u}_{st} is denoted as the *steady-state* response, whereas the additional solution \mathbf{u}_{tr} is called the initial *transient* part of the response. In case of static loading, $\mathbf{f}(t) = 0$, the transient response is completely determined by the initial conditions.

Frequency response analysis only studies the steady-state response [§ 48.2]. If the initial transient part is substantially important then you must analyse the complete transient response [§ 48.4]. If nonlinear phenomena, like plasticity or cracking, are present then a direct time integration with Module NONLIN must be used as described in Part IV.

48.1 General Phenomena

All types of structural dynamic analysis have to do with general phenomena like mass and damping, loading and base excitation as outlined in this section.

48.1.1 Mass

Either consistent or lumped mass matrices can be used in structural dynamic analysis. In practice, lumped or diagonal mass matrices are often employed due to their general economy and because they lead to some attractive time integration schemes like explicit methods. DIANA uses an adaptive lumping technique to lump mass matrices. This technique was developed by Hinton et al. [44]. The (diagonal) terms of the lumped mass matrix are scaled proportionally to the diagonal entries of the consistent mass in such a way that the total element mass is conserved. Caution:

Mass lumping may lead to inaccurate results in case of coarse meshes and/or irregular element shapes, for instance two-dimensional elements with very nonuniform thickness.

48.1.2 Damping

Either consistent or lumped damping matrices can be used in structural dynamic analysis. In practice the presence of damping reduces the steady-state response and damps out the transient response. A comprehensive discussion on damping in Finite Element Analysis is given by Spence & Kenchington [82]. A modal analysis [§ 48.2.1] assumes the application of proportional viscous damping and that the damping matrix \mathbf{C} satisfies the orthogonality condition. Modal damping can be employed for this and the magnitude of the damping has to be specified as a percentage of the critical damping factor

$$\phi_i^T \mathbf{C} \phi_j = 2\omega_i \xi_i \delta_{ij} \quad (48.4)$$

where ω is the natural angular frequency and ξ_i is the damping ratio. The critical damping factor is

$$c_{\text{crit}} = 2\sqrt{km} \quad (48.5)$$

Where k is the generalized stiffness $\phi_i^T \mathbf{K} \phi_j \delta_{ij}$ and m is the generalized mass $\phi_i^T \mathbf{M} \phi_j \delta_{ij}$.

48.1.2.1 Continuous Damping

When a form of continuous damping, like viscous dashpots, is included in the finite element model then the direct solution method [§ 48.2.2 p. 589], or the direct time integration procedure [§ 48.4 p. 592] must be used. Discrete places of damping are obtained by specifying properties for spring elements [§ 6.2 p. 104].

48.1.2.2 Rayleigh Damping

Applying a nonmodal solution technique, it is necessary to evaluate the damping matrix \mathbf{C} explicitly and usually viscous damping effects can be included by assumption of *Rayleigh* damping which is of the form

$$\mathbf{C} = a\mathbf{M} + b\mathbf{K} \quad (48.6)$$

where a and b are constants to be determined from given damping ratios.

48.1.2.3 Structural Damping

Another type of damping, frequently employed in dynamic analysis, is structural damping also called hysteretic damping. This type of damping is proportional to displacement but in-phase to velocity of a harmonically oscillating system. In that way the equation of motion is expressed as

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{K}(1 + i\gamma)\mathbf{u} = \mathbf{f}(t) \quad (48.7)$$

where γ is the structural damping factor. Equations like Eq. (48.7) can only be solved in the frequency domain.

48.1.3 Loading

In a frequency response analysis with DIANA the forcing function can be a (deterministic) *harmonic* one. That is, the forcing function is composed of sines and cosines with known amplitudes. Often, other types of loading, like short duration impulsive loads or general forms of long term loads, may be present. For those other types a *direct time integration*, i.e., a transient response analysis, should be applied [§ 48.4].

48.1.4 Base Excitation

The dynamic response like displacements, stresses of a finite element system may not only be induced by prescribed loads, but also by motions of its supported points like an applied base acceleration field in an earthquake analysis.

The displacements, velocities and accelerations can be partitioned into the known applied base movements \mathbf{u}_b and the unknown internal movements \mathbf{u}_i . The equations of motion can be expressed in the same partitioned form as:

$$\begin{bmatrix} \mathbf{M}_{ii} & \mathbf{M}_{ib} \\ \mathbf{M}_{bi} & \mathbf{M}_{bb} \end{bmatrix} \begin{Bmatrix} \ddot{\mathbf{u}}_i^{(a)} \\ \ddot{\mathbf{u}}_b^{(a)} \end{Bmatrix} + \begin{bmatrix} \mathbf{C}_{ii} & \mathbf{C}_{ib} \\ \mathbf{C}_{bi} & \mathbf{C}_{bb} \end{bmatrix} \begin{Bmatrix} \dot{\mathbf{u}}_i^{(a)} \\ \dot{\mathbf{u}}_b^{(a)} \end{Bmatrix} + \begin{bmatrix} \mathbf{K}_{ii} & \mathbf{K}_{ib} \\ \mathbf{K}_{bi} & \mathbf{K}_{bb} \end{bmatrix} \begin{Bmatrix} \mathbf{u}_i^{(a)} \\ \mathbf{u}_b^{(a)} \end{Bmatrix} = \begin{Bmatrix} \mathbf{0} \\ \mathbf{f}_b \end{Bmatrix} \quad (48.8)$$

with \mathbf{f}_b as the unknown reaction forces at the base.

The base accelerations $\ddot{\mathbf{u}}_b$ have a known time history and can be integrated to give the base velocities and displacements.

Taking the first partition, this set of equations, which represent the unknown internal movements, can be written as:

$$\mathbf{M}_{ii}\ddot{\mathbf{u}}_i^{(a)} + \mathbf{C}_{ii}\dot{\mathbf{u}}_i^{(a)} + \mathbf{K}_{ii}\mathbf{u}_i^{(a)} = -\mathbf{M}_{ib}\ddot{\mathbf{u}}_b^{(a)} - \mathbf{C}_{ib}\dot{\mathbf{u}}_b^{(a)} - \mathbf{K}_{ib}\mathbf{u}_b^{(a)} = \hat{\mathbf{f}}_i \quad (48.9)$$

where $\hat{\mathbf{f}}_i$ does represent the equivalent dynamic forces. In this formulation the absolute response $\mathbf{u}^{(a)}$ is directly calculated and the relative response $\mathbf{u}^{(r)}$ is found by subtracting the base movement from the absolute response.

The equations of motion defined by Eq. (48.9) can be solved by any relevant second order integration scheme like the *Newmark Beta* scheme to give the absolute response $\mathbf{u}_i^{(a)}$. The base velocities $\dot{\mathbf{u}}_b$ and base displacements \mathbf{u}_b are found by integrating the known base accelerations $\ddot{\mathbf{u}}_b$ by the same integration scheme as is used for solving the response \mathbf{u}_i from Eq. (48.9).

48.1.4.1 Rigid Base Movement

Under the assumption that all of the base movements in a given direction are identical, that is the base supports cannot move independently, then the movement of the structure $\mathbf{u}_i^{(r)}$ relative to the base is:

$$\begin{aligned} \mathbf{u}_i^{(r)} &= \mathbf{u}_i^{(a)} - \phi_i^{(j)}\alpha^{(j)} \\ \mathbf{u}_b^{(r)} &= \mathbf{u}_b^{(a)} - \phi_b^{(j)}\alpha^{(j)} = \mathbf{0} \\ \phi^{(j)} &= \begin{Bmatrix} \phi_i^{(j)} \\ \phi_b^{(j)} \end{Bmatrix} \end{aligned} \quad (48.10)$$

where $\phi^{(j)}$ is the rigid body movement in the j^{th} direction and $\alpha^{(j)}$ the corresponding time function of the base acceleration.

Since strain energy and viscous damping energy are proportional to differential displacements, it may be thought that:

$$\begin{aligned} [\mathbf{K}] \phi^{(j)} &= \mathbf{0} \\ [\mathbf{C}] \phi^{(j)} &= \mathbf{0} \end{aligned} \quad (48.11)$$

However, this statement is only correct when:

- 1 no rotations are present in the rigid body movement,
- 2 the system damping matrix \mathbf{C} must have viscous properties to satisfy the condition $\mathbf{C}\phi = \mathbf{0}$; in case of Rayleigh damping, $\mathbf{C} = a\mathbf{M} + b\mathbf{K}$, only the stiffness proportional part $b\mathbf{K}$ does satisfy this condition!

By further assuming that

$$\mathbf{M}_{ib} = \mathbf{M}_{bi} = \mathbf{0}$$

i.e. there is no coupling between the base degrees of freedom and the internal degrees of freedom in the mass matrix, then the equation of motion for the response relative to the base supports is

$$\mathbf{M}_{ii}\ddot{\mathbf{u}}_i^{(r)} + \mathbf{C}_{ii}\dot{\mathbf{u}}_i^{(r)} + \mathbf{K}_{ii}\mathbf{u}_i^{(r)} = -\mathbf{M}_{ii}\phi_i^{(j)}\ddot{\alpha}^{(j)} \quad (48.12)$$

and the relative base response $\mathbf{u}^{(r)}$ is now found.

This form is commonly used in seismic analysis and, as stated before, this form is only valid if the base supports move with the same amplitude and direction.

48.1.4.2 Equations of Motion in a Relative Coordinate System

The equations of motion for both the absolute and relative response have the same characteristics because the natural frequencies and corresponding mode shapes are identical.

If the results as obtained from the equations of motion for the absolute response [Eq. (48.9)] have to be equated to the results as found from the equations of motion for the response relative to the base supports [Eq. (48.12)], then the equations of motion for the absolute response have to be slightly modified.

By using the relations in Eq. (48.10), Eq. (48.9) can be rewritten as:

$$\begin{aligned} \mathbf{M}_{ii} \left(\ddot{\mathbf{u}}_i^{(r)} + \phi_i^{(j)} \ddot{\alpha}^{(j)} \right) + \mathbf{C}_{ii} \left(\dot{\mathbf{u}}_i^{(r)} + \phi_i^{(j)} \dot{\alpha}^{(j)} \right) + \mathbf{K}_{ii} \left(\mathbf{u}_i^{(r)} + \phi_i^{(j)} \alpha^{(j)} \right) &= \\ -\mathbf{M}_{ib} \left(\phi_b^{(j)} \ddot{\alpha}^{(j)} \right) - \mathbf{C}_{ib} \left(\phi_b^{(j)} \dot{\alpha}^{(j)} \right) - \mathbf{K}_{ib} \left(\phi_b^{(j)} \alpha^{(j)} \right) &= \hat{\mathbf{f}}_i \end{aligned} \quad (48.13)$$

To satisfy the assumptions which have been made in the equations of motion for the relative response, i.e. no coupling between the base degrees of freedom and the internal degrees of freedom in the mass matrix \mathbf{M} , and both $\mathbf{K} \phi^{(j)} = \mathbf{0}$ and $\mathbf{C} \phi^{(j)} = (a\mathbf{M} + b\mathbf{K}) \phi^{(j)} = \mathbf{0}$, the following corrections must be applied:

- 1 the contribution of $\mathbf{M}_{ib} \left(\phi_b^{(j)} \ddot{\alpha}^{(j)} \right)$ onto the equivalent forces $\hat{\mathbf{f}}_i$ must be ignored,
- 2 to neutralize the contribution of the mass proportional part $a\mathbf{M}$ in the Rayleigh damping matrix \mathbf{C} on the left hand side vector, the contribution $a\mathbf{M} \left(\phi^{(j)} \dot{\alpha}^{(j)} \right)$ must be added to the right hand side vector.

The equivalent force vector $\hat{\mathbf{f}}_i$ on the right hand side is now contributed by:

$$\begin{aligned} \hat{\mathbf{f}}_i &= -[\mathbf{C}_{ib}] \left(\phi_b^{(j)} \dot{\alpha}^{(j)} \right) - [\mathbf{K}_{ib}] \left(\phi_b^{(j)} \alpha^{(j)} \right) \\ &\quad + [a\mathbf{M}_{ii}] \left(\phi_{ii}^{(j)} \dot{\alpha}^{(j)} \right) + [a\mathbf{M}_{ib}] \left(\phi_{ib}^{(j)} \dot{\alpha}^{(j)} \right) \end{aligned} \quad (48.14)$$

When using this expression for the equivalent force vector $\hat{\mathbf{f}}_i$ [Eq. (48.14)], no damping energy due to the base movement is added into the system.

48.2 Frequency Response

Frequency response analysis predicts the linear response of a system subjected to a continuous series of harmonic excitations. When the loading is a deterministic frequency content, a *steady-state* linear dynamic analysis may be performed. Steady-state response is given as a frequency sweep through a specified range of frequencies. The analysis results are given terms of amplitudes and phase angles [§ 48.2.3 p. 590]. For more background theory on frequency response analysis see for instance the books by Craig [21] and by Clough & Penzien [20].

If the forcing function of Eq. (48.1) is periodic, the system can be transformed in terms of frequency components Ω_f and solved in the frequency domain. In this case, the steady-state solution can be obtained in a direct way and the initial conditions are irrelevant.

The steady-state response is determined for the same frequencies that defined the loading input.

$$\mathbf{M}\ddot{\mathbf{u}}(t) + \mathbf{C}\dot{\mathbf{u}}(t) + \mathbf{K}\mathbf{u}(t) = \mathbf{f}e^{i\Omega t} \quad (48.15)$$

$$\mathbf{u}(t) = \mathbf{u}e^{i\Omega t} \quad (48.16)$$

The response for each excitation frequency Ω_f can be determined from Eq. (48.15) where \mathbf{f} is the complex load vector. The real part $\Re\mathbf{f}$ is the *in-phase* component of the load and the imaginary part $\Im\mathbf{f}$ is the *out-of-phase* component. The corresponding solution can now be written in the same form Eq. (48.16) with \mathbf{u} as complex displacement vector with in- and out-of-phase components. Substitution of the solution Eq. (48.16) in the equation of motion Eq. (48.15) results in

$$(\mathbf{K} - \Omega^2\mathbf{M} + i\Omega\mathbf{C})\mathbf{u} = \mathbf{f} \quad (48.17)$$

$$\mathbf{u} = (\mathbf{K} - \Omega^2\mathbf{M} + i\Omega\mathbf{C})^{-1}\mathbf{f} = \mathbf{H}(\Omega)\mathbf{f} \quad (48.18)$$

This substitution may be repeated for each frequency component. The term $\mathbf{H}(\Omega)$, the ‘response function’, is the Fourier transformation of the response matrix. There are two methods to solve Eq. (48.17): an indirect method based on *mode superposition* [§ 48.2.1], and a *direct* method [§ 48.2.2].

48.2.1 Mode Superposition

The principle of the mode superposition technique may be employed providing the system of equations of Eq. (48.1) is *linear*, that is to say, \mathbf{M} , \mathbf{C} and \mathbf{K} remain constant during the response. In a modal analysis, the response will be found by superposition of the response in each mode:

$$\mathbf{u}(\Omega) = \sum_{i=1}^{N_p} \phi_i \alpha_i(\Omega) \quad (48.19)$$

where ϕ_i is the i -th eigenvector of the undamped eigenproblem [§ 45.2 p. 541] and $\alpha_i(\Omega)$ is the i -th generalized modal displacement. Substitution of this equation in Eq. (48.17) and pre-multiplication with the j th eigenvector yields an uncoupled system of equations, if the damping matrix \mathbf{C} is orthogonal with respect to the eigenvectors. In this case the inversion of a diagonal matrix will do and the response may be determined from

$$\mathbf{u} = \mathbf{H}(\Omega)\mathbf{f} = \sum_{i=1}^{N_p} \phi_i (K_{ii} - \Omega^2 M_{ii}, i\Omega C_{ii})^{-1} \phi_i^T \mathbf{f} \quad (48.20)$$

If the damping is not proportional but continuous, for instance when the model comprises discrete dampers like dashpots, the damping matrix \mathbf{C} is not ϕ -orthogonal and its diagonal structure is destroyed. In this case the damping is usually rather strong and, considering that the transformation of Eq. (48.20) is no longer correct, a direct solution method is more suitable.

48.2.2 Direct Solution

The direct solution method requires the solution of a complex system of equations. Complex arithmetic can be avoided by transformation of Eq. (48.17) to a system with real coefficients:

$$\begin{bmatrix} \mathbf{K} - \Omega^2\mathbf{M} & +\Omega\mathbf{C} \\ -\Omega\mathbf{C} & \mathbf{K} - \Omega^2\mathbf{M} \end{bmatrix} \begin{Bmatrix} \Im\mathbf{u} \\ \Re\mathbf{u} \end{Bmatrix} = \begin{Bmatrix} \Im\mathbf{f} \\ \Re\mathbf{f} \end{Bmatrix} \quad (48.21)$$

where Ω is the excitation frequency, $\Im\mathbf{f}$ and $\Re\mathbf{f}$ are the imaginary and real parts of the complex load vector \mathbf{f} , and $\Im\mathbf{u}$ and $\Re\mathbf{u}$ are the imaginary and real parts of the complex displacement vector \mathbf{u} . The system Eq. (48.21) is regular but not always positive definite and can only be solved with an LDU-decomposition [§ 45.1 p. 539].

This method has the advantage that it does not require the eigenfrequencies and eigenmodes to be determined. Unfortunately, there are also two disadvantages. In the

first place, a new system of equations must be solved for every excitation frequency which requires a lot of computing time, particularly if the number of excitations increases. Secondly, the system to be solved tends to a bad condition if an excitation exactly or nearly coincides with one of the (complex) eigenfrequencies. The disadvantages may be overcome by determination of the response function $\mathbf{H}(\Omega)$ via the mode superposition technique.

48.2.3 Analysis Results

To obtain the response of a system subjected to a series of harmonic excitations only the steady-state response is of interest, it is calculated by Modules MODAL and FREQUE. These modules present the resulting output in the frequency domain. Depending on the presence of damping and the type of the excitation function the response will not be in-phase or 180° out-of-phase with the excitation. The real part of a typical nodal degree of freedom direction i is given by

$$u_i(t) = \hat{u}_i \cos(\Omega t - \phi_i) \quad (48.22)$$

where \hat{u}_i is the amplitude and ϕ_i is the phase angle of the response relative to the excitation. DIANA represents all output results by their amplitude and corresponding phase angle with respect to the cosine function!

In addition, dynamic stresses are derived directly from the calculated displacement response vector \mathbf{u} . The steady-state solution of Eq. (48.1) may be assumed to have the form

$$\mathbf{u}(t) = \mathbf{u} e^{i\Omega t} \quad (48.23)$$

where $\mathbf{u} = \Re \mathbf{u} + i \Im \mathbf{u}$ is the complex spatial displacement vector. The element stresses have the same form

$$\boldsymbol{\sigma}_e(t) = \boldsymbol{\sigma}_e e^{i\Omega t} \quad (48.24)$$

where $\boldsymbol{\sigma} = \Re \boldsymbol{\sigma} + i \Im \boldsymbol{\sigma}$ is the complex stress vector. This complex stress vector $\boldsymbol{\sigma}_e$ is now determined via the usual relation

$$\boldsymbol{\sigma}_e = \mathbf{D} \mathbf{B} \mathbf{u}_e \quad (48.25)$$

where \mathbf{D} and \mathbf{B} respectively represent the stress-strain and strain-displacement relation, and \mathbf{u}_e is the complex element displacement response vector. In the same way as the nodal response, only the real part of the element stress components in DIANA are represented by their amplitude $\hat{\sigma}_i$ and phase angle ϕ_i with respect to the cosine function

$$\sigma_i(t) = \hat{\sigma}_i \cos(\Omega t - \phi_i) \quad (48.26)$$

48.3 Response Spectrum Analysis

Response Spectrum Analysis predicts the linear response of a system subjected to a base excitation spectrum. DIANA's Module SPECTR may perform a Response Spectrum Analysis on a finite element model. This section summarizes the background theory of Module SPECTR. The implementation of Module SPECTR is based on the Response Spectrum Method as described by Gupta [37]. When designing a structure we are particularly interested in the maximum forces, which can be evaluated from the maximum relative displacements. The results of a Response Spectrum Analysis are given in terms of individual and combined modal forces.

The implementation of SPECTR assumes that the contribution of damping is negligible and that the base acceleration spectrum has a working direction \mathbf{i} , i.e., a base acceleration value \ddot{u}_{su} . The system of equations for base excitation may then be written as

$$\mathbf{M}\ddot{\mathbf{u}}(t) + \mathbf{K}\mathbf{u}(t) = -\mathbf{M}\mathbf{i}\ddot{u}_{su}(t) \quad (48.27)$$

where \mathbf{i} is the contribution of the degrees of freedom in the working direction of the excitation. The forces acting on the structure are the internal set of forces

$$\mathbf{f}_k = \mathbf{K}\mathbf{u}(t) \quad (48.28)$$

and the pseudo-inertia forces

$$\mathbf{f}_m = \mathbf{M}(\ddot{\mathbf{u}}(t) + \mathbf{i} \ddot{u}_{su}(t)) = \mathbf{M}\mathbf{a}(t) \quad (48.29)$$

Equilibrium of the forces gives $\mathbf{f}_m = -\mathbf{f}_k$ which results in

$$\mathbf{a} = \omega^2 \mathbf{u} \quad (48.30)$$

The absolute maximum value of the pseudo-acceleration \mathbf{a} is called the spectral acceleration S_A

$$S_A(\omega) = \max |\mathbf{a}(t)| = \omega^2 S_D(\omega) \quad (48.31)$$

where the spectral displacement S_D is

$$S_D(\omega) = \max |\mathbf{u}(t)| \quad (48.32)$$

Response Spectrum analysis is based upon the mode superposition principle. Substitution of the mode superposition equation Eq. (48.19) in Eq. (48.27), premultiplication by ϕ_i^T , and modal vectors scaled such that

$$\phi_i^T \mathbf{M} \phi_i = 1 \quad \text{and} \quad \phi_i^T \mathbf{K} \phi_i = \omega_i^2 \quad (48.33)$$

gives

$$\ddot{\alpha}_i + \omega_i^2 \alpha_i = \gamma_i \ddot{u}_{su} \quad (48.34)$$

where α is the generalized modal displacement and γ_i is called the modal participation factor, which is equal to

$$\gamma_i = \phi_i^T \mathbf{M} \mathbf{i} \quad (48.35)$$

where \mathbf{i} is the contribution of the degrees of freedom in the working direction of the excitation. Thus, the maximum displacement vector in the i th mode can be written as

$$\mathbf{u}_{i,\max} = \gamma_i \phi_i S_{D_i} \quad (48.36)$$

Therefore, the maximum force vector for the i th mode can be calculated by

$$\mathbf{f}_{i,\max} = \mathbf{K} \mathbf{u}_{i,\max} = \mathbf{K} \gamma_i \phi_i S_{D_i} = \mathbf{M} \gamma_i \phi_i S_{A_i} \quad (48.37)$$

It is obvious that the upper bound of the combined forces is given by the absolute sum of the modal forces, the ABS rule:

$$\mathbf{f}_{\text{abs}} = \sum_{i=1}^N |\mathbf{f}_{i,\max}| \quad (48.38)$$

In general, it is unlikely that the maximum values of the modal forces would occur at the same time. Goodman, Rosenblueth & Newmark [35] showed that the probable maximum force or moment, in case the eigenfrequencies are not closely spaced, is approximately equal to the square root of the sum of the squares of modal values, the SRSS rule:

$$\mathbf{f}_s = \sqrt{\sum_{i=1}^N \mathbf{f}_{i,\max}^2} \quad (48.39)$$

A method to take the correlation among the modes into account is the standard Complete Quadratic Combination, the CQC rule:

$$\mathbf{f}_s = \sqrt{\left| \sum_{i=1}^N \sum_{j=1}^N \rho_{ij} \mathbf{f}_{i,\max} \mathbf{f}_{j,\max} \right|} \quad (48.40)$$

This formulations allows reduction of the superposed results when the responses of the modes have opposite signs. This may be the case with rotating machinery, where phase information is known.

DIANA also offers a Complete Quadratic Combination with absolute superposition terms:

$$\mathbf{f}_s = \sqrt{\sum_{i=1}^N \sum_{j=1}^N \rho_{ij} |\mathbf{f}_{i,\max} \mathbf{f}_{j,\max}|} \quad (48.41)$$

This formulation assures that the worst possible combination is assumed, i.e. both modes have their amplitude at the same time. This is being taken care of by taking the absolute value of the product in Eq. (48.41). This method can be used when there is no phase angle information between modes.

In both CQC methods the correlation among the modes is addressed explicitly by introducing correlation coefficients ρ_{ij} , which vary between zero and unity. The equation for the correlation coefficient due to Der Kiureghian is:

$$\rho_{ij} = \frac{8\sqrt{\xi_i \xi_j} (r_{ij} \xi_i + \xi_j) r_{ij}^{\frac{3}{2}}}{(1 - r_{ij}^2)^2 + 4\xi_i \xi_j r_{ij} (1 + r_{ij}^2) + 4(\xi_i^2 + \xi_j^2) r_{ij}^2} \quad (48.42)$$

where

$$r_{ij} = \frac{\omega_i}{\omega_j} \quad (48.43)$$

and ξ_i , ξ_j are the damping ratios for mode i and mode j , respectively. This equation implies that $\rho_{ij} = \rho_{ji}$ and $\rho_{ij} = 1$ for $i = j$ or for two modes with equal eigenfrequencies and equal damping ratios.

48.4 Transient Response

This section is an introduction to the background theory of dynamic analysis with time integration as performed by Module NONLIN [Ch. 7 p. 121]. Elaborate discussions on dynamics can be found in Bathe [5, Ch. 9], Craig [21], Hughes [47, §9], Zienkiewicz [94, §9], and G eradin & Rixen [34, Ch. 7]. Using Module NONLIN, such analysis may be combined with physical or geometrical nonlinearities, see part IV. The analysis results are presented in the time domain.

The system of governing equations for a transient dynamic problem at time t are generally written as

$$\mathbf{M}\ddot{\mathbf{u}}(t) + \mathbf{C}\dot{\mathbf{u}}(t) + \mathbf{f}_{\text{int}}(\mathbf{u}, \dot{\mathbf{u}}, \boldsymbol{\varepsilon}, \boldsymbol{\sigma}, t, \dots) = \mathbf{f}_{\text{ext}}(t) \quad (48.44)$$

where \mathbf{M} is the mass matrix, \mathbf{C} the damping matrix and \mathbf{f}_{ext} the external force vector or right-hand-side vector of forcing functions. Further, $\ddot{\mathbf{u}}$, $\dot{\mathbf{u}}$ and \mathbf{u} are the resulting acceleration, velocity and displacement vectors, $\boldsymbol{\varepsilon}$ and $\boldsymbol{\sigma}$ are the strain and stress fields. Vector \mathbf{f}_{int} is the internal set of forces opposing the displacements. For linear situations $\mathbf{f}_{\text{int}} = \mathbf{K}\mathbf{u}(t)$ and Eq. (48.44) can be written as

$$\mathbf{M}\ddot{\mathbf{u}}(t) + \mathbf{C}\dot{\mathbf{u}}(t) + \mathbf{K}\mathbf{u}(t) = \mathbf{f}_{\text{ext}}(t) \quad (48.44a)$$

For geometrical or physical nonlinear analysis or both, \mathbf{f}_{int} must be calculated for the actual stress distribution satisfying all nonlinear conditions.

$$\mathbf{f}_{\text{int}} = \int \mathbf{B}^T \boldsymbol{\sigma} \quad (48.45)$$

For the transient response of a nonlinear analysis, the solution of the second order differential equation Eq. (48.44) is obtained by *direct time integration* techniques. The solution for the dynamics problem will be determined at a number of discrete time points: $t_0, t_1, t_2, \dots, t - \Delta t, t, t + \Delta t, \dots, T$. We assume that we have the solution at time t so that Eq. (48.44) holds

$$\mathbf{M}^t \ddot{\mathbf{u}} + \mathbf{C}^t \dot{\mathbf{u}} + {}^t \mathbf{f}_{\text{int}}(\mathbf{u}, \dot{\mathbf{u}}, \boldsymbol{\varepsilon}, \boldsymbol{\sigma}, t, \dots) = {}^t \mathbf{f}_{\text{ext}} \quad (48.46)$$

If the time integration procedure being used requires

$$\mathbf{M}^{t+\Delta t} \ddot{\mathbf{u}} + \mathbf{C}^{t+\Delta t} \dot{\mathbf{u}} + {}^{t+\Delta t} \mathbf{f}_{\text{int}}(\mathbf{u}, \dot{\mathbf{u}}, \boldsymbol{\varepsilon}, \boldsymbol{\sigma}, t, \dots) = {}^{t+\Delta t} \mathbf{f}_{\text{ext}} \quad (48.47)$$

to obtain ${}^{t+\Delta t}\ddot{\mathbf{u}}$, then the time integration scheme will be called *implicit*. If, on the other hand, Eq. (48.47) is not required to obtain ${}^{t+\Delta t}\ddot{\mathbf{u}}$, the scheme will be called *explicit*. These methods use expressions which approximate the acceleration $\ddot{\mathbf{u}}$ and the velocity $\dot{\mathbf{u}}$ in terms of incremental displacement components $\Delta\mathbf{u} = {}^{t+\Delta t}\mathbf{u} - {}^t\mathbf{u}$.

48.4.1 Newmark

A widely used integration scheme is the Newmark method, which consists of the following equations.

$${}^{t+\Delta t}\dot{\mathbf{u}} = {}^t\dot{\mathbf{u}} + ((1 - \gamma) {}^t\ddot{\mathbf{u}} + \gamma {}^{t+\Delta t}\ddot{\mathbf{u}}) \Delta t \quad (48.48)$$

$${}^{t+\Delta t}\mathbf{u} = {}^t\mathbf{u} + {}^t\dot{\mathbf{u}} \Delta t + \left(\left(\frac{1}{2} - \beta\right) {}^t\ddot{\mathbf{u}} + \beta {}^{t+\Delta t}\ddot{\mathbf{u}}\right) \Delta t^2 \quad (48.49)$$

To find ${}^{t+\Delta t}\ddot{\mathbf{u}}$ and ${}^{t+\Delta t}\dot{\mathbf{u}}$ the equations are solved by using first Eq. (48.49)

$${}^{t+\Delta t}\ddot{\mathbf{u}} = \frac{1}{\beta \Delta t^2} \Delta\mathbf{u} - \frac{1}{\beta \Delta t} {}^t\dot{\mathbf{u}} - \frac{1 - 2\beta}{2\beta} {}^t\ddot{\mathbf{u}} \quad (48.50)$$

which is then substituted in Eq. (48.48) to obtain ${}^{t+\Delta t}\dot{\mathbf{u}}$.

In case of displacement controlled loading, a sudden rest ($\Delta\mathbf{u} = 0$) will be unstable.

For $\gamma \geq \frac{1}{2}$ and $\beta \geq \frac{1}{4}(\frac{1}{2} + \gamma)^2$ the Newmark method is unconditionally stable. For $\gamma = \frac{1}{2}$ no numerical damping is introduced, and the method is second order accurate. For $\gamma > \frac{1}{2}$ excessive low frequency mode damping is introduced, and the method is first order accurate.

48.4.2 Euler Backward

The Euler Backward method consists of the following equations.

$${}^{t+\Delta t}\dot{\mathbf{u}} = \frac{\Delta\mathbf{u}}{\Delta t} \quad (48.51)$$

$${}^{t+\Delta t}\ddot{\mathbf{u}} = \frac{{}^{t+\Delta t}\dot{\mathbf{u}} - {}^t\dot{\mathbf{u}}}{\Delta t} \quad (48.52)$$

48.4.3 Hilber–Hughes–Taylor

The Hilber–Hughes–Taylor method [43] (also called α -method) is an extension to the Newmark method. With the Hilber–Hughes–Taylor method it is possible to introduce numerical dissipation without degrading the order of accuracy. The Hilber–Hughes–Taylor method uses the same finite difference formulas Eq. (48.48) and Eq. (48.49) as the Newmark method with fixed γ and β ($\gamma = \frac{1}{2}(1 - 2\alpha)$, $\beta = \frac{1}{4}(1 - \alpha)^2$). The time-discrete equation of motion is modified as follows:

$$\begin{aligned} \mathbf{M} {}^{t+\Delta t}\ddot{\mathbf{u}} + (1 + \alpha) \mathbf{C} {}^{t+\Delta t}\dot{\mathbf{u}} - \alpha \mathbf{C} {}^t\dot{\mathbf{u}} + \\ (1 + \alpha) {}^{t+\Delta t}\mathbf{f}_{\text{int}} - \alpha {}^t\mathbf{f}_{\text{int}} = {}^{t+(1+\alpha)\Delta t}\mathbf{f}_{\text{ext}} \end{aligned} \quad (48.53)$$

For $\alpha = 0$ the method reduces to the Newmark method. For $-\frac{1}{3} \leq \alpha \leq 0$, $\gamma = \frac{1}{2}(1 - 2\alpha)$, and $\beta = \frac{1}{4}(1 - \alpha)^2$ the scheme is second order accurate and unconditionally stable. Decreasing α means increasing the numerical damping. This damping is low for low-frequency modes and high for the high-frequency modes.

48.4.4 Wilson

The Wilson- θ method is basically an extension of the Newmark scheme with $\gamma = \frac{1}{2}$ and $\beta = \frac{1}{6}$ (for which the Newmark method is conditionally stable), see Bathe [5]. In the Wilson- θ method the acceleration is assumed to vary linearly in time from t' to $t' + \theta\Delta t'$

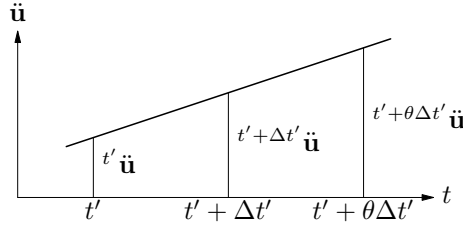


Figure 48.1: Linear acceleration

with $\theta \geq 1$ [Fig. 48.1]. DIANA determines the state variables at the user-specified $t + \Delta t$ which coincides with $t' + \theta \Delta t'$ for the purpose of postprocessing and for combination with other time integration methods. With this assumption we can derive the following equations for ${}^{t' + \theta \Delta t'}\dot{\mathbf{u}}$ and ${}^{t' + \theta \Delta t'}\ddot{\mathbf{u}}$.

$${}^{t' + \theta \Delta t'}\ddot{\mathbf{u}} = \frac{6}{\theta^2 \Delta t'^2} \left({}^{t' + \theta \Delta t'}\mathbf{u} - {}^{t'}\mathbf{u} \right) - \frac{6}{\theta \Delta t'} {}^{t'}\dot{\mathbf{u}} - 2 {}^{t'}\ddot{\mathbf{u}} \quad (48.54)$$

$${}^{t' + \theta \Delta t'}\dot{\mathbf{u}} = \frac{3}{\theta \Delta t'} \left({}^{t' + \theta \Delta t'}\mathbf{u} - {}^{t'}\mathbf{u} \right) - 2 {}^{t'}\dot{\mathbf{u}} - \frac{\theta \Delta t'}{2} {}^{t'}\ddot{\mathbf{u}} \quad (48.55)$$

The dynamic equilibrium equation is considered at $t' + \theta \Delta t'$.

$${}^{t' + \theta \Delta t'}\mathbf{M} {}^{t' + \theta \Delta t'}\ddot{\mathbf{u}} + {}^{t' + \theta \Delta t'}\mathbf{C} {}^{t' + \theta \Delta t'}\dot{\mathbf{u}} + {}^{t' + \theta \Delta t'}\mathbf{f}_{\text{int}} = {}^{t' + \theta \Delta t'}\mathbf{f}_{\text{ext}} \quad (48.56)$$

In this equation the external load vector on the right-hand side is just like the acceleration assumed to vary linearly in the time interval $t' \rightarrow t' + \theta \Delta t'$:

$${}^{t' + \theta \Delta t'}\mathbf{f}_{\text{ext}} = {}^{t'}\mathbf{f}_{\text{ext}} + \theta \left({}^{t' + \Delta t'}\mathbf{f}_{\text{ext}} - {}^{t'}\mathbf{f}_{\text{ext}} \right) \quad (48.57)$$

The Wilson- θ scheme is unconditional stable for $\theta \geq 1.37$. Like the Hilber–Hughes–Taylor method, the Wilson- θ scheme has a low damping for low-frequency modes and a high damping for the high-frequency (spurious) modes. However, for all frequencies the damping is usually higher than for the Hilber–Hughes–Taylor method. Furthermore, the Wilson- θ scheme show a comparatively large initial overshoot as a response to a ramp load.

48.4.5 Runge–Kutta

Runge–Kutta methods are a large class of methods for solving initial value problems of the form

$$\frac{d\mathbf{y}}{dt} = \mathbf{f}(\mathbf{y}, t) \quad (48.58)$$

By using the velocity $\mathbf{v} = \dot{\mathbf{u}}$ as an extra variable, the second order differential equation Eq. (48.44) is split into a system of two first order differential equations

$$\begin{aligned} \dot{\mathbf{u}} &= \mathbf{v} \\ \dot{\mathbf{v}} &= \mathbf{M}^{-1} \left(\mathbf{f}_{\text{ext}}(t) - \mathbf{C}\mathbf{v} - \mathbf{f}_{\text{int}}(\mathbf{u}, \mathbf{v}, \boldsymbol{\varepsilon}, \boldsymbol{\sigma}, t, \dots) \right) \end{aligned} \quad (48.59)$$

This system is of the same form as Eq. (48.58), so the Runge–Kutta method can be applied. A single step of an s -stage Runge–Kutta method is described by

$$\begin{aligned} {}^{t_i}\mathbf{y} &= {}^t\mathbf{y} + \Delta t \sum_{j=1}^s a_{ij} \mathbf{f}({}^{t_j}\mathbf{y}, t_j) \\ {}^{t + \Delta t}\mathbf{y} &= {}^t\mathbf{y} + \Delta t \sum_{j=1}^s b_j \mathbf{f}({}^{t_j}\mathbf{y}, t_j) \end{aligned} \quad (48.60)$$

with

$$t_i = t + c_i \Delta t \quad (48.61)$$

See for instance Hairer and Warner [38, §IV.3]. The coefficients a_{ij} , b_j , and c_i , that characterize the method, are conveniently represented in a Butcher tableau:

$$\begin{array}{c|c} c_i & a_{ij} \\ \hline & b_j \end{array}$$

If $a_{ij} = 0$ for $j \geq i$ the method is explicit, for example the classic 4th order method, which means that only function evaluations of \mathbf{f} are needed, but that no systems of equations have to be solved. Unfortunately, explicit methods cannot be used for Eq. (48.44) because of stability problems. In DIANA a two-stage Single Diagonal Implicit Runge–Kutta method, also known as SDIRK2, is implemented. This method is defined by the Butcher tableau

$$\begin{array}{c|cc} \alpha & \alpha & 0 \\ 1 & 1 - \alpha & \alpha \\ \hline & 1 - \alpha & \alpha \end{array}$$

with $\alpha = 1 - \sqrt{\frac{1}{2}}$. This method is unconditionally stable, second order accurate, and it efficiently damps high-frequency error modes. Moreover, the intermediate stag solution offers the possibility to estimate the time step error \mathbf{e} :

$$\mathbf{e} = \frac{\alpha - \hat{\alpha}}{\alpha} \left(({}^{t+\Delta t}\mathbf{y} - {}^t\mathbf{y}) - \frac{1}{\alpha} ({}^{t_1}\mathbf{y} - {}^t\mathbf{y}) \right) \quad (48.62)$$

with $\hat{\alpha} = 1 - \frac{5}{4}\sqrt{2}$. The error estimate \mathbf{e} can be used to control the time step size [§ 46.1.5.4 p. 565], see Hairer and Warner [38, §IV.8], or Ellsiepen and Hartmann [27].

48.5 Fluid–Structure Interaction Analysis

This section presents a brief overview of the background theory of the analysis of coupled fluid and structural systems, the so-called fluid–structure interaction analysis. For a more detailed description of the underlying theory see for instance Zienkiewicz & Bettles [92, 7] and Olson & Bathe [69]. Effects of large scale flow in the fluid are excluded. Attention is paid to the discretization method, the numerical solution techniques and simplifications.

Figure 48.2 shows a general fluid–structure interaction geometry. The solid extends

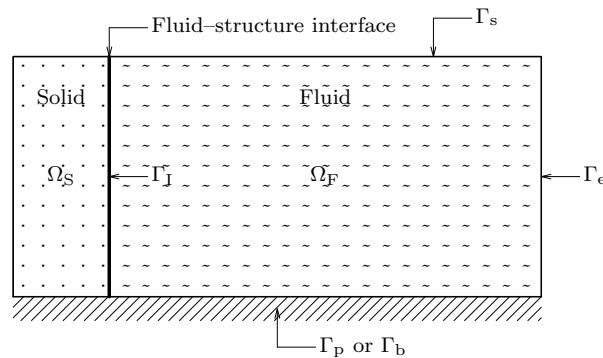


Figure 48.2: Fluid–structure interaction

throughout the region Ω_S , and Ω_F contains the fluid. Surface Γ_I defines the fluid–structure interface. The boundary of the fluid Γ_B may be separated in a fixed or prescribed part Γ_p , a part representing the bottom Γ_b , a part representing the free surface Γ_s and a part representing the infinite extent Γ_e .

48.5.1 Solid

In the solid, the discretization in the familiar form is given below and will be assumed throughout:

$$\mathbf{M}_S \ddot{\mathbf{u}} + \mathbf{C}_S \dot{\mathbf{u}} + \mathbf{K}_S \mathbf{u} + \mathbf{f}_I = \mathbf{f}_S^{\text{ex}}(t) \quad (48.63)$$

where \mathbf{M}_S , \mathbf{C}_S and \mathbf{K}_S are mass, damping and stiffness matrices respectively and \mathbf{u} is a set of unknowns describing the displacements of the structure. The vector \mathbf{f}_I stands for forces due to the interface interaction with the fluid and \mathbf{f}_S^{ex} represents the external force contributions.

48.5.2 Fluid

The fluid is characterized by a single pressure (or potential) variable p and the coupling with the structure is achieved by consideration of interface forces and a standard finite element idealization. Assuming the state of the fluid is linear, the governing equation is the wave or acoustic equation

$$\nabla^2 p = \frac{1}{c^2} \ddot{p} \quad (48.64)$$

where p is the pressure (compression positive) and c the wave speed, given by

$$c^2 = \frac{\beta}{\rho} \quad (48.65)$$

where β is the bulk modulus and ρ the density. Appropriate boundary conditions of the following form can be imposed.

48.5.2.1 Solid Boundary

The conditions applying to the surface Γ_I being the interface between the fluid and structure, can be written as

$$\frac{\partial p}{\partial n} = -\rho_F \mathbf{n}_F^T \ddot{\mathbf{u}}_F \quad \text{and} \quad \boldsymbol{\sigma} \mathbf{n}_S = p_F \mathbf{n}_F \quad \text{on} \quad \Gamma_I \quad (48.66)$$

where \mathbf{n}_F and \mathbf{n}_S are respectively the outward normal to the fluid domain and the outward normal to the structural domain. The coupling between the fluid domain and the structural domain is realized by continuity between the normal displacements with the condition $\ddot{\mathbf{u}}_F = \ddot{\mathbf{u}}_S$ and is obtained by combining this condition with Eq. (48.66)

$$\frac{\partial p}{\partial n} = -\rho_F \mathbf{n}_F^T \ddot{\mathbf{u}}_S \quad (48.67)$$

48.5.2.2 Prescribed Conditions

$$p = \bar{p} \quad \text{on} \quad \Gamma_p \quad (48.68)$$

where \bar{p} is a prescribed pressure often to be zero along part of the boundary Γ_p .

48.5.2.3 Free Surface

$$p = \rho_F g u_z \quad \text{on} \quad \Gamma_s \quad (48.69)$$

where g is the gravity acceleration and z is directed normal to the free surface. Noting that

$$\ddot{u}_z = -\frac{1}{\rho_F} \frac{\partial p}{\partial z} \quad (48.70)$$

Eq. (48.69) can be written as

$$\frac{\partial p}{\partial z} = -\frac{1}{g} \ddot{p} \quad \text{on} \quad \Gamma_s \quad (48.71)$$

which is the linearized free surface condition for first order waves.

48.5.2.4 Radiation for Boundary of Infinite Extent

If a boundary of infinite extent has been placed sufficiently far away, it may be assumed that only plane waves exist. In the existence of only outgoing waves, incoming waves are supposed to be absent, giving a solution of the form

$$p = f'(x - ct) \quad (48.72)$$

where a positive x is the outward direction. The radiation boundary condition is now obtained by eliminating f' and is given by

$$\frac{\partial p}{\partial x} = -\frac{1}{c} \dot{p} \quad \text{on } \Gamma_e \quad (48.73)$$

where c is the wave speed given by Eq. (48.65). This condition is denoted as the Sommerfeld radiation condition and in general, will be applied in a plane normal to the direction of the wave speed.

48.5.2.5 Bottom

The conditions applying to the surface Γ_b being the bottom of the fluid reservoir, can be written as

$$\frac{\partial p}{\partial n} = -\frac{1 - \alpha_B}{c(1 + \alpha_B)} \dot{p} \quad \text{on } \Gamma_b \quad (48.74)$$

where c is the wave speed given by Eq. (48.65) and α_B is the wave reflection coefficient of the bottom. The wave reflection coefficient α_B is the ratio of the amplitude of the reflected hydrodynamic pressure wave to the amplitude of a propagating pressure wave incident on the reservoir bottom. The wave reflection coefficient α_B may range within the limiting values of -1 and 1 . For rigid reservoir bottom materials $\alpha_B = 1$ and for very soft reservoir bottom materials $\alpha_B = -1$. For a more detailed description see for instance Fennes & Chopra [29] and Küçükarslan et al. [53].

48.5.3 Discretized Coupled Equations

A standard finite element discretization has used approximating p in terms of nodal values \mathbf{p}

$$p \approx \mathbf{N}_F \mathbf{p}^e \quad (48.75)$$

and the discretization gives a system of equations in a form

$$\mathbf{M}_F \ddot{\mathbf{p}} + \mathbf{C}_F \dot{\mathbf{p}} + \mathbf{K}_F \mathbf{p} + \mathbf{r}_I = 0 \quad (48.76)$$

where \mathbf{M}_F , \mathbf{C}_F , \mathbf{K}_F and \mathbf{r}_I are defined in terms of the following element matrices:

$$[M_{ij}]_F^e = \frac{1}{g} \int_{\Gamma_s^e} N_i N_j d\Gamma + \frac{1}{c^2} \int_{\Omega_F^e} N_i N_j d\Omega \quad (48.77)$$

with Γ_s the free surface and Ω_F the fluid reservoir

$$[C_{ij}]_F^e = \frac{1}{c_s} \int_{\Gamma_e^e} N_i N_j d\Gamma + \frac{1 - \alpha_B}{c(1 + \alpha_B)} \int_{\Gamma_b^e} N_i N_j d\Gamma \quad (48.78)$$

with Γ_e the radiation boundary and Γ_b the reservoir bottom

$$[K_{ij}]_F^e = \int_{\Omega_F^e} \nabla N_i \nabla N_j d\Omega \quad \text{with } \Omega_F \text{ the fluid domain} \quad (48.79)$$

$$[r_i]_I^e = \int_{\Gamma_I^e} N_i \rho_F n_k \ddot{u}_k d\Gamma \quad \text{with } \Gamma_I \text{ the fluid–structure interface} \quad (48.80)$$

The contribution \mathbf{f}_I from Eq. (48.63) can be written as

$$[f_{i_k}]_I^e = - \int_{\Gamma_I^e} N_i^u n_k p d\Gamma \quad (48.81)$$

or

$$\mathbf{f}_I^e = -\mathbf{R}^{eT} \mathbf{p}^e \quad (48.82)$$

and likewise the contribution \mathbf{r}_I from Eq. (48.76) as

$$[r_i]_I^e = \int_{\Gamma_I^e} N_i^p \rho_F n_k \ddot{u}_k \, d\Gamma \quad (48.83)$$

or

$$\mathbf{r}_I^e = \rho_F \mathbf{R}^e \ddot{\mathbf{u}}^e \quad (48.84)$$

After assembling contributions from each type of element (i.e., solid, fluid–structure interface, fluid, boundary fluid elements), the following coupled system of equations is obtained

$$\begin{bmatrix} \mathbf{M}_S & \mathbf{O}^T \\ \rho_F \mathbf{R} & \mathbf{M}_F \end{bmatrix} \begin{Bmatrix} \ddot{\mathbf{u}} \\ \ddot{\mathbf{p}} \end{Bmatrix} + \begin{bmatrix} \mathbf{C}_S & \mathbf{O}^T \\ \mathbf{O} & \mathbf{C}_F \end{bmatrix} \begin{Bmatrix} \dot{\mathbf{u}} \\ \dot{\mathbf{p}} \end{Bmatrix} + \begin{bmatrix} \mathbf{K}_S & -\mathbf{R}^T \\ \mathbf{O} & \mathbf{K}_F \end{bmatrix} \begin{Bmatrix} \mathbf{u} \\ \mathbf{p} \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}_S(t) \\ \mathbf{0} \end{Bmatrix} \quad (48.85)$$

48.5.4 Solution of Coupled System

The technique to be used for solving the system of equations Eq. (48.85) strongly depends on the form of the forcing function $\mathbf{f}_S(t)$.

48.5.4.1 Frequency Domain Analysis

If the forcing function of Eq. (48.85) has been expressed in, or can be transformed to a periodic form as

$$\mathbf{f}_S(t) = \hat{\mathbf{f}}_S e^{i\omega t} \quad (48.86)$$

then for linear problems the steady-state solution will exist in the same form, thus

$$\mathbf{u}(t) = \hat{\mathbf{u}} e^{i\omega t} \quad \text{and} \quad \mathbf{p}(t) = \hat{\mathbf{p}} e^{i\omega t} \quad (48.87)$$

Now a complex expression of the solution is obtained and can be written in the matrix form

$$\begin{bmatrix} -\omega^2 \mathbf{M}_S + \mathbf{K}_S + i\omega \mathbf{C}_S & -\mathbf{R}^T \\ -\omega^2 \rho_F \mathbf{R} & -\omega^2 \mathbf{M}_F + \mathbf{K}_F + i\omega \mathbf{C}_F \end{bmatrix} \begin{Bmatrix} \hat{\mathbf{u}} \\ \hat{\mathbf{p}} \end{Bmatrix} = \begin{Bmatrix} \hat{\mathbf{f}}_S \\ \mathbf{0} \end{Bmatrix} \quad (48.88)$$

from which the complex values of the amplitudes $\hat{\mathbf{u}}$ and $\hat{\mathbf{p}}$ can be found. A single set of complex equations for $\hat{\mathbf{u}}$ is obtained by eliminating the pressure values $\hat{\mathbf{p}}$ directly. The second subsystem of equations implied in Eq. (48.88) can be written

$$\hat{\mathbf{p}} = [-\omega^2 \mathbf{M}_F + \mathbf{K}_F + i\omega \mathbf{C}_F]^{-1} \omega^2 \rho_F \mathbf{R} \hat{\mathbf{u}} \quad (48.89)$$

or with $\mathbf{H}_F(\omega) = [-\omega^2 \mathbf{M}_F + \mathbf{K}_F + i\omega \mathbf{C}_F]^{-1}$ as frequency response function

$$\hat{\mathbf{p}} = \omega^2 \rho_F \mathbf{H}_F(\omega) \mathbf{R} \hat{\mathbf{u}} \quad (48.90)$$

On substitution for $\hat{\mathbf{p}}$ the result from above into the first subsystem of equations implied in Eq. (48.88) an additional fluid matrix $\hat{\mathbf{K}}_I$ is obtained and is given as

$$\hat{\mathbf{K}}_I = -\omega^2 \rho_F \mathbf{R}^T \mathbf{H}_F(\omega) \mathbf{R} \quad (48.91)$$

In the above the matrix $\hat{\mathbf{K}}_I$ is a complex quantity and can be written in a form

$$\mathbf{K}_I = -\omega^2 \rho_F \Re(\mathbf{R}^T \mathbf{H}_F(\omega) \mathbf{R}) - \omega^2 \rho_F \Im(\mathbf{R}^T \mathbf{H}_F(\omega) \mathbf{R}) \quad (48.92)$$

or

$$\mathbf{K}_I = -\omega^2 \tilde{\mathbf{M}}_F + i\omega \tilde{\mathbf{C}}_F \quad (48.93)$$

where $\tilde{\mathbf{M}}_F$ and $\tilde{\mathbf{C}}_F$ are denoted as the *added mass* matrix and the *added damping* matrix respectively:

$$\tilde{\mathbf{M}}_F = \Re(\mathbf{R}^T \mathbf{H}_F(\omega) \mathbf{R}) \quad (48.94)$$

$$\tilde{\mathbf{C}}_F = -\omega \Im(\mathbf{R}^T \mathbf{H}_F(\omega) \mathbf{R}) \quad (48.95)$$

The structural matrix now takes the form

$$\left[-\omega^2 (\mathbf{M}_S + \tilde{\mathbf{M}}_F) + \mathbf{K}_S + i\omega (\mathbf{C}_S + \tilde{\mathbf{C}}_F) \right] \hat{\mathbf{u}} = \hat{\mathbf{f}}_S \quad (48.96)$$

which can be solved with a direct solution procedure [§ 48.2.2 p. 589].

48.5.5 Simplification for Fixed Fluid Boundaries

If no surface waves are admitted and the effect of radiation waves at the infinite boundary is ignored, i.e.,

$$p = 0 \quad \text{respectively on } \Gamma_s \text{ and } \Gamma_e \quad (48.97)$$

and bottom absorption effects are neglected, i.e. $\alpha_B = 1$ on Γ_b , and compression effects are neglected, i.e., $c = \infty$ on Ω_F , then the matrices \mathbf{M}_F and \mathbf{C}_F of Eq. (48.76) as well as the second of Eq. (48.85) become zero. The pressure vector \mathbf{p} can now be obtained directly in terms of $\ddot{\mathbf{u}}$ as

$$\mathbf{p} = -\mathbf{K}_F^{-1} \rho_F \mathbf{R} \ddot{\mathbf{u}} \quad (48.98)$$

On substitution into the first of Eq. (48.85) the structural matrix now becomes of the general form

$$(\mathbf{M}_S + \tilde{\mathbf{M}}_F) \ddot{\mathbf{u}} + \mathbf{C}_S \dot{\mathbf{u}} + \mathbf{K}_S \mathbf{u} = \mathbf{f}_S(t) \quad (48.99)$$

where the added mass is simply given as

$$\tilde{\mathbf{M}}_F = \rho_F \mathbf{R}^T \mathbf{K}_F^{-1} \mathbf{R} \quad (48.100)$$

48.5.5.1 Frequency Domain Analysis

If the structural damping is absent or not strongly, the solution in the frequency domain can be obtained by the mode superposition technique [§ 48.2.1 p. 589]. Now the solution \mathbf{u} will be obtained by superposition of the response in each mode:

$$\mathbf{u}(\Omega) = \sum_{i=1}^{\infty} \phi_i \alpha_i(\Omega) \quad (48.101)$$

where ϕ_i is the i -th eigenvector and α_i is the i -th generalized modal displacement. Therefore, it requires first the solution of a sufficient number of eigenvalues and corresponding eigenvectors of the problem in Eq. (48.99) with damping neglected

$$(\mathbf{M}_S + \tilde{\mathbf{M}}_F) \ddot{\mathbf{u}} + \mathbf{K}_S \mathbf{u} = 0 \quad (48.102)$$

Next, the set of equations Eq. (48.99) are transformed to global coordinates and a decoupled set of equations is obtained provided that the damping matrix \mathbf{C}_S is proportional [§ 48.1.2.3 p. 586].

48.5.5.2 Time Domain Analysis

In case of an arbitrary transient loading, the response of the simplified problem in Eq. (48.99) can now be obtained by a direct time integration method [§ 48.4 p. 592]. After the *added mass* matrix $\tilde{\mathbf{M}}_F$, given by Eq. (48.100), has been determined via the solution of Eq. (48.98), the transient analysis can be carried out in the usual way by Module NONLIN [Ch. 13 p. 211].

Chapter 49

Hybrid Frequency Time Domain Analysis

The hybrid frequency time domain (HFTD) method as implemented in DIANA is mainly based on the procedure described by Darbre [24], and by Fenves and Chaves [30] and is described in detail in the work of Sirumbal [81].

49.1 Hybrid Frequency Time Domain Method

The HFTD method is generally applied in the solution of dynamic problems of systems with frequency dependent properties and nonlinear behaviour.

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{f}^{\text{int}}(t, \mathbf{u}, \dot{\mathbf{u}}) = \mathbf{f}^{\text{ext}}(t) \quad (49.1)$$

Frequency dependent properties, which are present in e.g. soil-structure and fluid-structure interaction systems, are properly taken into account by frequency domain solution methods. This type of analysis methods, however, is only applicable for linear systems. When some kind of nonlinearities, e.g. material, geometrical, contact, friction, sliding, are required to be included in the analysis, frequency domain solution methods are not applicable and a time domain solution should be used. Time domain analysis methods allow tracking the development of the nonlinear response and evaluating its magnitude in each time step. However, it is not possible to consider frequency dependent properties directly and consistently.

It is under these circumstances that the HFTD method is proposed as an alternative to study the nonlinear dynamic response of frequency dependent systems, by means of a hybrid formulation that combines the advantages of both, time and frequency domain analysis methods, and diminish their corresponding drawbacks. In general terms, the HFTD method transforms the nonlinear equation of motion defined in the time domain into a pseudo-linear equation in the frequency domain. For this to be achieved, the internal forces are divided into linear and nonlinear components. The linear components of the internal force are the summations of the linear elastic stiffness and damping forces. The nonlinear components are an unknown vector denominated pseudo force. Besides, the linear components of the internal force are kept in the left-hand side of the equation of motion, whereas the pseudo force is moved to the right-hand side. In this way, the left-hand side is composed exclusively by linear terms, whereas the right-hand side contains the applied load and the pseudo force which unknown value is determined iteratively.

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{C}\dot{\mathbf{u}} + \mathbf{K}\mathbf{u} = \mathbf{f}^{\text{ext}}(t) + \mathbf{q}(t) \quad (49.2)$$

As mentioned before, the HFTD method solves iteratively the pseudo-linear (or linearized) equation of motion in the frequency domain. However, the dynamic load applied to the system is defined by a signal in the time domain. In the same way, the pseudo force representing the nonlinear behaviour can only be evaluated in the time domain. Therefore, the HFTD method requires the transformation of both time dependent components

of the right-hand side of the equation in to a series of harmonic forces by means of a Discrete Fourier Transform (DFT).

$$[-\omega^2 \mathbf{M} + i\omega \mathbf{C} + \mathbf{K}] \hat{\mathbf{u}}(\omega) = \hat{\mathbf{f}}^{\text{ext}}(\omega) + \hat{\mathbf{q}}(\omega) \quad (49.3)$$

As a consequence the linearized equation of motion in the frequency domain can be solved for a range of loading frequencies, using the direct or mode superposition methods. The next step is to transform the solution obtained back to the time domain by means of the Inverse Fourier Transform.

With the dynamic response (displacements, velocities, and accelerations) in the time domain it is possible to use the constitutive laws ruling the nonlinear behaviour of the system and evaluate the internal forces. Then, a new value for the pseudo force is computed as the difference between the linear elastic force and the nonlinear internal force.

Finally, the new value of the pseudo force is transformed again to the frequency domain to solve one more time the linearized equation of motion in the frequency domain. This process is repeated until convergence is achieved.

49.2 Finite Element Equations

This section describes the HFTD analysis as implemented in DIANA for a fluid-structure interaction analysis. The linearized equation of motion in terms of relative displacements \mathbf{u}_r is defined as:

$$\left[-\omega^2 (\mathbf{M}_S + \tilde{\mathbf{M}}_F) + i\omega (\mathbf{C}_S + \tilde{\mathbf{C}}_F) + \mathbf{K}_S \right] \hat{\mathbf{u}}_r = - \left[\mathbf{M}_S + \tilde{\mathbf{M}}_F - i\frac{1}{\omega} \tilde{\mathbf{C}}_F \right] \mathbf{I}^{(s)} \hat{\mathbf{u}}_g(\omega) + \hat{\mathbf{q}}(\omega) \quad (49.4)$$

Where $\tilde{\mathbf{M}}_F$ [Eq. (48.94)] and $\tilde{\mathbf{C}}_F$ [Eq. (48.95)] are denoted as the *added mass* matrix and the *added damping* matrix of the fluid.

The mode superposition method can be employed by introducing the generalized coordinates $\hat{\mathbf{y}}(\omega)$:

$$\hat{\mathbf{u}}_r(\omega) = \Phi \hat{\mathbf{y}}(\omega) \quad (49.5)$$

Consequently, the pre-multiplication of the equation of motion [Eq. (49.4)] by the transposed matrix of the eigenmodes Φ^T , results in the equation of motion in terms of the generalized coordinates $\hat{\mathbf{y}}(\omega)$:

$$[-\omega^2 (\mathbf{B}_S + \mathbf{B}_F) + i\omega (\mathbf{D}_S + \mathbf{D}_F) + \mathbf{E}_S] \hat{\mathbf{y}}(\omega) = -\Phi^T \mathbf{F}_g \mathbf{I}^{(S)} \hat{\mathbf{U}}_g(\omega) + \Phi^T \hat{\mathbf{q}}(\omega) \quad (49.6)$$

Where

$$\mathbf{B}_S = \Phi^T \mathbf{M}_S \Phi \quad (49.7)$$

$$\mathbf{B}_F = \Phi^T \tilde{\mathbf{M}}_F \Phi \quad (49.8)$$

$$\mathbf{D}_S = \Phi^T \mathbf{C}_S \Phi \quad (49.9)$$

$$\mathbf{D}_F = \Phi^T \tilde{\mathbf{C}}_F \Phi \quad (49.10)$$

$$\mathbf{E}_S = \Phi^T \mathbf{K}_S \Phi \quad (49.11)$$

$$\mathbf{F}_g = \mathbf{M}_S + \tilde{\mathbf{M}}_F - i\frac{1}{\omega} \tilde{\mathbf{C}}_F \quad (49.12)$$

The linearized equation of motion in the generalized coordinates [Eq. (49.6)] constitutes a condensed system of equations in which the unknowns are reduced to the number of eigenmodes included in the response calculation. In the general case only the matrices \mathbf{B}_S and \mathbf{E}_S are diagonal, which means that the system is not uncoupled.

49.3 Solution Procedure Sequence

The HFTD method as implemented in DIANA can be summarized in the following general steps:

1. Select the number of eigenmodes to be included in the response calculation.
2. Perform the eigenvalue analysis of the structure to obtain the eigenmodes Φ and calculate the generalized mass \mathbf{B}_S , damping \mathbf{D}_S , and stiffness \mathbf{E}_S matrices.
3. Determine the range of excitation frequencies to be used in the solution of the linearized equation of motion [Eq. (49.6)].
4. Determine for each excitation frequency the generalized added mass \mathbf{B}_F , added damping \mathbf{D}_F and load \mathbf{F}_g .
5. Initialize the pseudo-force in the time domain $\mathbf{q}(t)$. Only for the first iteration, the pseudo-force should be set equal to zero.
6. Perform the Fourier transformation of the base excitation signals from the time domain $\ddot{\mathbf{u}}_g(t)$ to the frequency domain $\hat{\ddot{\mathbf{u}}}_g(\omega)$.
7. Set up the current generalized pseudo-force in the time domain $\Phi^T \mathbf{q}(t)$.
8. Perform the Fourier transformation of the generalized pseudo-force to the frequency domain $\Phi^T \hat{\mathbf{q}}(\omega)$.
9. Solve the linearized equation [Eq. (49.6)] for each excitation frequency and determine the generalized displacements in the frequency domain $\hat{\mathbf{y}}(\omega)$.
10. Perform the inverse Fourier transformation to obtain the generalized displacements $\mathbf{y}(t)$, velocities $\dot{\mathbf{y}}(t)$, and accelerations $\ddot{\mathbf{y}}(t)$ in the time domain.
11. Expand the generalized response in the time domain to the nodal relative displacements \mathbf{u}_r , velocities $\dot{\mathbf{u}}_r$, and accelerations $\ddot{\mathbf{u}}_r$ [Eq. (49.5)].
12. Perform the state determination in the time domain: Calculation of the nonlinear internal forces of the system using the appropriate constitutive laws for the response obtained in step 11.
13. Determine the linear restoring forces $(\mathbf{K}_S \mathbf{u}_r + \mathbf{C}_S \dot{\mathbf{u}}_r)$ based on the response calculated in step 11.
14. Evaluate the new pseudo-force using the nonlinear internal forces of the system and the linear restoring forces.
15. Check the convergence of the displacement and pseudo-force. If convergence is not achieved, go to step 7 and perform a new iteration with the new pseudo-force.

49.4 Fourier Transform Period

The period of time used in the Discrete Fourier Transform (DFT) indicates the period of the periodic loading (both ground acceleration and pseudo-force) acting on the system. Therefore, if the loading duration t_0 is used to transform the ground acceleration and pseudo-force to the frequency domain, the results obtained from the HFTD method would be the steady-state response of the system subject to a periodic loading with a period equal to t_0 , which harmonic components are given by its DFT.

The purpose of the HFTD method is to determine the actual transient response of the subject to an arbitrary ground acceleration loading, and not the steady-state response of the system subject to a periodic loading. Therefore, to obtain an accurate approximation of the transient response, the period used in the DFT should be extended sufficiently long that the effects of the steady-state response and loading periodicity become negligible [87].

The Fourier period T_p should be defined as the summation of the ground acceleration t_0 plus an additional extended period called "quiet zone" T_q . The quiet zone is a band of zeros added at the end of the ground acceleration signal, which length depends on the free vibration properties of the system, i.e. the fundamental period of vibration and viscous damping ratio of the structure.

Steps 6, 8, and 10 of the HFTD solution procedure [§ 49.3] indicate that the Fourier transformation is applied to the ground acceleration (direct) and the pseudo-force (direct), and the response of the system (inverse), respectively. All these transformations should always be performed over the same Fourier period T_p in terms of fundamental vibration period T_1 , and the viscous damping ratio of the system (ξ) [30]:

$$T_p = t_0 + T_q \quad (49.13)$$

where

$$T_q \geq 0.75 \frac{T_1}{\xi} \quad (49.14)$$

Eq. (49.14) shows that for long oscillation periods, i.e. low frequency systems, and a low damping ratio, the quiet zone T_q is longer. In the extreme case of damping ratio equal to zero, T_q tends to infinity, which means that the HFTD method is not applicable for undamped systems [87]. In DIANA a damping ratio ξ of 0.5 is assumed to calculate the quiet zone T_q .

As mentioned before, the quiet zone is necessary to satisfy the initial conditions and consequently obtain a better approximation of the transient response instead of the steady state response. On the contrary, "the use of a Fourier transformation period that is too short provides inaccurate results, no matter that they be stable and apparently reasonable" [25]. In conclusion, the insertion of a quiet zone should be understood as a numerical presentation of what theoretically should be an infinite period in the analytic Fourier integral.

49.5 Time Segmentation

The HFTD procedure converges in a time progressive manner, which means that "convergence at any time is reached only after the solution has converged at all previous times" [25]. If the HFTD procedure [§ 49.3] is executed directly along the whole time span of interest, convergence and stability problems are likely to occur.

For this reason, the HFTD method is performed following a time segmentation approach in which the displacements and pseudo-forces are evaluated for a limited number of consecutive time steps contained in time segments. Once convergence is achieved in all the time steps within a segment, the iterative procedure is repeated for the next time segment, until the total time span of interest is covered.

The maximum number of time steps in a segment allowed depends on stability conditions. For a detailed discussion of HFTD method stability criteria see Darbre and Wolf [25]. When a system is highly stable, which is the case for linear frequency independent systems, the number of time steps per segment increases and the iterations required to obtain convergence diminish. Nevertheless, the minimum number of iterations required to obtain convergence is more sensitive to the degree of stability than the maximum number of time steps per segment. This means that two different systems can converge with the same number of time steps per segment, but the less stable system will require a higher number of iterations.

In conclusion, the time span of interest has to be divided into segments containing one or several time steps. It is important to mention that the number of time steps per segment does not affect the accuracy of the results obtained but leads to a convergent (small number of time steps) or divergent (large number of time steps) solution. When stability or convergence problems occur, then the number of time steps per segment should be reduced.

On the other hand, the number of iterations and the computational effort are proportional to the number of time segments used. If many segments consisting of a few steps each are defined for the iterative procedure, then the computational effort and the total

number of iterations required to reach convergence is high. However, this proportional dependence is not critic unless considerably smaller number of time steps per segment than the optimal are used.

Using the segmentation approach implies the repetition of the HFTD solution procedure (step 6 to step 15 of [§ 49.3]) for all the segments defined in the Fourier period T_p . The HFTD solution procedure indicates that, for the first iteration of the first time segment only, the pseudo-force vector is set equal to zero (step 5) and transformed to the frequency domain (step 8) using the Fourier transformation period T_p . Once the pseudo-linear system has been solved in the frequency domain (step 9), the generalized response is transformed back to the time domain (step 10).

For the subsequent iterations, the segmentation approach requires that the pseudo-forces and displacements in the time domain are calculated and updated (step 11 to step 14) only for the non-converged time steps belonging to the current time segment k :

$$t_c < t < T_{k+1} - \Delta t \quad (49.15)$$

where t_c is the last converged time step, T_{k+1} is the initial time of the next time segment, and Δt the time step size. The iterative process (step 7 to step 14) is repeated until the convergence criteria (step 15) [§ 49.7 p. 606] are fulfilled for all the time steps inside the current segment k , i.e. $t_c = T_{k+1} - \Delta t$.

When all the time steps inside the current segment k have converged, the next segment is investigated. However, in the first iteration of the new segment (step 6) the values of the pseudo-forces belonging to the previous converged time segments are maintained, i.e. the pseudo-forces of the converged previous time steps are not modified nor updated anymore, whereas for the current and subsequent segments the values of the pseudo-forces are set to zero. Alternatively, a decaying appended function is also possible [§ 49.6].

In this way, the iterative procedure is continued until the last segment of the Fourier period T_p is covered, doing always the Fourier transformation through all the Fourier period T_p , but updating progressively the displacements and pseudo-forces only for the non-converged time steps inside the current segment of analysis.

As a final point, special attention should be given to the fact that the equation of motion of the pseudo-linear system (not the real nonlinear system) is the one solved in the frequency domain. Therefore, the eigenvalues, eigenmodes, and critical damping ratios, required to perform the mode superposition method and to determine the length of the Fourier transformation period T_p [Eqs. (49.13) (49.14)], should be based on the properties of the pseudo-linear system defined in Eq. (49.6).

49.6 Decaying Functions

Due to the segmentation approach for solving the pseudo-linear equation in the frequency domain, both load vectors in the right-hand side of Eq. (49.6), the ground acceleration $\ddot{\mathbf{u}}_g(t)$ and the pseudo-forces $\mathbf{q}(t)$, are filled with zeros for all time steps belonging to the time segments coming after the current. Therefore, the expressions defining $\ddot{\mathbf{u}}_g(t)$ and $\mathbf{q}(t)$, for the current time segment k and iteration j , are given in terms of the Heaviside function H :

$$\ddot{\mathbf{u}}_g^k(t) = \ddot{\mathbf{u}}_g(t) [H(t) - H(t - T_{k+1})], \quad 0 < t < T_p \quad (49.16)$$

$$\mathbf{q}^{k,j}(t) = \mathbf{q}(t) [H(t) - H(t - T_{k+1})], \quad 0 < t < T_p \quad (49.17)$$

This means that both time load vectors die out suddenly between the time steps $T_{k+1} - \Delta t$ and T_{k+1} . In order to avoid this sudden unloading and consequent inaccurate results, a decaying function may be appended to both load components. The decaying function must start in the last point of the current time segment $T_{k+1} - \Delta t$ and its extension ΔT_d is recommended not to be longer than one segment length ΔT_s , due to implementation more than performance or accuracy reasons [23].

One type of decaying function that can be used for this purpose is the sinusoidal function $f(\tau)$ proposed by Darbre [23]:

$$f(\tau) = \left[f(0) + \tau \dot{f}(0) + \frac{\tau^2}{2} \ddot{f}(0) \right] \left[1 - \frac{\tau}{\Delta t_d} + \frac{\sin(2\pi\tau/\Delta t_d)}{2\pi} \right], \quad 0 < \tau < \Delta T_d \leq \Delta T_s \quad (49.18)$$

As expressed in Eq. (49.16), the ground acceleration signal is taken from 0 to $T_{k+1} - \Delta t$, neglecting the subsequent part. Instead the decaying function [Eq. (49.18)] is appended to the signal in the time point $T_{k+1} - \Delta t$, followed by a band of zeros until the Fourier period T_p is completed.

Therefore, the modified ground acceleration signal for the current time segment k can be expressed as:

$$\ddot{\mathbf{u}}_g^k(t) = \ddot{\mathbf{u}}_g(t) [H(t) - H(t - T_{k+1})] + f_{\ddot{\mathbf{u}}_g}(\tau_k), \quad 0 < t < T_p \quad (49.19)$$

with

$$\tau_k = t - T_{k+1} + \Delta t \quad (49.20)$$

On the other hand, for the first iteration the psuedo-forces are set equal to zero for all time steps between 0 and T_p . For the next iterations, the values of the pseudo-forces corresponding to the converged time steps (from 0 to t_c) are kept fixed, whereas for the other non-converged time steps inside the current segment k (from t_c to $T_{k+1} - \Delta t$), the pseudo-forces corresponding to the last iteration are assigned.

Similarly as in the case of the ground acceleration, the pseudo-forces of the following tiem segments correspond to the appended decaying function $f(\tau)$, continuing with a band of zeros until the Fourier period T_p is completed. The modified pseudo-force vector for the current time segment k can be expressed as:

$$\bar{\mathbf{q}}^{k,j+1}(t) = \bar{\mathbf{q}}^j(t) [H(t) - H(t - T_{k+1})] + f_q(\tau_k), \quad 0 < t < T_p \quad (49.21)$$

49.7 Convergence Criteria

The convergence criteria ceck is performed for both the pseudo-force and the displacement vector in natural or generalized coordinates. For each time step inside the current segment k , the convergence criteria verification after iteration j is given by:

$$\frac{|\mathbf{u}_r^j(t) - \mathbf{u}_r^{j-1}(t)|}{|\mathbf{u}_r^j(t)|} \leq tol, \quad t_c + \Delta t \leq t \leq T_{k+1} - \Delta t \quad (49.22)$$

$$\frac{|\mathbf{q}^j(t) - \mathbf{q}^{j-1}(t)|}{|\mathbf{q}^j(t)|} \leq tol, \quad t_c + \Delta t \leq t \leq T_{k+1} - \Delta t \quad (49.23)$$

Chapter 50

Nonlinear Vibration Analysis

Nonlinear or large amplitude vibration problems are discussed in this chapter. At large vibration amplitude (in the order of shell or plate thickness), due to geometrically nonlinear effects, the vibration frequency varies with vibration amplitude. As a result, a softening or a hardening effect is observed (the frequency decreases or increases with increasing amplitude, respectively) which is visualized by so called backbone curves. Many studies in the field of nonlinear vibration to describe this type of vibration amplitude–frequency relation are based on semi-analytical (Galerkin or Rayleigh–Ritz) formulation. However such formulations are restricted to structures with a relatively simple geometry. A finite element discretization makes it possible to analyse structures with arbitrary geometry. However, standard finite element based transient analysis procedures for nonlinear vibration problems are still highly time consuming. As an alternative approach, a finite element based perturbation approach for nonlinear vibrations has been implemented in DIANA that is analogous to initial postbuckling analysis described in the stability analysis background theory [§ 54.3 p. 621]. Based on Rehfield’s work [73], the method is formulated in a single mode context. Necessary extensions for multi mode case are also discussed. The perturbation approach yields dynamic b coefficient b_D (analogous to static b coefficient in case of static postbuckling problems) which corresponds to the initial curvature of the frequency–amplitude relation and accounts for the most important nonlinear effect. See § 54.4.1 for a description of the functional notation used in this chapter.

50.1 The Perturbation Method

Application of perturbation method for nonlinear vibration problem (in a similar way as initial postbuckling analysis) was done by Rehfield [73] in a single mode context using functional notation. Tiso [85] made a finite element implementation of that approach. In the following only the essential equations following Rehfield and Tiso will be mentioned. The dynamics of a system under periodic motion with radial frequency ω is governed by Hamilton’s principle, which can be written as

$$\int_0^{2\pi/\omega} \left[\left(\frac{1}{2} M \left(\frac{\partial \mathbf{u}}{\partial t} \right) \cdot \frac{\partial \mathbf{u}}{\partial t} \right) - \boldsymbol{\sigma} \cdot \delta \boldsymbol{\varepsilon} \right] dt = 0 \quad (50.1)$$

where t denotes time, and the ‘dot’ operation implies the inner multiplication of variables and the integration over the entire domain. The mass operator M and its properties are defined in Eq. (54.72) in § 54.5.2. Starting from Eq. (50.1) and introducing the new time variable $\tau = \omega t$, and considering the periodicity one obtains

$$\int_0^{2\pi} (\boldsymbol{\sigma} \cdot \delta \boldsymbol{\varepsilon} + M(\ddot{\mathbf{u}}) \cdot \delta \mathbf{u}) d\tau = 0 \quad (50.2)$$

The vibration mode and the corresponding strain and stress are assumed as

$$\begin{aligned} \mathbf{u} &= \xi \mathbf{u}_1 \\ \boldsymbol{\varepsilon} &= \xi \boldsymbol{\varepsilon}_1 \\ \boldsymbol{\sigma} &= \xi \boldsymbol{\sigma}_1 \end{aligned} \quad (50.3)$$

where ξ is an amplitude parameter associated with mode \mathbf{u}_1 . If the proposed form in Eq. (50.3) is substituted in Eq. (50.2) and only linear terms are retained, and finally by letting $\delta\mathbf{u} = \mathbf{u}_1$, the natural frequency squared ω_0^2 is obtained.

$$\int_0^{2\pi} (\omega_0^2 M(\ddot{\mathbf{u}}_1) \cdot \delta\mathbf{u} + \boldsymbol{\sigma}_1 \cdot \delta\boldsymbol{\epsilon}) d\tau = 0 \quad (50.4)$$

$$\omega_0^2 = \frac{\int_0^{2\pi} \boldsymbol{\sigma}_1 \cdot \boldsymbol{\epsilon}_1 d\tau}{\int_0^{2\pi} M(\dot{\mathbf{u}}_1) \cdot \dot{\mathbf{u}}_1 d\tau} \quad (50.5)$$

It is assumed for now that only one mode \mathbf{u}_1 is associated with the frequency ω_0 .

To find how the structure behaves when the amplitude of vibration becomes finite the solution is expanded as follows

$$\mathbf{u} = \xi \mathbf{u}_1 + \xi^2 \mathbf{u}_2 + \xi^3 \mathbf{u}_3 + \dots \quad (50.6)$$

$$\boldsymbol{\epsilon} = \xi \boldsymbol{\epsilon}_1 + \xi^2 \boldsymbol{\epsilon}_2 + \xi^3 \boldsymbol{\epsilon}_3 + \dots \quad (50.7)$$

$$\boldsymbol{\sigma} = \xi \boldsymbol{\sigma}_1 + \xi^2 \boldsymbol{\sigma}_2 + \xi^3 \boldsymbol{\sigma}_3 + \dots \quad (50.8)$$

In order to make the expansion unique, the second order mode \mathbf{u}_2 is orthogonalized to \mathbf{u}_1 with respect to the mass operator

$$M(\dot{\mathbf{u}}_1) \cdot \dot{\mathbf{u}}_k = M(\ddot{\mathbf{u}}_1) \cdot \mathbf{u}_k = 0, \quad k \neq 1 \quad (50.9)$$

By substituting the expansion Eq. (50.6) in the equilibrium Eq. (50.2) and letting $\delta\mathbf{u} = \mathbf{u}_1$ and accordingly $\delta\boldsymbol{\epsilon} = \boldsymbol{\epsilon}_1$, and introducing the expression for ω_0^2 , Eq. (50.5) one finds

$$\begin{aligned} & \int_0^{2\pi} [\xi(\omega^2 M(\ddot{\mathbf{u}}_1) \cdot \delta\mathbf{u} + \boldsymbol{\sigma}_1 \cdot \delta\boldsymbol{\epsilon}) \\ & + \xi^2(\omega^2 M(\ddot{\mathbf{u}}_2) \cdot \delta\mathbf{u} + \boldsymbol{\sigma}_2 \cdot \delta\boldsymbol{\epsilon} + \boldsymbol{\sigma}_1 \cdot L_{11}(\mathbf{u}_1, \delta\mathbf{u})) \\ & + \xi^3(\omega^2 M(\ddot{\mathbf{u}}_3) \cdot \delta\mathbf{u} + \boldsymbol{\sigma}_3 \cdot \delta\boldsymbol{\epsilon} + \boldsymbol{\sigma}_1 \cdot L_{11}(\mathbf{u}_2, \delta\mathbf{u}) \\ & + \boldsymbol{\sigma}_2 \cdot L_{11}(\mathbf{u}_1, \delta\mathbf{u})) + \dots] d\tau = 0 \end{aligned} \quad (50.10)$$

Finally, the relation between the frequency ω and the amplitude ξ is found:

$$\frac{\omega^2}{\omega_0^2} = 1 + a_D \xi + b_D \xi^2 + \dots \quad (50.11)$$

where

$$a_D = \frac{\int_0^{2\pi} \frac{3}{2} \boldsymbol{\sigma}_1 \cdot L_2(\mathbf{u}_1) d\tau}{\omega_0^2 \int_0^{2\pi} M(\dot{\mathbf{u}}_1) \cdot \dot{\mathbf{u}}_1 d\tau} \quad (50.12)$$

and

$$b_D = \frac{\int_0^{2\pi} (2\boldsymbol{\sigma}_1 \cdot L_{11}(\mathbf{u}_1, \mathbf{u}_2) + \boldsymbol{\sigma}_2 \cdot L_2(\mathbf{u}_1)) d\tau}{\omega_0^2 \int_0^{2\pi} M(\dot{\mathbf{u}}_1) \cdot \dot{\mathbf{u}}_1 d\tau} \quad (50.13)$$

Eq. (50.11) is a compact representation of the effect of the vibration amplitude on the frequency. The calculation of the second order coefficient b_D requires the calculation of the second order field \mathbf{u}_2 . By equating the term multiplying ξ^2 in Eq. (50.10) to zero the second order state problem is obtained

$$\omega^2 M(\ddot{\mathbf{u}}_2) \cdot \delta\mathbf{u} + \boldsymbol{\sigma}_2 \cdot \delta\boldsymbol{\epsilon} + \boldsymbol{\sigma}_1 \cdot L_{11}(\mathbf{u}_1, \delta\mathbf{u}) = 0 \quad (50.14)$$

The second order field \mathbf{u}_2 is time dependent and it is actually constituted by two parts. In order to obtain the two contributions following the same line as considered by Tiso [85], the time dependence of the vibration mode \mathbf{u}_1 can be written explicitly as

$$\begin{aligned} \mathbf{u}_1 &= \hat{\mathbf{u}}_1 \cos \tau \\ \boldsymbol{\epsilon}_1 &= \hat{\boldsymbol{\epsilon}}_1 \cos \tau \\ \boldsymbol{\sigma}_1 &= \hat{\boldsymbol{\sigma}}_1 \cos \tau \end{aligned} \quad (50.15)$$

where the hatted quantities are the spatial shapes, which are multiplied by a harmonic time response. By substituting Eq. (50.15) into the second order state problem (50.14) one obtains

$$\omega^2 M(\ddot{\mathbf{u}}_2) \cdot \delta \mathbf{u} + \boldsymbol{\sigma}_2 \cdot \delta \boldsymbol{\epsilon} = -\frac{1}{2}(1 + \cos 2\tau) \hat{\boldsymbol{\sigma}}_1 \cdot L_{11}(\hat{\mathbf{u}}_1, \delta \mathbf{u}) \quad (50.16)$$

It can be noticed that the right hand side of Eq. (50.16) consists of a constant forcing term and a harmonic forcing term, respectively. The solution can therefore be split into two parts:

$$\mathbf{u}_2 = \hat{\mathbf{u}}_{2_1} + \hat{\mathbf{u}}_{2_2} \cos 2\tau \quad (50.17)$$

which are the solution of the two problems

$$\begin{aligned} \hat{\boldsymbol{\sigma}}_{2_1} \cdot \delta \boldsymbol{\epsilon} &= -\frac{1}{2} \hat{\boldsymbol{\sigma}}_1 \cdot L_{11}(\hat{\mathbf{u}}_1, \delta \mathbf{u}) \\ -4\omega^2 M(\hat{\mathbf{u}}_{2_2}) \cdot \delta \mathbf{u} + \hat{\boldsymbol{\sigma}}_{2_2} \cdot \delta \boldsymbol{\epsilon} &= -\frac{1}{2} \hat{\boldsymbol{\sigma}}_1 \cdot L_{11}(\hat{\mathbf{u}}_1, \delta \mathbf{u}) \end{aligned} \quad (50.18)$$

where the mass operator of the first problem has been dropped since $\hat{\mathbf{u}}_{2_1}$ does not depend on time. By accounting for the two different contributions of the second order field \mathbf{u}_2 and carrying out the time integrations the coefficients a_D and b_D assume the form

$$a_D = 0 \quad (50.19)$$

$$\begin{aligned} b_D &= [2\boldsymbol{\sigma}_1 \cdot L_{11}(\hat{\mathbf{u}}_1, \hat{\mathbf{u}}_{2_1}) + \boldsymbol{\sigma}_1 \cdot L_{11}(\hat{\mathbf{u}}_1, \hat{\mathbf{u}}_{2_2}) \\ &\quad + H(L_1(\hat{\mathbf{u}}_{2_1})) \cdot L_2(\hat{\mathbf{u}}_1) + \frac{1}{2} H(L_1(\hat{\mathbf{u}}_{2_2})) \cdot L_2(\hat{\mathbf{u}}_1) \\ &\quad + \frac{3}{8} H(L_2(\hat{\mathbf{u}}_1)) \cdot L_2(\hat{\mathbf{u}}_1)] / \omega_0^2 M(\hat{\mathbf{u}}_1) \cdot \hat{\mathbf{u}}_1 \end{aligned} \quad (50.20)$$

Unlike the coefficient in the perturbation expansion for the load parameter in the initial postbuckling analysis, the first coefficient of the perturbation expansion for the frequency in the dynamic analysis, a_D , is always zero, also for non-symmetric structures. A positive coefficient b_D represents a hardening behaviour, i.e. the frequency of vibration increases with increasing amplitude. Conversely, a negative coefficient b_D indicates a softening behaviour.

An important case is constituted by multiple vibration modes associated with the same frequency. The vibration modes can interact and modify the frequency-amplitude curve. This situation can be treated by assuming the displacement field as a linear combination of the m modes \mathbf{u}_i ($i = 1, 2, \dots, m$) plus contribution of the second order modes \mathbf{u}_{ij} , as

$$\mathbf{u} = \xi_i \mathbf{u}_i + \xi_i \xi_j \mathbf{u}_{ij} + \dots \quad (50.21)$$

with the corresponding strain and stress fields

$$\boldsymbol{\epsilon} = \xi_i \boldsymbol{\epsilon}_i + \xi_i \xi_j \boldsymbol{\epsilon}_{ij} + \dots \quad (50.22)$$

$$\boldsymbol{\sigma} = \xi_i \boldsymbol{\sigma}_i + \xi_i \xi_j \boldsymbol{\sigma}_{ij} + \dots \quad (50.23)$$

where the summation convention is used for repeated indices. The derivation follows the same line as the multi-mode analysis for initial postbuckling [14]. Only the main results are reported here. The nonlinear frequency-amplitude relations are obtained in the following form:

$$\xi_I \left(1 - \frac{\omega^2}{\omega_{0I}^2} \right) + \xi_i \xi_j a_{ijI} + \xi_i \xi_j \xi_k b_{ijkI} = 0, \quad I = 1, 2, \dots, m \quad (50.24)$$

The a_D and b_D coefficients are found as

$$a_{D_{ijI}} = \frac{1}{\omega_{0I}^2 \Delta_I} \int_0^{2\pi} [\boldsymbol{\sigma}_I \cdot L_{11}(\mathbf{u}_i, \mathbf{u}_j) + 2\boldsymbol{\sigma}_i \cdot L_{11}(\mathbf{u}_j, \mathbf{u}_I)] d\tau \quad (50.25)$$

$$b_{D_{ijkI}} = \frac{1}{\omega_{0I}^2 \Delta_I} \int_0^{2\pi} \frac{1}{2} [\boldsymbol{\sigma}_{Ii} \cdot L_{11}(\mathbf{u}_j, \mathbf{u}_k) + \boldsymbol{\sigma}_{ij} \cdot L_{11}(\mathbf{u}_k, \mathbf{u}_I) + \boldsymbol{\sigma}_I \cdot L_{11}(\mathbf{u}_i, \mathbf{u}_{jk}) + \boldsymbol{\sigma}_i \cdot L_{11}(\mathbf{u}_I, \mathbf{u}_{jk}) + 2\boldsymbol{\sigma}_i \cdot L_{11}(\mathbf{u}_j, \mathbf{u}_{kI})] d\tau \quad (50.26)$$

where

$$\Delta_I = \int_0^{2\pi} M(\dot{\mathbf{u}}_I) \cdot \dot{\mathbf{u}}_I d\tau \quad (50.27)$$

The second order fields \mathbf{u}_{JK} are the solutions of the second order state problems

$$\omega^2 M(\ddot{\mathbf{u}}_{jk}) \cdot \delta \mathbf{u} + \sigma_{jk} \cdot \delta \epsilon = -\frac{1}{2} [\sigma_j \cdot L_{11}(\mathbf{u}_k, \delta \mathbf{u}) + \sigma_k \cdot L_{11}(\mathbf{u}_j, \delta \mathbf{u})] \quad (50.28)$$

50.2 Finite Element Implementation

In finite element notation, the eigenvalue problem defined by Eq. (50.4) giving the natural frequency and the corresponding single vibration mode $\hat{\mathbf{u}}_1$ can be written as

$$[\mathbf{K}_M - \omega_1^2 \mathbf{M}] \hat{\mathbf{u}}_1 = \mathbf{0} \quad (50.29)$$

where \mathbf{K}_M is the material stiffness matrix and \mathbf{M} is the mass matrix. After the calculation of the natural frequency and corresponding vibration mode, the initial curvature of the frequency–amplitude relation can be computed with a modest additional computational cost, by solving for the corresponding second order modes. The linear problem for the second order modes $\hat{\mathbf{u}}_{2_1}$, $\hat{\mathbf{u}}_{2_2}$ are

$$[\mathbf{K}_M] \hat{\mathbf{u}}_{2_1} = \mathbf{g}(\hat{\mathbf{u}}_1) \quad (50.30)$$

$$[\mathbf{K}_M - 4\omega_1^2 \mathbf{M}] \hat{\mathbf{u}}_{2_2} = \mathbf{g}(\hat{\mathbf{u}}_1) \quad (50.31)$$

with the orthogonality conditions

$$\hat{\mathbf{u}}_1^T \mathbf{M} \hat{\mathbf{u}}_{2_1} = 0 \quad (50.32)$$

$$\hat{\mathbf{u}}_1^T \mathbf{M} \hat{\mathbf{u}}_{2_2} = 0 \quad (50.33)$$

The forcing term $\mathbf{g}(\hat{\mathbf{u}}_1)$ is defined at element level as

$$\mathbf{g}(\hat{\mathbf{u}}_1) = -\frac{1}{2} [\mathbf{B}_L^T \mathbf{H} \mathbf{B}_{NL}(\hat{\mathbf{u}}_1) \hat{\mathbf{u}}_1 + 2\mathbf{B}_{NL}^T(\hat{\mathbf{u}}_1) \mathbf{H} \mathbf{B}_L \hat{\mathbf{u}}_1] \quad (50.34)$$

The global $\mathbf{g}(\hat{\mathbf{u}}_1)$ is formed through an assembly operation of contributions calculated at element level.

Chapter 51

Strength Reduction Method

In strength reduction method the strength characteristics of the structural materials are reduced by a factor until the loss of stability or failure of the structure occurs. The reciprocal of this reduction factor is identified as the factor of safety associated with the structure under investigation. In DIANA strength reduction method is implemented as a separate module, named REDUCT. Strength reduction method is typically used for the assessment of slope stability where dominantly a Mohr-Coulomb or similar material model is used [36, 26]. Therefore, cohesion c and friction angle ϕ are reduced to assess the slope stability.

In a strength reduction analysis, first self weight and additional loads are applied on the structure using a standard phased or non-phased nonlinear static analysis and an equilibrium state is obtained. Next the module REDUCT is run to determine the factor of safety iteratively. At each iteration a nonlinear analysis is carried out. In the following, first the iterative procedure to determine factor of safety (FS) is discussed. Next the nonlinear static analysis that is carried out at each iteration is considered with more details.

51.1 Iterative Procedure for Factor of Safety

The flow-chart for the determination of the factor of safety (FS) is shown in Figure 51.1. The iteration starts with the factor of safety, $FS_n = FS_0$. The user can specify FS_0 , by default $FS_0 = 1$. Next FS_n is incremented by ΔFS to FS_{n+1} where ΔFS is a user input. With the updated FS , the cohesion c , and the tangent of the friction angle $\tan \phi$ are scaled down. The resulting yield surface is shown in Figure 51.2 where c and ϕ are the scaled down respectively to c_f and ϕ_f . Now with the reduced strength parameters a new equilibrium is sought by carrying out a nonlinear static analysis. If a new equilibrium is found, i.e. the analysis converges, then FS is incremented again and a new equilibrium is sought again with further reduced strength parameters. The process is repeated until the analysis diverges or does not converge within a certain number of iterations, e.g. 50 or some user-defined failure criteria, e.g. maximum equivalent plastic strain is exceeded. At this stage ΔFS is set to $\Delta FS = \Delta FS/2$ which reduces FS to a lower value so that equilibrium can be found again. This iterative procedure continues until ΔFS becomes smaller than a user specified tolerance, so that FS is determined with sufficient accuracy.

51.2 Nonlinear Analysis at Each Iteration

The nonlinear static analysis to find the equilibrium with reduced strength parameters is discussed in this section. First the reduced stress σ due to reduction of the strength parameters is computed. In order to do that, the elastic strain corresponding to the stress σ_0 from the preceding nonlinear analysis is calculated as

$$\epsilon_0 = \mathbf{SE}^{-1} \sigma_0 \quad (51.1)$$

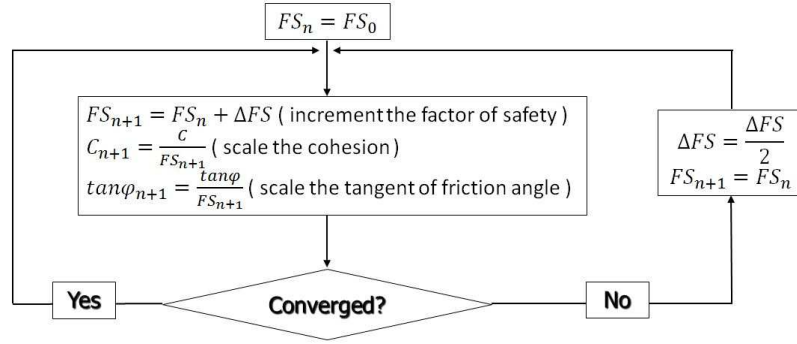


Figure 51.1: Flow-chart for the determination of FS

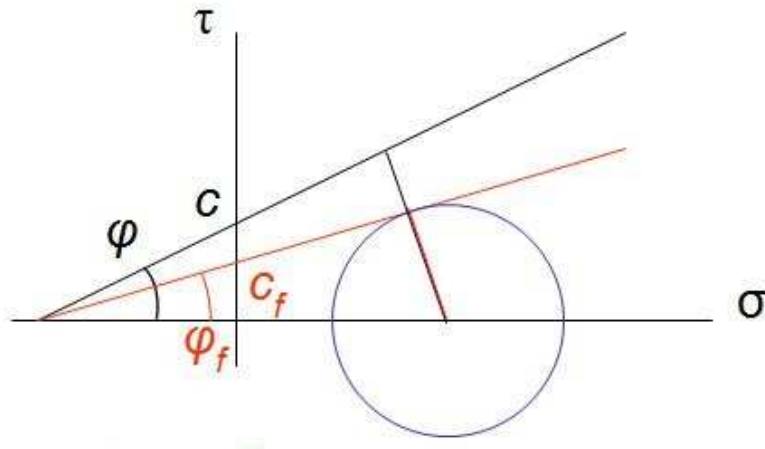


Figure 51.2: Mohr-Coulomb yield surface after strength reduction

where \mathbf{SE} is the elastic stress-strain matrix. Once $\boldsymbol{\varepsilon}_0$ is available the reduced stress $\boldsymbol{\sigma}$ is obtained by invoking the Mohr-Coulomb (or Mohr-Coulomb like) material routine with $\boldsymbol{\varepsilon}_0$ as the total strain. Finally the out-of-balance force (residual force) \mathbf{g} to find the equilibrium is computed as

$$\mathbf{g} = \int \mathbf{B}^T (\boldsymbol{\sigma}_0 - \boldsymbol{\sigma}) \quad (51.2)$$

where \mathbf{B} is the strain-displacement relation matrix and the integration sign indicates integration over element level leading to element force vector followed by an assembly process to obtain the global force vector.

Chapter 52

Engineering Liquefaction Analysis

Engineering liquefaction analysis is an engineering approach to compute liquefaction induced deformation after Yasuda et al. [90]. It is assumed that the liquefaction induced deformation occurs due to the reduction of the shear modulus of the ground. The stress-

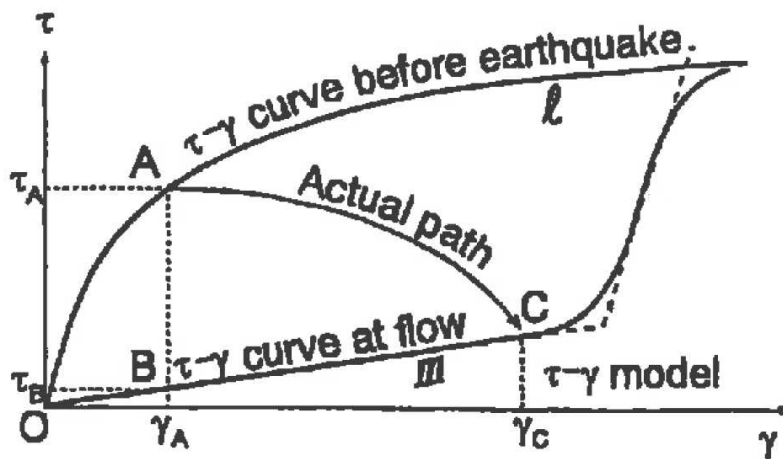


Figure 52.1: Concept of engineering liquefaction analysis

strain curves in Figure 52.1 shows the concept behind the engineering liquefaction analysis. The curve which starts from O and passes A denotes the curve l which is the material response of the ground before liquefaction caused by an earthquake. The point A is supposed to be initial state of soil element in the ground. After liquefaction the curve l moves to curve m due to the drop of the shear modulus. As a result the state point A on the curve l moves to C on the curve m along with liquefaction induced strain, $\gamma_C - \gamma_A$. The stress in the other part of the ground which is not subject to liquefaction must increase to hold the applied load on the ground (such as self weight).

In DIANA this approach is implemented as a separate module, named ELIQUE. In an engineering liquefaction analysis, first self weight and additional loads are applied on the ground using a standard phased or non-phased nonlinear static analysis and an equilibrium state is obtained. Next the module ELIQUE is run where the shear modulus is reduced by a used defined reduction factor and a new equilibrium is obtained which gives additional deformation i.e. the liquefaction induced deformation. It is noticeable in Figure 52.1 that curve m regains its stiffness after a threshold strain C . This behaviour is simulated by a bilinear elasticity model (see Modified Elasticity in Volume *Material Library*). The following steps are done during the analysis:

1. Computation of the reduced stress-strain matrix

It is assumed that the shear modulus drops due to liquefaction but the bulk modulus remains the same. Let α be the user specified shear modulus reduction factor. Then the resulting shear modulus at liquefied state will be

$$G_m = \alpha G_l = \frac{\alpha E_l}{2(1 + \nu_l)} \quad (52.1)$$

The bulk modulus is multiplied by a factor β :

$$K_m = \beta K_l \quad (52.2)$$

Therefore, Young's modulus and Poisson's ratio in the liquefied state will be

$$E_m = \frac{9K_m G_m}{3K_m + G_m} \quad (52.3)$$

$$\nu_m = \frac{3K_m - 2G_m}{2(3K_m + G_m)} \quad (52.4)$$

Here subscripts l and m stands for the elastic properties on curve l and m in Figure 52.1, respectively. With E_m and ν_m at hand the reduced elastic stress-strain matrix \mathbf{SE}_m can be setup. The factors α and β are user input specified by the material parameter **SHRRED** (see Volume *Material Library*). Generally $\alpha \ll 1$ and $\beta \gg 1$ to make K_m significantly higher than G_m so that they can simulate the elastic behaviour of the liquefied soil.

2. Computation of the reduced stress

The elastic strain corresponding to σ_A is

$$\epsilon_A = \mathbf{SE}_l^{-1} \sigma_A \quad (52.5)$$

where σ_A is available from the previous nonlinear analysis. The reduced stress σ_B can be expressed as

$$\sigma_B = \mathbf{SE}_m \epsilon_A \quad (52.6)$$

Here subscripts A and B stand for the state points A and B shown in Figure 52.1. In Figure 52.1, τ_A , τ_B , and γ_A are the shear stress and strain components of the stress and strain tensors σ_A , σ_B , and ϵ_A respectively.

3. Finding new equilibrium

Finally the out-of-balance force (residual force) \mathbf{g} to find the new equilibrium is computed as

$$\mathbf{g} = \int \mathbf{B}^T (\sigma_A - \sigma_B) \quad (52.7)$$

where \mathbf{B} is the strain-displacement relation matrix and the integration sign indicates integration over element level leading to element force vector followed by an assembly process to obtain the global force vector.

Chapter 53

Engineering Creep Analysis

Engineering creep analysis is an engineering approach to compute deformation due to creep. It is assumed that the creep deformation occurs due to the reduction of Young's modulus and Poisson's ratio of the ground.

In DIANA this approach is implemented as a separate module, named ECREEP. In an engineering creep analysis, first self weight and additional loads are applied on the ground using a standard phased or non-phased nonlinear static analysis and an equilibrium state is obtained. Next, module ECREEP is run where the Young's modulus and Poisson's ratio are replaced by a user-defined long term Young's modulus and Poisson's ratio, specified by the material parameters YOUNLT and POISLT respectively, see Volume *Material Library*, and a new equilibrium is obtained, which gives additional deformation i.e. the deformation caused by creep effect. The following steps are done during the analysis:

1. **Computation of the reduced stress-strain matrix**

The reduced (long term) elastic stress-strain matrix \mathbf{SE}_B is computed using the long term Young's modulus and Poisson's ratio. The original elastic stress-strain matrix \mathbf{SE}_A is retained in the database as it will be necessary later for the computation of the reduced stress due to the reduction of the elastic stiffness.

2. **Computation of the reduced stress**

The elastic strain corresponding to σ_A is

$$\epsilon_A = \mathbf{SE}_A^{-1}(\sigma_A - \sigma_0) \quad (53.1)$$

where σ_A is available from the previous nonlinear analysis and σ_0 is a reference stress (for instance stress caused by self weight). The influence of σ_0 is excluded from the creep analysis by deducting it from σ_A . The reference stress σ_0 has to be made available from the previous nonlinear analysis by using the command REFERENCE STRESS, see [§ 13.3.8 p. 244]. The reduced stress σ_B can be expressed as

$$\sigma_B = \mathbf{SE}_B \epsilon_A + \sigma_0 \quad (53.2)$$

3. **Finding new equilibrium**

Finally, the out-of-balance force (residual force) \mathbf{g} to find the new equilibrium is computed as

$$\mathbf{g} = \int \mathbf{B}^T (\sigma_A - \sigma_B) \quad (53.3)$$

where \mathbf{B} is the strain-displacement relation matrix and the integration sign indicates integration over element level leading to element force vector followed by an assembly process to obtain the global force vector.

Chapter 54

Stability Analysis

The set of equations representing nodal equilibrium are written as

$$\mathbf{r}(\mathbf{u}) = \mathbf{f}(\mathbf{u}) \quad (54.1)$$

where \mathbf{r} represents the internal force vector, \mathbf{f} the external force vector and \mathbf{u} represents the vector of nodal degrees of freedom (displacements). Displacements \mathbf{u}_{crit} are searched for such that

$$\mathbf{r}(\mathbf{u}_{\text{crit}}) = \mathbf{f}(\mathbf{u}_{\text{crit}}) \quad (54.2)$$

and

$$\mathbf{r}(\mathbf{u}_{\text{crit}} + \delta\mathbf{u}) = \mathbf{f}(\mathbf{u}_{\text{crit}} + \delta\mathbf{u}) \quad (54.3)$$

i.e., incremental variations $\delta\mathbf{u}$ to the solution \mathbf{u}_{crit} exist such that the equations of equilibrium remain satisfied. The displacement vector \mathbf{u}_{crit} is called a stability point in the space of possible displacement vectors. As $\mathbf{u}_{\text{crit}} + \delta\mathbf{u}$ is ‘close to’ \mathbf{u}_{crit} , Eq. (54.3) can be linearized with respect to the nodal degrees of freedom:

$$\mathbf{r}(\mathbf{u}_{\text{crit}}) + \left(\frac{\partial \mathbf{r}}{\partial \mathbf{u}} \right)_{\text{crit}} \delta\mathbf{u} \approx \mathbf{f}(\mathbf{u}_{\text{crit}}) + \left(\frac{\partial \mathbf{f}}{\partial \mathbf{u}} \right)_{\text{crit}} \delta\mathbf{u} \quad (54.4)$$

Introducing the tangent stiffness matrix

$$\mathbf{K} = \left(\frac{\partial \mathbf{r}}{\partial \mathbf{u}} \right)_{\text{crit}} \quad (54.5)$$

leads to

$$\mathbf{r}(\mathbf{u}_{\text{crit}}) + \mathbf{K}\delta\mathbf{u} \approx \mathbf{f}(\mathbf{u}_{\text{crit}}) + \left(\frac{\partial \mathbf{f}}{\partial \mathbf{u}} \right)_{\text{crit}} \delta\mathbf{u} \quad (54.6)$$

And subtracting Eq. (54.2) yields

$$\mathbf{K}\delta\mathbf{u} \approx \left(\frac{\partial \mathbf{f}}{\partial \mathbf{u}} \right)_{\text{crit}} \delta\mathbf{u} \quad (54.7)$$

For conservative loading, the external force vector \mathbf{f} does not depend on \mathbf{u} and thus

$$\mathbf{K}\delta\mathbf{u} \approx \mathbf{0} \quad (54.8)$$

54.1 Linear Buckling (Euler)

Only geometrical nonlinear effects are taken into account (any physical nonlinearities are neglected), and thus the tangent stiffness matrix reads (see Bathe [5])

$$\mathbf{K} = \mathbf{K}_L + \mathbf{K}_{NL} \quad (54.9)$$

where

$$\mathbf{K}_L = \int_{V_0} \mathbf{B}_L^T \mathbf{D} \mathbf{B}_L dV_0 \quad \text{with:} \quad \mathbf{B}_L = \mathbf{B}_{L0} + \mathbf{B}_{L1} \quad (54.10)$$

with \mathbf{B}_{L1} due to an initial displacement effect. See Bathe [5] for a definition of the matrices \mathbf{B}_{L0} and \mathbf{B}_{L1} . Furthermore

$$\mathbf{K}_{NL} = \int_{V_0} \mathbf{B}_{NL}^T \boldsymbol{\tau} \mathbf{B}_{NL} dV_0 \quad (54.11)$$

where \mathbf{B}_{NL} is defined in Bathe [5]. Both the matrices \mathbf{K}_L and \mathbf{K}_{NL} have up to second order displacement contributions. In linear buckling analysis only first order displacement contributions are recollected however. This is described in the following.

Suppose a solution \mathbf{u}_{lin} from Eq. (54.1) linearized is known, i.e., \mathbf{u}_{lin} results from

$$\mathbf{K}_{L0} \mathbf{u}_{lin} = \mathbf{f} \quad (54.12)$$

where

$$\mathbf{K}_{L0} = \int_{V_0} \mathbf{B}_{L0}^T \mathbf{D} \mathbf{B}_{L0} dV_0 \quad (54.13)$$

with constant external loading \mathbf{f} . It can be questioned whether solutions \mathbf{u}_{crit} satisfying Eq. (54.2) and Eq. (54.8) exist such that

$$\mathbf{u}_{crit} = \lambda_{crit} \mathbf{u}_{lin} \quad (54.14)$$

Note that $\lambda_{crit} \mathbf{u}_{lin}$ would result from a loading $\mathbf{f}_{crit} = \lambda_{crit} \mathbf{f}$ on the linear system, as can be shown by multiplication of Eq. (54.12) by λ_{crit} :

$$\lambda_{crit} \mathbf{K}_{L0} \mathbf{u}_{lin} = \lambda_{crit} \mathbf{f} \quad (54.15)$$

Parameter λ_{crit} will be determined from the instability condition Eq. (54.8). Doing so, the following approximations are used:

- Only first order displacement contributions in matrix \mathbf{K}_L are taken into account. These are stored in the matrix \mathbf{K}_{LL} :

$$\mathbf{K}_{LL}(\mathbf{u}_{lin}) = \int_{V_0} (\mathbf{B}_{L1}^T \mathbf{D} \mathbf{B}_{L0} + \mathbf{B}_{L0}^T \mathbf{D} \mathbf{B}_{L1}) dV_0 \quad (54.16)$$

- The second Piola–Kirchhoff stresses:

$$\boldsymbol{\tau} \approx \lambda_{crit} \boldsymbol{\sigma}_{lin} \quad (54.17)$$

Application of these approximations leads to

$$\left(\mathbf{K}_{L0} + \lambda_{crit} (\mathbf{K}_{LL}(\mathbf{u}_{lin}) + \mathbf{K}_G(\mathbf{u}_{lin})) \right) \delta \mathbf{u} = \mathbf{0} \quad (54.18)$$

where

$$\mathbf{K}_G(\mathbf{u}_{lin}) = \int_{V_0} \mathbf{B}_{NL}^T \boldsymbol{\sigma}_{lin} \mathbf{B}_{NL} dV_0 \quad (54.19)$$

and

$$\mathbf{K}_{L0} = \int_{V_0} \mathbf{B}_{L0}^T \mathbf{D} \mathbf{B}_{L0} dV_0 \quad (54.20)$$

Eq. (54.18) is satisfied by nontrivial solutions for $\delta \mathbf{u}$ if

$$\det \left(\mathbf{K}_{L0} + \lambda_{crit} (\mathbf{K}_{LL}(\mathbf{u}_{lin}) + \mathbf{K}_G(\mathbf{u}_{lin})) \right) = 0 \quad (54.21)$$

And so, the general stability conditions Eq. (54.2) and Eq. (54.3) are replaced by the relatively simple conditions Eq. (54.15) and Eq. (54.21) in linear stability analysis. Eq. (54.21) is solved as a generalized eigenproblem written as

$$\mathbf{K}_{L0} \phi_i = -\lambda_{i,crit} (\mathbf{K}_{LL}(\mathbf{u}_{lin}) + \mathbf{K}_G(\mathbf{u}_{lin})) \phi_i \quad (54.22)$$

where ϕ_i is the i -th buckling mode and λ_i is the appropriate buckling value. The theory discussed in this section is referred to as *linear buckling theory*, or *Euler buckling theory*.

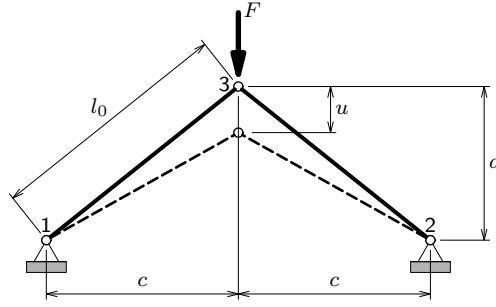


Figure 54.1: Arch, loading and deformation

Tutorial example. The theory of the linear Euler stability analysis will be illustrated with an example. Consider a simple arch structure made from two bars as shown in Figure 54.1 and loaded by a force F . Nodes 1 and 2 are fixed in space. The deformed configuration is shown with a dashed line. The two displacements of the joint are the only degrees of freedom. For reasons of symmetry, it is assumed a priori that the joint does not displace laterally. A vertical displacement u_{crit} , corresponding with a force F_{crit} , is searched for such that Eq. (54.2) and Eq. (54.8) are satisfied. The force F_{crit} is called the critical load. Firstly, the equilibrium condition Eq. (54.1) will be examined in more detail. Load F is in equilibrium with the normal forces N in the bars as shown in Figure 54.2. Mathematically this reads:

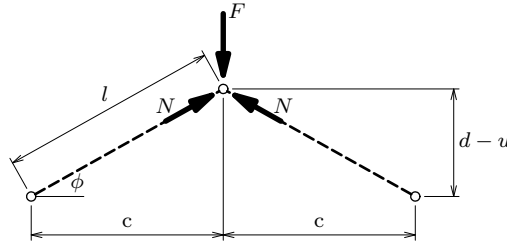


Figure 54.2: Arch, equilibrium node 3 deformed state

$$2N \sin \phi = 2N \frac{d - u}{l} = F \quad (54.23)$$

where l denotes the deformed length of the bars, in contrast with the initial length which is denoted by l_0 . The forces N only depends on the shortening or elongation of the bars. If geometrical nonlinearities (large deformations) are taken into account, this relation reads:

$$N = -A_0 \frac{l}{l_0} \tau \quad (54.24)$$

where A_0 is the initial cross-section of the bars. The second Piola–Kirchhoff stress τ is given by

$$\tau = E \gamma \quad (54.25)$$

where E denotes Young's modulus (the bars are assumed to be isotropic) and γ represents the Green–Lagrange strain given by

$$\gamma = \frac{1}{2} \left(\frac{c^2 + (d - u)^2}{l_0^2} - 1 \right) \quad (54.26)$$

Combination of Eq. (54.15) and Eq. (54.18) yields

$$-2 \frac{A_0 \tau}{l_0} (d - u) = F \quad (54.27)$$

Taking the first variation of Eq. (54.27) leads to:

$$2 \frac{A_0}{l_0} \left(\left(\frac{Ed^2}{l_0^2} \right) - \left(2 \frac{Edu}{l_0^2} - \frac{Eu^2}{l_0^2} \right) + (\tau) \right) = 0 \quad (54.28)$$

The first term in the left-hand-side of Eq. (54.28) is the linear stiffness; the second term is referred to as the stiffness due to the initial displacements and the last term is the geometrical stress–stiffness of the structure. For *Euler stability analysis* the following approximations are made:

1. Small displacements, i.e., u^2 is neglected.
2. $\tau \approx \sigma$ where σ is the Cauchy stress following from a geometrically linear analysis:

$$\tau = \sigma = E \varepsilon \quad (54.29)$$

where ε is the geometrical linear (small) strain:

$$\varepsilon = -\frac{d}{l_0^2} u \quad (54.30)$$

These assumptions lead to a more simple expression for Eq. (54.28):

$$2 \frac{A_0 E d}{l_0^3} (d - 3u) = 0 \quad (54.31)$$

It can be seen that applying an external load F_{crit} such that $u_{\text{crit}} = \frac{1}{3}d$ leads to instability because Eq. (54.31) becomes satisfied for such F . The required value for F_{crit} is determined by substituting $u = \frac{1}{3}d$ into Eq. (54.27).

It should be realized that due to the approximations the critical load F_{crit} as obtained from an Euler stability analysis may be in error. In order to demonstrate this, the exact solution of the stability problem will now be derived. Elaborating Eq. (54.28) without applying any approximations leads to:

$$\frac{EA_0}{l_0^3} (c^2 - l_0^2 + 3(d - u)^2) = 0 \quad (54.32)$$

Eq. (54.32) is satisfied if:

$$u_{\text{crit}} = d \left(1 + \frac{1}{\sqrt{3}} \right) \quad \text{or:} \quad u_{\text{crit}} = d \left(1 - \frac{1}{\sqrt{3}} \right) \quad (54.33)$$

with a load F_{crit} following from Eq. (54.27). Thus, exactly solving the stability problem renders two solutions. The most critical of both is approximately $0.42d$ whereas the linear buckling analysis rendered $0.33d$. Neglect of the initial displacement terms in the linear buckling analysis would have given d .

54.2 Imperfections

The constant part of the stiffness matrix of the imperfect structure is denoted by \mathbf{K}_{L0}^I . Once the imperfect geometry is known, this matrix \mathbf{K}_{L0}^I can be determined. The imperfect geometry is defined by a displacement field \mathbf{u}^I from the perfect structure to the imperfect structure. Thus the matrix \mathbf{K}_{L0}^I actually depends on this displacement field:

$$\mathbf{K}_{L0}^I = \mathbf{K}_{L0}^I(\mathbf{u}^I) \quad (54.34)$$

The field \mathbf{u}^I can be chosen in different ways:

Compatible with critical buckling mode. A buckling mode from an analysis without imperfections is used as imperfection pattern. Generally, the lowest buckling mode is most critical. For this pattern the critical buckling value may be decreased more substantially than for an arbitrary pattern.

Random. Random imperfections may be used to simulate random imperfections of reality.

User-specified. A specific field, for instance taken from measurements, may be defined by the user.

Summarizing, the buckling criterion for the imperfect structure reads:

$$\det\left(\mathbf{K}_{L0}^I(\mathbf{u}^I) + \lambda(\mathbf{K}_{LL}(\mathbf{u}_{lin}) + \mathbf{K}_G(\mathbf{u}_{lin}))\right) = 0 \quad (54.35)$$

For a fixed imperfection pattern (i.e., user-specified or random), an eigenvalue calculation on Eq. (54.35) is performed to calculate the critical buckling load. However, for the lowest buckling mode compatible imperfection pattern, a two-step strategy is necessary. First, an eigenvalue determination on Eq. (54.35) without imperfections ($\mathbf{u}^I = \mathbf{0}$) is performed. The corresponding critical buckling mode is used to set up the imperfections \mathbf{u}^I . A second eigenvalue analysis of Eq. (54.35), now including \mathbf{u}^I , is performed to add the effect of the imperfections.

54.3 Postbuckling Analysis

The displacement vector \mathbf{u}_{crit} is a stability point in the space of possible displacement vectors. The definition of \mathbf{u}_{crit} is given by Eq. (54.3) which must be valid for arbitrary ‘small’ $\delta\mathbf{u}$. The purpose of perturbation analysis is to calculate a postbuckling displacement field \mathbf{u}_{pb} satisfying Eq. (54.1) $\mathbf{r}(\mathbf{u}_{pb}) = \mathbf{f}(\mathbf{u}_{pb})$ but being different from the primary path $\mathbf{u} = \lambda\mathbf{u}_{lin}$ with λ being the load parameter.

Assumption: there are M coinciding or nearly coinciding interacting buckling modes, denoted by ϕ_k with $k = 1, \dots, M$. In the theory of mode interaction the initial postbuckling displacement field \mathbf{u}_{pb} is written as

$$\mathbf{u}_{pb} = \lambda\mathbf{u}_{lin} + a_i\phi_i + a_j\phi_j \quad (54.36)$$

where \mathbf{u}_{ij} is called the second order displacement vector and a_i should be interpreted as amplitude of the respective mode. In literature¹ it is shown that \mathbf{u}_{ij} must be calculated by solving the system

$$(\mathbf{K}_{L0} + \lambda_p\lambda_1\mathbf{K}_G(\mathbf{u}_{lin}))\mathbf{u}_{ij} = \mathbf{f}_{ij} \quad (54.37)$$

where $\lambda_p \neq 1$ must be specified by the user. Applying the orthogonality conditions

$$\phi_k^T \mathbf{K}_{L0} \mathbf{u}_{ij} = 0 \quad \text{with } k = 1, \dots, M \quad (54.38)$$

where \mathbf{f}_{ij} is defined as the mode interaction load vector

$$\mathbf{f}_{ij} = \int_{V_0} \left(\mathbf{B}_{L0}^T \boldsymbol{\sigma}_{ij} + \mathbf{B}_{NL}^T(\phi_i) \boldsymbol{\sigma}_j + \mathbf{B}_{NL}^T(\phi_j) \boldsymbol{\sigma}_i \right) dV_0 \quad (54.39)$$

where $\boldsymbol{\sigma}_{ij}$ is the stress related to interaction of modes i and j and $\boldsymbol{\sigma}_i$ is the stress related to mode i . Further, the potential can be written as a function of the load parameter λ and the mode amplitudes a_i

$$P(a_i, \lambda) = \frac{1}{2} \sum_{I=1}^M \left(1 - \frac{\lambda}{\lambda_I} \right) a_I a_I + A_{ijk} a_i a_j a_k + A_{ijkl} a_i a_j a_k a_l \quad (54.40)$$

where

$$A_{ijk} = \frac{1}{2} \int_{V_0} \boldsymbol{\sigma}_{ij} \boldsymbol{\varepsilon}_k dV_0 \quad (54.41)$$

and

$$A_{ijkl} = \frac{1}{8} \int_{V_0} \boldsymbol{\sigma}_{ij} \boldsymbol{\varepsilon}_{kl} dV_0 - \frac{1}{2} \mathbf{u}_{ij} \mathbf{f}_{kl} \quad (54.42)$$

are the third and fourth order potential terms, respectively.

From the potential function the nonlinear equilibrium equations after buckling, the *continuation analysis*, can be calculated in a stepwise approach. Points of equilibrium are indicated by terms of a_i and λ . From these data, the postbuckling displacement field \mathbf{u}_{pb} can be derived using Eq. (54.36).

¹See for instance Koiter [52], Van Erp [28], Byskov & Hutchinson [14].

54.4 Extensions of Postbuckling Analysis

The following extensions of postbuckling analysis are discussed in this section:

- Inclusion of prebuckling nonlinearity
- Effect of imperfection
- Dynamic buckling analysis

These extensions are based on the work of Tiso [85] and Rahman [72]. In these works functional notation introduced by Budiansky [12] was used. Additionally, the meaning of many of the symbols used in this section are different than in the previous sections. Therefore, along with the discussion about the underlying background theory, the necessary symbols and notations will be reintroduced briefly.

54.4.1 Functional Notation

If \mathbf{u} and $\boldsymbol{\epsilon}$ denote displacement and strain fields respectively, then in functional notation strain–displacement relation is written as

$$\boldsymbol{\epsilon} = L_1(\mathbf{u}) + \frac{1}{2}L_2(\mathbf{u}) \quad (54.43)$$

Here L_1 is a linear operator and L_2 is a quadratic operator. Therefore, $L_1(\mathbf{u})$ is a linear functional representing the linear part of strain and $L_2(\mathbf{u})$ is a quadratic functional representing the nonlinear part of strain. Further, the bilinear operator L_{11} is defined such that

$$L_2(\mathbf{u} + \mathbf{v}) = L_2(\mathbf{u}) + 2L_{11}(\mathbf{u}, \mathbf{v}) + L_2(\mathbf{v}) \quad (54.44)$$

From Eq. (54.44) it follows that

$$L_{11}(\mathbf{u}, \mathbf{v}) = L_{11}(\mathbf{v}, \mathbf{u}) \quad (54.45)$$

$$L_{11}(\mathbf{u}, \mathbf{u}) = L_2(\mathbf{u}) \quad (54.46)$$

54.4.2 The Perturbation Method

The perturbation method based postbuckling analysis [§ 54.3 p.621] is based on a linear prebuckling analysis. In the following the perturbation method is discussed with the inclusion of prebuckling nonlinearity. The variables $(\mathbf{u}, \boldsymbol{\epsilon}, \boldsymbol{\sigma})$ denoting displacement, strain, and stress of the postbuckling equilibrium state can be expanded in a multi-mode form about the prebuckling equilibrium state $(\mathbf{u}_0, \boldsymbol{\epsilon}_0, \boldsymbol{\sigma}_0)$

$$\begin{aligned} \mathbf{u} &= \mathbf{u}_0(\lambda) + \mathbf{u}_i\xi_i + \mathbf{u}_{ij}\xi_i\xi_j + \dots \\ \boldsymbol{\epsilon} &= \boldsymbol{\epsilon}_0(\lambda) + \boldsymbol{\epsilon}_i\xi_i + \boldsymbol{\epsilon}_{ij}\xi_i\xi_j + \dots \\ \boldsymbol{\sigma} &= \boldsymbol{\sigma}_0(\lambda) + \boldsymbol{\sigma}_i\xi_i + \boldsymbol{\sigma}_{ij}\xi_i\xi_j + \dots \end{aligned} \quad (54.47)$$

where $i, j = 1, 2 \dots m$. Here m is the number of buckling modes considered for the multi-mode analysis. \mathbf{u}_0 is the prebuckling state, \mathbf{u}_i is the buckling mode i , \mathbf{u}_{ij} is the second order mode, and $\boldsymbol{\epsilon}$ and $\boldsymbol{\sigma}$ are the corresponding strains and stresses. In Eq. (54.47) lower case repeated indices imply summation while upper case indices do not imply summation unless otherwise stated.

The necessary equations for the bifurcation buckling loads λ_i and the corresponding buckling modes are:

$$\boldsymbol{\epsilon}_i = L_1(\mathbf{u}_i) + L_{11}(\mathbf{u}_c, \mathbf{u}_i) \quad (54.48)$$

$$\boldsymbol{\sigma}_i = H(\boldsymbol{\epsilon}_i) \quad (54.49)$$

$$\boldsymbol{\sigma}_c \cdot \delta\boldsymbol{\epsilon}_c + \boldsymbol{\sigma}_c \cdot L_{11}(\mathbf{u}_i, \delta\mathbf{u}) = 0 \quad (54.50)$$

where the prebuckling quantities with the subscript $()_c$ are evaluated at the lowest critical or bifurcation load and H is the linear elastic stress-strain operator.

In addition it will be assumed that $(\lambda - \lambda_I)$ admits the asymptotic perturbation expansion:

$$\xi_I(\lambda - \lambda_I) = a_{Ijk}\lambda_I\xi_j\xi_k + b_{Ijkl}\lambda_I\xi_j\xi_k\xi_l + \dots \quad (54.51)$$

In view of Eq. (54.51), if a plot of load parameter (λ) versus the mode amplitudes (ξ_i) is made then a_{Ijk} and b_{Ijkl} coefficients respectively indicate the slopes and curvatures of the postbuckling curve.

With some manipulations one can finally obtain the necessary equations for the determination of the second order modes \mathbf{u}_{ij}

$$\epsilon_{ij} = L_1(\mathbf{u}_{ij}) + L_{11}(\mathbf{u}_c, \mathbf{u}_{ij}) + \frac{1}{2}L_{11}(\mathbf{u}_i, \mathbf{u}_j) \quad (54.52)$$

$$\sigma_{ij} = H(\epsilon_{ij}) \quad (54.53)$$

$$\sigma_{ij} \cdot \delta \epsilon_c + \sigma_c \cdot L_{11}(\mathbf{u}_{ij}, \delta \mathbf{u}) + \frac{1}{2}[\sigma_i \cdot L_{11}(\mathbf{u}_j, \delta \mathbf{u}) + \sigma_j \cdot L_{11}(\mathbf{u}_i, \delta \mathbf{u})] = 0 \quad (54.54)$$

The second order modes \mathbf{u}_{ij} are further subject to the following orthogonality condition

$$\dot{\sigma}_c \cdot L_{11}(\mathbf{u}_i, \mathbf{u}_{ij}) + \sigma_1 \cdot L_{11}(\dot{\mathbf{u}}_c, \mathbf{u}_{ij}) + \sigma_2 \cdot L_{11}(\dot{\mathbf{u}}_c, \mathbf{u}_i) = 0 \quad (54.55)$$

where $(\dot{}) = \frac{\partial}{\partial \lambda}()$.

In order to obtain the expression for b_{Ijkl} the following expansion of the total potential energy, $P(\xi, \lambda)$ of the structure in the postbuckling regime given by Van Erp [28], and Byskov & Hutchinson [14] will be used. However, this expansion was derived based on linear prebuckling state, here the additional terms resulting from nonlinearity of the prebuckling state will be accounted for.

$$P(\xi, \lambda) = \frac{1}{2} \sum_{I=1}^m \left(1 - \frac{\lambda}{\lambda_I}\right) (\lambda_I \hat{\Delta}_I) \xi_I^2 + A_{ijk} \xi_i \xi_j \xi_k + A_{ijkl} \xi_i \xi_j \xi_k \xi_l \quad (54.56)$$

where $\hat{\Delta}_I$, A_{ijk} and A_{ijkl} are defined as

$$\hat{\Delta}_I = 2\sigma_I \cdot L_{11}(\dot{\mathbf{u}}_c, \mathbf{u}_I) + \dot{\sigma}_c \cdot L_2(\mathbf{u}_I) \quad (54.57)$$

$$A_{ijk} = \frac{1}{2} L_1(\mathbf{u}_i) \cdot \mathbf{H} L_{11}(\mathbf{u}_j, \mathbf{u}_k) \quad (54.58)$$

$$A_{ijkl} = \frac{1}{4} \{ L_1(\mathbf{u}_k) \cdot \mathbf{H}(L_{11}(\mathbf{u}_l, \mathbf{u}_{ij})) + L_1(\mathbf{u}_l) \cdot \mathbf{H}(L_{11}(\mathbf{u}_k, \mathbf{u}_{ij})) \\ + L_1(\mathbf{u}_{ij}) \cdot \mathbf{H}(L_{11}(\mathbf{u}_k, \mathbf{u}_l)) + \frac{1}{2} L_{11}(\mathbf{u}_i, \mathbf{u}_j) \cdot \mathbf{H}(L_{11}(\mathbf{u}_k, \mathbf{u}_l)) \} \quad (54.59)$$

The reduced set of Eq. (54.51) can be obtained from Eq. (54.56) by setting $\frac{\delta P}{\delta \xi_I} = 0$ and noting the fact that the lowercase indices denote summation and A_{ijk} is symmetric between (j, k) and A_{ijkl} is symmetric between (i, j) and (k, l). Finally, one obtains the expression for b_{Ijkl} :

$$b_{Ijkl} = \frac{2}{\lambda_I \hat{\Delta}_I} (A_{Ijkl} + A_{jklI}) \quad (54.60)$$

54.5 Effect of Imperfection

In § 54.2 the effect of imperfection on the buckling load was considered based on the linearized buckling analysis of the imperfect structure. The effect of imperfection on the buckling load can also be assessed in a more precise way by making use of the postbuckling

coefficients. In case of an imperfect structure, the asymptotic expansion as defined by Eq. (54.51) is modified to

$$\xi_I(\lambda - \lambda_I) = a_{Ijk}\lambda_I\xi_j\xi_k + b_{Ijkl}\lambda_I\xi_j\xi_k\xi_l - \alpha_I\lambda_I\bar{\xi} - \beta_I(\lambda - \lambda_I)\bar{\xi} + \dots \quad (54.61)$$

where $\bar{\xi}$ is the imperfection amplitude and the coefficients α_I and β_I are known as the first and second imperfection form factors respectively. The expressions for α_I and β_I are obtained as

$$\alpha_I = (1/\lambda_I\hat{\Delta}_I)[\boldsymbol{\sigma}_I \cdot L_{11}(\hat{\mathbf{u}}, \mathbf{u}_c) + \boldsymbol{\sigma}_c \cdot L_{11}(\hat{\mathbf{u}}, \mathbf{u}_I)] \quad (54.62)$$

$$\begin{aligned} \beta_I = & (1/\hat{\Delta}_I)\{\boldsymbol{\sigma}_I \cdot L_{11}(\hat{\mathbf{u}}, \dot{\mathbf{u}}_c) + \dot{\boldsymbol{\sigma}}_c \cdot L_{11}(\hat{\mathbf{u}}, \mathbf{u}_I) + H[L_{11}(\dot{\mathbf{u}}_c, \mathbf{u}_I)] \cdot L_{11}(\hat{\mathbf{u}}, \mathbf{u}_c) \\ & - \alpha_I\lambda_I[\boldsymbol{\sigma}_I \cdot L_{11}(\dot{\mathbf{u}}_c, \mathbf{u}_I) + (1/2)\dot{\boldsymbol{\sigma}}_c \cdot L_{11}(\mathbf{u}_I, \mathbf{u}_I) \\ & + H[L_{11}(\dot{\mathbf{u}}_c, \mathbf{u}_I)] \cdot L_{11}(\dot{\mathbf{u}}_c, \mathbf{u}_I)]\} \end{aligned} \quad (54.63)$$

In case of a linear prebuckling state $\alpha_I = \beta_I = 1$. Eq. (54.61) is a small set of m nonlinear algebraic equation. Once the coefficients a_{Ijk} , b_{Ijkl} , α_I , and β_I are computed one can carry out imperfection sensitivity analysis by solving Eq. (54.61) with varying imperfection amplitude $\bar{\xi}$ at very little additional computational expense.

54.5.1 Finite Element Implementation

After some algebraic manipulation of Eqs.(54.48), (54.49), and (54.50) and replacing the L_1 and L_{11} operators and the continuous displacement fields \mathbf{u}_1 , \mathbf{u}_c , $\delta\mathbf{u}$ respectively, with the finite element matrices \mathbf{B}_L , \mathbf{B}_{NL} and nodal displacements \mathbf{q}_1 , \mathbf{q}_c , $\delta\mathbf{q}$ one can formulate the nonlinear buckling problem as

$$[\mathbf{K}_{tb} + (\lambda_i - \lambda_b)[\mathbf{K}_D(\mathbf{q}_b, \dot{\mathbf{q}}_b) + \mathbf{K}_G(\dot{\boldsymbol{\sigma}}_b)]]\mathbf{q}_i = 0 \quad (54.64)$$

where λ_i and q_i are the buckling loads and modes, respectively. \mathbf{K}_t , \mathbf{K}_D , and \mathbf{K}_G are tangent stiffness matrix, initial displacement matrix, and geometric stiffness matrix, respectively. Further more, In order to reach to Eq. (54.64) one can proceed in the following way:

First, a standard nonlinear analysis is performed to reach as close as possible to the critical (bifurcation buckling) point without encountering any negative diagonal term in the system stiffness matrix. Let that state be defined as the base state which occurs at $\lambda = \lambda_b$ with the corresponding displacement and stress states \mathbf{q}_b , $\boldsymbol{\sigma}_b$ and tangent stiffness matrix \mathbf{K}_{tb} . Then, a linearized buckling analysis is done at the base state, which is nothing but the solution of the eigenvalue problem [Eq. (54.64)].

Eqs. (54.52), (54.53), (54.54) are the equations for determination of the second order modes in functional notation. In terms of finite element matrices they can be written as

$$[\mathbf{K}_{tb} + \phi(\lambda_c - \lambda_b)[\dot{\mathbf{K}}_D(\mathbf{q}_b, \dot{\mathbf{q}}_b) + \dot{\mathbf{K}}_G(\dot{\boldsymbol{\sigma}}_b)]]\mathbf{q}_{ij} = \mathbf{g}_{ij} \quad (54.65)$$

where λ_c is the first (lowest) buckling load and ϕ is a user-defined scalar ($0 < \phi < 1$), which by default is set to the optimal value of 0.99. The right hand side force vector \mathbf{g}_{ij} is defined as

$$\begin{aligned} \mathbf{g}_{ij} = & \frac{1}{2} [[\mathbf{B}_L + \mathbf{B}_{NL}(\mathbf{q}_c)]^T \mathbf{H} \mathbf{B}_{NL}(\mathbf{q}_i) \mathbf{q}_j \\ & + \mathbf{B}_{NL}^T(\mathbf{q}_i) \mathbf{H} [\mathbf{B}_L + \mathbf{B}_{NL}(\mathbf{q}_c)] \mathbf{q}_j \\ & + \mathbf{B}_{NL}^T(\mathbf{q}_j) \mathbf{H} [\mathbf{B}_L + \mathbf{B}_{NL}(\mathbf{q}_c)] \mathbf{q}_i] \end{aligned} \quad (54.66)$$

where \mathbf{H} is the stress-strain matrix. The associated orthogonality constraint as defined by Eq. (54.55) can be translated to finite element context as

$$\mathbf{q}_i^T [\mathbf{K}_D(\mathbf{q}_b, \dot{\mathbf{q}}_b) + \mathbf{K}_G(\dot{\boldsymbol{\sigma}}_b)] \mathbf{q}_{ij} + \frac{1}{2} [\mathbf{H} \mathbf{B}_{NL}(\mathbf{q}_i) \mathbf{q}_i]^T [\mathbf{B}_{NL}(\mathbf{q}_i) \dot{\mathbf{q}}_b] = 0 \quad (54.67)$$

Solution of Eq. (54.65) together with the Eq. (54.67) gives the second order modes \mathbf{q}_{ij} .

A concise expression for A_{Ijkl} defined in Eq. (54.59) can be obtained as

$$A_{Ijkl} = \frac{1}{8} \int_v [\mathbf{H}\mathbf{B}_{NL}(\mathbf{q}_i)\mathbf{q}_j]^T [\mathbf{B}_{NL}(\mathbf{q}_k)\mathbf{q}_l] dv - \frac{1}{2} \mathbf{q}_{ij}^T \mathbf{g}_{kl} \quad (54.68)$$

In Eq. (54.68) the integration sign implies integration over the entire structure. The post-buckling coefficients b_{Ijkl} are then computed according to Eq. (54.60). The imperfection form factors (α_I , β_I) are evaluated at each integration point using Eqs. (54.62), (54.63) and summed up over the entire structure. In order to compute α_I , β_I one needs to compute $\dot{\mathbf{q}}_c$, $\ddot{\mathbf{q}}_c$ first. Because the base load level λ_b is quite close to the critical or bifurcation buckling load λ_c , the following approximation holds:

$$\mathbf{q}_c \approx \mathbf{q}_b, \quad \dot{\mathbf{q}}_c \approx \dot{\mathbf{q}}_b \quad (54.69)$$

Now $\ddot{\mathbf{q}}_b$ can be approximated as

$$\ddot{\mathbf{q}}_b = \frac{\dot{\mathbf{q}}_b - \dot{\mathbf{q}}_{b-1}}{\Delta\lambda} \quad (54.70)$$

where $\Delta\lambda = \lambda_b - \lambda_{b-1}$. Here λ_{b-1} is a load step preceding the final load step λ_b in the nonlinear prebuckling analysis and $\dot{\mathbf{q}}_b$ and $\dot{\mathbf{q}}_{b-1}$ are available from the first linear solution during the Newton–Raphson iteration process at each load step. By making λ_{b-1} and λ_b sufficiently close a reasonable estimation of $\ddot{\mathbf{q}}_b$ is possible.

54.5.2 Dynamic Buckling Analysis

In case of dynamic buckling problem time dependent loading is considered, where, with increasing magnitude of the load an increasing displacement of the structure results, and often at a particular load level the displacement exhibits a sharp increase with respect to the load increment. That particular load level can be identified as the dynamic buckling load of the structure. This criterion for the dynamic buckling load is known as the Budiansky–Roth criterion and will be used here. The perturbation approach used for static postbuckling problems can be extended to cover dynamic buckling problems taking the effect of inertia into account. The extension is done following the approach proposed by Budiansky [12] in a multi-mode context with the inclusion of prebuckling nonlinearity. Due to inertial forces the equilibrium equation takes the form

$$\boldsymbol{\sigma} \cdot \delta \boldsymbol{\epsilon} - \mathbf{f} \cdot \delta \mathbf{u} + M(\ddot{\mathbf{u}}) \cdot \delta \mathbf{u} = 0 \quad (54.71)$$

where $(\dot{}) = \frac{\partial}{\partial t}()$ in contrast to the static postbuckling problem where $(\dot{}) = \frac{\partial}{\partial \lambda}()$ and $M(\ddot{\mathbf{u}})$ which is linear in $\ddot{\mathbf{u}}$ represents the inertial loading. It is assumed that the reciprocal relation

$$M(\mathbf{u}) \cdot \mathbf{v} = M(\mathbf{v}) \cdot \mathbf{u} \quad (54.72)$$

holds. The dynamic loading is assumed to take the form $\mathbf{f} = \lambda F(t) \mathbf{f}_0$ where the time variation $F(t)$ is normalized so that its maximum value is unity. Now one can write the dynamic counter part of the Eq. (54.47) as

$$\begin{aligned} \mathbf{u} &= \lambda F(t) \mathbf{u}_0 + \mathbf{u}_i \xi_i(t) + \mathbf{u}_{ij} \xi_i(t) \xi_j(t) + \dots \\ \boldsymbol{\epsilon} &= \lambda F(t) \boldsymbol{\epsilon}_0 + \boldsymbol{\epsilon}_i \xi_i(t) + \boldsymbol{\epsilon}_{ij} \xi_i(t) \xi_j(t) + \dots \\ \boldsymbol{\sigma} &= \lambda F(t) \boldsymbol{\sigma}_0 + \boldsymbol{\sigma}_i \xi_i(t) + \boldsymbol{\sigma}_{ij} \xi_i(t) \xi_j(t) + \dots \end{aligned} \quad (54.73)$$

By repeating the same procedure as for the static case and neglecting the inertial forces associated with the prebuckling displacements one obtains

$$\begin{aligned} &\left(\frac{1}{\omega_I^2}\right) \ddot{\xi}_I(t) + \left[1 - \frac{\lambda F(t)}{\lambda_I}\right] \xi_I(t) + a_{Ijk} \xi_j(t) \xi_k(t) \\ &+ b_{Ijkl} \xi_j(t) \xi_k(t) \xi_l(t) = \left[\frac{\lambda F(t)}{\lambda_I}\right] \bar{\xi}_I \end{aligned} \quad (54.74)$$

where ω_I^2 is defined as

$$\omega_I^2 = \frac{\boldsymbol{\sigma}_I \cdot \boldsymbol{\epsilon}_I}{M(\mathbf{u}_I) \cdot \mathbf{u}_I} \quad (54.75)$$

If \mathbf{u}_I happens to be a natural vibration mode then ω_I is its natural circular frequency otherwise ω_I^2 has an interpretation as a Rayleigh quotient for circular frequency squared based on the buckling mode \mathbf{u}_I . In DIANA, Eq. (54.74) is solved with a standard Runge–Kutta scheme of the 4th order.

In order to account for prebuckling nonlinearity Eq. (54.75) is written as

$$\omega_I^2 = \frac{-\lambda_I \hat{\Delta}_I}{M(\mathbf{u}_I) \cdot \mathbf{u}_I} \quad (54.76)$$

It can be noticed that, the quantity $\boldsymbol{\sigma}_I \cdot \boldsymbol{\epsilon}_I$ in Eq. (54.75) is replaced by $-\lambda_c \hat{\Delta}_I$ in Eq. (54.76) where $\hat{\Delta}_I$ is defined by Eq. (54.57). Finally, Eq. (54.74) is modified to

$$\begin{aligned} & \left(\frac{1}{\omega_I^2} \right) \ddot{\xi}_I(t) + \left[1 - \frac{\lambda F(t)}{\lambda_I} \right] \xi_I(t) + a_{Ijk} \xi_j(t) \xi_k(t) + b_{Ijkl} \xi_j(t) \xi_k(t) \xi_l(t) \\ & = \alpha_I \bar{\xi}_I - \beta_I \left[1 - \frac{\lambda F(t)}{\lambda_I} \right] \bar{\xi}_I \end{aligned} \quad (54.77)$$

where α_I and β_I are imperfection form factors which are evaluated using Eqs. (54.62) and (54.63). In case of linear prebuckling state both α_I and β_I are unity and Eq. (54.77) becomes identical to Eq. (54.74). Further, in case of nonlinear prebuckling state the buckling analysis is carried out at a state close to bifurcation buckling load in contrast to the undeformed state considered in case of linear prebuckling state. Now with the background established so far a step by step procedure for the perturbation type dynamic buckling analysis can be set up as following

1. Computation of the prebuckling state \mathbf{u}_0 .
2. Computation of the buckling loads λ_I and the corresponding buckling modes \mathbf{u}_I .
3. Computation of the second order modes \mathbf{u}_{ij} .
4. Computation of the a_{ijk} and b_{ijkl} coefficients.
5. Computation of the mode amplitude $\xi_I(t)$ corresponding to the applied dynamic load by solving Eq. (54.74) and identification of the dynamic buckling load level $\lambda = \lambda_d$ at which $\xi_I(t)$ shows a sharp rise or no solution is found.
6. Recovering the displacement, stress and strain by substituting the already computed terms in Eq. (54.73).

Chapter 55

Potential Flow Analysis

55.1 Basic Equations

55.1.1 Convection–Diffusion Equation

Potential flow generally obeys the following convection–diffusion equation.

$$\operatorname{div} \mathbf{q} + \beta \nabla \phi + c \dot{\phi} = q_V \quad (55.1)$$

$$\mathbf{q} = -\mathbf{k} \nabla \phi \quad (55.2)$$

with ϕ the potential, \mathbf{q} the specific flux vector, \mathbf{k} the diffusivity tensor, β the field convection vector, c the capacitance, and q_V the external flux per volume.

55.1.2 Boundary Conditions

Three conditions can be applied to boundaries or parts thereof: *essential*, *natural*, or *mixed*.

Essential (Dirichlet). The essential boundary condition is a prescribed boundary potential ϕ_B :

$$\phi = \phi_B \quad (55.3)$$

Natural (Neumann). The natural boundary condition is a prescribed boundary flux q_B :

$$\mathbf{q} \mathbf{n} = -q_B \quad (55.4)$$

with \mathbf{n} the vector pointing outwards normal to the boundary.

Mixed (Neumann/Robin). The mixed boundary condition is a boundary flux which depends on a free boundary potential and a prescribed environment potential ϕ_E :

$$\mathbf{q} \mathbf{n} = K(\phi - \phi_E) \quad (55.5)$$

with K the conduction coefficient, which is typically used to model for example heat convection or radiation at boundaries.

If no boundary condition is specified then the natural condition of *insulation* is valid:

$$\mathbf{q} \mathbf{n} = 0 \quad (55.6)$$

55.1.3 Finite Element Formulation

The finite element formulation can be derived according to the *Galerkin* procedure. Eq. (55.1) is multiplied by a test function ν and integrated over the considered volume V .

$$\int_V \nu \left(\operatorname{div} \mathbf{q} + \beta \nabla \phi + c \dot{\phi} \right) dV = \int_V \nu q_V dV \quad (55.7)$$

According to the theorem of *Gauss* we can write

$$\int_V \text{div} \mathbf{q} \nu \, dV + \int_V \mathbf{q} \nabla \nu \, dV = \int_B \mathbf{q} \mathbf{n} \nu \, dB \quad (55.8)$$

With Eq. (55.8), Eq. (55.4), and Eq. (55.5) we can rewrite Eq. (55.7) as

$$\begin{aligned} \int_V -\mathbf{q} \nabla \nu \, dV + \int_V \beta \nabla \phi \nu \, dV + \int_V c \dot{\phi} \nu \, dV + \int_B K \phi \nu \, dB \\ = \int_V q_V \nu \, dV + \int_B (K \phi_E + q_B) \nu \, dB \end{aligned} \quad (55.9)$$

We now assume the potential field and the test function inside a finite element as a linear function of the values in the element nodes:

$$\phi(\mathbf{x}) = \mathbf{N} \phi, \quad \nabla \phi(\mathbf{x}) = \mathbf{B} \phi \quad (55.10)$$

$$\nu(\mathbf{x}) = \mathbf{N} \nu, \quad \nabla \nu(\mathbf{x}) = \mathbf{B} \nu \quad (55.11)$$

The vector ϕ contains the node potentials, the vector ν contains the node test function values, and \mathbf{N} and \mathbf{B} are interpolation matrices. Substitution of Eq. (55.10), Eq. (55.11), and Eq. (55.2) in Eq. (55.9) finally results in the following finite element equation:

$$\boxed{\mathbf{K} \phi + \mathbf{C} \dot{\phi} = \mathbf{Q}} \quad (55.12)$$

with

$$\mathbf{K} = \int_V \mathbf{B}^T \mathbf{k} \mathbf{B} \, dV + \int_V \mathbf{N}^T \beta \mathbf{B} \, dV + \int_B \mathbf{N}^T K \mathbf{N} \, dB \quad (55.13)$$

$$\mathbf{C} = \int_V \mathbf{N}^T c \mathbf{N} \, dV \quad (55.14)$$

$$\mathbf{Q} = \int_V \mathbf{N}^T q_V \, dV + \int_B \mathbf{N}^T q_B \, dB + \int_B \mathbf{N}^T K \phi_E \, dB \quad (55.15)$$

where \mathbf{K} is the conduction/convection matrix, \mathbf{C} the capacity matrix, and \mathbf{Q} the nodal discharge or external flux vector.

55.1.3.1 Time Integration

DIANA solves the potential flow finite element equations Eq. (55.12) stepwise with direct time integration, using a generalized trapezoidal rule. This means that for each step Δt , the equation is solved at time $t + \alpha \Delta t$. Various methods are available depending on the value of α .

$\alpha = 0$ for *Euler Forward* integration (explicit). This method has first order accuracy and a limited step size for stable solution. With λ_{\max} for the largest eigenvalue, the condition is

$$\Delta t < \frac{2}{(1 - 2\alpha)\lambda_{\max}} \quad \text{for} \quad \alpha < \frac{1}{2} \quad (55.16)$$

$\alpha = \frac{1}{2}$ for *Crank-Nicolson* integration. This method has second order accuracy and no numerical damping.

$\alpha = \frac{2}{3}$ for *Galerkin* integration, a method with first order accuracy.

$\alpha = 1$ for *Euler Backward* integration. This method has first order accuracy without oscillations. It has numerical damping.

Combination of the finite element equations and direct time integration results in an equation of the following form.

$$\mathbf{K}^* \phi^* = \mathbf{Q}^* \quad (55.17)$$

With \mathbf{K}^* the effective matrix, ϕ^* the vector of nodal potentials, and \mathbf{Q}^* the effective vector of nodal discharges. In DIANA an implicit and an explicit integration scheme have been implemented.

Implicit time integration. The implicit time integration is characterized by

$$\boxed{\alpha > 0 \quad \text{and} \quad \phi^* = \phi^{t+\alpha \Delta t}} \quad (55.18)$$

With the time derivative:

$$\dot{\phi}^{t+\alpha \Delta t} = \frac{\phi^{t+\alpha \Delta t} - \phi^t}{\alpha \Delta t} \quad (55.19)$$

The effective matrix:

$$\mathbf{K}^* = \mathbf{C}^{t+\alpha \Delta t} + \alpha \Delta t \mathbf{K}^{t+\alpha \Delta t} \quad (55.20)$$

The effective nodal discharge vector:

$$\mathbf{Q}^* = \alpha \Delta t \mathbf{Q}^{t+\alpha \Delta t} + \mathbf{C}^{t+\alpha \Delta t} \phi^t \quad (55.21)$$

In case of nonlinearity the iteration scheme is:

$$\mathbf{K}_{(i)}^* \phi_{(i+1)}^* = \mathbf{Q}_{(i)}^* \quad (55.22)$$

The extrapolation for $\alpha < 1$ is:

$$\phi^{t+\Delta t} = \phi^t + \dot{\phi}^{t+\alpha \Delta t} \Delta t \quad (55.23)$$

Explicit time integration. The explicit time integration is characterized by

$$\boxed{\alpha = 0 \quad \text{and} \quad \phi^* = \phi^{t+\Delta t}} \quad (55.24)$$

With the time derivative:

$$\dot{\phi}^t = \frac{\phi^{t+\Delta t} - \phi^t}{\Delta t} \quad (55.25)$$

The effective matrix is:

$$\mathbf{K}^* = \mathbf{C}^t \quad (55.26)$$

The effective nodal discharge vector is:

$$\mathbf{Q}^* = \Delta t \mathbf{Q}^t - \Delta t \mathbf{K}^t \phi^t + \mathbf{C}^t \phi^t \quad (55.27)$$

55.1.3.2 Nonlinearity

If the value of conductivity, capacity or boundary condition is a function of the potential, the set of equations must be solved iteratively.

Iteration method. DIANA uses an incremental iterative method to solve the set of nonlinear equations. This means that in iteration $(i+1)$ the incremental potential $\Delta\phi$ is calculated from

$$\mathbf{K}_{\text{incr}}^* \Delta\phi = \mathbf{Q}_{(i)}^* - \mathbf{K}_{\text{sec.}(i)}^* \phi_{(i)}^* \quad (55.28)$$

$$\phi_{(i+1)}^* = \phi_{(i)}^* + \Delta\phi \quad (55.29)$$

The effective secant conduction matrix is updated in every iteration. The default *Regular Newton-Raphson* method updates the effective incremental conductivity in every iteration:

$$\mathbf{K}_{\text{incr}} = \mathbf{K}_{\text{sec.}(i)} \quad (55.30)$$

The *Modified Newton-Raphson* method updates the effective incremental conductivity only at the start of a step:

$$\mathbf{K}_{\text{incr}} = \mathbf{K}_{\text{sec.}(1)} \quad (55.31)$$

Accuracy. During the iteration process, DIANA uses the norm of the incremental vector of potentials to determine if accuracy is reached:

$$|\Delta\phi| \leq \epsilon \times |\phi_{(1)}^*| \quad (55.32)$$

Chapter 56

Soil–Pore Fluid Analysis

This chapter summarizes the basic equations and assumptions used to derive the implemented finite element equations for pore fluid interaction with porous media like soil or rock, also known as ‘Mixture analysis’. For a more comprehensive treatment of the theoretical and analytical aspects of consolidation see the well known textbooks and articles, for instance Terzaghi [84], or Biot [8]. For general aspects of Finite Element Analysis of coupled flow–stress analysis see for instance the textbook by Zienkiewicz [94, Ch. 11]. The Finite Element Analysis of soil–pore fluid interaction with partial saturation has also been addressed in various articles, for instance by Li & Zienkiewicz [56].

56.1 Basic Equations and Assumptions

The equations and assumptions in this chapter relate the stresses and pore pressures to deformations and fluid flow, using material parameters for elastic compressibility, drained elastoplasticity, density, porosity and permeability.

Notations. For general notation conventions see the *Glossary of Symbols* on page xxix. Following notations are specific for this chapter. The operator ∇ denotes the gradient of a function. For example

$$\nabla p = \left\{ \frac{\partial p}{\partial x}, \frac{\partial p}{\partial y}, \frac{\partial p}{\partial z} \right\}$$

The gradient of a scalar is a vector, the gradient of a vector is a matrix. The divergence of a vector or matrix is denoted as

$$\text{div}(\mathbf{u}) = \frac{\partial u_x}{\partial x} + \frac{\partial u_y}{\partial y} + \frac{\partial u_z}{\partial z}$$

The divergence of a vector is a scalar, the divergence of a matrix is a vector.

Sign conventions. Stresses and strains are assumed to be positive for tension. Pressures are assumed to be positive for compression.

56.1.1 Porosity, Saturation and Apparent Density

Assuming that all pores are filled with fluid or air, the fluid fraction depends on porosity and saturation. The apparent density can therefore be formulated as follows.

$$\rho = \rho_{\text{dry}} + S n \rho_f \quad (56.1)$$

where ρ is the total density of the mixture, n the porosity (equal to the fluid fraction), S the pore pressure dependent degree of saturation ($0 \leq S \leq 1$), ρ_f the density of the fluid, and ρ_{dry} the density of the dry porous solid. DIANA uses this equation to determine the deformation and saturation dependent density during the analysis, assuming a deformation dependent porosity.

$$n = 1 + \frac{n_{\text{init}} - 1}{J} \quad (56.2)$$

with J the determinant of the deformation gradient:

$$J = \det \nabla \mathbf{x} \quad (56.3)$$

For geometric linear analysis J reduces to $J = 1 + \varepsilon_1 + \varepsilon_2 + \varepsilon_3$.

56.1.2 Undrained Compressibility

A compression modulus relates hydrostatic stress to elastic volumetric strain. The elastic compressible behaviour of the undrained mixture is determined by the following compression moduli.

K_D Compression modulus of the (drained) porous soil skeleton, related to the effective stress. This compression modulus is accompanied by the shear modulus G for determination of drained elastic deformation.

K_f Intrinsic compression modulus of the fluid.

K_s Intrinsic compression modulus of the non-porous solid.

K_p Compression modulus of the solid fraction, related to the inter-granular stress. Assuming average stress distribution in the solid particles, this modulus is directly related to K_s :

$$K_p = (1 - n)K_s \quad (56.4)$$

56.1.3 Stress Separation and Pore Pressure Potential

The total stress is separated into inter-granular stress and pore pressure. The inter-granular stress is closely related to Terzaghi's effective stress. The effective stress governs the strength and deformation of the soil skeleton.

$$\begin{aligned} \bar{\boldsymbol{\sigma}} &= \boldsymbol{\sigma} + S \mathbf{I} p \\ \boldsymbol{\sigma}' &= \bar{\boldsymbol{\sigma}} - \gamma S \mathbf{I} p \end{aligned} \quad (56.5)$$

where $\boldsymbol{\sigma}$ is the total stress, $\bar{\boldsymbol{\sigma}}$ the inter-granular stress, $\boldsymbol{\sigma}'$ the effective stress, p the pore pressure, and γ a function of compressibility. Widely used is $\gamma = K_D/K_s$, with for soils $\gamma \rightarrow 0$. Furthermore \mathbf{I} is an identity (unity) matrix and S the degree of saturation. Substitution gives:

$$\boldsymbol{\sigma} = \boldsymbol{\sigma}' - (1 - \gamma)S \mathbf{I} p \quad (56.6)$$

Which can also be written as:

$$\boldsymbol{\sigma} = \boldsymbol{\sigma}' - \alpha S \mathbf{I} p \quad (56.7)$$

where α is the first Biot parameter:

$$\alpha = 1 - \frac{K_D}{K_s} = 1 - \gamma \quad (56.8)$$

For soils $\alpha \rightarrow 1$.

Note, that conform the sign convention in DIANA where compressive stresses are negative, the total stresses are the effective stresses minus the pore pressures.

DIANA determines the pore pressure p from the pore pressure potential ϕ , using the initial position vector $\mathbf{x}_{\text{init}} - \mathbf{x}_{\text{ref}}$ and the displacement \mathbf{u} :

$$p = \phi + \rho_f \mathbf{g} (\mathbf{x} - \mathbf{x}_{\text{ref}}) \quad (56.9)$$

with

$$\mathbf{x} = \mathbf{x}_{\text{init}} + \mathbf{u} \quad (56.10)$$

56.1.4 Momentum Conservation

The basic equilibrium law in stress analysis is momentum conservation. If a gravity field is present, DIANA uses the following momentum balance:

$$\operatorname{div}(\boldsymbol{\sigma}) + \rho \mathbf{g} + \rho \ddot{\mathbf{u}} = \mathbf{0} \quad (56.11)$$

where \mathbf{g} is the gravity acceleration vector and $\ddot{\mathbf{u}}$ the acceleration of the solid. Convective terms and the relative accelerations between solid and fluid are neglected. Using Eq. (56.5) this equation reads

$$\operatorname{div}(\boldsymbol{\sigma}') - S(1 - \gamma) \nabla \mathbf{I} p + \rho \mathbf{g} + \rho \ddot{\mathbf{u}} = \mathbf{0} \quad (56.12)$$

56.1.5 Mass Conservation

The influence of mass conservation depends on the degree of saturation of the porous medium: *full* or *partial*.

Fully saturated. Mass conservation in a saturated context means that both fluid flow and undrained compression or expansion will cause volumetric changes. The homogeneous form of the so-called *storage equation* is derived from the general mass conservation law

$$\alpha \operatorname{div}(\dot{\mathbf{u}}) + \operatorname{div}(\mathbf{q}) + \frac{\dot{p}}{Q} = 0 \quad (56.13)$$

where $\dot{\mathbf{u}}$ is the velocity vector and \mathbf{q} the fluid flux vector. Variables α and Q are the *Biot material parameters*:

$$\frac{1}{Q} = \frac{n}{K_f} + (1 - n) \left(\frac{1}{K_s} - \frac{\gamma}{K_p} \right) \quad ; \quad \alpha = 1 - \gamma \quad (56.14)$$

Assuming Eq. (56.4) this equation reduces to

$$\frac{1}{Q} = \frac{n}{K_f} + \frac{\alpha - n}{K_s} \quad (56.15)$$

For soils $\alpha \rightarrow 1$ and $Q \rightarrow K_f/n$. The value of $1/Q$ is often called the *hydraulic capacity*.

Partially saturated. For partially saturated areas, DIANA uses the following extension to the storage equation, assuming that air pressures are zero.

$$S \alpha \operatorname{div}(\dot{\mathbf{u}}) + \operatorname{div}(\mathbf{q}) + \frac{\dot{p}}{\tilde{Q}} = 0 \quad (56.16)$$

DIANA uses the pore pressure dependent degree of saturation S to determine the modified Biot parameter \tilde{Q} , with $\lim_{S \rightarrow 1} \tilde{Q} = Q$

$$\frac{1}{\tilde{Q}} = C_s + \frac{nS}{K_f} + \frac{(\alpha - n)S \left(S + \frac{C_s p}{n} \right)}{K_s} \quad (56.17)$$

where the so-called moisture capacity C_s is derived from the relation between degree of saturation and pore pressure.

$$C_s = n \frac{\partial S}{\partial p} \quad (56.18)$$

56.1.6 Darcy Flow

Darcy's law relates the fluid flow in porous media to gradients in pore pressure potentials by permeability.

$$\mathbf{q} = -\mathbf{k}' \nabla \phi \quad (56.19)$$

with

$$\mathbf{k}' = \frac{\mathbf{k}}{\mu} \quad \text{and} \quad \phi = p - \rho_f \mathbf{g} (\mathbf{x} - \mathbf{x}_{\text{ref}}) \quad (56.20)$$

where \mathbf{q} is the specific discharge, \mathbf{k} the permeability, \mathbf{k}' the modified permeability, $\mu = \nu \times \rho$ the dynamic viscosity, ϕ the pore pressure potential, \mathbf{g} the gravity acceleration vector, and $\mathbf{x} - \mathbf{x}_{\text{ref}}$ the current position vector.

Note that DIANA uses the permeability \mathbf{k}' , which relates the specific discharge to the pore pressure potential. However, in geotechnical practice also the permeability $\mathbf{k}'_{\text{head}}$ is in use, which relates the specific discharge to gradients in the pore pressure potential head.

$$\mathbf{k}' = \frac{\mathbf{k}'_{\text{head}}}{\rho_f |\mathbf{g}|} \quad (56.21)$$

with

$$\mathbf{q} = -\mathbf{k}'_{\text{head}} \nabla \phi_{\text{head}} \quad \text{and} \quad \phi = \rho_f |\mathbf{g}| \phi_{\text{head}} \quad (56.22)$$

Dependency of the permeability. In general, the modified permeability depends on porosity and saturation. DIANA supports a porosity dependent relative permeability k_{rel_n} and a saturation dependent relative permeability k_{rel_s} using

$$k' = k_{\text{rel}_n} \times k_{\text{rel}_s} \times k'_{\text{init}} \quad (56.23)$$

DIANA can derive the relative permeabilities from user-specified diagrams. In case of k_{rel_n} DIANA also supports the relation given by Kozeny & Karman in [54]:

$$k_{\text{rel}_n} = \frac{n \left(\frac{n}{1-n} \right)^2}{n_{\text{init}} \left(\frac{n_{\text{init}}}{1-n_{\text{init}}} \right)^2} \quad (56.24)$$

Dynamic behaviour. In case of dynamic behaviour, DIANA uses the following extended equation for the Darcy flow, assuming isothermal behaviour, and omitting convective terms and relative accelerations between fluid and solid.

$$\mathbf{q} = -\mathbf{k}' (\nabla \phi - \rho_f \ddot{\mathbf{u}}) \quad (56.25)$$

56.1.7 Elastoplastic Stiffness

Elastoplasticity relates stress $\boldsymbol{\sigma}$ to strain $\boldsymbol{\varepsilon}$, using parameters which are typically shear modulus G , compression modulus K_D , cohesion c , friction angle ϕ , and dilatancy angle ψ .

$$\dot{\boldsymbol{\sigma}}' = \mathbf{D}(\boldsymbol{\sigma}') \cdot \dot{\boldsymbol{\varepsilon}}(\nabla \dot{\mathbf{u}}) \quad (56.26)$$

where \mathbf{D} is the drained elastoplastic stiffness of the soil skeleton. See Volume *Material Library* for the effective stress models that are available in DIANA.

56.2 Finite Element Equations

The equations derived in this section, relate nodal displacements, pore pressure potentials and their velocities to nodal forces and fluid discharges.

56.2.1 Space Discretization

The basic finite element assumption is the interpolation of coordinates, displacements and pore pressure potentials for each element from the values in nodes

$$u_i = \mathbf{N}_u \mathbf{u}_i \quad ; \quad \phi = \mathbf{N}_\phi \phi \quad (56.27)$$

where ϕ is the vector of nodal values for the pore pressure potential ϕ , \mathbf{u}_i the vector of the nodal values for the displacement component u_i , \mathbf{N}_ϕ the pore pressure potential interpolation vector, and \mathbf{N}_u the displacement interpolation vector. Based on Eq. (56.4) to Eq. (56.26) a finite element space discretization is derived via the standard Galerkin procedure.

$$\mathbf{M}_{\phi u} \ddot{\mathbf{u}} + \mathbf{C}_{\phi u} \dot{\mathbf{u}} + \mathbf{C}_{\phi\phi} \dot{\phi} + \mathbf{K}_{\phi\phi} \phi = \mathbf{f}_{b.\phi\phi} \quad (56.28)$$

$$\mathbf{M}_{uu} \ddot{\mathbf{u}} + \mathbf{K}_{uu} \mathbf{u} - \mathbf{K}_{u\phi} \phi = \mathbf{f}_{b.uu} + \mathbf{f}_{g.uu} - \mathbf{f}_{u\phi} \quad (56.29)$$

where \mathbf{u} is the nodal displacement vector, ϕ the nodal pore pressure potential vector, \mathbf{M}_{uu} the mass matrix, $\mathbf{M}_{\phi u}$ the coupling mass matrix for flow by acceleration of displacement, \mathbf{K}_{uu} the stiffness matrix, $\mathbf{K}_{\phi\phi}$ the permeability matrix, $\mathbf{C}_{\phi\phi}$ the hydraulic capacity matrix, $\mathbf{f}_{b.\phi\phi}$ the boundary flux vector, $\mathbf{f}_{g.uu}$ the dead weight force vector, $\mathbf{f}_{u\phi}$ the buoyancy force vector, $\mathbf{f}_{b.uu}$ the boundary force vector, $\mathbf{K}_{u\phi}$ the coupling matrix for stress by pore pressure potential, and $\mathbf{C}_{\phi u}$ the coupling matrix for flow by displacements.

All these matrices and vectors are determined as follows:

$$\mathbf{K}_{uu} = \int_V \mathbf{B}_u^T \mathbf{D} \mathbf{B}_u \, dV \quad (56.30)$$

$$\mathbf{C}_{\phi\phi} = \int_V \frac{1}{Q} \mathbf{N}_\phi^T \mathbf{N}_\phi \, dV \quad (56.31)$$

$$\mathbf{K}_{\phi\phi} = \int_V \mathbf{B}_\phi^T \mathbf{k}' \mathbf{B}_\phi \, dV \quad (56.32)$$

$$\mathbf{K}_{u\phi} = \int_V \alpha \mathbf{B}_u^T \mathbf{m} \mathbf{N}_\phi \, dV \quad ; \quad \mathbf{m}^T \equiv \{1 \ 1 \ 1 \ 0 \ 0 \ 0\} \quad (56.33)$$

$$\mathbf{C}_{\phi u} = \mathbf{K}_{u\phi}^T \quad (56.34)$$

$$\mathbf{M}_{uu} = \int_V \mathbf{N}_u^T \rho \mathbf{N}_u \, dV \quad (56.35)$$

$$\mathbf{M}_{\phi u} = \int_V \mathbf{B}_\phi^T \rho_f \mathbf{k}' \mathbf{N}_u \, dV \quad (56.36)$$

$$\mathbf{f}_{g.uu} = \int_V \rho \mathbf{g} \mathbf{N}_u \, dV \quad (56.37)$$

$$\mathbf{f}_{u\phi} = \int_V \rho_f \mathbf{g} \mathbf{x} \mathbf{B}_u^T \mathbf{m} \, dV \quad (56.38)$$

where \mathbf{B}_u is the strain vector interpolation matrix, \mathbf{D} the material stiffness matrix, \mathbf{N}_ϕ the pore pressure potential interpolation vector, \mathbf{N}_u the displacement interpolation matrix, and \mathbf{B}_ϕ the pore pressure potential gradient interpolation matrix $\nabla \mathbf{N}_\phi$.

56.2.2 Time Discretization

The space discretization is evaluated at or between several time steps, mostly using an Euler Backward time integration for consolidation analysis, and a finite difference approximation for the time derivatives for dynamic analysis: Newmark–Hilber–Hughes–Taylor or Wilson- θ [§ 48.4 p. 592].

This section illustrates how the Euler Backward approximation of velocities is incorporated in a nonlinear iteration scheme. Euler Backward assumes constant velocities within each time step

$$\dot{\mathbf{u}} = \frac{\Delta \mathbf{u}}{\Delta t} = \frac{^{t+\Delta t} \mathbf{u} - ^t \mathbf{u}}{\Delta t} \quad (56.39)$$

$$\dot{\phi} = \frac{\Delta \phi}{\Delta t} = \frac{^{t+\Delta t} \phi - ^t \phi}{\Delta t} \quad (56.40)$$

The preceding superscript t or $^{t+\Delta t}$ denotes the time of evaluation. Combination with Eq. (56.28), Eq. (56.29) and a nonlinear iteration scheme leads to the following incremental iterative formulation (subscript (i) denoting the iteration number, starting with 1).

$$\begin{aligned} {}^{t+\Delta t} \left(\frac{1}{\Delta t} \mathbf{C} + \mathbf{K} \right)_{(i-1)} \times \begin{Bmatrix} \Delta \mathbf{u}_{(i)} - \Delta \mathbf{u}_{(i-1)} \\ \Delta \phi_{(i)} - \Delta \phi_{(i-1)} \end{Bmatrix} = \\ {}^{t+\Delta t} \mathbf{f}_{\text{extern}} - {}^{t+\Delta t} \mathbf{f}_{\text{intern},(i-1)} \end{aligned} \quad (56.41)$$

With

$$\mathbf{C} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{C}_{\phi u} & \mathbf{C}_{\phi \phi} \end{bmatrix} \quad (56.42)$$

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_{uu} & -\mathbf{K}_{u\phi} \\ \mathbf{0} & \mathbf{K}_{\phi\phi} \end{bmatrix} \quad (56.43)$$

$$\mathbf{f}_{\text{extern}} = \begin{Bmatrix} \mathbf{f}_{b,uu} + \mathbf{f}_{g,uu} - \mathbf{f}_{u\phi} \\ \mathbf{f}_{b,\phi\phi} \end{Bmatrix} \quad (56.44)$$

$$\mathbf{f}_{\text{intern}} = \begin{Bmatrix} \int_V \mathbf{B}_u^T \boldsymbol{\sigma}' dV - \mathbf{K}_{u\phi} \phi \\ \mathbf{K}_{\phi\phi} \phi + \mathbf{C}_{\phi\phi} \frac{\Delta \phi}{\Delta t} + \mathbf{C}_{\phi u} \frac{\Delta \mathbf{u}}{\Delta t} \end{Bmatrix} \quad (56.45)$$

56.2.2.1 Consolidation Analysis

In general, accuracy is assured if the time step is small enough. However, if the time step is chosen too small, spatial instabilities will occur at the almost undrained and incompressible situation immediately after loading. Due to numerical diffusion, these instabilities will disappear during the integration. The spatial instabilities are suppressed either by a larger time step or by a small element size. Vermeer & Verruyt [88] derived a criterion for one-dimensional consolidation with assumption of linearly interpolated elements and equally spaced time steps. For Euler Backward time integration this equation reads

$$\Delta t_{\min} \geq \frac{l^2}{6 c_c} \quad (56.46)$$

with c_c the so-called coefficient of consolidation, and l the element size.

$$c_c = \frac{k'}{\frac{1}{K_D} + \frac{1}{Q}} \quad (56.47)$$

56.2.2.2 Dynamic Analysis

The derivation of the equations for dynamic behaviour with different time integrations is basically analogous. DIANA however does not include the $\mathbf{M}_{\phi u}$ contribution in the total stiffness matrix, but only in the internal force vector $\mathbf{f}_{\text{intern}}$. This means that always iteration is required in case of dynamic analysis with Mixture elements.

56.2.3 Undrained Behaviour

DIANA uses a penalty formulation

to model completely undrained behaviour [Vol. *Material Library*]. You can use the penalty method as a substitute for Mixture analysis to model the behaviour of layers with practically no dissipation of excess pore pressure during the period of analysis. You can also use the penalty method in combination with a Mixture analysis, during the almost undrained and incompressible situation immediately after loading, to avoid spatial instabilities.

The penalty method derives the excess pore pressure p_e directly from the volumetric strain ε_v .

$$p_e = -K_f \varepsilon_v \quad (56.48)$$

By default, DIANA sets the value of the bulk modulus of the pore fluid K_f equal to the drained bulk modulus K_D times a large penalty factor: $K_f \gg K_D$. In practice, this implies that the fluid fraction will prevent almost all volumetric deformations and that only the soil fraction will resist against deviatoric deformations.

56.2.4 Hydraulic Pore Pressure Load

As an alternative to coupled Mixture analysis, you can also apply a hydraulic pore pressure potential load ϕ_h on the soil skeleton, which you may combine with undrained behaviour via the penalty method. DIANA uses:

$$\mathbf{f}_{h.u\phi} - \mathbf{f}_{u\phi} = \mathbf{K}_{u\phi} \phi_h - \int_V \rho_f \mathbf{g} \mathbf{x} \mathbf{B}_u^T \mathbf{m} \, dV \quad (56.49)$$

By applying reduced density ρ_{red} instead, you can simulate the buoyancy load:

$$\rho_{red} = \rho - \rho_f \quad (56.50)$$

$$- \int_V \rho_f \mathbf{g} \mathbf{N}_u \, dV \equiv \mathbf{f}_{h.u\phi} - \mathbf{f}_{u\phi} \quad (56.51)$$

56.2.4.1 Determination of Hydraulic Pressure Load

Besides a Mixture analysis, DIANA can also determine a nonuniform distribution of the hydraulic pore pressure potential via a separate potential flow analysis [Part VI]. The equations for determination of a hydraulic pore pressure load can simply be derived from the general equations by removing the deformation dependent part from the mass conservation law:

$$\text{div}(\mathbf{q}) + \frac{\dot{p}_h}{Q} = 0 \quad (56.52)$$

$$\mathbf{C}_{\phi\phi} \dot{\phi}_h + \mathbf{K}_{\phi\phi} \phi_h = \mathbf{f}_{b.\phi\phi} \quad (56.53)$$

Chapter 57

Fracture Mechanics Analysis

DIANA offers some special ‘crack tip’ elements to be applied for Linear Elastic Fracture Mechanics Analysis (LEFM) [Vol. *Element Library*]. In a regular linear static analysis these elements can produce output of the LEFM parameters: *energy release rate* and *stress intensity factor*. The calculation of these results is done via a virtual crack extension method according to Parks [70].

57.1 Stress Intensity Factor

In Linear Elastic Fracture Mechanics the problem of calculating the stresses around a crack in a structure is analysed by using linear elastic constitutive relations. The solution to the linear elastic problem is singular at the crack tip. In this case, the stresses and strains around a crack tip tends to infinity. The magnitude of this singularity can be described with the *stress intensity factor*. There are three basic stress intensity factors [Fig. 57.1]: K_I for opening mode loading, K_{II} for sliding mode loading and K_{III} for tearing

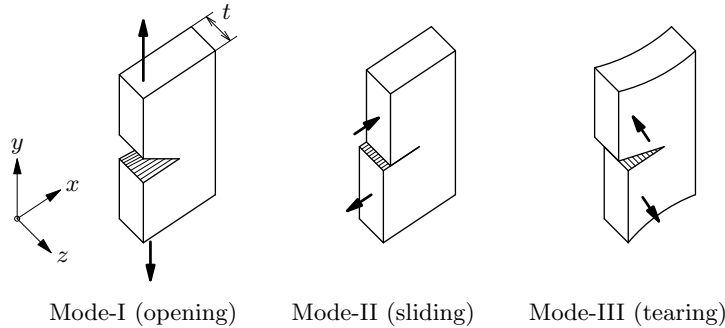


Figure 57.1: Loading modes

mode loading. In case of an opening mode loading, the stresses just before the crack tip (in the crack plane) are:

$$\sigma_{yy} = \frac{K_I}{\sqrt{2\pi r}} \quad (57.1)$$

where r is the distance from the crack tip. The stress intensity factor for an infinite plate with a through the plate crack of size $2a$ and a far away stress field $\sigma_{yy} = \sigma$ is:

$$K_I = \sigma\sqrt{\pi a} \quad (57.2)$$

In Eq. (57.2) we see that K_I depends on the crack geometry and the loading. There are many textbooks about fracture mechanics and handbooks with stress intensity factors for specific structures and crack geometries.

57.1.1 Mesh Adaptation

As stated above the stresses and strains at the crack tip are singular. In the vicinity of the crack tip, this singularity can be described with:

$$\sigma = \frac{K_I}{\sqrt{2\pi r}} f(\theta) \quad (57.3)$$

where K_I is the stress intensity factor for Mode-I opening. Modelling this singularity in a finite element mesh with the usual elements, would require a considerably refined mesh near the crack tip. It can be shown however, for instance for two-dimensional quadratic elements, that when the midside nodes of the elements at the crack tip are at a quarter distance from that crack tip, the element can describe this $1/\sqrt{r}$ singularity. The same holds for three-dimensional elements. For more information see for instance Zienkiewicz [93, pp. 189–191], Barsoum [3], or Henshell & Shaw [41]. With midside nodes on quarterpoint position the singularity at the crack tip is best described.

57.2 Energy Release Rate

The stress intensity factor can be calculated from the *energy release rate*. This energy release rate G , equals the release of the elastic energy W_{in} minus the external potential P_{ex} when the cracked area A_{cr} increases with one unit area, with constant displacements at the outer boundary.

$$G = -\frac{\partial W_{\text{in}}}{\partial A_{\text{cr}}} + \frac{\partial P_{\text{ex}}}{\partial A_{\text{cr}}} \quad (57.4)$$

See for instance Hellen [39]. For convenience, the contribution of the external potential is ignored in the sequel of this chapter. However, the contribution of thermal and chemical expansion, initial stresses and crack face loads have been included in DIANA.

57.2.1 Two-dimensional Analysis

In case of a two-dimensional analysis, the increase in cracked area can be related to the increase in crack length a . If W_{in} is the elastic energy and t is the thickness of a plate, then the energy release rate is defined as

$$G = -\frac{1}{t} \frac{\partial W_{\text{in}}}{\partial a} \quad (57.5)$$

at constant displacements. In finite element analysis the elastic energy W can be calculated from pre- and postmultiplication of the stiffness matrix \mathbf{K} by the displacement vector \mathbf{u} .

$$W_{\text{in}} = \frac{1}{2} \mathbf{u}^T \mathbf{K} \mathbf{u} \quad (57.6)$$

Since the displacements are kept constant, the derivative in the right hand side of Eq. (57.5) can be calculated from:

$$\frac{\partial W_{\text{in}}}{\partial a} = \frac{1}{2} \mathbf{u}^T \frac{\partial \mathbf{K}}{\partial a} \mathbf{u} \quad (57.7)$$

The derivative $\partial \mathbf{K} / \partial a$ is determined analytically with reference to the known shape functions, see Van Den Boogaard [10]. Since the system stiffness matrix is only affected by the elements that are at the crack tip, only these elements are to be re-analysed. The relation between the stress intensity factor and the energy release rate is known for plane stress and plane strain conditions. In the case of Mode-I loading (opening) this relation is:

$$G_I = \frac{K_I^2}{E^*} \quad (57.8)$$

where for plane stress

$$E^* = E \quad (57.9)$$

and for plane strain

$$E^* = \frac{E}{1 - \nu^2} \quad (57.10)$$

It is noted here that the calculation of K_I from G only holds if we have a purely Mode-I loading. Therefore, DIANA derives the stress intensity factor from the energy release rate as if the loading of the crack is purely Mode-I (opening).

57.2.2 Three-dimensional Analysis

The analysis of the energy release rate in two-dimensional structures is straightforward. For three-dimensional analysis the energy release rate can vary along the crack tip. DIANA calculates the energy release rate in three-dimensional models (with solid elements) by independent virtual crack extension at different nodes at the crack tip. The virtual crack extension always takes place in the plane of the crack itself. It was shown that, for quadratic elements, the most accurate results are obtained if the corner nodes of every element are the nodes that will have independent crack extensions and if the midpoint nodes of an element are given half the crack extension of a corner node. In this way, the crack grows around a chosen corner node in a shape that is triangular in the isoparametric coordinates [Fig. 57.2]. From the original structure and the perturbed structure, the

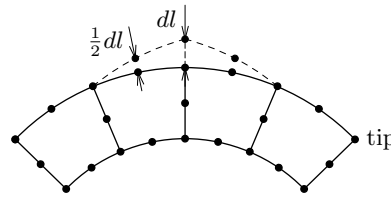


Figure 57.2: Virtual crack extension in three-dimensional analysis

decrease in elastic energy $-\Delta W_{\text{in}}$ can be calculated in the same way as for two-dimensional structures. The integral of $G \times \Delta a$ along the crack tip must be equal to this decrease in elastic energy.

$$\int_{\text{tip}} G \Delta a \, ds = -\Delta W_{\text{in}} \quad (57.11)$$

In the left hand side of this relation, s is the length along the crack tip. The energy release rate can be calculated by assuming that G varies linearly between two corner nodes. This yields a set of equations in G , from which G can be calculated at the different corner points. For three-dimensional calculations DIANA applies the plane strain equation Eq. (57.10).

The implementation of the virtual crack extension method has been corrected for the thermal and concentration loads and initial stress at the crack tip elements, as well as for external loading on the crack faces like pressure loading.

Part XIV

Appendix

Appendix A

Using User-supplied Subroutines

DIANA offers you, as an end-user, the opportunity to supply Fortran source code of some predefined subroutines. A description of the predefined subroutines is given in the various Volumes of the DIANA User's Manual, see for instance Chapter *User-supplied Models* in Volume *Material Library*. In this section, only the general working of the user-supplied subroutine mechanism will be discussed.

Cautionary note

*Redefining or changing the variables which are used to pass information to a user-supplied subroutine will have unpredictable effects. Only those variables which are defined as output variable should be updated in the subroutine. If user-supplied subroutines call other subroutines or use **COMMON**-blocks to pass data, it is mandatory to name these subroutines and **COMMON**-blocks with an identifier starting with a double capital X: **XX----**.*

It is good practice to develop user-supplied subroutines and then test the functionality on a single-element example before using the subroutines in production analyses. The only complicated factor in the small analyses is the developed subroutine and attention can be focused on the proper working of the user-supplied model.

Requirements. The user-supplied subroutine option requires the following:

- A basic knowledge of programming in general and of programming in Fortran in particular. For general aspects of programming style see for instance Kernighan & Plauger [51].
- A system with a Fortran compiler and the Fortran run-time libraries. A list of supported Fortran compilers can be found on the DIANA product pages of the DIANA FEA BV website: dianafea.com.
- A DIANA installation including the necessary object library files.

In case of any doubt contact your DIANA support.

A.1 Supplying the Source Code

This section describes the analysis commands for Module FORTRAN which indicates that user-supplied subroutines should be used. The user-supplied subroutine can be provided by using an external shared library.

syntax

```

*FORTRAN
USE usrlibs
*modulew
...           analysis commands
*END

```

USE indicates to use routines from precompiled shared library file *usrlib*. The file name must contain the extension *.dll* on MS-Windows systems and *.so* on Linux systems.

The name of file *usrlib* can be chosen freely. Externals which are not available in any of DIANA's standard service libraries, must follow the code of the user-supplied routine. If more than one user-routine is supplied, all routines must be included in the same shared library file *usrlib* specified in the USE command.

**module* is the module command to start the analysis, with the *analysis commands*.

A.1.1 Building Shared Libraries

External shared libraries can be build in many ways. Describing all available manners is not the aim of this section. This is a description on how to build a shared library with the supported compiler using the command line. To create a shared library from source using the supported fortran compiler *ifort* use the following command:

```
ifort <flags> -o usrlib {sourcefile(s)} {lib(s)}
```

Where:

ifort indicates the supported fortran compiler.

<flags> indicates the compiler flags. For the *ifort* compiler, these flags can be found in the file *uss.flags* in the *lib* directory of the DIANA installation.

-o *usrlib* indicates the file name of the compiled shared library file. The file name must contain the extension *.dll* on MS-Windows systems and *.so* on Linux systems.

{sourcefile(s)} indicates the file name(s) of the source file(s) for the compiled shared library file.

{lib(s)} indicates the names of the required programmer's service libraries [§ A.3]. These libraries can be found in the *lib* directory of the DIANA installation.

To ensure that the correct system libraries are used the following command must be given on MS-Windows platforms after generating the shared library:

```
mt.exe -manifest usrlib.manifest -outputresource:usrlib;2
```

Example. First of all we need to provide the source code. In this example we use a Fortran-90 source file that contains two user-supplied routines.

Fortran

```

      SUBROUTINE ELSEUS( EPS, NSTR, SE )
!DEC$ ATTRIBUTES DLLEXPORT::ELSEUS
!
! PURPOSE : User supplied tangential [SE] - matrix
!
      INTEGER          NSTR
      DOUBLE PRECISION EPS(*), SE(NSTR,*)

```



```

!
      DOUBLE PRECISION YOUNG
!
      YOUNG = 1.D3
      DO 100, I=1,NSTR
      DO 100, J=1,NSTR
100 SE(J,I) = 0.0D0
!
      IF ( NSTR.EQ.4 ) THEN
        SE(1,1) = YOUNG + 2000.DO * YOUNG * EPS(1)
        SE(2,2) = YOUNG + 2000.DO * YOUNG * EPS(2)
        SE(3,3) = YOUNG + 2000.DO * YOUNG * EPS(3)
        SE(4,4) = 0.5 * YOUNG
      END IF
!
      END
      SUBROUTINE ELSGUS( EPS, NSTR, TSIG )
!DEC$ ATTRIBUTES DLLEXPORT::ELSGUS
!
! PURPOSE : USER SUPPLIED TOTAL STRESSES
!
      INTEGER          NSTR
      DOUBLE PRECISION EPS(NSTR) , TSIG(NSTR)
!
      DOUBLE PRECISION YOUNG
!
      YOUNG = 1.D3
      IF ( NSTR.EQ.4 ) THEN
        TSIG(1) = YOUNG * EPS(1) + 1000.DO * YOUNG * EPS(1) ** 2
        TSIG(2) = YOUNG * EPS(2) + 1000.DO * YOUNG * EPS(2) ** 2
        TSIG(3) = YOUNG * EPS(3) + 1000.DO * YOUNG * EPS(3) ** 2
        TSIG(4) = 0.5 * YOUNG * EPS(4)
      END IF
!
      END

```

Please note the "`!DEC$ ATTRIBUTES DLLEXPORT`" lines in both routines. Without those lines the user supplied routines will not be visible to DIANA on MS-Windows platforms. For generation of the `uscou1.dll` shared library file from the sourcefile `elseus.f90` containing two user-supplied subroutines, `ELSEUS` and `ELSGUS`, on a Windows 64-bit platform the following command has to be given in the DIANA Command Box:

```
ifort <flags>.../lib/uss.flags -o uscou1.dll elseus.f90.../lib/lbcx40.lib
```

Where `<flags>` indicates the compiler flags, i.e. the content of `.../lib/uss.flags` and `...` indicates the directory of the DIANA installation. To ensure that the correct system libraries are being used the following command must be given on a MS-Windows platform in the DIANA Command Box:

```
mt.exe -manifest uscou1.dll.manifest -outputresource:uscou1.dll;2
```

A.2 Available Routines

The available routines are listed in the file `$DIASLIB/userinfo`. For DIANA-10.1 this file looks like:

```
# user supplied routine information
# layout:
```

```
# <routine_name>: <used_in_segment> ...
uslafr: ap/la02
elsgus: ap/nl41
elseus: ap/nl41
bowliq: ap/nl41
nisliq: ap/nl41
towliq: ap/nl41
usrbea: ap/ls41
usrliq: ap/nl41
usryou: ap/nl41
usrpoi: ap/nl41
usrten: ap/nl41
usrcrk: ap/nl41
usrcrv: ap/nl41
usrepu: ap/nl41
usrifc: ap/nl41
usrmat: ap/nl41
usrmnl: ap/nl41
usrshl: ap/ls41
usrshr: ap/nl41
usrcrp: ap/nl41
usrcst: ap/nl41
usrgf1: ap/nl41
usrtst: ap/nl41
usrbet: ap/nl41
usrrub: ap/ls41 ap/nl41
usrbou: ap/pe41
usrpar: ap/pe41
usrhtp: ap/ht40
usrshk: ap/nl41
```

Each line contains the name and location of a pre-defined user-supplied subroutine. For instance the line starting with `elsgus` indicates that subroutine `ELSGUS` is user-supplied and called in the executable `ap/nl41`. The directory name `ap` refers to the DIANA application executables and the file name `nl41` to version 41 of Module `NONLIN`.

A.3 Programmer's Service Libraries

The DIANA programming environment comes with a set of service libraries of utility routines. These libraries are especially helpful when coding user-supplied subroutines because the routines are extensively tested and often optimized. When applying user-supplied subroutines, you are encouraged to call the routines of the DIANA service libraries whenever appropriate. DIANA programmer's service libraries are subdivided for some basic tasks such as matrix and vector manipulations, I/O operations etc. This section presents the subroutines and functions which are particularly useful in combination with the user-supplied subroutine option. See the DIANA Programmer's Manual for full description of the service library subroutines.¹

A.3.1 Matrix Manipulation

DIANA's Matrix library (UNIX: `liblbgs30.so`; MS-Windows: `lbgs30.lib`) offers a set of subroutines to perform matrix manipulations like multiplication, triple products, inversion, and initialization. In this library we use the following naming convention:

R----- Result is stored in output matrix **R**.

S----- Result is superposed to input matrix **R**.

¹The DIANA Programmer's Manual is available interactively via the documentation processor `ddoc`.

A----- Result is stored back in input matrix **A**.
 B----- Result is stored back in input matrix **B**.
 -A---- Input matrix **A**.
 -B---- Input matrix **B**.
 --T--- Preceding input matrix is processed transposed.
 INV--- Input matrix is inverted.
 D----- Decomposition of input matrix.
 DET--- Determinant of input matrix.
 ---#--- Dimension of input matrix.

For example: subroutine **RAB** performs the matrix multiplication $\mathbf{R} = \mathbf{A} \times \mathbf{B}$.

A.3.1.1 Simple Products

Multiply $\mathbf{R} = \mathbf{A} \times \mathbf{B}$

Fortran

```
CALL RAB( a, n, m, b, l, r )
```

in	DBL	a(n,m)	$\mathbf{A}_{n \times m}$.
in	DBL	b(m,l)	$\mathbf{B}_{m \times l}$.
out	DBL	r(n,l)	Result $\mathbf{R}_{n \times l}$.
in	INT	n m l	Dimensions n , m and l .

Subroutine **RAB** performs a simple matrix multiplication:

$$\begin{bmatrix} l \\ n \end{bmatrix} \mathbf{R} = \begin{bmatrix} m \\ n \end{bmatrix} \mathbf{A} \begin{bmatrix} l \\ m \end{bmatrix} \mathbf{B}$$

file.f

```
DOUBLE PRECISION AMAT(4,3), BMAT(3,5), RMAT(4,5)
CALL RAB( AMAT, 4, 3, BMAT, 5, RMAT )
```

See also **AAB**, **BAB** and **SAB**.

Multiply $\mathbf{R} = \mathbf{R} + \mathbf{A} \times \mathbf{B}$

Fortran

```
CALL SAB( a, n, m, b, l, r, k )
```

in	DBL	a(n,m)	$\mathbf{A}_{n \times m}$.
in	DBL	b(m,l)	$\mathbf{B}_{m \times l}$.
in	DBL	r(k,l)	Input $\mathbf{R}_{n \times l}$.
out	DBL	r(k,l)	Added result $\mathbf{R}_{n \times l}$.
in	INT	n m l	Dimensions n , m and l .
in	INT	k	Column size k of array r .

Subroutine **SAB** superposes a matrix product to the result matrix:

$$\begin{bmatrix} l \\ n \\ k \end{bmatrix} \mathbf{R} = \begin{bmatrix} l \\ n \end{bmatrix} \mathbf{R} + \begin{bmatrix} m \\ n \end{bmatrix} \mathbf{A} \begin{bmatrix} l \\ m \end{bmatrix} \mathbf{B}$$

The result matrix $\mathbf{R}_{n \times l}$ may be stored in a part of array **r**. If $k < n$ the result is ($k \geq n$) erroneous.

file.f

```

DOUBLE PRECISION AMAT(3,2), BMAT(2,4), RMAT(6,8)
CALL RSET( 0.DO, RMAT, 6*8 )
CALL SAB( AMAT, 3, 2, BMAT, 4, RMAT(1,1), 6 )
CALL SAB( AMAT, 3, 2, BMAT, 4, RMAT(4,1), 6 )
CALL SAB( AMAT, 3, 2, BMAT, 4, RMAT(1,5), 6 )
CALL SAB( AMAT, 3, 2, BMAT, 4, RMAT(4,5), 6 )

```

See also RAB.

Multiply $\mathbf{A} = \mathbf{A} \times \mathbf{B}$

Fortran

```
CALL AAB( a, n, m, b, l, w )
```

in DBL a(n,m) $\mathbf{A}_{n \times m}$.
in DBL b(m,l) $\mathbf{B}_{m \times l}$.
out DBL a(n,l) Result $\mathbf{A}_{n \times l}$.
in INT n m l Dimensions n , m and l .
spc DBL w(l) Workspace.

Subroutine AAB returns a matrix product in the first input array:

$$\begin{bmatrix} l \\ n \\ \mathbf{A} \end{bmatrix} = \begin{bmatrix} m \\ n \\ \mathbf{A} \end{bmatrix} ? \begin{bmatrix} l \\ m \\ \mathbf{B} \end{bmatrix}$$

This subroutine may be used instead of RAB to save memory space. The result matrix $\mathbf{A}_{n \times l}$ is returned in array **a** itself, so the input matrix $\mathbf{A}_{n \times m}$ is destroyed. The rows of array **a** must be dimensioned to the maximum of l and m .

file.f

```

DOUBLE PRECISION AMAT(3,5), BMAT(2,5), WSPACE(5)
DATA AMAT/ 3*1.DO, 3*2.DO, 9*0.DO /
DATA BMAT/ 2*1.DO, 2*2.DO, 2*3.DO, 2*4.DO, 2*5.DO /
CALL PRIMAT( AMAT, 3, 2, 'AMAT1 ' )
CALL PRIMAT( BMAT, 2, 5, 'BMAT ' )
CALL AAB( AMAT, 3, 2, BMAT, 5, WSPACE )
CALL PRIMAT( AMAT, 3, 5, 'AMAT2 ' )
END

```

This example yields the following output:

file.out

```

AMAT1 :      1      2
1  1.000E+00  2.000E+00
2  1.000E+00  2.000E+00
3  1.000E+00  2.000E+00
BMAT :      1      2      3      4      5
1  1.000E+00  2.000E+00  3.000E+00  4.000E+00  5.000E+00
2  1.000E+00  2.000E+00  3.000E+00  4.000E+00  5.000E+00
AMAT2 :      1      2      3      4      5
1  3.000E+00  6.000E+00  9.000E+00  1.200E+01  1.500E+01
2  3.000E+00  6.000E+00  9.000E+00  1.200E+01  1.500E+01
3  3.000E+00  6.000E+00  9.000E+00  1.200E+01  1.500E+01

```

See also RAB and BAB.

Multiply $B = A \times B$ *Fortran*

```

      CALL BAB( a, n, m, b, l, w )

in  DBL a(n,m)       $A_{n \times m}$ .
in  DBL b(m,l)       $B_{m \times l}$ .
out DBL b(n,l)      Result  $B_{n \times l}$ .
in  INT n m l       Dimensions  $n$ ,  $m$  and  $l$ .
spc DBL w(n)        Workspace.

```

Subroutine BAB returns a matrix product in the second input array:

$$\begin{bmatrix} l \\ n \\ \mathbf{B} \\ ? \end{bmatrix} = \begin{bmatrix} m \\ n \\ \mathbf{A} \end{bmatrix} \begin{bmatrix} l \\ m \\ \mathbf{B} \end{bmatrix}$$

This subroutine may be used instead of RAB to save memory space. The result matrix $B_{n \times l}$ is returned in array `b` itself, so the input matrix $B_{m \times l}$ is destroyed. The following example calls subroutine PRIMAT to show how the various matrixes have been filled. ($m \geq n$)

file.f

```

DOUBLE PRECISION AMAT(2,3), BMAT(12), WSPACE(2)
DATA AMAT/ 2*1.D0, 2*2.D0, 2*3.D0 /
DATA BMAT/ 3*1.D0, 3*2.D0, 3*3.D0, 3*4.D0 /
CALL PRIMAT( AMAT, 2, 3, 'AMAT ' )
CALL PRIMAT( BMAT, 3, 4, 'BMAT1 ' )
CALL BAB( AMAT, 2, 3, BMAT, 4, WSPACE )
CALL PRIMAT( BMAT, 2, 4, 'BMAT2 ' )
END

```

Note that array BMAT is declared one-dimensional because it is used with two different column lengths. The address space of the output array BMAT is filled without gaps as shown in the example's output:

file.out

```

AMAT :      1      2      3
1  1.000E+00  2.000E+00  3.000E+00
2  1.000E+00  2.000E+00  3.000E+00
BMAT1 :      1      2      3      4
1  1.000E+00  2.000E+00  3.000E+00  4.000E+00
2  1.000E+00  2.000E+00  3.000E+00  4.000E+00
3  1.000E+00  2.000E+00  3.000E+00  4.000E+00
BMAT2 :      1      2      3      4
1  6.000E+00  1.200E+01  1.800E+01  2.400E+01
2  6.000E+00  1.200E+01  1.800E+01  2.400E+01

```

See also RAB and AAB.

Multiply $R = A \times B^T$ *Fortran*

```

      CALL RABT( a, n, m, b, l, r )

in  DBL a(n,m)       $A_{n \times m}$ .
in  DBL b(l,m)       $B_{l \times m}$ .
out DBL r(n,l)      Result  $R_{n \times l}$ .
in  INT n m l       Dimensions  $n$ ,  $m$  and  $l$ .

```

Subroutine RABT performs a simple matrix multiplication with the transpose of the second input matrix:

$$\begin{bmatrix} l \\ n \\ \mathbf{R} \end{bmatrix} = \begin{bmatrix} m \\ n \\ \mathbf{A} \end{bmatrix} \begin{bmatrix} l \\ m \\ \mathbf{B}^T \end{bmatrix}$$

file.f

```
DOUBLE PRECISION AMAT(4,3), BMAT(5,3), RMAT(4,5)
CALL RABT( AMAT, 4, 3, BMAT, 5, RMAT )
```

See also RATB.

Multiply $\mathbf{R} = \mathbf{A}^T \times \mathbf{B}$

Fortran

```
CALL RATB( a, n, m, b, l, r )
```

in DBL a(n,m) $\mathbf{A}_{n \times m}$.
in DBL b(n,l) $\mathbf{B}_{n \times l}$.
out DBL r(m,l) Result $\mathbf{R}_{m \times l}$.
in INT n m l Dimensions n , m and l .

Subroutine RATB performs a simple matrix multiplication with the transpose of the first input matrix:

$$\begin{bmatrix} l \\ m \end{bmatrix} \mathbf{R} = \begin{bmatrix} n \\ m \end{bmatrix} \mathbf{A}^T \begin{bmatrix} l \\ n \end{bmatrix} \mathbf{B}$$

file.f

```
DOUBLE PRECISION AMAT(3,4), BMAT(3,5), RMAT(4,5)
CALL RATB( AMAT, 3, 4, BMAT, 5, RMAT )
```

See also RABT.

Multiply $\mathbf{R} = \mathbf{R} + \mathbf{A}^T \times \mathbf{B}$

Fortran

```
CALL SATB( a, n, m, b, l, r, k )
```

in DBL a(n,m) $\mathbf{A}_{n \times m}$.
in DBL b(n,l) $\mathbf{B}_{n \times l}$.
in DBL r(k,l) Input $\mathbf{R}_{m \times l}$.
out DBL r(k,l) Added result $\mathbf{R}_{m \times l}$.
in INT n m l Dimensions n , m and l .
in INT k Column size k of array \mathbf{r} .

Subroutine SATB superposes a matrix product with the transpose of the first input matrix to the result matrix:

$$\begin{matrix} \uparrow \\ k \end{matrix} \begin{bmatrix} l \\ m \end{bmatrix} \mathbf{R} = \begin{bmatrix} l \\ m \end{bmatrix} \mathbf{R} + \begin{bmatrix} n \\ m \end{bmatrix} \mathbf{A}^T \begin{bmatrix} l \\ n \end{bmatrix} \mathbf{B}$$

The result matrix $\mathbf{R}_{m \times l}$ may be stored in a part of array \mathbf{r} . If $k < m$ the result is ($k \geq m$) erroneous. See also RABT and SAB.

A.3.1.2 Triple Products

Triple product $\mathbf{R} = \mathbf{A}^T \times \mathbf{B} \times \mathbf{A}$ *Fortran*

```

      CALL RATBA( a, n, m, b, r, w, k )
in  DBL a(n,m)       $\mathbf{A}_{n \times m}$ .
in  DBL b(n,n)       $\mathbf{B}_{n \times n}$ .
out DBL r(k,m)      Result  $\mathbf{R}_{m \times m}$ .
spc DBL w(m)        Workspace.
in  INT n m         Dimensions  $n$  and  $m$ .
in  INT k           Column size  $k$  of array  $\mathbf{r}$ .

```

Subroutine RATBA performs a triple matrix multiplication:

$$\begin{array}{c} \begin{array}{|c|} \hline m \\ \hline \end{array} \left[\begin{array}{c} m \\ \hline m \\ \hline \end{array} \right] \mathbf{R} = \begin{array}{c} n \\ \left[\begin{array}{c} m \\ \hline m \\ \hline \end{array} \right] \mathbf{A}^T \end{array} \begin{array}{c} n \\ \left[\begin{array}{c} n \\ \hline n \\ \hline \end{array} \right] \mathbf{B} \end{array} \begin{array}{c} m \\ \left[\begin{array}{c} n \\ \hline n \\ \hline \end{array} \right] \mathbf{A} \end{array}$$

k

The result of the triple matrix product is returned in the first m rows of array $\mathbf{r}(k, m)$. If $k > m$ then the remainder is filled with zeros. In practice $k = m$ in many cases. If $k < m$ the result is erroneous. See also SATBA.

 $(k \geq m)$ **Triple product $\mathbf{R} = \mathbf{R} + \mathbf{A}^T \times \mathbf{B} \times \mathbf{A}$** *Fortran*

```

      CALL SATBA( a, n, m, b, r, w, k )
in  DBL a(n,m)       $\mathbf{A}_{n \times m}$ .
in  DBL b(n,n)       $\mathbf{B}_{n \times n}$ .
in  DBL r(k,m)      Input  $\mathbf{R}_{m \times m}$ .
out DBL r(k,m)      Added result  $\mathbf{R}_{m \times m}$ .
spc DBL w(n)        Workspace.
in  INT n m         Dimensions  $n$  and  $m$ .
in  INT k           Column size  $k$  of array  $\mathbf{r}$ .

```

Subroutine SATBA superposes a triple matrix product to the result matrix:

$$\begin{array}{c} \begin{array}{|c|} \hline m \\ \hline \end{array} \left[\begin{array}{c} m \\ \hline m \\ \hline \end{array} \right] \mathbf{R} = \begin{array}{c} m \\ \left[\begin{array}{c} m \\ \hline m \\ \hline \end{array} \right] \mathbf{R} \end{array} + \begin{array}{c} n \\ \left[\begin{array}{c} m \\ \hline m \\ \hline \end{array} \right] \mathbf{A}^T \end{array} \begin{array}{c} n \\ \left[\begin{array}{c} n \\ \hline n \\ \hline \end{array} \right] \mathbf{B} \end{array} \begin{array}{c} m \\ \left[\begin{array}{c} n \\ \hline n \\ \hline \end{array} \right] \mathbf{A} \end{array}$$

k

The input and output matrix $\mathbf{R}_{m \times m}$ may be stored in a part of array \mathbf{r} . The result of the triple matrix product is superposed to the first m rows of this array. If $k < m$ the result is erroneous. See also RATBA.

 $(k \geq m)$

A.3.1.3 Inversion

Invert 2×2 matrix*Fortran*

```

      CALL INV2( a, r, det )
in  DBL a(2,2)       $\mathbf{A}_{2 \times 2}$ .
out DBL r(2,2)      Result  $\mathbf{R}_{2 \times 2} = \mathbf{A}_{2 \times 2}^{-1}$ .
out DBL det         Determinant  $\det \mathbf{A}$ .

```

Subroutine INV2 returns the inverse and the determinant of a 2×2 matrix \mathbf{A} . If \mathbf{A} is not positive definite, the inverse cannot be calculated and a fatal error occurs. Positive definiteness of \mathbf{A} may be checked beforehand by subroutine DET2. See also INV3 and INVMTX.

Invert 3×3 matrix*Fortran*

 CALL INV3(a, r, det)

in	DBL	a(3,3)	$\mathbf{A}_{3 \times 3}$.
out	DBL	r(3,3)	Result $\mathbf{R}_{3 \times 3} = \mathbf{A}_{3 \times 3}^{-1}$.
out	DBL	det	Determinant det \mathbf{A} .

Subroutine INV3 returns the inverse and the determinant of a 3×3 matrix \mathbf{A} . If \mathbf{A} is not positive definite, the inverse cannot be calculated and a fatal error occurs. Positive definiteness of \mathbf{A} may be checked beforehand by subroutine DET3. See also INV2 and INVMTX.

Invert matrix by partial pivoting*Fortran*

 CALL INVMTX(n, a, ai, dodet, det, silent, errflg)

in	INT	n	Dimension n .
in	DBL	a(n,n)	$\mathbf{A}_{n \times n}$.
out	DBL	ai(n,n)	$\mathbf{A}_{n \times n}^{-1}$.
in	LOG	dodet	Compute determinant?
out	DBL	det	Determinant of \mathbf{A} .
in	LOG	silent	Work silently?
out	LOG	errflg	Error encountered?

Subroutine INVMTX calculates the inverse of a matrix \mathbf{A} by a partial pivoting algorithm:

$$\begin{bmatrix} & n \\ n & \mathbf{A} \end{bmatrix} \xrightarrow[\text{errflg} = \text{.FALSE.}]{\text{errflg}} \begin{bmatrix} & n \\ n & \mathbf{A}^{-1} \end{bmatrix}$$

If the input \mathbf{A} is (nearly) singular, the inverse cannot be calculated. In this case **errflag** is returned as **.TRUE.** and \mathbf{A}^{-1} is undetermined. The value of the determinant is returned in **det** only if **dodet** is input as **.TRUE.** and **errflag** is returned as **.FALSE.**. If **silent** is input as **.TRUE.** then no error messages will be given. In this case only **errflag** will indicate whether or not the inversion has succeeded. See also INV2, INV3, and INVPGB.

Invert matrix by Gauss–Jordan*Fortran*

 CALL INVPGB(a, n, w, v, eps, err)

in	DBL	a(n,n)	$\mathbf{A}_{n \times n}$.
out	DBL	a(n,n)	$\mathbf{A}_{n \times n}^{-1}$.
in	DBL	eps	Tolerance ε for pivot.
in	INT	n	Dimension n .
spc	INT	w(n)	Integer workspace.
spc	DBL	v(n)	Real workspace.
out	INT	err	Error code.

Subroutine INVPGB calculates the inverse of a matrix $\mathbf{A}_{n \times n}$ by a Gauss–Jordan algorithm with partial pivoting:

$$\begin{bmatrix} & n \\ n & \mathbf{A} \end{bmatrix} \xrightarrow[\text{err} = 0]{\text{err}} \begin{bmatrix} & n \\ n & \mathbf{A}^{-1} \end{bmatrix}$$

The inverse \mathbf{A}^{-1} is returned in array **a** which means that the original matrix is lost. The error code returns the status of the inversion: **err** = 0 indicates successful inversion, **err** = -1 indicates singularity. See also INVMTX.

A.3.1.4 Decomposition

Polar decomposition*Fortran*

 CALL DGEPPD(n, a, s, q, r, d, w, info)

in	DBL	a(n,n)	$\mathbf{A}_{n \times n}$.
in	INT	n	Dimension n .
out	DBL	s(n,n)	Stretch matrix $\mathbf{S}_{n \times n}$.
out	DBL	q(n,n)	Rotation matrix $\mathbf{Q}_{n \times n}$.
out	DBL	r(n,n)	Square root of \mathbf{S} .
out	DBL	d(n)	Singular values of \mathbf{A} .
out	INT	info	Job information.
spc	DBL	w(n)	Work space.

Subroutine DGEPPD performs a polar decomposition of the matrix \mathbf{A} according to Strang [83]:

$$\mathbf{A}_{n \times n} = \mathbf{Q}_{n \times n} \mathbf{S}_{n \times n}$$

If `info` = 0 then the polar decomposition has been successful.

A.3.1.5 Determinants

Determinant of 2×2 matrix*Fortran*

 det = DET2(a)

in	DBL	a(2,2)	$\mathbf{A}_{2 \times 2}$.
out	DBL	det	Determinant $\det \mathbf{A}$.

Subroutine DET2 returns the determinant of a 2×2 matrix \mathbf{A} . See also DET3 and INVMTX.

Determinant of 3×3 matrix*Fortran*

 det = DET3(a)

in	DBL	a(3,3)	$\mathbf{A}_{3 \times 3}$.
out	DBL	det	Determinant $\det \mathbf{A}$.

Subroutine DET3 returns the determinant of a 3×3 matrix \mathbf{A} . See also DET2 and INVMTX.

A.3.1.6 Miscellaneous

Clear matrix*Fortran*

 CALL CLEAR(e, a, n, m)

in	DBL	e	Round-off value ε .
in	DBL	a(n,m)	$\mathbf{A}_{n \times m}$.
out	DBL	a(n,m)	Cleared $\mathbf{A}_{n \times m}$.
in	INT	n,m	Dimensions n and m .

Subroutine CLEAR clears a matrix $\mathbf{A}_{n \times m}$. It sets the elements of \mathbf{A} to zero if their absolute value is $\leq \varepsilon$:

$$\text{if } |A(i,j)| \leq \varepsilon \text{ then } A(i,j) = 0$$

Fill symmetric matrix*Fortran*

 CALL FILMA(kod, a, n)

in INT kod Fill code.
 in DBL a(n,n) One half of $\mathbf{A}_{n \times n}$.
 out DBL a(n,n) Filled $\mathbf{A}_{n \times n}$.
 in INT n Dimension n .

Subroutine FILMA fills the empty half of a symmetric matrix $\mathbf{A}_{n \times n}$:

$$\begin{bmatrix} n & & \\ & \mathbf{A}_u & \\ n & ? & \end{bmatrix} \xrightarrow{\text{kod} > 0} \begin{bmatrix} n & & \\ & \mathbf{A} & \\ n & & \end{bmatrix} \qquad \begin{bmatrix} n & & \\ & ? & \\ n & \mathbf{A}_l & \end{bmatrix} \xrightarrow{\text{kod} < 0} \begin{bmatrix} n & & \\ & \mathbf{A} & \\ n & & \end{bmatrix}$$

The value of **kod** determines which half is to be filled. If **kod** > 0 the lower half is made identical to the upper, if **kod** < 0 the upper identical to the lower. If **kod** = 0 nothing is done.

Set unity matrix*Fortran*

 CALL UNITMX(r, n)

out DBL r(n,n) Unity matrix $\mathbf{I}_{n \times n}$.
 in INT n Dimension n .

Subroutine UNITMX returns a unity matrix $\mathbf{I}_{n \times n}$:

$$\begin{bmatrix} n & & \\ & ? & \\ n & & \end{bmatrix} \longrightarrow \begin{bmatrix} n & & \\ & \mathbf{I} & \mathbf{0} \\ n & \mathbf{0} & \end{bmatrix}$$

A.3.2 Vector Manipulation

DIANA's Vector library (UNIX: liblbs30.so; MS-Windows: lbs30.lib) offers a set of subroutines to perform vector manipulations such as superposition, scaling, inner- and outer-products. In this library we use the following naming convention:

U----- Result is vector **u**.
 -V----- Input is vector **v**.
 --P---- Addition (plus).
 --M---- Subtraction (minus).
 ---W--- Result is vector **w**.
 ----S- Input is scalar *S*.

A.3.2.1 Superposition**Add vectors $\mathbf{u} = \mathbf{v} + \mathbf{w}$** *Fortran*

 CALL UVPW(v, w, n, u)

in DBL v(n) \mathbf{v}_n .
 in DBL w(n) \mathbf{w}_n .
 out DBL u(n) Result \mathbf{u}_n .
 in INT n Dimension n .

Subroutine UVPW returns the sum of two column vectors **v** and **w** in a third column vector **u**:

$$\begin{bmatrix} n \\ \mathbf{u} \end{bmatrix} = \begin{bmatrix} n \\ \mathbf{v} \end{bmatrix} + \begin{bmatrix} n \\ \mathbf{w} \end{bmatrix}$$

See also UVMW and UVPWS.

Subtract vectors $\mathbf{u} = \mathbf{v} - \mathbf{w}$ *Fortran*

 CALL UVMW(v, w, n, u)

in	DBL	v(n)	\mathbf{v}_n .
in	DBL	w(n)	\mathbf{w}_n .
out	DBL	u(n)	Result \mathbf{u}_n .
in	INT	n	Dimension n .

Subroutine UVMW subtracts a column vector \mathbf{w} from a column vector \mathbf{v} and returns the result in a column vector \mathbf{u} :

$$\begin{bmatrix} \vdots \\ n \\ \vdots \end{bmatrix} \mathbf{u} = \begin{bmatrix} \vdots \\ n \\ \vdots \end{bmatrix} \mathbf{v} - \begin{bmatrix} \vdots \\ n \\ \vdots \end{bmatrix} \mathbf{w}$$

See also UVPW.

Add scaled vector $\mathbf{u} = \mathbf{v} + S \mathbf{w}$ *Fortran*

 CALL UVPWS(v, w, n, s, u)

in	DBL	v(n)	\mathbf{v}_n .
in	DBL	w(n)	\mathbf{w}_n .
in	DBL	s	Scale factor S .
out	DBL	u(n)	Result \mathbf{u}_n .
in	INT	n	Dimension n .

Subroutine UVPWS adds a scaled vector \mathbf{w} to a column vector \mathbf{v} and returns the result in a column vector \mathbf{u} :

$$\begin{bmatrix} \vdots \\ n \\ \vdots \end{bmatrix} \mathbf{u} = \begin{bmatrix} \vdots \\ n \\ \vdots \end{bmatrix} \mathbf{v} + S \begin{bmatrix} \vdots \\ n \\ \vdots \end{bmatrix} \mathbf{w}$$

See also UVPW and UVS.

A.3.2.2 Scaling**Scale vector $\mathbf{u} = S \mathbf{v}$** *Fortran*

 CALL UVS(v, n, s, u)

in	DBL	v(n)	\mathbf{v}_n .
in	DBL	s	Scale factor S .
out	DBL	u(n)	Result \mathbf{u}_n .
in	INT	n	Dimension n .

Subroutine UVS calculates the product vector \mathbf{u} of a scalar S and a vector \mathbf{v} :

$$\begin{bmatrix} \vdots \\ n \\ \vdots \end{bmatrix} \mathbf{u} = S \begin{bmatrix} \vdots \\ n \\ \vdots \end{bmatrix} \mathbf{v}$$

See also UVPWS.

A.3.2.3 Inner-products

Simple inner-product $P = \mathbf{u}^T \mathbf{v}$

Fortran

```

      p = UV( u, v, n )

in   DBL u(n)            $\mathbf{u}_n$ .
in   DBL v(n)            $\mathbf{v}_n$ .
in   INT  n             Dimension  $n$ .
out  DBL p              Inner-product  $P$ .

```

Subroutine UV calculates the inner-product P of two column vectors \mathbf{u} and \mathbf{v} :

$$P = \begin{bmatrix} \vdots \\ \mathbf{u} \\ \vdots \end{bmatrix}^T \begin{bmatrix} \vdots \\ \mathbf{v} \\ \vdots \end{bmatrix}$$

See also PRODIN and VINPRO.

Inner-product of row vectors $P = \mathbf{A}_{(i,)} \mathbf{B}_{(j,)}^T$

Fortran

```

      p = VINPRO( a, m, b, k, n )

in   DBL a(n)           Row  $i$  of  $\mathbf{A}_{m \times n}$ .
in   DBL b(n)           Row  $j$  of  $\mathbf{B}_{k \times n}$ .
in   INT  m n k         Dimensions  $m$ ,  $n$  and  $k$ .
out  DBL p              Inner-product  $P$ .

```

Subroutine VINPRO calculates the inner-product P of two row vectors:

$$P = \begin{bmatrix} & n \\ & \mathbf{A} \\ i & \end{bmatrix} \begin{bmatrix} n \\ j & \\ k & \mathbf{B} \end{bmatrix}$$

Input arrays \mathbf{a} and \mathbf{b} are the start addresses of the rows.

file.f

```

      DOUBLE PRECISION VINPRO, AMAT(4,5), BMAT(6,5), PROD
      PROD = VINPRO( AMAT(3,1), 4, BMAT(2,1), 6, 5 )

```

This example returns the inner-product of the third row of **AMAT** and the second row of **BMAT**. See also UV.

A.3.2.4 Outer-products

Outer product in 3-D space $\mathbf{u} = \mathbf{v} \otimes \mathbf{w}$

Fortran

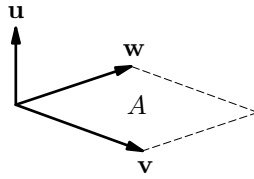
```

      CALL OUTER( v, w, u )

in   DBL v(3)            $\mathbf{v}_3$ .
in   DBL w(3)            $\mathbf{w}_3$ .
out  DBL u(3)           Result  $\mathbf{u}_3$ .

```

Subroutine OUTER calculates the outer product of two vectors \mathbf{v} and \mathbf{w} in three-dimensional space:



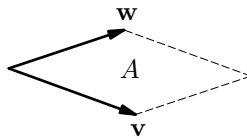
The result vector \mathbf{u} is the vector perpendicular to \mathbf{v} and \mathbf{w} . The length of \mathbf{u} is equal to the area A of the parallelogram spanned by \mathbf{v} and \mathbf{w} . See also AREA.

Area spanned in 3-D space $A = \|\mathbf{v}_3 \otimes \mathbf{w}_3\|_{L_2}$ *Fortran*

`a = AREA(v, w)`

in	DBL	v(3)	\mathbf{v}_3 .
in	DBL	w(3)	\mathbf{w}_3 .
out	DBL	a	Area A .

Subroutine AREA calculates the area A spanned by two vectors \mathbf{v} and \mathbf{w} in three-dimensional space:



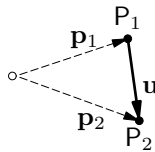
It is assumed that the two vectors \mathbf{v} and \mathbf{w} have the same origin. See also OUTER.

Arrow between two points $\mathbf{u} = \mathbf{p}_2 - \mathbf{p}_1$ *Fortran*

`CALL ARROW(p1, p2, u)`

in	DBL	p1(3)	Coordinates x, y, z of point P_1 .
in	DBL	p2(3)	Coordinates x, y, z of point P_2 .
out	DBL	u(3)	Vector \mathbf{u}_3 from P_1 to P_2 .

Subroutine ARROW returns the vector \mathbf{u} between two points P_1 and P_2 in three-dimensional space:



The vector is determined by subtraction of two vectors \mathbf{p}_1 and \mathbf{p}_2 . See also UVMW.

A.3.2.5 Norms

Normalize 3-D vector *Fortran*

`CALL NORM(v, len)`

in	DBL	v(3)	\mathbf{v}_3 .
out	DBL	v(3)	Normalized \mathbf{v}_3 .
out	DBL	len	L_2 -norm of input \mathbf{v}_3 .

Subroutine NORM normalizes a vector \mathbf{v} in three-dimensional space to a length equal to one. If \mathbf{v} is a null-vector, it cannot be normalized and NORM gives a fatal error. To prevent this, NULLVC should be called prior to NORM whenever \mathbf{v} may be a null-vector.

file.f

```

      DOUBLE PRECISION V(3), VCLEN
      LOGICAL          NULLVC
      IF ( NULLVC( V ) ) THEN
C...      GIVE AN ERROR MESSAGE,
           CALL PRINT *, 'WARNING: VECTOR EQUAL TO NULL-VECTOR'
           RETURN
      END IF
C...      NOW SAFELY CALL NORM TO NORMALIZE THE VECTOR.
           CALL NORM( V, VCLEN )

```

See also NULLVC and NORM2.

Check for 3-D null-vector

Fortran

```

      isnull = NULLVC( v )

```

in DBL v(3) \mathbf{v}_3 .
out LOG isnull Indicates if \mathbf{v}_3 is a null-vector.

Subroutine NULLVC checks if a vector in three-dimensional space is a null-vector. See also NORM.

Normalize 2-D vector

Fortran

```

      CALL NORM2( v, len )

```

in DBL v(2) \mathbf{v}_2 .
out DBL v(2) Normalized \mathbf{v}_2 .
out DBL len L_2 -norm of input \mathbf{v}_2 .

Subroutine NORM2 normalizes a vector \mathbf{v} in two-dimensional space to a length equal to one. If \mathbf{v} is a null-vector, it cannot be normalized and NORM2 gives a fatal error. See also NORM.

A.3.3 Miscellaneous

These libraries contain a set of subroutines to perform various tasks like printing of values, initialization of arrays, interpolation, sorting, error handling etc.

A.3.3.1 Printing

This section of DIANA's General Service library (UNIX: liblbs30.so; MS-Windows: lbs30.lib) describes subroutines to print values on DIANA's standard output file.

Print real value

Fortran

```

      CALL PRIVAL( val, label )

```

in DBL val Real value v to print.
in CHA label*6 Label to print.

Subroutine PRIVAL prints a real value v preceded by a label. The value is printed in E12.3 format. See also PRIIVL, PRILVL and PRIVEC.

file.f

```

      DOUBLE PRECISION TEMP
      CALL PRIVAL( TEMP, 'TEMPER' )

```

file.out

```

TEMPER=  -2.730E+02

```

Print real vector*Fortran*

```
CALL PRIVEC( v, n, label )
```

```

in  DBL  v(n)      Real vector  $\mathbf{v}_n$  to print.
in  INT   n         Dimension  $n$ .
in  CHA  label*6    Label to print.

```

Subroutine PRIVEC prints a real vector \mathbf{n} preceded by a label. The values are printed in 10E12.3 format. If $n > 10$ the vector is split in lines of at most ten values. See also PRIVAL, PRIIVL and PRILVL.

file.f

```

DOUBLE PRECISION TEMP(4)
CALL PRIVEC( TEMP, 4, 'TEMPER' )

```

file.out

```

TEMPER:  -2.730E+02  -1.710E+02   0.000E+00   1.000E+02

```

Print real matrix*Fortran*

```
CALL PRIMAT( a, n, m, label )
```

```

in  DBL  a(n,m)     Real matrix  $\mathbf{A}_{n \times m}$  to print.
in  INT   n m        Dimensions  $n$  and  $m$ .
in  CHA  label*6     Label to print.

```

Subroutine PRIMAT prints a real matrix \mathbf{A} preceded by a label. The values are printed in 10E12.3 format with row and column indices aside and above. If $m > 10$ the rows are split in lines of at most ten values. See also PRIVEC and MATPRI.

file.f

```

DOUBLE PRECISION SE(4,4)
CALL PRIMAT( SE, 4, 4, 'SE-MAT' )

```

file.out

```

SE-MAT :      1      2      3      4
  1  3.333E+04  8.333E+03  8.333E+03  0.000E+00
  2  8.333E+03  3.333E+04  8.333E+03  0.000E+00
  3  8.333E+03  8.333E+03  3.333E+04  0.000E+00
  4  0.000E+00  0.000E+00  0.000E+00  1.250E+04

```

Print integer value*Fortran*

```
CALL PRIIVL( val, label )
```

```

in  INT  val        Integer value  $v$  to print.
in  CHA  label*6     Label to print.

```

Routine PRIIVL prints an integer value v preceded by a label. The value is printed in I6 format. See also PRIVAL, PRILVL and PRIIVC.

file.f

```

      INTEGER KODE
      CALL PRIIVL( KODE, 'KODE  ' )

```

file.out

```

KODE = -273

```

Print integer vector*Fortran*

```

      CALL PRIIVC( v, n, label )

```

in INT v(n) Integer vector \mathbf{v}_n to print.
in INT n Dimension n .
in CHA label*6 Label to print.

Subroutine PRIIVC prints an integer vector \mathbf{n} preceded by a label. The values are printed in 20I6 format. If $n > 20$ the vector is split in lines of at most twenty values. See also PRIVEC and PRIIVL.

file.f

```

      INTEGER V(6)
      CALL PRIIVC( V, 6, 'VALUES' )

```

file.out

```

VALUES:   101   202   303   404   505   606

```

Print integer matrix*Fortran*

```

      CALL MATPRI( a, n, m, label )

```

in INT a(n,m) Integer matrix $\mathbf{A}_{n \times m}$ to print.
in INT n m Dimensions n and m .
in CHA label*6 Label to print.

Subroutine MATPRI prints an integer matrix \mathbf{A} preceded by a label. The values are printed in 20I5 format with row and column indices aside and above. If $m > 20$ the rows are split in lines of at most twenty values. See also PRIIVC and PRIMAT.

file.f

```

      INTEGER KP(3,4)
      CALL MATPRI( KP, 3, 4, 'KP-MAT' )

```

file.out

```

KP-MAT    1    2    3    4
  1   111 -222  333  444
  2   111 -222  333  444
  3   111 -222  333  444

```


Print logical value*Fortran*

 CALL PRILVL(val, label)

in LOG val Logical value v to print.
 in CHA label*6 Label to print.

Subroutine PRILVL prints a logical value v preceded by a label. The value is printed in L3 format. See also PRIVAL, PRIIVL and PRILVC.

file.f

```

LOGICAL DONE
CALL PRILVL( DONE, 'DONEIT' )

```

file.out

```

DONEIT=  T

```

Print logical vector*Fortran*

 CALL PRILVC(v, n, label)

in LOG v(n) Logical vector \mathbf{v}_n to print.
 in INT n Dimension n .
 in CHA label*6 Label to print.

Subroutine PRILVC prints a logical vector \mathbf{v} preceded by a label. The values are printed in 40L3 format. If $n > 40$ the vector is split in lines of at most forty values. See also PRIVEC and PRIIVL.

file.f

```

LOGICAL FL(6)
CALL PRILVC( FL, 6, 'FLAGS ' )

```

file.out

```

FLAGS :  T  F  F  T  T  F

```

A.3.3.2 Initializing

This library (UNIX: liblbcx40.so; MS-Windows: lbcx40.lib) describes subroutines to initialize various types of arrays and values. Especially for large arrays, these subroutines use less processing time than ordinary Fortran DO-loops.

Initialize real array*Fortran*

 CALL RSET(s, u, n)

in DBL s Value s .
 out DBL u(n) Initialized array $\mathbf{u}(n)$.
 in INT n Number of array elements n to be initialized.

Subroutine RSET initializes the first n elements of real array \mathbf{u} to the value s . If $n \leq 0$ the call to RSET has no effect. See also ISET and LSET.

Initialize integer array*Fortran*

 CALL ISET(*s*, *u*, *n*)

in INT *s* Value *s*.
 out INT *u*(*n*) Initialized array *u*(*n*).
 in INT *n* Number of array elements *n* to be initialized.

Subroutine ISET initializes the first *n* elements of integer array *u* to the value *s*. If $n \leq 0$ the call to ISET has no effect. See also RSET and LSET.

Initialize logical array*Fortran*

 CALL LSET(*s*, *u*, *n*)

in LOG *s* Value *s*.
 out LOG *u*(*n*) Initialized array *u*(*n*).
 in INT *n* Number of array elements *n* to be initialized.

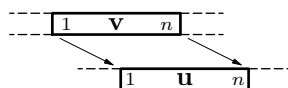
Subroutine LSET initializes the first *n* elements of logical array *u* to the value *s*. If $n \leq 0$ the call to LSET has no effect. See also RSET and ISET.

Move (copy) real array*Fortran*

 CALL RMOVE(*v*, *u*, *n*)

in DBL *v*(*n*) Array *v*(*n*) to be moved.
 out DBL *u*(*n*) Array *u*(*n*) to receive.
 in INT *n* Number of array elements *n* to be moved.

Subroutine RMOVE copies the first *n* elements of a real array *v* to the first *n* elements of a real array *u*:



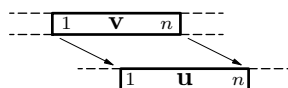
The input array *v* is left unchanged. If the arrays *v* and *u* overlap, the result is undefined. See also IMOVE and LMOVE.

Move (copy) integer array*Fortran*

 CALL IMOVE(*v*, *u*, *n*)

in INT *v*(*n*) Array *v*(*n*) to be moved.
 out INT *u*(*n*) Array *u*(*n*) to receive.
 in INT *n* Number of array elements *n* to be moved.

Subroutine IMOVE copies the first *n* elements of an integer array *v* to the first *n* values of an integer array *u*:



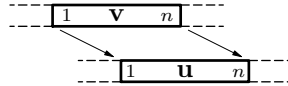
Routine IMOVE copies the first *n* elements of an integer array *v* to the first *n* values of an integer array *u*. The input array *v* is left unchanged. If the arrays *v* and *u* overlap, the result is undefined. See also RMOVE and LMOVE.

Move (copy) logical array*Fortran*

 CALL LMOVE(v, u, n)

in LOG v(n) Array **v**(*n*) to be moved.
 out LOG u(n) Array **u**(*n*) to receive.
 in INT n Number of array elements *n* to be moved.

Subroutine LMOVE copies the first *n* elements of a logical array **v** to the first *n* elements of a logical array **u**:



The input array **v** is left unchanged. If the arrays **v** and **u** overlap, the result is undefined. See also RMOVE and IMOVE.

Random real*Fortran*

 CALL DSRAND(izeed)
 ranvl = DRRAND()

in INT izeed Initializer.
 out DBL ranvl Random value.

Subroutine DRRAND returns a random real value in the range 0 to 1. To get a reproducible series of random values, the function value must be initialized by a call to DSRAND.

file.f

```
DOUBLE PRECISION DRRAND, VAL(5)
INTEGER I
CALL DSRAND( 12345 )
DO 10, I = 1, 5
    VAL(I) = DRRAND( )
    VI = VAL(I)
10 CONTINUE
CALL PRIVEC( VAL, 5, 'RANVL ' )
END
```

file.out

```
RANVL :   1.333E-01   3.599E-01   8.986E-01   7.494E-01   9.123E-01
```

Get machine parameters*Fortran*

 par = DPMPAR(kod)

in DBL kod Indicator.
 out DBL par Machine parameter.

Subroutine DPMPAR provides double precision machine parameters. It returns a machine parameter depending on the value of **kod**: the machine precision if **kod** = 1, the lowest double precision value if **kod** = 2 or the highest double precision value if **kod** = 3.

file.f

```
DOUBLE PRECISION DPMPAR
CALL PRIVAL( DPMPAR( 1 ), 'DBLEPS' )
CALL PRIVAL( DPMPAR( 2 ), 'DBLMIN' )
CALL PRIVAL( DPMPAR( 3 ), 'DBLMAX' )
```

file.out

```
DBLEPS=  2.500E-16
DBLMIN=  5.000-308
DBLMAX=  1.500+308
```

A.3.3.3 Interpolation

This library (UNIX: `liblbgs30.so`; MS-Windows: `lbgs30.lib`) describes subroutines to interpolation of piecewise linear diagrams.

One-dimensional interpolation

Fortran

```
CALL INTER1( x, vx, wz, l, z, dzdx )
```

in	DBL x	x -coordinate.
in	DBL vx(1)	Array $\mathbf{v}(l)$ containing the x -values.
in	DBL wz(1)	Array $\mathbf{w}(l)$ containing the z -values.
out	DBL z	Interpolated value $z = f(x)$.
out	DBL dzdx	Gradient $df(x)/dx$.
in	INT l	Dimension l .

Subroutine INTER1 interpolates in a piecewise linear one-dimensional diagram. See also INTER2.

Two-dimensional interpolation

Fortran

```
CALL INTER2( x, y, ux, vy, wz, l, m, z, grad )
```

in	DBL x	x -coordinate.
in	DBL y	y -coordinate.
in	DBL ux(1)	Array $\mathbf{v}(l)$ containing the x -values.
in	DBL uy(m)	Array $\mathbf{v}(m)$ containing the y -values.
in	DBL wz(1,m)	Array $\mathbf{w}(l \times m)$ containing the z -values.
out	DBL z	Interpolated value $z = f(x, y)$.
out	DBL grad(2)	Gradient ∇ .
in	INT l m	Dimensions l and m .

Subroutine INTER2 interpolates in a piecewise linear two-dimensional diagram. See also INTER1.

A.3.3.4 Sorting

This library (UNIX: `liblbcx40.so`; MS-Windows: `lbcx40.lib`) describes subroutines to sort arrays.

Sort real array

Fortran

```
CALL SORTR( kod, u, n )
```

in	INT kod	Indicates sort direction.
in	DBL u(n)	Array $\mathbf{u}(n)$ to sort.
in	INT n	Dimension n .
out	DBL u(n)	Sorted array.

Subroutine SORTR sorts the first n elements of array \mathbf{u} . Argument `kod` indicates the sort direction: descending (high to low) if `kod = -1`, ascending (low to high) if `kod + 1`. See also SORTRI and SORTI.

file.f

```

DOUBLE PRECISION VALUE(4)
DATA VALUE / 0.6300D0, 0.1240D3, -0.4200D0, 0.1000D2 /
CALL SORTR( +1, VALUE, 4 )
CALL PRIVEC( VALUE, 4, 'VL-ASC' )
CALL SORTR( -1, VALUE, 4 )
CALL PRIVEC( VALUE, 4, 'VL-DES' )

```

file.out

```

VL-ASC:  -4.200E-01   6.300E-01   1.000E+01   1.240E+02
VL-DES:   1.240E+02   1.000E+01   6.300E-01  -4.200E-01

```

Sort real array with indices*Fortran*

```
CALL SORTRI( kod, u, n, ind )
```

in	INT	kod	Indicates sort direction.
in	DBL	u(n)	Array $u(n)$ to sort.
in	INT	n	Dimension n .
out	DBL	u(n)	Sorted array.
out	INT	ind(n)	Original positions in $u(n)$.

Subroutine **SORTRI** sorts the first n elements of array **u**. Argument **kod** indicates the sort direction: descending (high to low) if **kod** = -1, ascending (low to high) if **kod** + 1. The original index of each sorted array element is returned in **ind** such that **ind(i)** contains the original index in **u** of **u(i)**. See also **SORTR** and **SORTII**.

file.f

```

DOUBLE PRECISION VALUE(4)
INTEGER IND(4)
DATA VALUE / 0.6300D0, 0.1240D3, -0.4200D0, 0.1000D2 /
CALL PRIVEC( VALUE, 4, 'VL-ORG' )
CALL SORTRI( +1, VALUE, 4, IND )
CALL PRIVEC( VALUE, 4, 'VL-ASC' )
CALL PRIIVC( IND, 4, 'INDEX ' )

```

file.out

```

VL-ORG:   6.300E-01   1.240E+02  -4.200E-01   1.000E+01
VL-ASC:  -4.200E-01   6.300E-01   1.000E+01   1.240E+02
INDEX :     3       1       4       2

```

Sort integer array*Fortran*

```
CALL SORTI( kod, u, n )
```

in	INT	kod	Indicates sort direction.
in	INT	u(n)	Array $u(n)$ to sort.
in	INT	n	Dimension n .
out	INT	u(n)	Sorted array.

Subroutine **SORTI** sorts the first n elements of array **u**. Argument **kod** indicates the sort direction: descending (high to low) if **kod** = -1, ascending (low to high) if **kod** + 1. See also **SORTII** and **SORTR**.

file.f

```

INTEGER VALUE(4)
DATA VALUE / 63, 124, -42, 10 /
CALL SORTI( +1, VALUE, 4 )
CALL PRIIVC( VALUE, 4, 'VL-ASC' )
CALL SORTI( -1, VALUE, 4 )
CALL PRIIVC( VALUE, 4, 'VL-DES' )

```

file.out

```

VL-ASC:  -42   10   63   124
VL-DES:  124   63   10   -42

```

Sort integer array with indices*Fortran*

```
CALL SORTII( kod, u, n, ind )
```

in	INT	kod	Indicates sort direction.
in	INT	u(n)	Array $u(n)$ to sort.
in	INT	n	Dimension n .
out	INT	u(n)	Sorted array.
out	INT	ind(n)	Original positions in $u(n)$.

Subroutine SORTII sorts the first n elements of array u . Argument kod indicates the sort direction: descending (high to low) if $kod = -1$, ascending (low to high) if $kod + 1$. The original index of each sorted array element is returned in ind such that $ind(i)$ contains the original index in u of $u(i)$. See also SORTI and SORTRI.

file.f

```

INTEGER VALUE(4), IND(4)
DATA VALUE / 63, 124, -42, 10 /
CALL PRIIVC( VALUE, 4, 'VL-ORG' )
CALL SORTII( +1, VALUE, 4, IND )
CALL PRIIVC( VALUE, 4, 'VL-ASC' )
CALL PRIIVC( IND, 4, 'INDEX ' )

```

file.out

```

VL-ORG:  63  124  -42  10
VL-ASC:  -42  10   63  124
INDEX :   3   4   1   2

```

A.3.3.5 Error Handling

This library (UNIX: liblber50.so; MS-Windows: lber50.lib) describes a subroutine for DIANA's error handling.

If the control of execution runs in an erroneous situation, for instance in the ELSE branch of an IF-THEN-ELSE structure, it is convenient and good programming practice to call DIANA's error handling subroutine PRGERR instead of a Fortran STOP statement.

Abort in error situation*Fortran*

```
CALL PRGERR( usnam, errnr )
```

in	CHA	usnam	Subroutine name.
in	INT	errnr	Error number.

Subroutine PRGERR writes the subroutine name and the error number on DIANA's standard output file, closes the FILOS file and aborts the executing of the current job.

file.f

```

SUBROUTINE USRMAT( ... )
  INTEGER NS
C...
  IF ( NS .EQ. 3 ) THEN
C...    do something
  ELSE
    PRINT *, 'NUMBER OF STRESS COMPONENTS NOT CORRECT'
    CALL PRGERR( 'USRMAT', 27 )
  END IF

```

file.out

```

NUMBER OF STRESS COMPONENTS NOT CORRECT

SEVERITY   : ABORT
ERROR CODE: /DIANA/LB/GS30/0014
ERRORMSG.A: Severe program error detected in USRMAT, reference number 27.
Contact your DIANA support.
DIANA-JOB ABORTED

```

A.3.4 Database I/O

DIANA's Database library (UNIX: `liblbfl44.so`; MS-Windows: `lbfl44.lib`) offers a set of subroutines to access the data in the FILOS file, for instance getting data records from the various Data Structures.

A.3.4.1 Access

Get record, check length

Fortran

```

CALL GTC( path, gval, len )

in  CHA path*(*)    Pathname.
out GEN gval(len)   Data values.
in  INT len         Expected length of data item.

```

Subroutine `GTC` checks the length of data item `path` and reads the data if the expected length is equal to the actual length. If the expected length is not equal to the actual length on the database, a fatal error message occurs. The type of output array `gval` is generic: it depends on the type of the data.

file.f

```

DOUBLE PRECISION YOUNG, B(3,8)
INTEGER          USRIND(2)
CALL GTC( '../MATERI/YOUNG', YOUNG, 1 )
CALL GTC( 'B', B, 24 )
CALL GTC( 'USRIND', USRIND, 2 )

```


Appendix B

Available Element Types

This appendix is an alphabetically ordered list of all elements available in DIANA. See Volume *Element Library* for a comprehensive description of these elements, including input data and background theory.

B2AGW	Axisymmetric groundwater flow, boundary line, 2 nodes, linear.		nodes, quadratic.
B2AHT	Axisymmetric potential flow, boundary line, 2 nodes, linear.	BT3GW	Groundwater flow, boundary triangle, 3 nodes, linear.
B2GW	Groundwater flow, boundary line, 2 nodes, linear.	BT3HT	Potential flow, boundary triangle, 3 nodes, linear.
B2HT	Potential flow, boundary line, 2 nodes, linear.	BT9S3	Fluid–structure triangular interface, 6 nodes, linear.
BC3AG	Axisymmetric groundwater flow, boundary line, 3 nodes, quadratic.	CHX20G	Groundwater flow, 3-D, brick, 20 nodes, quadratic.
BC3AHT	Axisymmetric potential flow, boundary line, 3 nodes, quadratic.	CHX20H	Potential flow, 3-D, brick, 20 nodes, quadratic.
BC3GW	Groundwater flow, boundary line, 3 nodes, quadratic.	CHX60	Solid brick, 20 nodes, quadratic.
BC3HT	Potential flow, boundary line, 3 nodes, quadratic.	CHX64	Solid brick, 20 nodes, quadratic, hyperelastic.
BCL6S2	Fluid–structure line interface, 5 nodes, quadratic-linear.	CHX96	Solid brick, 32 nodes, cubic.
BCL6S3	Fluid–structure line interface, 6 nodes, quadratic.	CL10T	Curved truss bar, 2-D, 5 nodes, quartic.
BCQ8GW	Groundwater flow, boundary quadrilateral, 8 nodes, quadratic.	CL12B	Curved beam, 2-D, 4 nodes, degenerated cubic.
BCQ8HT	Potential flow, boundary quadrilateral, 8 nodes, quadratic.	CL12I	Line interface, 2-D, 6 nodes, quadratic.
BCT6GW	Groundwater flow, boundary triangle, 6 nodes, quadratic.	CL12T	Curved truss bar, 3-D, 4 nodes, cubic.
BCT6HT	Potential flow, boundary triangle, 6 nodes, quadratic.	CL15B	Curved beam, 2-D, 5 nodes, degenerated quartic.
BL4S2	Fluid–structure line interface, 4 nodes, linear.	CL15T	Curved truss bar, 3-D, 5 nodes, quartic.
BQ12S4	Fluid–structure quadrilateral interface, 8 nodes, linear.	CL18B	Curved beam, 3 nodes, 3-D, quadratic.
BQ24S4	Fluid–structure quadrilateral interface, 12 nodes, quadratic-linear.	CL18I	Curved line interface, 6 nodes, quadratic, line–solid connection.
BQ24S8	Fluid–structure quadrilateral interface, 16 nodes, quadratic.	CL20I	Curved line interface, 10 nodes, quartic.
BQ4GW	Groundwater flow, boundary quadrilateral, 4 nodes, linear.	CL24B	Curved beam, 4 nodes, 3-D, cubic.
BQ4HT	Potential flow, boundary quadrilateral, 4 nodes, linear.	CL24I	Line interface, to shell, 6 nodes, quadratic.
BT18S3	Fluid–structure triangular interface, 9 nodes, quadratic-linear.	CL30B	Curved beam, 5 nodes, 3-D, quartic.
BT18S6	Fluid–structure triangular interface, 12 nodes, quadratic.	CL32I	Line interface, to shell, 8 nodes, cubic.
		CL3CM	Curved composed line, 3 nodes.
		CL3CR	Crack tip, 3-D, 3 nodes.
		CL6CT	Line contact interface, 2-D, 3 nodes.
		CL6TB	Line bounding, 3 nodes, quadratic, 2-D.
		CL6TM	Curved line mass, 2-D, 3 nodes, quadratic.
		CL6TR	Curved truss bar, 2-D, 3 nodes, quadratic.
		CL8TR	Curved truss bar, 2-D, 4 nodes, cubic.

CL9AX	Axisymmetric shell, 3 nodes, quadratic.	CQ60S	Quadrilateral curved shell, 12 nodes, cubic.
CL9BE	Curved beam, 3 nodes, 2-D, quadratic.	CQ8AG	Axisymmetric groundwater flow, quadrilateral, 8 nodes, quadratic.
CL9PE	Infinite plane strain shell, 3 nodes, quadratic.	CQ8AHT	Axisymmetric potential flow, quadrilateral, 8 nodes, quadratic.
CL9TM	Curved line mass, 3-D, 3 nodes, quadratic.	CQ8CM	Quadrilateral composed surface, 8 nodes.
CL9TR	Curved truss bar, 3-D, 3 nodes, quadratic.	CQ8GW	Groundwater flow, quadrilateral, 8 nodes, quadratic.
CPY13G	Groundwater flow, 3-D, pyramid, 13 nodes, quadratic.	CQ8HT	Potential flow, quadrilateral, 8 nodes, quadratic.
CPY13H	Potential flow, 3-D, pyramid, 13 nodes, quadratic.	CQ8KD	Layered groundwater flow, quadrilateral, 8 nodes, quadratic.
CPY39	Solid pyramid, 13 nodes, quadratic.	CQ8RE	Reynolds flow, quadrilateral, 8 nodes, quadratic.
CQ12C	Quadrilateral composed surface, 12 nodes.	CQ8TO	Cross-section torsion, quadrilateral, 8 nodes, quadratic.
CQ16A	Quadrilateral axisymmetric, 8 nodes, quadratic.	CT12A	Triangular axisymmetric, 6 nodes, quadratic.
CQ16E	Quadrilateral plane strain, 8 nodes, quadratic.	CT12E	Triangular plane strain, 6 nodes, quadratic.
CQ16M	Quadrilateral plane stress, 8 nodes, quadratic.	CT12M	Triangular plane stress, 6 nodes, quadratic.
CQ16O	Quadrilateral plane stress, 8 nodes, quadratic, orthotropic.	CT12O	Triangular plane stress, 6 nodes, quadratic, orthotropic.
CQ18M	Quadrilateral plane stress, 9 nodes, quadratic, Lagrange.	CT18C	Triangular contact interface, 3-D, 6 nodes.
CQ20A	Quadrilateral axisymmetric, 8 nodes, quadratic, hyperelastic.	CT18GE	Triangular complete plane strain, 6 nodes, quadratic.
CQ20E	Quadrilateral plane strain, 8 nodes, quadratic, hyperelastic.	CT18GM	Triangular plane stress, 3-D, 6 nodes, quadratic.
CQ22A	Quadrilateral axisymmetric, 9 nodes, quadratic, hyperelastic.	CT18P	Triangular plate bending, 6 nodes, quadratic, Mindlin.
CQ22E	Quadrilateral plane strain, 9 nodes, quadratic, hyperelastic.	CT18T	Triangular bounding, 6 nodes, quadratic, 3-D.
CQ24C	Quadrilateral contact interface, 3-D, 8 nodes.	CT18TM	Triangular curved plane mass, 3-D, 6 nodes, quadratic.
CQ24GE	Quadrilateral complete plane strain, 8 nodes, quadratic.	CT27GE	Triangular complete plane strain, 9 nodes, cubic.
CQ24GM	Quadrilateral plane stress, 3-D, 8 nodes, quadratic.	CT27T	Triangular bounding, 9 nodes, cubic, 3-D.
CQ24P	Quadrilateral plate bending, 8 nodes, quadratic, Mindlin.	CT30A	Triangular axisymmetric, 15 nodes, quartic, Lagrange.
CQ24T	Quadrilateral bounding, 8 nodes, quadratic, 3-D.	CT30E	Triangular plane strain, 15 nodes, quartic, Lagrange, hyperelastic.
CQ24TM	Quadrilateral curved plane mass, 3-D, 8 nodes, quadratic.	CT30F	Triangular flat shell, 6 nodes, quadratic, Mindlin.
CQ36GE	Quadrilateral complete plane strain, 12 nodes, cubic.	CT30L	Triangular curved shell, 6 nodes, quadratic, layered.
CQ36T	Quadrilateral bounding, 12 nodes, cubic, 3-D.	CT30S	Triangular curved shell, 6 nodes, quadratic.
CQ40F	Quadrilateral flat shell, 8 nodes, quadratic, Mindlin.	CT36F	Triangular flat shell, 6 nodes, quadratic, Mindlin + ϕ_z d.o.f.
CQ40L	Quadrilateral curved shell, 8 nodes, quadratic, layered.	CT36I	Triangular interface, 3-D, 12 nodes, quadratic.
CQ40S	Quadrilateral curved shell, 8 nodes, quadratic.	CT36L	Triangular curved shell, 6 nodes, quadratic, layered, Mindlin + ϕ_z d.o.f.
CQ48F	Quadrilateral flat shell, 8 nodes, quadratic, Mindlin + ϕ_z d.o.f.	CT36S	Triangular curved shell, 6 nodes, quadratic, Mindlin + ϕ_z d.o.f.
CQ48I	Quadrilateral interface, 3-D, 16 nodes, quadratic.	CT45S	Triangular curved shell, 9 nodes, cubic.
CQ48L	Quadrilateral curved shell, 8 nodes, quadratic, layered, Mindlin + ϕ_z d.o.f.	CT6AG	Axisymmetric groundwater flow, triangle, 6 nodes, quadratic.
CQ48S	Quadrilateral curved shell, 8 nodes, quadratic, Mindlin + ϕ_z d.o.f.	CT6AHT	Axisymmetric potential flow, triangle, 6 nodes, quadratic.

CT6CM	Triangular composed surface, 6 nodes.		nonlinear.
CT6GW	Groundwater flow, triangle, 6 nodes, quadratic.	L6AXI	Axisymmetric shell, 2 nodes, linear.
CT6HT	Potential flow, triangle, 6 nodes, quadratic.	L6BEA	Straight beam, 2 nodes, 2-D, linear.
CT6KD	Layered groundwater flow, triangle, 6 nodes, quadratic.	L6BEN	Bending beam, 2 nodes, 2-D, Timoshenko or Bernoulli.
CT6RE	Reynolds flow, triangle, 6 nodes, quadratic.	L6PE	Infinite plane strain shell, 2 nodes, linear.
CT6TO	Cross-section torsion, triangle, 6 nodes, quadratic.	L6TM	Line mass, 3-D, 2 nodes, linear.
CT9CM	Triangular composed surface, 9 nodes.	L6TRU	Truss bar, 2 nodes, 3-D geometrically nonlinear.
CTE10G	Groundwater flow, 3-D, tetrahedron, 10 nodes, quadratic.	L7BEN	Bending beam, 2 nodes, 2-D, isoparametric.
CTE10H	Potential flow, 3-D, tetrahedron, 10 nodes, quadratic.	L8IF	Line interface, 2-D, 4 nodes, linear.
CTE30	Solid tetrahedron, 10 nodes, quadratic.	N4IF	Node interface, 2-D, 2 nodes, linear.
CTE48	Solid tetrahedron, 16 nodes, cubic.	N6IF	Node interface, 3-D, 2 nodes, linear.
CTP15G	Groundwater flow, 3-D, wedge, 15 nodes, quadratic.	N6SPR	Matrix spring, 3-D, 1 node.
CTP15H	Potential flow, 3-D, triangular prism (wedge), 15 nodes, quadratic.	PT1CR	Crack tip, 2-D, 1 node.
CTP45	Solid wedge, 15 nodes, quadratic.	PT3RO	Point mass, rotation, 1 node.
CTP72	Solid wedge, 24 nodes, cubic.	PT3T	Point mass, translation, 1 node.
HX24L	Solid brick, 8 nodes, linear.	PY15L	Solid pyramid, 5 nodes, linear.
HX25L	Solid brick, 8 nodes, linear, hyperelastic.	PY5GW	Groundwater flow, 3-D, pyramid, 5 nodes, linear.
HX8GW	Groundwater flow, 3-D, brick, 8 nodes, linear.	PY5HT	Potential flow, 3-D, pyramid, 5 nodes, linear.
HX8HT	Potential flow, 3-D, brick, 8 nodes, linear.	Q12CT	Quadrilateral contact interface, 3-D, 4 nodes.
ICL6H	Potential flow, line interface, 6 nodes, quadratic.	Q12GME	Quadrilateral plane stress, 3-D, 4 nodes, linear.
ICQ16H	Potential flow, quadrilateral interface, 16 nodes, quadratic.	Q12ME	Quadrilateral plane stress, 4 nodes, linear, drilling d.o.f.
ICT12H	Potential flow, triangular interface, 12 nodes, quadratic.	Q12PL	Quadrilateral plate bending, 4 nodes, linear, Mindlin.
IL4HT	Potential flow, line interface, 4 nodes, linear.	Q12TB	Quadrilateral bounding, 4 nodes, linear, 3-D.
IPT2H	Potential flow, point interface, 2 nodes.	Q12TM	Quadrilateral plane mass, 3-D, 4 nodes, linear.
IQ8HT	Potential flow, quadrilateral interface, 8 nodes, linear.	Q20SF	Quadrilateral flat shell, 4 nodes, linear, Mindlin.
IT6HT	Potential flow, triangular interface, 6 nodes, malinear.	Q20SH	Quadrilateral curved shell, 4 nodes, linear.
L12BE	Bending beam, 2 nodes, 3-D, Timoshenko or Bernoulli.	Q24IF	Quadrilateral interface, 3-D, 8 nodes, linear.
L12BEA	Straight beam, 2 nodes, 3-D, linear.	Q24SF	Quadrilateral flat shell, 4 nodes, linear, Mindlin + ϕ_z d.o.f.
L12IF	Line interface, 4 nodes, linear, line-solid connection.	Q24SH	Quadrilateral curved shell, 4 nodes, linear, Mindlin + ϕ_z d.o.f.
L13BE	Bending beam, 2 nodes, 3-D, isoparametric.	Q48SPL	Rectangular spline (strip), 8 nodes, 3 sections.
L16IF	Line interface, to shell, 4 nodes, linear.	Q4AGW	Axisymmetric groundwater flow, quadrilateral, 4 nodes, linear.
L20IF	Line interface, to shell, 3+2 nodes, quadratic/linear.	Q4AHT	Axisymmetric potential flow, quadrilateral, 4 nodes, linear.
L2CMP	Straight composed line, 2 nodes.	Q4CMP	Quadrilateral composed surface, 4 nodes.
L2HT	Cooling pipe, 2 nodes, linear.	Q4GW	Groundwater flow, quadrilateral, 4 nodes, linear.
L2TRU	Truss bar, 1-D, 2 nodes.	Q4HT	Potential flow, quadrilateral, 4 nodes, linear.
L4CT	Line contact interface, 2-D, 2 nodes.	Q4KD	Layered groundwater flow, quadrilateral, 4 nodes, linear.
L4HT	Cooling pipe, 4 nodes, linear, nonsymmetric.	Q4RE	Reynolds flow, quadrilateral, 4 nodes, linear.
L4TB	Line bounding, 2 nodes, linear, 2-D.	Q4TO	Cross-section torsion, quadrilateral, 4 nodes, linear.
L4TM	Line mass, 2-D, 2 nodes, linear.		
L4TRU	Truss bar, 2 nodes, 2-D geometrically		

Q56SPL	Rectangular spline (strip), 10 nodes, 4 sections.	T3HT	Potential flow, triangle, 3 nodes, linear.
Q8AXI	Quadrilateral axisymmetric, 4 nodes, linear.	T3KD	Layered groundwater flow, triangle, 3 nodes, linear.
Q8EPS	Quadrilateral plane strain, 4 nodes, linear.	T3RE	Reynolds flow, triangle, 3 nodes, linear.
Q8MEM	Quadrilateral plane stress, 4 nodes, linear.	T3TO	Cross-section torsion, triangle, 3 nodes, linear.
Q8OME	Quadrilateral plane stress, 4 nodes, linear, orthotropic geometry.	T6AXI	Triangular axisymmetric, 3 nodes, linear.
SP12BA	Base spring, 2 nodes, 3-D.	T6EPS	Triangular plane strain, 3 nodes, linear.
SP1RO	Rotation spring/dashpot, 1 node.	T6MEM	Triangular plane stress, 3 nodes, linear.
SP1TR	Translation spring/dashpot, 1 node.	T6OME	Triangular plane stress, 3 nodes, linear, orthotropic geometry.
SP2RO	Rotation spring/dashpot, 2 nodes.	T9CT	Triangular contact interface, 3-D, 3 nodes.
SP2TR	Translation spring/dashpot, 2 nodes.	T9GME	Triangular plane stress, 3-D, 3 nodes, linear.
SP6BA	Base spring, 2 nodes, 2-D.	T9MEM	Triangular plane stress, 3 nodes, linear, drilling d.o.f.
T15SF	Triangular flat shell, 3 nodes, linear, Mindlin.	T9PLA	Triangular plate bending, 3 nodes, linear, Kirchhoff.
T15SH	Triangular curved shell, 3 nodes, linear.	T9TB	Triangular bounding, 3 nodes, linear, 3-D.
T18FSH	Triangular flat shell, 3 nodes, Kirchhoff, analytically integrated.	T9TM	Triangular plane mass, 3-D, 3 nodes, linear.
T18IF	Triangular interface, 3-D, 6 nodes, linear.	TE12L	Solid tetrahedron, 4 nodes, linear.
T18SF	Triangular flat shell, 3 nodes, linear, Mindlin + ϕ_z d.o.f.	TE4GW	Groundwater flow, 3-D, tetrahedron, 4 nodes, linear.
T18SH	Triangular curved shell, 3 nodes, linear, Mindlin + ϕ_z d.o.f.	TE4HT	Potential flow, 3-D, tetrahedron, 4 nodes, linear.
T3AGW	Axisymmetric groundwater flow, triangle, 3 nodes, linear.	TP18L	Solid wedge, 6 nodes, linear.
T3AHT	Axisymmetric potential flow, triangle, 3 nodes, linear.	TP6GW	Groundwater flow, 3-D, wedge, 6 nodes, linear.
T3CMP	Triangular composed surface, 3 nodes.	TP6HT	Potential flow, 3-D, wedge, 6 nodes, linear.
T3GW	Groundwater flow, triangle, 3 nodes, linear.		

Appendix C

Physical Properties Forms

This appendix describes the interactive input of physical properties for elements in the Design working environment of iDIANA. See Volume *iDIANA* for a description of the iDIANA interface. There are two ways of properties specification: via commands on the iDIANA command line or via so-called ‘forms’. In both cases, the basic choice is for the ‘aspect’ of the properties. This appendix only describes the input via forms with cross references to the appropriate section in this volume. See also Volume *iDIANA* for a general description of physical properties input via forms.

Depending on the model type that you specified with the **FEMGEN** command, DIANA offers you a selection out of the following aspects. You may activate an aspect by clicking on its tab.

Geometry to specify geometrical properties for various elements families [Vol. *Element Library*].

Mobile to specify physical properties required for mobile loads [§ C.1].

Wind and Water to specify physical properties required for wind and water loads in beam elements [Vol. *Element Library*].

External to read data from an external file [Vol. *Element Library*].

C.1 Mobile

The **Mobile** aspect enables the specification of physical properties required for mobile loads [§ 2.3.6 p. 36].

C.1.1 Dutch VBB Code

The Dutch VBB code concept enables the specification of geometrical parameters for mobile load according to the Dutch VBB code [§ 2.3.6.4 p. 40].

Default position		[§ 2.3.6.4 p. 40]
Width of carriage way w	BWIDTH w	[§ 2.3.6.4 p. 40]
Length for bump and reduction .. l	LOALEN l	[§ 2.3.6.4 p. 42]
Load reduction factor B	LOARED b	[§ 2.3.6.4 p. 42]
Bump factor S	BUMP s	[§ 2.3.6.4 p. 42]
Transverse position		[§ 2.3.6.4 p. 41]
Width of carriage way w	BWIDTH w	[§ 2.3.6.4 p. 40]
Distance between beams b	TWOBEA b	[§ 2.3.6.4 p. 41]
Point i lorry influence line v_i	TRVINP v_i	[§ 2.3.6.4 p. 41]
Factor for lorry forces f_F	LODFAC ff	[§ 2.3.6.4 p. 41]
Factor for distributed load f_q	DISFAC f_q	[§ 2.3.6.4 p. 41]
Length for bump and reduction .. l	LOALEN l	[§ 2.3.6.4 p. 42]
Load reduction factor B	LOARED b	[§ 2.3.6.4 p. 42]
Bump factor S	BUMP s	[§ 2.3.6.4 p. 42]

C.1.2 Dutch VOSB Code

The Dutch VOSB code concept enables the specification of geometrical parameters for mobile load according to the Dutch VOSB code [§ 2.3.6.4 p. 40].

Default position			[§ 2.3.6.4 p. 40]
Width of carriage way	w	BWIDTH w	[§ 2.3.6.4 p. 40]
Width of the lanes	l	WLANE l	[§ 2.3.6.4 p. 40]
Length for bump and reduction ..	l	LOALEN l	[§ 2.3.6.4 p. 42]
Transverse position			[§ 2.3.6.4 p. 41]
Width of carriage way	w	BWIDTH w	[§ 2.3.6.4 p. 40]
Width of the lanes	l	WLANE l	[§ 2.3.6.4 p. 40]
Distance between beams	b	TWOBEA b	[§ 2.3.6.4 p. 41]
Factor for lorry forces	f_F	LODFAC ff	[§ 2.3.6.4 p. 41]
Factor for distributed load	f_q	DISFAC fq	[§ 2.3.6.4 p. 41]
Length for bump and reduction ..	l	LOALEN l	[§ 2.3.6.4 p. 42]

C.1.3 European ENV Code

The European ENV code concept enables the specification of geometrical parameters for mobile load according to the European ENV 1991-3 code [§ 2.3.6.5 p. 42].

European ENV code			
Width of carriage way	w	BWIDTH w	[§ 2.3.6.5 p. 44]
Factor for lorry forces	f_F	LODFAC ff	[§ 2.3.6.5 p. 44]
Factor for distributed load	f_q	DISFAC fq	[§ 2.3.6.5 p. 44]

C.1.4 General

The General concept enables the explicit specification of geometrical parameters for mobile load [§ 2.3.6.3 p. 38].

General			
Width of carriage way	w	BWIDTH w	[§ 2.3.6.3 p. 38]
Factor for lorry forces	f_F	LODFAC ff	[§ 2.3.6.3 p. 39]
Factor for distributed load	f_q	DISFAC fq	[§ 2.3.6.3 p. 39]

Bibliography

- [1] AKI, K. Generation and propagation of G waves from the Niigata earthquake of June 16, 1964: Part 2. Estimation of earthquake moment, released energy and stress drop from the G wave spectra. Tech. Rep. 44 (73-88), Bull Earthq. Res. Inst., Univ. of Tokyo, 1966.
- [2] AZZI, V. D., AND TSAI, S. W. Anisotropic Strength of Composite Materials. *Experimental Mechanics* 5, 9 (1965), 283–288.
- [3] BARSOUM, R. S. On the use of isoparametric finite elements in Linear Fracture Mechanics. *Int. J. Num. Meth. Eng.* 10 (1976), 25–37.
- [4] BATHE, K.-J. *Finite Element Procedures in Engineering Analysis*. Prentice-Hall, 1982.
- [5] BATHE, K.-J. *Finite Element Procedures*. Prentice-Hall, 1996.
- [6] BECKER, A. A. *Understanding Non-Linear Finite Element Analysis*. National Agency for Finite Element Methods & Standards (NAFEMS), Glasgow, 2001.
- [7] BETTES, P. B., AND ZIENKIEWICZ, O. C. Diffraction and refraction of surface waves using finite and infinite elements. *Int. J. Num. Meth. Eng.* 11 (1977), 1271–1290.
- [8] BIOT, M. A. General theory of three-dimensional consolidation. *J. Appl. Phys.* 12, 2 (1941), 155–164.
- [9] BOER, DE, A. *Design strategy structural concrete in 3D*. PhD thesis, Delft University of Technology, 2010.
- [10] BOOGAARD, VAN DEN, A. H. The Computation of J-integrals Using the Finite Element Method. Tech. Rep. BI-93-022, TNO Building and Construction Research, Rijswijk, The Netherlands, 1993.
- [11] BOOGAARD, VAN DEN, A. H. Arbitrary Lagrangian Eulerian Analysis in DIANA. Tech. Rep. 95-NM-R1002, TNO Building and Construction Research, Rijswijk, The Netherlands, 1995.
- [12] BUDIANSKY, B. Dynamic buckling of elastic structures: criteria and estimates. In *Dynamic Stability of Structures* (1965), Pergamon Press. Proceedings of International Conference, Northwestern University, Evanston, Illinois.
- [13] BUILDING SEISMIC SAFETY COUNCIL. NEHRP Guidelines for the Seismic Rehabilitation of Buildings. Tech. Rep. FEMA-273, Federal Emergency Management Agency, Washington DC, 1997.
- [14] BYSKOV, E., AND HUTCHINSON, J. W. Mode interaction in axially stiffened cylindrical shells. *AIAA Journal* 15, 7 (1977), 941–948.
- [15] CARDARELLI, F. *Scientific Unit Conversion*. No. ISBN 3-540-76022-9. Springer-Verlag, London, 1997.
- [16] CEB-FIP. *CEB-FIP Model Code 1990*. Comité Euro-International du Béton, 1993.

- [17] CEN. Eurocode 1: Basis of design and actions on structures – Part 3: Traffic loads on bridges. Tech. Rep. ENV 1991-3, European Committee for Standardization (CEN), 1995.
- [18] CHAN, T. F., AND MATHEW, T. P. Domain decomposition algorithms. *Acta Numerica* (1994), 61–143.
- [19] CHOPRA, A. K., AND GOEL, R. K. A modal pushover analysis procedure for estimating seismic demands for buildings. *Earthquake Eng. Struct. Dyn.* 31 (2002), 561–582.
- [20] CLOUGH, R. W., AND PENZIEN, J. *Dynamics of Structures*. McGraw-Hill, 1982.
- [21] CRAIG, R. *Structural Dynamics*. John Wiley & Sons, 1981.
- [22] CRISFIELD, M. A. *Non-linear Finite Element Analysis of Solids and Structures*, vol. 1: *Essentials*. John Wiley & Sons, 1991.
- [23] DARBRE, G. R. Seismic analysis of non-linearly base-isolated soil-structure interacting reactor building by way of the hybrid frequency-time domain procedure. *Earthquake Eng. Struct. Dyn.* 19 (1990), 725–738.
- [24] DARBRE, G. R. Nonlinear dam–reservoir interaction analysis. In *Proc. 11th World Conference on Earthquake Engineering, Acapulco* (1996), vol. 6. Paper No. 760.
- [25] DARBRE, G. R., AND WOLF, J. P. Criterion of stability and implementation issues of hybrid frequency-time domain procedure for nonlinear dynamic analysis. *Earthquake Eng. Struct. Dyn.* 16 (1988), 569–581.
- [26] DAWSON, E. M., ROTH, W. H., AND DRESCHER, A. Slope Stability Analysis by Finite Elements. *Geotechnique* 49, 3 (1999), 387–403.
- [27] ELLSIEPEN, P., AND HARTMANN, S. Remarks on the interpretation of current non-linear finite element analyses as differential–algebraic equations. *Int. J. Num. Meth. Eng.* 51 (2001), 679–707.
- [28] ERP, VAN, E. M. *Advanced Buckling Analysis of Beams with Arbitrary Cross-sections*. PhD thesis, Eindhoven University of Technology, 1989.
- [29] FENVES, G., AND CHOPRA, A. K. Earthquake Analysis and Response of Concrete Gravity Dams. Tech. Rep. UCBEERC-8410, University of California, Berkeley, 1984.
- [30] FENVES, G. L., AND CHAVEZ, J. W. Hybrid Frequency-Time Domain Analysis of Nonlinear Fluid-Structure Systems. In *Proc. Fourth U.S. Nat. Conf. on Earthquake Engineering* (1990), vol. 11, pp. 809–829.
- [31] FIB. *fib Model Code for Concrete Structures 2010*, 1st ed. fédération internationale du béton/International Federation for Structural Concrete, 2013.
- [32] FORDE, B. W. R., AND STIEMER, S. F. Improved Arc-length Orthogonality methods for nonlinear Finite Element Analysis. *Computers & Structures* 27, 5 (1987), 625–630.
- [33] GEORGE, A., AND LIU, J. W. *Computer Solution of Large Sparse Positive Definite Systems*. Prentice-Hall, 1981.
- [34] GÉRADIN, M., AND RIXEN, D. *Mechanical Vibrations – Theory and Application to Structural Dynamics*, 2nd ed. John Wiley & Sons, 1997.
- [35] GOODMAN, L. E., ROSENBLUETH, E., AND NEWMARK, N. M. Aseismic design of elastic structures founded on firm ground. *Proc. ASCE* (November 1953), 349/1–349/27.
- [36] GRIFFITHS, D. V., AND LANE, P. A. Slope Stability Analysis by Strength Reduction. *Geotechnique* 49, 6 (1999), 835–840.

- [37] GUPTA, A. K. *Response Spectrum Method In Seismic Analysis and Design of Structures*. CRC Press, Inc., 1992.
- [38] HAIRER, E., AND WANNER, G. *Solving Ordinary Differential Equations II*, 2nd ed. Springer-Verlag, 1996.
- [39] HELLEN, T. K. On the method of virtual crack extension. *Int. J. Num. Meth. Eng.* 9 (1975), 187–207.
- [40] HENDRIKS, M. A. N. *Identification of the Mechanical Behavior of Solid Materials*. PhD thesis, Eindhoven University of Technology, 1991.
- [41] HENSHELL, R. D., AND SHAW, K. G. Crack tip finite elements are unnecessary. *Int. J. Num. Meth. Eng.* 9 (1975), 495–507.
- [42] HESTENES, M. R., AND STIEFEL, E. Methods of conjugate gradients for solving linear systems. *J. Res. Nat. Bur. Standards* 49, 6 (1952), 409–436.
- [43] HILBER, H. M., HUGHES, T. J. R., AND TAYLOR, R. L. Improved numerical dissipation for time integration algorithms in structural dynamics. *Earthquake Engineering and Structural Dynamics* 5 (1977), 283–292.
- [44] HINTON, E., ROCK, T., AND ZIENKIEWICZ, O. C. A note on mass lumping and related processes in the Finite Element Method. *Earthquake Engineering and Structural Dynamics* 4 (1976), 245–249.
- [45] HORVATH, A. L. *Conversion Tables of Units for Science and Engineering*. No. ISBN 0-333-40857-8. Macmillan Reference Books, London, 1987.
- [46] HUÉTINK, J. *On the Simulation of Thermo-mechanical Forming Processes*. PhD thesis, University of Twente, 1986.
- [47] HUGHES, T. J. R. *The Finite Element Method – Linear Static and Dynamic Finite Element Analysis*. Prentice-Hall, 1987.
- [48] JSCE. JSCE Guidelines for Concrete No. 15: Standard Specifications for Concrete Structures - 2007 "Design". Tech. rep., Japan Society of Civil Engineers, 2010.
- [49] JSCE. Standard Specifications for Concrete Structures - 2012 "Design". Tech. rep., Japan Society of Civil Engineers, 2012.
- [50] KARYPIS, G., AND KUMAR, V. A fast and highly quality multilevel scheme for partitioning irregular graphs. *SIAM J. Sc. Comp.* 20, 1 (1999).
- [51] KERNIGHAN, B. W., AND PLAUGER, P. J. *The Elements of Programming Style*, 2nd ed. McGraw-Hill, 1978.
- [52] KOITER, W. T. *Over de Stabiliteit van Elastisch Evenwicht*. PhD thesis, Delft University of Technology, 1945.
- [53] KÜÇÜKARSLAN, S., COŞKUN, S. B., AND TAŞKIN, B. Transient analysis of dam-reservoir interaction including the reservoir bottom effects. *Journal of Fluids and Structures* 20 (2005), 1073–1084.
- [54] LAMBE, T. W., AND WHITMAN, R. V. *Soil Mechanics*. John Wiley & Sons, 1969.
- [55] LEHOUCQ, R. B., SORENSEN, D. C., AND YANG, C. *ARPACK Users' Guide: Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods*. SIAM Series: Software, Environments, and Tools. SIAM: Society for Industrial and Applied Mathematics, 1998. <http://www.caam.rice.edu/software/ARPACK/UG/ug.html>.
- [56] LI, XIKUI, AND ZIENKIEWICZ, O. C. Multiphase flow in deforming porous media and finite element solutions. *Computers & Structures* 45, 2 (1992), 211–227.

- [57] LINGEN, F., BONNIER, P., BRINKGREVE, R., GIJZEN, VAN, M., AND VUIK, C. A parallel linear solver exploiting the physical properties of the underlying mechanical problem. Report 12-12, Delft University of Technology, Delft Institute of Applied Mathematics, 2012.
- [58] MEIJERINK, J. A., AND VORST, VAN DER, H. A. An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix. *Math. Comp.* 31 (1977), 148–162.
- [59] MERKS, P. J. G. Wapeningsgrootheden bij Plaatconstructies. Tech. Rep. BSW 88-05, Dutch Public Works, Rijswijk, The Netherlands, 1988.
- [60] NAFEMS. *Guidelines to Finite Element Practice*. National Agency for Finite Element Methods & Standards (NAFEMS), Glasgow, 1984.
- [61] NAFEMS. The Standard NAFEMS Benchmarks. Tech. Rep. TNSB Rev. 3, National Agency for Finite Element Methods & Standards (NAFEMS), 1990.
- [62] NAFEMS. *A Finite Element Primer*. National Agency for Finite Element Methods & Standards (NAFEMS), Glasgow, 1992.
- [63] NEN. Concrete Bridges (VBB 1995) (Structural requirements and calculation methods). Tech. Rep. NEN 6723, Nederlands Normalisatie-instituut, 1995.
- [64] NEN. The Design of Steel Bridges (VOSB 1995) (Basic requirements and simple rules). Tech. Rep. NEN 6788, Nederlands Normalisatie-instituut, 1995.
- [65] NEN. Eurocode 8: Design of structures for earthquake resistance - Part 1: General rules, seismic actions and rules for buildings. Tech. Rep. NEN-EN 1998-1, Nederlands Normalisatie-instituut, 2005.
- [66] NEN. Eurocode 2: Design of concrete structures – Part 1-1: General rules and rules for buildings. Tech. Rep. NEN-EN 1992-1-1, Nederlands Normalisatie-instituut, 2007.
- [67] NEN. Eurocode 2: Design of concrete structures – Part 1-2: General rules - Structural fire design. Tech. Rep. NEN-EN 1992-1-2, Nederlands Normalisatie-instituut, 2007.
- [68] NEN. Nederlandse praktijkrichtlijn: Beoordeling van de constructieve veiligheid van een gebouw bij nieuwbouw, verbouw en afkeuren – Grondslagen voor aardbevingsbelastingen: Geïnduceerde aardbevingen (Assessment of buildings in case of erection, reconstruction, disapproval – Basic rules for seismic actions: induced earthquakes). Tech. Rep. NPR 9998:2015, Nederlands Normalisatie-instituut, 2015.
- [69] OLSON, L. G., AND BATHE, K.-J. Analysis of fluid–structure interactions. A direct symmetric coupled formulation based on the fluid velocity potential. *Computers & Structures* 21 (1985), 21–32.
- [70] PARKS, D. M. A stiffness derivative finite element technique for determination of crack tip stress intensity factors. *Int. J. of Fracture* 10, 4 (1974), 487–502.
- [71] POLIZZI, E. FEAST Eigenvalue Solver. <http://www.ecs.umass.edu/~polizzi/feast/index.htm>. Accessed: 2014/02/19.
- [72] RAHMAN, T. *A Perturbation Approach for Geometrically Nonlinear Structural Analysis Using a General Purpose Finite Element Code*. PhD thesis, Delft University of Technology, 2009.
- [73] REHFELD, L. Nonlinear vibration of elastic structures. *International Journal of Solids Structures* 9 (1973), 581–590.
- [74] RODDEMAN, D. G., AND JANSEN, L. F. An a priori geometry check for a single isoparametric finite element. *Computers & Structures* 47, 1 (1993), 69–72.

- [75] SAAD, Y. ILUT: a dual threshold incomplete ILU factorization. *Numerical Linear Algebra with Applications 1* (1994), 387–402.
- [76] SAAD, Y. *Iterative Methods for Sparse Linear Systems*, 2nd ed. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2003.
- [77] SAAD, Y., AND SCHULTZ, M. H. GMRES: a Generalized Minimal Residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.* 7, 3 (1986), 856–869.
- [78] SANGERS, A. Enhancing iterative solution methods for general FEM computations using rigid body modes. Master’s thesis, Delft University of Technology, 2014.
- [79] SCHENK, O. *Scalable Parallel Sparse LU Factorization Methods on Shared Memory Multiprocessors*. PhD thesis, Eidgenössische Technische Hochschule Zürich.
- [80] SCHENK, O., GARTNER, K., AND FICHTNER, W. Efficient Sparse LU Factorization with Left–right Looking Strategy on Shared Memory Multiprocessors. *BIT* 40, 1 (2000), 158–176.
- [81] SIRUMBAL ZAPATA, L. F. Numerical modeling of dam-reservoir interaction seismic response using the Hybrid Frequency-Time Domain (HFTD) method. Master’s thesis, Delft University of Technology, 2013.
- [82] SPENCE, P. W., AND KENCHINGTON, C. J. The Role of Damping in Finite Element Analysis. Tech. Rep. R0021, National Agency for Finite Element Methods & Standards (NAFEMS), Glasgow, 1993.
- [83] STRANG, G. *Linear Algebra and Its Applications*, 3rd ed. Harcourt Brace Jovanovich College Publishers, 1988.
- [84] TERZAGHI, K. V. *Theoretical Soil Mechanics*. John Wiley & Sons, 1943.
- [85] TISO, P. *Finite Element Based Reduction Methods for Static and Dynamic Analysis of Thin-Walled Structures*. PhD thesis, Delft University of Technology, 2006.
- [86] TSAI, S. W., AND WU, E. M. A general theory of strength for anisotropic materials. *Journal of Composite Materials* 5 (1971), 58–80.
- [87] VELETOS, A. S., AND VENTURA, C. E. Dynamic analysis of structures by the DFT method. *J. Struct. Eng., ASCE* 111 (1985), 2625–2642.
- [88] VERMEER, P. A., AND VERRUYT, A. An accuracy condition for consolidation by finite elements. *Int. J. Num. and Analytical Methods in Geomechanics* 5 (1981), 1–14.
- [89] WAN, R. G. Stress return solution algorithm for generalized Hoek-Brown plasticity model. In *Computer Methods and Advances in Geomechanics* (1994), H. J. Siriwardane and M. M. Zaman, Eds., Balkema.
- [90] YASUDA, S., YOSHIDA, N., KIKU, H., ADACHI, K., AND GOSE, S. A simplified method to evaluate liquefaction-induced deformation. In *Earthquake Geotechnical Engineering: Proceedings of the Second International Conference, Lisbon, Portugal, 21-25 June, 1999* (1999), P. S. Seco e Pinto, Ed., Balkema, Rotterdam, pp. 555–560.
- [91] ZHANG, Z. Introduction to the Intel MKL Extended Eigensolver. <http://software.intel.com/en-us/articles/introduction-to-the-intel-mkl-extended-eigensolver>. Accessed: 2014/02/19.
- [92] ZIENKIEWICZ, O. C., AND BETTES, P. B. Fluid–structure dynamic interaction and wave forces. An introduction to numerical treatment. *Int. J. Num. Meth. Eng.* 13 (1978), 1–16.

- [93] ZIENKIEWICZ, O. C., AND TAYLOR, R. L. *The Finite Element Method*, 4th ed., vol. 1: *Basic Formulations and Linear Problems*. McGraw-Hill, 1989.
- [94] ZIENKIEWICZ, O. C., AND TAYLOR, R. L. *The Finite Element Method*, 4th ed., vol. 2: *Solid and Fluid Mechanics, Dynamics and Non-linearity*. McGraw-Hill, 1991.

Index

Page numbers. Bold face numbers indicate pages with formal information about the entry, e.g., a syntax description (**36**). Italic numbers point to an instructive example of how the concept in question might be used (*132*). Underlined numbers refer to theoretical backgrounds on the subject (95).

Keywords. Sans serif type style refers to the interactive interface (EYE). Typewriter style refers to the batch interface (YOUNG).

A

- A output, frequency response analysis, **158**
- A output, response spectrum analysis, **176**
- AAB subroutine, **650**
- ABS option, **173**
- ABS rule, **169**, **173**, 591
- Absolute sum rule, **169**, 591
- ACCELE command, frequency response analysis, **157**
- ACCELE command, HFTD analysis, **197**
- ACCELE command, mixture analysis, **394**
- ACCELE command, modal response analysis, **148**
- ACCELE command, nonlinear analysis output, **247**
- ACCELE command, nonlinear stop criterion, **242**
- ACCELE command, response spectrum analysis, **176**
- ACCELE command, transient analysis output, **129**
- ACCELE input, pushover analysis, **106**
- ACCELE subtable of 'LOADS', **108**
- Acceleration of gravity, **3**, **4**, **6**
- Acceleration of gravity, staggered analysis, **382**
- Acceleration vector length, response spectrum analysis, **176**
- Acceleration vector length, transient analysis, **130**
- Acceleration, frequency response analysis, **157**
- Acceleration, prescribed, **107**
- Acceleration, relative, **58**
- Acceleration, response spectrum analysis, **175**
- Acceleration, transient analysis, **129**
- Accelerogram, HFTD analysis, **189**
- Accuracy, consolidation analysis, 636
- Accuracy, eigenvalue analysis, **437**, 545
- Accuracy, nonlinear potential flow analysis, 629
- ACTCRK output, **266**
- ACTIVE command, **405**, **417**
- Adaptive loading, 563
- Adaptive loading, cutback based, **227**, 565
- Adaptive loading, energy based, **226**, **289**, 564
- Adaptive loading, iteration based, **226**, 563
- Adaptive time increments, 563
- Adaptive time increments, cutback based, **233**, 565
- Adaptive time increments, iteration based, **232**, 563
- ADD parameter, initial conditions, **220**
- Aggregate size, stiffness adaptation analysis, **473**
- Aging, phased potential flow analysis, **418**
- ALLDIR input, **106**
- ALPHA command, CMOD Arc-length control, **230**
- ALPHA parameter, nonlinear analysis, **217**
- ALPHA parameter, transient groundwater flow, **358**
- ALPHA parameter, transient heat flow, **341**
- α -method, *see* Hilber–Hughes–Taylor
- Ambient influence, **110**
- AMPIMP command, **204**, **309**
- AMPLIT command, **204**, **309**
- AMPLIT input, **114**
- AMPLIT option, **153**
- Analysis commands, *see* Commands
- ANGLE input, **4**
- ANGLE parameter, **53**
- Angle units, **4**
- Anisotropy, mixture analysis, **378**
- Anisotropy, staggered analysis, **377**
- ANNEX command, **452**
- Append input, **402**
- APPEND option, tabular output, **64**
- APPEND option, input reading, **50**
- APPEND option, output file, **56**
- APPEND option, phased analysis, **402**
- APPEND option, postprocessing output, **67**
- Aquifer, **352**, **357**
- Ar output, frequency response analysis, **158**
- Arc-length control, **215**, **227**, 561
- ARCLEN command, **227**
- ARCLEN command, engineering liquefaction analysis, **290**
- ARCLEN command, nonlinear analysis, **224**
- AREA subroutine, **659**
- ARNOLD command, **432**
- Arnoldi method based eigenvalue analysis, **432**
- ARROW subroutine, **659**
- INTPNT command, **456**
- ASAPxy output, **456**
- ASCII option, **56**, **66**
- Aspect Ratio test, **52**
- ASRAT command, **456**
- ASREQ command, **456**

- ASRQxy output, [456](#)
 ASRXY command, [456](#)
 ASSEMB command, [54](#), [534](#)
 ASSEMB command, transient heat flow, [338](#)
 Assumed strain, [209](#), [385](#)
 AtXYZ output, response spectrum analysis, [176](#)
 AUTARC option, [228](#)
 AUTO option, [424](#)
 AUTOMA command, engineering liquefaction analysis, [289](#)
 AUTOMA command, nonlinear initial state, [224](#)
 AUTOMA command, nonlinear load step, [225](#), [227](#)
 AUTOMA command, nonlinear time step, [231](#)
 AUTOMA command, strength reduction analysis, [283](#), [284](#)
 AUTOMA command, transient analysis, [124](#)
 AUTOMA command, transient time step, [125](#)
 Automatic load increments, [563](#)
 Automatic load increments, cutback based, [227](#), [565](#)
 Automatic load increments, energy based, [226](#), [289](#), [564](#)
 Automatic load increments, iteration based, [226](#), [563](#)
 Automatic time increments, [563](#)
 Automatic time increments, cutback based, [233](#), [565](#)
 Automatic time increments, iteration based, [232](#), [563](#)
 Automatic tying, [28](#)
 AUTOTY option, [54](#)
 AVERAG option, [53](#)
 AVERAG option, nonlinear strain, [250](#)
 AVJSCE command, [60](#)
 AVJSCE option, [56](#)
 AXES input, [23–25](#)
 AXES option, [58](#), [66](#)
 Axisymmetric elements, options, [298](#)
- ## B
- BAB subroutine, [651](#)
 Backup of FILOS file, [49](#)
 BACKWA command, [217](#)
 Backward analysis, [489](#), [490](#)
 Backward substitution, [540](#)
 *BALANC command, [514](#)
 BALANC command, [515](#)
 BANDWI option, [274](#)
 Base excitation, [586](#)
 Base excitation, dynamic analysis, [104](#)
 Base excitation, relative acceleration, [130](#)
 Base excitation, relative displacement, [127](#)
 Base excitation, relative velocity, [128](#)
 Base excitation, Response Spectrum Analysis, [590](#)
 BASE subtable of 'LOADS', [104](#)
 BASNOD command, [59](#)
 *BCROSS command, [366](#)
 Beam cross-section analysis, [365](#)
 Beam elements, stability analysis, [296](#), [297](#)
 Beam elements, tying to plane stress, [29](#), [30](#)
 Beam elements, tying to solid, [29](#), [30](#)
 BEHAVI input, Eurocode 8 EN 1998-1, [116](#)
 BEHAVI input, NPR 9998:2015, [118](#)
 Bending moments, frequency response analysis, [165](#)
 Bending moments, linear static analysis, [86](#)
 Bending moments, nonlinear analysis, [259](#)
 Bending moments, response spectrum analysis, [183](#)
 Bending moments, transient analysis, [136](#)
 BETA parameter, [217](#)
 BETWEE subtable of 'TYINGS', [21](#), [324](#)
 BFGS iteration, [237](#), [555](#)
 BFGS option, [237](#)
 BIAXFE option, linear static analysis, [83](#)
 BIAXFE option, nonlinear analysis, [256](#)
 BIAXFE option, transient analysis, [134](#)
 Biaxial failure envelope, linear static analysis, [83](#)
 Biaxial failure envelope, nonlinear analysis, [256](#)
 Biaxial failure envelope, transient analysis, [134](#)
 Bifurcation, [215](#), [564](#)
 BILIN input, [500](#)
 BINARY option, [56](#), [64](#), [66](#), [67](#)
 Biot material parameters, [633](#)
 Biot theory (full), [378](#)
 BOND command, [235](#)
 Bottom absorption, [597](#)
 BOUNDA command, cross-section analysis, [367](#)
 BOUNDA command, lubrication analysis, [372](#)
 BOUNDA command, steady-state groundwater flow, [351](#), [352](#), [360](#)
 BOUNDA command, steady-state heat flow, [333](#), [334](#), [343](#)
 BOUNDA command, transient groundwater flow, [356](#)
 BOUNDA command, transient heat flow, [338](#)
 BOUNDA input, [318](#), [319](#)
 'BOUNDA' table, [317](#)
 Boundary case, [317](#)
 Boundary conditions, [627](#)
 Boundary conditions, essential, [531](#)
 Boundary conditions, natural, [531](#)
 Boundary conditions, potential flow, [317](#)
 Boundary conditions, transient potential flow, [318](#)
 Boundary elements, groundwater flow, [362](#)
 Boundary elements, heat flow, [345](#)
 BOUNDS input, [499](#)
 Bowl liquefaction, status output, [265](#)
 BOWL option, [265](#)
 BRIEF command, HFTD analysis, [195](#)
 BRIEF command, nonlinear analysis, [243](#)
 British Imperial units, [3](#)
 BRITTL input, stiffness adaptation analysis, [472](#)
 BROUDE option, [237](#)
 Broyden iteration, [237](#), [555](#)
 BTWMPD subtable of 'TYINGS', [22](#), [325](#)
 BUCKLI command, [302](#)
 BUCKLI option, [304](#)
 Buckling analysis, [300](#), [617](#)
 Buckling modes, [305](#)
 Buckling values, [305](#), [437](#)
 Bump factor, [42](#)

BUMP input, [42](#)
 Buoyancy, [637](#)
 Buoyancy, mixture analysis, [389](#)
 Buoyancy, staggered analysis, [383](#)
 BWIDTH input, Dutch Code, [40](#)
 BWIDTH input, European Code, [44](#)
 BWIDTH input, explicit input, [39](#)

C

CALCUL command, [515](#)
 CALCUL option, direct response analysis, [151](#)
 CALCUL option, eigenvalue analysis, [430](#), [431](#)
 CALCUL option, HFTD analysis, [191](#)
 CALCUL option, initial displacements, [222](#)
 CALCUL option, initial stresses, [223](#)
 CALCUL option, steady-state groundwater flow, [354](#)
 CALCUL option, steady-state heat flow, [336](#)
 CALCUL option, transient groundwater flow, [357](#)
 CALCUL option, transient heat flow, [339](#)
 CALCULATE option, [520](#)
 CAPACI command, [338](#)
 Capacity, [628](#)
 Capacity matrices, [338](#)
 Capacity, heat flow, [330](#)
 Capacity, hydrostatic, [377](#)
 CASE input, mobile load, [38](#)
 CASE input, potential flow analysis, [317](#)
 CASE input, pushover analysis, [105](#)
 CASE input, structural analysis, [32](#)
 CASE input, transient analysis, [104](#)
 CASE parameter, steady-state groundwater flow, [351](#), [352](#), [354](#)
 CASE parameter, steady-state heat flow, [334](#), [336](#)
 CASE parameter, transient groundwater flow, [357](#)
 CASE parameter, transient heat flow, [339](#)
 CASES command, cross-section analysis, [369](#)
 CASES command, design checking, [453](#)
 CASES command, lubrication analysis, [373](#)
 CAUCHY option, [57](#)
 Cauchy stress, [568](#)
 Cauchy stress, frequency response analysis, [163](#)
 Cauchy stress, linear static analysis, [85](#)
 Cauchy stress, nonlinear analysis, [257](#)
 Cauchy stress, response spectrum analysis, [181](#)
 Cauchy stress, transient analysis, [135](#)
 CBSPEC input, parameter output, [274](#)
 CEB-FIP fiber reinforced concrete, stiffness adaptation analysis, [473](#)
 CEB-FIP Model Code 1990, stiffness adaptation analysis, [473](#), [478](#)
 CEB-FIP Model Code 1990, units, [4](#)
 CELSIU input, [4](#)
 Celsius, temperature unit, [4](#)
 Cement hydration, *see* Hydration
 Cement hydration, staggered analysis, [382](#)
 Cement hydration, transient heat flow analysis, [340](#)
 CENTAX option, [58](#)
 CENTAX option, linear static analysis, [84](#)
 CENTAX option, nonlinear analysis, [257](#)

CENTER option, [56](#)
 Centimeter, length unit, [4](#)
 CENTRI subtable of 'LOADS', [35](#)
 Centrifugal load, [35](#)
 CG, [547](#)
 CGS units, [3](#), [4](#)
 CHECK command, element evaluation, [51](#)
 CHECK command, FILOS file, [49](#)
 Chemical expansion, staggered analysis, [382](#)
 CHOLE command, [421](#), [422](#)
 CHOLE command, eigenvalue analysis, [433](#)
 Cholesky solver, *see* Sparse Cholesky
 CIRCUM input, [500](#)
 CLASS input, [40](#)
 CLASS input, Eurocode 2 EN 1992-1-1, [449](#)
 Class-I beams, stability analysis, [296](#), [297](#)
 Class-II beams, stability analysis, [296](#)
 Class-III beams, stability analysis, [297](#)
 CLASSE input, [44](#)
 Clay, [235](#)
 Clay, Egg Cam-clay model, [260](#)
 CLEAR command, [235](#)
 CLEAR subroutine, [655](#)
 CM input, [4](#)
 CMOD command, [229](#)
 CMOD control, [229](#), [563](#)
 CO output, [270](#)
 CO parameter, [60](#)
 CODE command, [451](#)
 CODE input, [38](#)
 CODE input, design spectrum, [116](#), [118](#)
 CODE input, Dutch Code, [40](#)
 CODE input, elastic response spectrum, [114](#), [117](#)
 CODE input, Eurocode 8 EN 1998-1, [114](#), [116](#)
 CODE input, European Code, [43](#)
 CODE input, NPR 9998:2015, [117](#), [118](#)
 COEFF command, [309](#)
 COHESI option, linear static analysis, [93](#)
 COHESI option, nonlinear analysis, [276](#)
 COHESI output, [266](#)
 COHESI output, linear static analysis, [95](#)
 COHESI output, nonlinear analysis, [278](#)
 COLLIN command, [65](#)
 COMBIN command, [66](#)
 COMBIN command, design checking, [453](#)
 COMBIN option, design checking, [457](#)
 COMBIN subtable of 'LOADS', [45](#)
 Combination of load cases, [45](#)
 Combination of load cases, design checking, [453](#)
 Combination of output items, [66](#)
 Combined reinforcement forces, [457](#)
 COMCRV input, stiffness adaptation analysis, [469](#), [475](#)
 Commands, [47](#)
 Commands, loads optimization, [514](#)
 Commands, parameter estimation, [505](#)
 COMPLETE option, [153](#)
 Complex loads, [113](#)
 COMPOS command, element evaluation, [53](#)
 Composed element evaluation, [53](#)
 Composites, failure criteria, [84](#)
 COMPRE command, [49](#)
 Compressibility, undrained, [632](#)

- Compression modulus, [632](#)
- Compression of FILOS file, [48](#), [49](#)
- Compressive failure, Kotsovos concrete model, [266](#)
- Compressive strength, stiffness adaptation analysis, [478](#)
- COMSTO input, stiffness adaptation analysis, [477](#)
- COMSTR input, stiffness adaptation analysis, [476–478](#)
- COMSTR option, linear static analysis, [93](#)
- COMSTR option, nonlinear analysis, [276](#)
- COMSTR output, linear static analysis, [95](#)
- COMSTR output, nonlinear analysis, [278](#)
- CONCEN command, [269](#)
- CONCEN command, HFTD analysis, [198](#)
- CONCEN command, nonlinear analysis, [215](#)
- CONCEN option, [250](#)
- Concentrated mass, dynamic analysis, [103](#)
- Concentration, [269](#)
- Concentration flow, [329](#)
- Concentration forces, nonlinear analysis, [259](#)
- Concentration load, staggered analysis, [382](#)
- Concentration strain, [250](#), [252](#)
- CONCRE input, Eurocode 2 EN 1992-1-1, [449](#)
- CONCRE input, stiffness adaptation analysis, [470](#)
- Concrete biaxial failure envelope, linear static analysis, [83](#), [85](#)
- Concrete biaxial failure envelope, nonlinear analysis, [256](#), [258](#)
- Concrete biaxial failure envelope, transient analysis, [134](#), [136](#)
- Concrete hydration, [378](#)
- Concrete safety factors, linear static analysis, [83](#)
- Concrete safety factors, nonlinear analysis, [256](#)
- Concrete safety factors, transient analysis, [134](#)
- Concrete, staggered analysis, [382](#)
- Condition number, [546](#)
- CONDUCT command, cross-section analysis, [366](#)
- CONDUCT command, lubrication analysis, [372](#)
- CONDUCT command, steady-state groundwater flow, [351](#)
- CONDUCT command, steady-state heat flow, [333](#)
- CONDUCT command, transient groundwater flow, [356](#)
- CONDUCT command, transient heat flow, [338](#)
- Conduction, [628](#)
- Conduction, staggered analysis, [378](#)
- Conductivity matrices, steady-state groundwater flow, [351](#)
- Conductivity matrices, steady-state heat flow, [333](#)
- Conductivity matrices, transient groundwater flow, [356](#)
- Conductivity matrices, transient heat flow, [338](#)
- Conductivity, heat flow, [330](#)
- Conductivity, hydraulic, [347](#)
- Conjugate Gradient, [547](#)
- Conservative loading, [617](#)
- CONSIS option, cracking, [215](#)
- CONSIS option, direct response analysis, [151](#)
- CONSIS option, eigenvalue analysis, [430](#)
- CONSIS option, HFTD analysis, [191](#)
- CONSIS option, nonlinear analysis, [218](#)
- CONSIS option, transient groundwater flow, [356](#)
- CONSIS option, transient heat flow, [338](#)
- Consistent mass analysis, [542](#)
- Consolidation of soil, [348](#), [377](#)
- Consolidation of soil, mixture analysis, [385](#), [390](#)
- CONSTA command, [237](#)
- CONSTA input, stiffness adaptation analysis, [471](#), [476](#)
- Constant dilatation, mixture analysis, [385](#)
- Constant shear, mixture analysis, [385](#)
- Constant Stiffness iteration, [237](#), [556](#)
- CONSTR command, [452](#)
- Constrained Minimization, [571](#)
- CONTAC command, [215](#)
- CONTAC option, [265](#)
- Contact analysis, [571](#)
- Contact status, [265](#)
- Contamination transport, [348](#)
- CONTIN command, nonlinear analysis, [236](#)
- CONTIN command, nonlinear vibration analysis, [204](#)
- CONTIN command, postbuckling analysis, [308](#)
- CONTIN option, HFTD analysis, [195](#)
- CONTIN option, nonlinear analysis, [239](#)
- CONTIN option, nonlinear load step, [226](#), [227](#)
- CONTIN option, nonlinear time step, [232](#)
- Continuation analysis, [204](#), [296](#), [308](#), [621](#)
- Continuation iteration, [236](#), [557](#)
- CONTOUR option, [520](#)
- Contour plots, stress, [520](#)
- Control commands, [47](#)
- CONVEC command, [351](#)
- Convection, [330](#)
- Convection–diffusion, [627](#)
- CONVER command, heat transfer, [335](#)
- CONVER command, HFTD analysis, [195](#)
- CONVER command, nonlinear analysis, [236](#), [238](#)
- CONVER command, parameter estimation, [509](#)
- CONVER command, steady-state groundwater flow, [353](#)
- CONVER command, strength reduction analysis, [285](#)
- CONVER command, transient groundwater flow, [359](#)
- CONVER command, transient heat flow, [342](#)
- CONVER option, [246](#)
- CONVER parameter, [204](#), [309](#)
- Convergence criteria, [236](#), [238](#), [558](#)
- Convergence criteria, HFTD analysis, [194](#)
- Convergence rate, [553](#), [556](#), [557](#)
- Convergence, eigenvalue analysis, [433–435](#)
- Convergence, iterative solution, [549](#)
- Cooling pipe elements, [331](#), [343](#)
- Cooling pipe elements, internal temperature, [341](#), [346](#)
- COORD option, [58](#)
- COORD option, [65](#)
- 'COORDI' table, [11](#)
- COPY command, [49](#)
- Copy FILOS file, *see* FILOS file
- Correlation factors, [187](#)

CORROS command, [214](#)
 Coulomb friction, *see* Friction
 Coulomb friction failure criterion, [581](#), [583](#)
 Coulomb friction failure criterion, linear static analysis, [83](#), [87](#)
 Coulomb friction failure criterion, nonlinear analysis, [256](#), [268](#)
 Coupled flow–stress analysis, [377](#)
 CPOWER input, stiffness adaptation analysis, [473](#)
 CQC option, [172](#)
 CQC rule, [169](#), [172](#), [591](#)
 CQCABS option, [173](#)
 Crack bandwidth, output, [274](#)
 Crack index, linear static analysis, [82](#)
 Crack index, nonlinear analysis, [255](#), [258](#)
 Crack indicator, linear static analysis, [82](#)
 Crack indicator, nonlinear analysis, [255](#)
 CRACK option, nonlinear stress, [254](#)
 CRACK option, nonlinear status, [261](#)
 CRACK option, strains, [249](#)
 Crack status, [261](#)
 Crack strains, output, [249](#), [251](#)
 Crack strains, summation, [249](#), [251](#)
 Crack stress, [258](#)
 Crack tip, [566](#)
 Crack tip elements, [639](#)
 Crack tip elements, linear static analysis, [95](#)
 Crack tip elements, nonlinear analysis, [278](#)
 Crack width, [252](#), [483](#)
 Crack width, output, [249](#), [483](#)
 CRACKB input, parameter output, [274](#)
 CRACKI command, [215](#)
 CRACKI command, HFTD analysis logging, [196](#)
 CRACKI command, nonlinear analysis logging, [243](#)
 CRACKN input, stiffness adaptation analysis, [470](#)
 Crank–Nicolson integration, potential flow analysis, [628](#)
 Crank–Nicolson integration, transient groundwater flow, [358](#)
 Crank–Nicolson integration, transient heat flow, [341](#)
 CREEP command, [214](#)
 CREEP option, [249](#)
 Creep strain, [249](#)
 CRISFI option, [237](#)
 Crisfield iteration, [237](#), [556](#)
 Critical damping factor, [148](#), [172](#), [436](#), [586](#)
 CRKIND option, linear static analysis, [82](#)
 CRKIND option, nonlinear analysis, [255](#)
 CRKSUM option, [249](#)
 CRKWDT option, [249](#)
 CRKWDT option, stiffness adaptation analysis, [483](#)
 Cross-section analysis, [365](#)
 Cross-section analysis, input, [315](#), [365](#)
 Cross-section analysis, output, [368](#)
 CRUSHD output, [266](#)
 CUMSHR output, [266](#)
 Cumulative energy, normalized, [262](#)
 Curvature, frequency response analysis, [161](#)
 Curvature, linear static analysis, [80](#)
 Curvature, nonlinear analysis, [252](#)
 Curvature, response spectrum analysis, [179](#)

Curvature, transient analysis, [133](#)
 Curved shell elements, stability analysis, [298](#)
 CUSTOM option, design spectrum, [116](#)
 CUSTOM option, elastic response spectrum, [114](#)
 CUTBCK parameter, nonlinear load step, [227](#)
 CUTBCK parameter, nonlinear time step, [234](#)
 CUTBCK parameter, strength reduction analysis, [284](#)

D

D output, buckling analysis, [305](#)
 D output, eigenvalue analysis, [440](#)
 D output, frequency response analysis, [154](#)
 D output, linear static analysis, [77](#)
 D output, response spectrum analysis, [174](#)
 D–min soil model, [275](#)
 D2 output, [203](#), [307](#)
 Damage Index, [262](#)
 DAMIND option, [262](#)
 DAMPIN command, direct response analysis, [151](#)
 DAMPIN command, eigenvalue analysis, [436](#)
 DAMPIN command, HFTD analysis, [191](#)
 DAMPIN command, modal response analysis, [148](#)
 DAMPIN command, nonlinear analysis, [218](#)
 DAMPIN command, response spectrum analysis, [172](#)
 DAMPIN input, Eurocode 8 EN 1998-1, [114](#)
 DAMPIN input, NPR 9998:2015, [117](#)
 DAMPIN option, nonlinear analysis, [271](#), [273](#)
 DAMPIN option, transient analysis, [139](#), [141](#)
 Damping, [103](#), [586](#)
 Damping, added, [599](#)
 Damping, continuous, [586](#), [589](#)
 Damping, discrete, [586](#)
 Damping, forces, [139](#), [141](#), [271](#), [273](#)
 Damping, modal, [148](#), [172](#), [586](#)
 Damping, Rayleigh, [6](#), [218](#), [435](#), [544](#), [586](#)
 Damping, strain energy based, [438](#), [544](#)
 Damping, structural, [586](#)
 Damping, Viscous, [586](#)
 Darcy flow, [633](#)
 Darcy flux, [396](#)
 Darcy’s law, staggered analysis, [377](#)
 DARCYP command, [396](#)
 Dashpots, *see* Damping, continuous
 Data structures access, [669](#)
 Database, *see* FILOS file
 DAY input, [4](#)
 Day, time unit, [4](#)
 ddcc utility program, [648](#)
 Dead weight, *see* Weight load, [33](#)
 Decimeter, length unit, [4](#)
 Decomposition, [539](#), [655](#)
 DEFORM subtable of ‘LOADS’, [35](#)
 DEFORM subtable of ‘LOADS’, mixture analysis, [388](#)
 DEFORM subtable of ‘LOADS’, phased analysis, [403](#), [408](#)
 Deformation load, *see* Prescribed displacements
 Deformation, frequency response analysis, [161](#)
 Deformation, linear static analysis, [80](#)
 Deformation, nonlinear analysis, [252](#)

- Deformation, response spectrum analysis, [178](#)
- Deformation, transient analysis, [132](#)
- Degree Celsius, temperature unit, [4](#)
- Degree Fahrenheit, temperature unit, [4](#)
- DEGREE input, [4](#)
- Degree of reaction, initial, [340](#)
- Degree of reaction, output, [345](#)
- Degree of reaction, staggered analysis, [378](#), [382](#)
- Degree, angle unit, [4](#)
- Degrees of freedom, [531](#)
- DELETE command, [50](#), [402](#)
- DELTAT parameter, [233](#)
- DENSFL input, [6](#)
- DENSIT option, linear static analysis, [93](#)
- DENSIT option, nonlinear analysis, [275](#)
- DENSIT output, linear static analysis, [95](#)
- DENSIT output, nonlinear analysis, [278](#)
- Density, apparent, [631](#)
- Density, reduced, [637](#)
- Der Kiureghian, A., [592](#)
- Design checking, [443](#), [445](#)
- *DESIGN command, [451](#)
- DESIGN input, [447](#)
- DESIGN input, Eurocode 8 EN 1998-1, [116](#)
- DESIGN input, NPR 9998:2015, [118](#)
- Design spectrum, Eurocode 8 EN 1998-1, [116](#)
- Design spectrum, NPR 9998:2015, [118](#)
- DESSPC input, [118](#)
- DET2 subroutine, [655](#)
- DET3 subroutine, [655](#)
- DETA parameter, [237](#)
- Detailed groundwater flow, [347](#), [361](#)
- Determinants subroutines, [655](#)
- Deviatoric deformation, [637](#)
- Deviatoric strains, [573](#)
- Deviatoric stresses, [575](#)
- DGEPD subroutine, [655](#)
- DGR output, [346](#)
- DGRINI parameter, [340](#)
- DI parameter, directory size, [11](#)
- DIAGON command, [423](#)
- Diana Interactive Environment, output, [56](#)
- DianaIE, output, [56](#)
- DianaIE, output for postprocessing, [63](#)
- DIASLIB symbol, [647](#)
- Differential matrix, [532](#)
- DIGITS command, [65](#)
- DIRECT command, CMOD Arc-length control, [230](#)
- DIRECT input, mobile load, [38](#)
- DIRECT input, pushover analysis, [106](#)
- DIRECT option, FILOS file, [49](#)
- Direct response analysis, [149](#)
- Direct solution, [422](#), [539](#), [589](#)
- 'DIRECT' table, [11](#)
- Direct time integration, [592](#)
- Direction dependent participation factor, [543](#)
- Direction of gravity, [6](#)
- Directions, [11](#)
- Dirichlet, [627](#)
- DISC option, [261](#)
- Discharge, Darcy flow, [633](#)
- Discharge, groundwater flow, [362](#)
- Discharge, heat flow, [345](#)
- Discharge, input, [317](#), [389](#)
- Discharge, output, [395](#)
- Discharge, prescribed, [318](#)
- DISFAC input, [39](#)
- DISFAC input, Dutch Code, [42](#)
- DISFAC input, European Code, [44](#)
- DISFOR option, [57](#)
- DISFOR option, design checking, [457](#)
- Disk space, [48](#)
- DISMOM option, [57](#)
- DISMOM option, design checking, [457](#)
- DISPLA command, buckling analysis, [301](#), [304](#)
- DISPLA command, eigenvalue analysis, [439](#)
- DISPLA command, frequency response analysis, [153](#)
- DISPLA command, HFTD analysis, [197](#)
- DISPLA command, linear static analysis, [76](#), [519](#)
- DISPLA command, mixture analysis, [390](#), [394](#)
- DISPLA command, nonlinear analysis, [222](#), [239](#)
- DISPLA command, nonlinear analysis output, [247](#)
- DISPLA command, nonlinear stop criterion, [242](#)
- DISPLA command, nonlinear vibration analysis, [202](#), [205](#)
- DISPLA command, phased analysis, [409](#)
- DISPLA command, postbuckling analysis, [306](#), [309](#)
- DISPLA command, response spectrum analysis, [174](#)
- DISPLA command, stiffness adaptation analysis, [482](#)
- DISPLA command, transient analysis output, [126](#)
- DISPLA option, [235](#)
- DISPLA subtable of 'INIVAR', stability analysis, [296](#)
- DISPLA subtable of 'INIVAR', transient analysis, [107](#)
- Displacement, [533](#)
- Displacement control, [227](#), [560](#)
- Displacement norm, [239](#), [559](#)
- Displacement vector length, linear static analysis, [77](#)
- Displacement vector length, response spectrum analysis, [174](#)
- Displacement vector length, transient analysis, [127](#)
- Displacement, frequency response analysis, [153](#)
- Displacement, linear static analysis, [76](#)
- Displacement, phased analysis, [410](#)
- Displacement, prescribed, [35](#), [408](#)
- Displacement, relative, [58](#)
- Displacement, response spectrum analysis, [174](#)
- Displacement, transient analysis, [126](#)
- DISPX input, loads optimization, [513](#)
- DISPX input, parameter estimation, [496](#)
- DISPY input, loads optimization, [513](#)
- DISPY input, parameter estimation, [496](#)
- DISPZ input, loads optimization, [513](#)
- DISPZ input, parameter estimation, [496](#)
- DISSEI option, [574](#)
- DISSEI option, linear static analysis, [79](#), [81](#)
- DISSEI option, nonlinear analysis, [250](#), [253](#)

- DISSEI option, response spectrum analysis, [177](#), [179](#)
- DISTRI input, [38](#)
- Distributed forces, design checking, [457](#)
- Distributed mass, dynamic analysis, [103](#)
- Distributed moments, design checking, [457](#)
- Distributed seismic moment, *see* Seismic moment, [179](#)
- Divergence, [239](#), [554](#), [558](#)
- Divergence, HFTD analysis, [195](#)
- Divergence, soil–pore fluid, [631](#)
- DM input, [4](#)
- Dm output, [440](#)
- DMAX input, stiffness adaptation analysis, [473](#)
- DMIN option, [275](#)
- Domain decomposition, [423](#), [547](#)
- DP output, [206](#), [310](#)
- Dp output, [440](#)
- DPMPAR subroutine, [665](#)
- Dr output, frequency response analysis, [154](#)
- Dr output, linear static analysis, [77](#)
- DRAINE command, [235](#)
- DRAINE command, mixture analysis, [392](#)
- DRJ output, linear static analysis, [81](#)
- DRJ output, nonlinear analysis, [253](#)
- DRRAND subroutine, [665](#)
- Drucker–Prager plasticity, parameter estimation, [500](#)
- dS output, [258](#)
- dSE output, [258](#)
- dSELod output, [258](#)
- dSEP output, [258](#)
- dSEQ output, [258](#)
- DSNX option, linear static analysis, [93](#)
- DSNX option, nonlinear analysis, [276](#)
- DSNX output, linear static analysis, [95](#)
- DSNX output, nonlinear analysis, [278](#)
- DSNY option, linear static analysis, [93](#)
- DSNY option, nonlinear analysis, [276](#)
- DSNY output, linear static analysis, [95](#)
- DSNY output, nonlinear analysis, [278](#)
- DSNZ option, linear static analysis, [93](#)
- DSNZ option, nonlinear analysis, [276](#)
- DSNZ output, linear static analysis, [95](#)
- DSNZ output, nonlinear analysis, [278](#)
- dSP output, [258](#)
- dSQ output, [258](#)
- DSRAND subroutine, [665](#)
- DSSX option, linear static analysis, [93](#)
- DSSX option, nonlinear analysis, [276](#)
- DSSX output, linear static analysis, [95](#)
- DSSX output, nonlinear analysis, [278](#)
- DSSY option, linear static analysis, [93](#)
- DSSY option, nonlinear analysis, [276](#)
- DSSY output, linear static analysis, [95](#)
- DSSY output, nonlinear analysis, [278](#)
- DSSZ option, linear static analysis, [93](#)
- DSSZ option, nonlinear analysis, [276](#)
- DSSZ output, linear static analysis, [95](#)
- DSSZ output, nonlinear analysis, [278](#)
- DTX command, [454](#)
- DtXYZ output, response spectrum analysis, [174](#)
- DTY command, [454](#)
- DTZ command, [454](#)
- DU output, frequency response analysis, [162](#)
- DU output, linear static analysis, [81](#)
- DU output, nonlinear analysis, [253](#)
- DU output, response spectrum analysis, [179](#)
- DU output, transient analysis, [133](#)
- DUe output, nonlinear analysis, [253](#)
- DUJ output, linear static analysis, [81](#)
- DUJ output, nonlinear analysis, [253](#)
- DUp output, nonlinear analysis, [253](#)
- Dupuit assumption, [349](#)
- Dutch codes, *see* NEN, VBB, VOSB
- DUTSIG input, stiffness adaptation analysis, [474](#)
- DX parameter, [23–25](#)
- DXYZ output, linear static analysis, [77](#)
- DY parameter, [23–25](#)
- DYBUCK command, [309](#)
- DYN input, [4](#)
- DYNAMI command, [218](#)
- Dynamic analysis, [103](#)
- Dynamic analysis, Darcy flow, [633](#)
- Dynamic analysis, flow–stress, [635](#), [636](#)
- Dynamic analysis, mixture elements, [378](#)
- Dynamic analysis, transient nonlinear, [218](#)
- Dynamic concrete biaxial failure envelope, linear static analysis, [83](#)
- Dynamic concrete biaxial failure envelope, nonlinear analysis, [256](#)
- Dynamic concrete biaxial failure envelope, transient analysis, [134](#)
- Dynamic pressures, frequency response analysis, [167](#)
- Dynamic pressures, nonlinear analysis, [279](#)
- Dynamic pressures, transient analysis, [142](#)
- Dynamic structural analysis, linear transient, [121](#)
- Dyne, force unit, [4](#)
- DZ parameter, [23–25](#)

E

- E output, frequency response analysis, [160](#)
- E output, linear static analysis, [80](#)
- E output, nonlinear analysis, [250](#)
- E output, response spectrum analysis, [178](#)
- E output, transient analysis, [132](#)
- Earthquake, [104](#)
- Earthquakes, induced, [117](#), [118](#)
- Ec output, [251](#)
- EC2 input, stiffness adaptation analysis, [478](#)
- ECCENT option, linear static analysis, [94](#)
- ECCENT option, nonlinear analysis, [276](#)
- ECCENT output, linear static analysis, [95](#)
- ECCENT output, nonlinear analysis, [278](#)
- ECCENT subtable of 'TYINGS', [23](#)
- Eccentricity, [23](#), [28](#), [534](#)
- ECCMPC subtable of 'TYINGS', [25](#)
- Eceq output, [251](#)
- ECHO option, [50](#)
- Econ output, [252](#)
- *ECREEP command, [291](#)
- Ecw output, [252](#), [483](#)
- Ecweq output, [252](#), [483](#)

- Editor, *see* Text editor
- Ee output, [250](#)
- Eeeq output, [251](#)
- Eeq output, frequency response analysis, [161](#)
- Eeq output, linear static analysis, [80](#)
- Eeq output, nonlinear analysis, [251](#)
- Eeq output, response spectrum analysis, [178](#)
- Eeq output, transient analysis, [132](#)
- EFFECT option, nonlinear stress, [254](#)
- Effective force vector, [561](#)
- Effective mass, [438](#), [543](#)
- Effective stresses, [254](#)
- EFFMAS command, [436](#)
- EHAR input, stiffness adaptation analysis, [476](#), [477](#)
- *EIGEN command, [425](#)
- EIGEN command, buckling analysis, [300](#)
- EIGEN command, HFTD analysis, [191](#)
- EIGEN command, modal response analysis, [145](#)
- EIGEN command, nonlinear vibration analysis, [200](#)
- EIGEN command, response spectrum analysis, [170](#)
- Eigenfrequencies, shifting, [544](#)
- Eigenfrequency, [437](#), [545](#)
- Eigenmode, [439](#)
- Eigenvalue, [437](#), [545](#)
- Eigenvalue analysis, [425](#), [541](#)
- Eigenvalues, shifting, [544](#)
- Ek output, crack strains, [251](#)
- Ek output, summed crack strains, [251](#)
- Ekeq output, [251](#)
- ELASTI command, [215](#)
- ELASTI input, NPR 9998:2015, [117](#)
- ELASTI input, stiffness adaptation analysis, [471](#)
- ELASTI option, [249](#)
- ELASTI option, Eurocode 8 EN 1998-1, [114](#)
- Elastic energy, Kotsovos concrete model, [266](#)
- Elastic response spectrum, Eurocode 8 EN 1998-1, [114](#)
- Elastic response spectrum, NPR 9998:2015, [117](#)
- Elastic strain, [249](#), [250](#)
- Elastoplasticity, [634](#)
- Elastoplasticity, staggered analysis, [377](#)
- ELEMEN command, output selection, [60](#)
- ELEMEN command, output selection, [59](#)
- ELEMEN command, phased analysis, [405](#)
- ELEMEN command, reinforcement selection, [62](#)
- ELEMEN input, mobile load, [38](#)
- ELEMEN option, automatic tying, [54](#)
- ELEMEN option, linear static analysis, [90](#), [91](#)
- ELEMEN option, nonlinear analysis, [271](#), [273](#)
- ELEMEN option, phased analysis, [417](#)
- ELEMEN option, response spectrum analysis, [185](#), [187](#)
- ELEMEN option, transient analysis, [139](#), [141](#)
- ELEMEN subtable of 'GROUPS', [9](#)
- ELEMEN subtable of 'TYINGS', [30](#)
- Element assembling, [534](#)
- Element damping forces, nonlinear analysis, [273](#)
- Element damping forces, transient analysis, [141](#)
- Element forces, nonlinear analysis, [272](#)
- Element forces, transient analysis, [140](#)
- Element inertia forces, nonlinear analysis, [273](#)
- Element inertia forces, transient analysis, [141](#)
- Element internal forces, [583](#)
- Element interpolation matrix, [533](#)
- Element load, [536](#)
- Element mass matrix, direct response analysis, [151](#)
- Element mass matrix, eigenvalue analysis, [429](#)
- Element mass matrix, HFTD analysis, [190](#)
- Element rigidity matrix, [534](#)
- Element stiffness matrix, [535](#)
- Element tractions, [534](#)
- Element transformation matrix, [533](#)
- Element types, [671](#)
- ELF modal distribution, [104](#)
- *ELIQUE command, [287](#)
- ELMFOR command, HFTD analysis, [198](#)
- ELMFOR command, linear static analysis, [91](#)
- ELMFOR command, nonlinear analysis, [273](#)
- ELMFOR command, response spectrum analysis, [187](#)
- ELMFOR command, stiffness adaptation analysis, [483](#)
- ELMFOR command, transient analysis, [141](#)
- EMAX parameter, [435](#)
- EMIN parameter, [435](#)
- EN 1992-1-1, stiffness adaptation analysis, [478](#)
- EN 1992-1-2, stiffness adaptation analysis, [478](#)
- EN1992 input, [449](#)
- EN1992 input, stiffness adaptation analysis, [478](#)
- ENERGY command, engineering liquefaction analysis, [289](#)
- ENERGY command, nonlinear analysis, [239](#)
- ENERGY command, nonlinear initial state, [224](#)
- ENERGY command, nonlinear load step, [225](#), [226](#), [289](#)
- ENERGY command, transient analysis, [124](#)
- Energy load increments, [564](#)
- Energy norm, [559](#)
- ENERGY option, [137](#), [138](#), [263–265](#)
- Energy release rate, [640](#)
- Energy release rate, linear static analysis, [95](#)
- Energy release rate, nonlinear analysis, [278](#)
- Energy status, normalized cumulative, [262](#)
- Energy status, specific gravitational, [137](#), [264](#)
- Energy status, specific kinetic, [137](#), [263](#)
- Energy status, specific pressure, [264](#)
- Energy status, specific stress, [263](#)
- Engineering creep, [615](#)
- Engineering creep analysis, [291](#)
- Engineering liquefaction, [613](#)
- Engineering liquefaction analysis, [287](#)
- Engineering Masonry model, linearized element status, [267](#)
- Engineering Masonry model, status output, [266](#)
- Engineering strains, [573](#)
- ENGMAS option, [266](#)
- Enthalpy, [331](#)
- ENV 1991-3 code, [42](#), [43](#)
- ENV input, [38](#), [43](#)
- ENVELO command, design checking, [451](#), [454](#)
- ENVIRO command, [452](#)
- ENVIRO input, [448](#), [449](#)

- Ep output, [251](#)
 Epeq output, [251](#)
 EPSC1 input, stiffness adaptation analysis, [478](#)
 EPSCU input, stiffness adaptation analysis, [478](#)
 EPSIGC input, stiffness adaptation analysis, [476](#)
 EPSIGT input, stiffness adaptation analysis, [472](#)
 EPSTU input, stiffness adaptation analysis, [473](#)
 EPSULT input, stiffness adaptation analysis, [472](#)
 EqA output, [346](#)
 EQAINI parameter, [340](#)
 EQUAGE command, [340](#), [343](#), [346](#)
 EQUAL subtable of 'TYINGS', [18](#), [19](#), [322](#)
 EQUAL subtable of 'TYINGS', mixture analysis, [389](#)
 Equality tying, [18](#), [322](#)
 EQUIAC subtable of 'LOADS', [34](#)
 Equilibrium, [532](#), [536](#)
 Equivalent acceleration load, model load, [34](#)
 Equivalent age, [331](#)
 Equivalent age, initial, [340](#)
 Equivalent age, output, [346](#)
 Equivalent age, staggered analysis, [382](#)
 Equivalent lateral force modal distribution, [104](#)
 Equivalent mass, [438](#), [544](#)
 Equivalent nodal forces, [536](#)
 EQUMPC subtable of 'TYINGS', [20](#), [323](#)
 ERATIO option, [255](#)
 Error handling subroutine, [668](#)
 Error messages, [47](#)
 ERROR option, [58](#)
 ERROR option, linear static analysis, [84](#)
 ERRORS command, [47](#)
 Eshrvo output, [252](#)
 Essential boundary condition, [627](#)
 ESTIMA command, loads optimization, [515](#)
 ESTIMA command, steady-state groundwater flow, [352](#), [354](#)
 ESTIMA command, steady-state heat flow, [334](#), [336](#)
 'ESTIMA' table, [496](#)
 Estimation report, loads optimization, [515](#)
 Estimation report, parameter estimation, [510](#), [526](#)
 ETAMAX parameter, [237](#)
 ETAMIN parameter, [237](#)
 Etmp output, [252](#)
 Euler Backward integration, potential flow analysis, [628](#)
 Euler Backward integration, transient groundwater flow, [358](#)
 Euler Backward integration, transient heat flow, [341](#)
 Euler Backward time integration, [593](#)
 Euler Backward time integration, nonlinear analysis, [217](#)
 Euler Backward time integration, soil-pore fluid analysis, [635](#)
 Euler buckling, [300](#), [617](#)
 *EULER command, [299](#)
 Euler Forward integration, potential flow analysis, [628](#)
 Euler Forward integration, transient groundwater flow, [358](#)
 Euler Forward integration, transient heat flow, [341](#)
 EURO8 input, design spectrum, [116](#)
 EURO8 input, elastic response spectrum, [114](#)
 Eurocode 2, stiffness adaptation analysis, [478](#)
 Eurocode 8 EN 1998-1, [111](#)
 European code, mobile load, *see* ENV 1991-3
 EVALUA command, [51](#)
 EVALUA command, transient heat flow, [337](#)
 Evaporation, [331](#)
 Evol output, linear static analysis, [80](#)
 Evol output, nonlinear analysis, [250](#)
 Evol output, response spectrum analysis, [178](#)
 Evol output, transient analysis, [132](#)
 Excitation, HFTD analysis, [189](#)
 Excitation, response spectrum analysis, [169](#)
 EXECUT command, buckling analysis, [300](#)
 EXECUT command, cross-section analysis, [367](#)
 EXECUT command, design checking, [451](#), [452](#)
 EXECUT command, direct response analysis, [152](#)
 EXECUT command, eigenvalue analysis, [432](#)
 EXECUT command, HFTD analysis, [191](#), [192](#), [194](#)
 EXECUT command, loads optimization, [514](#)
 EXECUT command, lubrication analysis, [372](#), [372](#)
 EXECUT command, mixture analysis, [391](#), [392](#)
 EXECUT command, modal response analysis, [145](#), [146](#), [147](#)
 EXECUT command, nonlinear analysis, [219](#)
 EXECUT command, nonlinear structural analysis, [211](#)
 EXECUT command, nonlinear vibration analysis, [200](#), [201](#), [204](#)
 EXECUT command, parameter estimation, [506](#)
 EXECUT command, postbuckling analysis, [309](#)
 EXECUT command, response spectrum analysis, [171](#)
 EXECUT command, steady-state groundwater flow, [350](#)
 EXECUT command, steady-state heat flow, [332](#)
 EXECUT command, stiffness adaptation analysis, [482](#)
 EXECUT command, strength reduction analysis, [281](#)
 EXECUT command, transient analysis, [121](#), [123](#)
 EXECUT command, transient groundwater flow, [358](#)
 EXECUT command, transient heat flow, [340](#)
 EXECUT option, nonlinear analysis logging, [243](#)
 EXPLIC command, direct response analysis, [152](#)
 EXPLIC command, eigenvalue analysis, [436](#)
 EXPLIC command, engineering liquefaction analysis, [289](#)
 EXPLIC command, HFTD analysis, [194](#)
 EXPLIC command, modal response analysis, [148](#)
 EXPLIC command, nonlinear initial state, [224](#)
 EXPLIC command, nonlinear load step, [225](#)
 EXPLIC command, nonlinear time step, [231](#), [232](#)
 EXPLIC command, strength reduction analysis, [283](#)
 EXPLIC command, transient analysis, [124](#)
 EXPLIC command, transient time step, [125](#)

- Explicit time integration, phased analysis, [417](#)
 Explicit time integration, transient analysis, [593](#)
 Explicit time integration, transient flow analysis, [628](#)
 EXPONE command, [125](#), [231](#), [233](#)
 EXPONE input, stiffness adaptation analysis, [472](#)
 Exponential time increments, [233](#)
 EXTERN option, groundwater flow, [363](#)
 EXTERN option, heat transfer, [345](#)
 EXTERN option, load vectors, [88](#)
 External flows, groundwater flow, [363](#)
 External flows, heat transfer, [345](#)
 Extreme values, steady-state groundwater flow, [360](#)
 Extreme values, steady-state heat flow, [343](#)
 Extreme values, transient groundwater flow, [361](#)
 Extreme values, transient heat flow, [344](#)
- ## F
- FACTOR command, buckling analysis, [303](#)
 FACTOR command, design checking, [453](#)
 FACTOR command, factorization, [422](#), [423](#)
 FACTOR command, phased analysis, [406](#)
 FACTOR command, postbuckling analysis, [309](#)
 FACTOR command, response spectrum analysis, [173](#)
 FACTOR input, frequency domain analysis, [112](#)
 FACTOR input, frequency-load diagram, [111](#)
 FACTOR input, spatial functions, [13](#)
 FACTOR input, transient analysis, [109](#), [110](#)
 FACTOR input, transient potential flow, [319](#)
 Factor of safety, [282](#)
 FACTOR parameter, direct response analysis, [151](#)
 FACTOR parameter, eigenvalue analysis, [430](#), [431](#)
 FACTOR parameter, HFTD analysis, [191](#)
 FACTOR parameter, initial conditions, [220](#)
 FACTOR parameter, nonlinear initial state, [222](#)
 FACTOR parameter, nonlinear time step, [233](#)
 FACTOR parameter, steady-state groundwater flow, [351](#), [352](#), [354](#)
 FACTOR parameter, steady-state heat flow, [334](#), [336](#)
 FACTOR parameter, transient groundwater flow, [357](#)
 FACTOR parameter, transient heat flow, [339](#)
 Factorization, [540](#)
 Factorization, out-of-core direct solution, [423](#)
 Factorization, Parallel Direct Sparse Solver, [422](#)
 Factorization, Sparse Cholesky based solution method, [422](#)
 FAHREN input, [4](#)
 Fahrenheit, temperature unit, [4](#)
 Failure criteria, composites, [84](#)
 Fatigue failure, [97](#), [98](#)
 FB output, frequency response analysis, [166](#)
 FB output, linear static analysis, [89](#)
 FB output, nonlinear analysis, [271](#)
 FB output, response spectrum analysis, [184](#)
 FB output, transient analysis, [139](#)
 FBXYZ output, response spectrum analysis, [184](#)
 FCK input, design checking, [448](#)
 FCTM input, design checking, [448](#)
 FD output, [140](#), [272](#)
 FE output, [89](#)
 FEAST command, [432](#)
 FEAST method based eigenvalue analysis, [434](#)
 FED output, nonlinear analysis, [273](#)
 FED output, transient analysis, [141](#)
 FEE output, linear static analysis, [91](#)
 FEE output, nonlinear analysis, [273](#)
 FEE output, response spectrum analysis, [187](#)
 FEE output, transient analysis, [141](#)
 FEM output, nonlinear analysis, [273](#)
 FEM output, transient analysis, [141](#)
 FEMA 273, [104](#)
 FEMVIE output device, [56](#), [66](#)
 FEMVIEW command, [520](#)
 FER output, linear static analysis, [91](#)
 FER output, nonlinear analysis, [273](#)
 FER output, response spectrum analysis, [187](#)
 FER output, transient analysis, [141](#)
 FET output, linear static analysis, [91](#)
 FET output, nonlinear analysis, [273](#)
 FET output, response spectrum analysis, [187](#)
 FET output, transient analysis, [141](#)
 FI output, [89](#)
 fib fiber reinforced concrete, stiffness adaptation analysis, [473](#)
 fib Model Code for Concrete Structures 2010, stiffness adaptation analysis, [473](#), [478](#)
 Fiber reinforced concrete, stiffness adaptation analysis, [473](#)
 FIELD command, buckling analysis, [303](#)
 FIELD parameter, nonlinear initial state, [222](#)
 FIELD parameter, steady-state groundwater flow, [354](#)
 FIELD parameter, steady-state heat flow, [336](#)
 FIELD parameter, transient groundwater flow, [357](#)
 FIELD parameter, transient heat flow, [339](#)
 FILE option, [49](#)
 FILE parameter, FILOS file, [49](#)
 FILE parameter, input reading, [50](#)
 FILE parameter, output, [50](#), [57](#)
 FILMA subroutine, [656](#)
 *FILOS command, [49](#)
 FILOS file, access subroutines, [669](#)
 FILOS file, maintenance, [48](#)
 FILOS file, commands, [49](#)
 FINDEQ command, [282](#)
 Finite element formulation, potential flow analysis, [627](#)
 FIRST option, creep, [214](#)
 FIRST option, plasticity, [214](#)
 First order displacement, [618](#)
 FIX subtable of 'TYINGS', [26](#), [326](#), [326](#)
 FIXED command, loads optimization, [515](#)
 Fixed displacements, *see* Deformation load
 'FIXHEA' table, [315](#)
 FIXMPC subtable of 'TYINGS', [27](#), [326](#)
 'FIXPOT' table, [316](#)
 'FIXTEM' table, [315](#)
 FL output, groundwater flow, [362](#)
 FL output, heat flow, [345](#)
 FL output, lubrication analysis, [374](#)

- Flat shell elements, stability analysis, [298](#)
- FLB output, [345](#), [363](#)
- FLE output, [345](#), [363](#)
- FLOW command, groundwater flow, [361](#), [363](#)
- FLOW command, heat flow, [343](#)
- FLOW command, heat transfer, [345](#)
- Flow–stress analysis, [377](#)
- Flow–stress analysis, mixture analysis, [377](#), [378](#)
- Flow–stress analysis, one-directional, [377](#), [378](#)
- Flow–stress analysis, staggered analysis, [377](#), [378](#)
- Flow–stress analysis, two-directional, [377](#), [378](#)
- Flows, external, [345](#), [363](#)
- Flows, groundwater flow, [362](#)
- Flows, heat transfer, [345](#)
- Flows, reaction, [345](#), [363](#)
- Flows, residual, [345](#), [363](#)
- FLR output, [345](#), [363](#)
- Fluid density, [6](#), [631](#)
- Fluid density, fluid–structure interaction, [93](#), [276](#)
- Fluid density, staggered analysis, [382](#)
- Fluid flow, [348](#)
- Fluid–structure interaction, background theory, [595](#)
- Fluid–structure interaction, dynamic pressures, [142](#), [167](#), [279](#)
- Fluid–structure interaction, eigenvalue analysis, [425](#)
- Fluid–structure interaction, fluid density, [93](#), [276](#)
- Fluid–structure interaction, frequency response analysis, [143](#)
- Fluid–structure interaction, HFTD analysis, [189](#)
- Fluid–structure interaction, response spectrum analysis, [169](#)
- Fluid–structure interaction, transient analysis, [122](#)
- FLUX command, groundwater flow, [361](#), [362](#)
- FLUX command, heat flow, [343](#), [344](#)
- FLUX command, lubrication analysis, [373](#), [374](#)
- Flux, external, [628](#)
- Flux, groundwater flow, [362](#)
- Flux, heat flow, [344](#)
- Flux, lubrication analysis, [374](#)
- Flux, mixture analysis, [396](#)
- FM output, [140](#), [272](#)
- FMAX parameter, [434](#)
- FMIN parameter, [434](#)
- FNE output, linear static analysis, [90](#)
- FNE output, nonlinear analysis, [272](#)
- FNE output, response spectrum analysis, [186](#)
- FNE output, transient analysis, [140](#)
- FNR output, linear static analysis, [90](#)
- FNR output, nonlinear analysis, [272](#)
- FNR output, response spectrum analysis, [186](#)
- FNR output, transient analysis, [140](#)
- FNT output, linear static analysis, [90](#)
- FNT output, nonlinear analysis, [272](#)
- FNT output, response spectrum analysis, [186](#)
- FNT output, transient analysis, [140](#)
- Foot, length unit, [4](#)
- FORCE command, frequency response analysis, [166](#)
- FORCE command, HFTD analysis, [195](#), [198](#)
- FORCE command, linear static analysis, [88](#)
- FORCE command, mixture analysis, [396](#)
- FORCE command, nonlinear analysis, [239](#), [270](#)
- FORCE command, nonlinear stop criterion, [242](#)
- FORCE command, response spectrum analysis, [184](#)
- FORCE command, stiffness adaptation analysis, [482](#)
- FORCE command, transient analysis, [138](#)
- FORCE input, structural analysis, [33](#), [46](#)
- FORCE input, units, [4](#)
- Force norm, [558](#)
- FORCE option, [57](#)
- Force output, frequency response analysis, [164](#)
- Force output, linear static analysis, [86](#)
- Force output, transient analysis, [136](#)
- Force units, [4](#)
- Force, internal vector, [532](#)
- Force, nonlinear analysis, [270](#)
- Force, nonlinear analysis output, [259](#)
- Force, response spectrum analysis, [182](#)
- Force, transient analysis, [138](#)
- Forchheimer’s law, [348](#)
- *FORTRAN command, [645](#)
- Fortran compiler, [645](#)
- Forward analysis, logging, [509](#)
- Forward substitution, [540](#)
- FR output, linear static analysis, [89](#)
- FR output, nonlinear analysis, [271](#)
- FR output, response spectrum analysis, [185](#)
- FR output, transient analysis, [139](#)
- Fraction model, [250](#)
- Fraction model, parameter estimation, [499](#)
- FRACTU command, HFTD analysis, [198](#)
- FRACTU command, linear static analysis, [95](#)
- FRACTU command, nonlinear analysis, [278](#)
- Fracture mechanics, [639](#)
- Fracture mechanics, energy release rate, [95](#), [278](#)
- Fracture mechanics, linear static analysis, [95](#)
- Fracture mechanics, nonlinear analysis, [278](#)
- Fracture mechanics, stress intensity factor, [95](#), [278](#)
- FRCCMD input, stiffness adaptation analysis, [473](#)
- FRCCON input, stiffness adaptation analysis, [473](#)
- FRCEPS input, stiffness adaptation analysis, [473](#)
- Free vibration, [145](#), [200](#), [437](#), [542](#)
- Free vibration eigenvalue analysis, [425](#), [429](#)
- Free vibration eigenvalue analysis, response spectrum analysis, [170](#)
- FREEVI command, [429](#)
- ’FREQLQ’ table, [111](#)
- *FREQUE command, [149](#)
- FREQUE command, [153](#)
- FREQUE command, eigenvalue analysis, [436](#)
- FREQUE input, complex loads, [113](#), [114](#)
- FREQUE input, direct input, [111](#)
- FREQUE input, external file, [112](#)
- FREQUE parameter, [204](#)
- FREQUE parameter, direct response analysis, [152](#)

- FREQUE parameter, modal response analysis, [148](#)
 Frequency domain analysis, input, [110](#)
 Frequency motion, [545](#)
 Frequency response analysis, [143](#), [149](#), [588](#)
 Frequency–acceleration spectrum, [169](#)
 Frequency–load diagram, [111](#)
 FRICTI option, linear static analysis, [83](#), [87](#)
 FRICTI option, nonlinear analysis, [256](#), [268](#)
 FRXYZ output, response spectrum analysis, [185](#)
 FSDyex output, linear static analysis, [85](#)
 FSDyex output, nonlinear analysis, [258](#)
 FSDyex output, transient analysis, [136](#)
 FSDyun output, linear static analysis, [85](#)
 FSDyun output, nonlinear analysis, [258](#)
 FSDyun output, transient analysis, [136](#)
 FSHIFT parameter, eigenvalue analysis, [433](#), [435](#)
 FSPRES command, frequency response analysis, [167](#)
 FSPRES command, HFTD analysis, [198](#)
 FSPRES command, nonlinear analysis, [279](#)
 FSPRES command, transient analysis, [142](#)
 FSstun output, linear static analysis, [85](#)
 FSstun output, nonlinear analysis, [258](#)
 FSstun output, transient analysis, [136](#)
 FSstus output, linear static analysis, [85](#)
 FSstus output, nonlinear analysis, [258](#)
 FSstus output, transient analysis, [136](#)
 FT input, [4](#)
 FT output, nonlinear analysis, [270](#)
 FT output, transient analysis, [138](#)
 FTALPH output, [266](#)
 Ftu output, [258](#)
 FTX output, [266](#)
 FULL command, HFTD analysis, [195](#)
 FULL command, nonlinear analysis, [243](#)
 FULL option, [509](#)
 FULLEL option, [53](#)
 'FUNCTI' table, nodal loads, [12](#)
 'FUNCTI' table, prescribed accelerations, [12](#)
 'FUNCTI' table, prescribed displacements, [12](#)
 FX⁺, output for postprocessing, [67](#)
 FXPLUS output device, [56](#), [67](#)
- ## G
- G input, [4](#)
 G output, fracture mechanics, [95](#), [278](#)
 G output, frequency response analysis, [160](#)
 G output, linear static analysis, [80](#)
 G output, nonlinear analysis, [250](#)
 G output, response spectrum analysis, [178](#)
 G output, transient analysis, [132](#)
 Galerkin integration, potential flow analysis, [627](#), [628](#)
 Galerkin integration, soil–pore fluid analysis, [635](#)
 Galerkin integration, transient groundwater flow, [358](#)
 Galerkin integration, transient heat flow, [341](#)
 GAMMA input, stiffness adaptation analysis, [477](#)
 GAMMA option, [265](#)
 GAMMA parameter, engineering liquefaction analysis, [289](#)
 GAMMA parameter, nonlinear analysis, [217](#)
 GAMMA parameter, nonlinear load step, [226](#)
 GAMMA parameter, nonlinear time step, [233](#)
 GAMMA parameter, strength reduction analysis, [284](#)
 GAMMAR option, linear static analysis, [93](#)
 GAMMAR option, nonlinear analysis, [276](#)
 Gauss theorem, [628](#)
 Gauss–Jordan inversion, [654](#)
 GAUSSIAN option, results selection, [520](#)
 GC input, stiffness adaptation analysis, [478](#)
 Gc output, [251](#)
 Gcon output, [252](#)
 Gcrw output, [252](#), [483](#)
 Ge output, [250](#)
 GENEL command, [421](#), [422](#)
 General connections, [26](#), [325](#)
 Generalized eigenproblem, [542](#)
 Generalized mass, [437](#), [542](#)
 Generalized Minimal Residual, [547](#)
 GEOMET command, [212](#), [215](#)
 GEOMET input, [499](#)
 Geometric nonlinearity, [215](#), [566](#)
 Geometric stress–stiffness matrix, [542](#)
 Geotechnics, [255](#)
 Geotechnics, Darcy flow, [633](#)
 Geotechnics, initial state, [219](#), [244](#)
 GF1 input, stiffness adaptation analysis, [472](#), [473](#)
 GF1 option, linear static analysis, [93](#)
 GF1 option, nonlinear analysis, [276](#)
 GF1 output, linear static analysis, [95](#)
 GF1 output, nonlinear analysis, [278](#)
 Gk output, crack strains, [251](#)
 Gk output, summed crack strains, [251](#)
 GLOBAL option, [57](#)
 GMRES, [547](#)
 GOVIND option, parameter output, [274](#)
 Gp output, [251](#)
 GRADIE option, [87](#)
 Gram, mass unit, [4](#)
 Gram–force, force unit, [4](#)
 GRAVAC input, [6](#)
 GRAVDI input, [6](#)
 GRAVIT option, [138](#), [265](#)
 Gravitational energy, specific, [137](#), [264](#)
 Gravity acceleration, [3](#), [4](#), [6](#)
 Gravity acceleration, staggered analysis, [382](#)
 Gravity direction, [6](#)
 Gravity, soil–pore fluid analysis, [633](#)
 GREEN option, [57](#)
 Green–Lagrange strain, frequency response analysis, [160](#)
 Green–Lagrange strain, linear static analysis, [80](#)
 Green–Lagrange strain, nonlinear analysis, [250](#)
 Green–Lagrange strain, response spectrum analysis, [178](#)
 Green–Lagrange strain, transient analysis, [132](#)
 GRID command, element evaluation, [53](#)
 GROUND input, design spectrum, [116](#)
 GROUND input, elastic response spectrum, [114](#)
 Groundwater flow analysis, [347](#)
 Groundwater flow, staggered analysis, [382](#)

Groups, [7](#)
 'GROUPS' table, [9](#), [10](#)
 *GROWSS command, [350](#)
 *GROWTR command, [355](#)
 GSTAR option, [265](#)
 GTC subroutine, [669](#)
 Gtmp output, [252](#)

H

H input, prescribed head, [318](#)
 H output, groundwater flow, [362](#)
 Hcr output, [278](#)
 HEAD command, groundwater flow, [361](#), [361](#)
 HEAD subtable of 'INIVAR', [316](#)
 Heat flow, [542](#)
 Heat flow analysis, [329](#)
 Heat flow analysis, nonlinear transient, [340](#)
 *HEATSS command, [332](#)
 *HEATTR command, [337](#)
 HFTD, *see* Hybrid frequency time domain analysis
 *HFTD command, [189](#)
 HHT command, nonlinear analysis, [217](#)
 Hilber–Hughes–Taylor time integration, [217](#), [593](#), [635](#)
 HILL output, [85](#)
 Hoek-Brown failure criterion, [582](#)
 Hoek-Brown failure criterion, linear static analysis, [83](#), [87](#)
 Hoek-Brown failure criterion, nonlinear analysis, [256](#), [268](#)
 Hoek-Brown failure criterion, transient analysis, [135](#)
 HOEKBR option, linear static analysis, [83](#), [87](#)
 HOEKBR option, nonlinear analysis, [256](#), [268](#)
 HOEKBR option, transient analysis, [135](#)
 Hooke's law, [534](#)
 Hordijk tension softening, stiffness adaptation analysis, [472](#)
 HORDYK input, stiffness adaptation analysis, [472](#)
 HORIZO input, Eurocode 8 EN 1998-1, [114](#), [116](#)
 HORIZO input, NPR 9998:2015, [117](#), [118](#)
 HOUR input, [4](#)
 Hour, time unit, [4](#)
 HPR output, [362](#)
 Hybrid frequency time domain analysis, [189](#), [601](#)
 HYDRAT command, [340](#), [342](#)
 Hydration, staggered analysis, [378](#), [382](#)
 Hydration, transient heat flow, [331](#), [341](#), [345](#)
 Hydration, transient heat flow analysis, [340](#)
 Hydraulic capacity, [633](#)
 Hydraulic conductivity, [347](#)
 Hydraulic fracture, [386](#)
 Hydraulic head, groundwater flow, [361](#)
 Hydraulic head, staggered analysis, [377](#)
 Hydraulic pore pressure, [637](#)
 Hydrostatic capacity, staggered analysis, [377](#)
 Hydrostatic pressure capacity, [581](#), [582](#)
 Hydrostatic pressure capacity Coulomb friction, [583](#)

Hydrostatic pressure capacity Coulomb friction, linear static analysis, [87](#)
 Hydrostatic pressure capacity Coulomb friction, nonlinear analysis, [268](#)
 Hydrostatic pressure capacity Hoek-Brown, linear static analysis, [87](#)
 Hydrostatic pressure capacity Hoek-Brown, nonlinear analysis, [268](#)
 Hydrostatic pressure capacity Mohr-Coulomb, linear static analysis, [87](#)
 Hydrostatic pressure capacity Mohr-Coulomb, nonlinear analysis, [268](#)
 HYPERE command, [215](#)
 Hyperelasticity, [215](#), [566](#)
 Hysteresis, *see* Damping, structural

I

IApr output, [395](#)
 Icr output, [258](#)
 ID output, [127](#)
 IDENTI command, [430](#)
 Identification, [490](#)
 Identity matrix, [430](#), [542](#)
 IDpr output, [395](#)
 IDXYZ output, transient analysis, [127](#)
 Ill-conditioned system, [549](#)
 ILU option, [423](#)
 ILU preconditioning, [423](#), [548](#)
 IMAGIN input, frequency domain analysis, [113](#)
 IMOVE subroutine, [664](#)
 IMPERF command, [300](#), [302](#)
 Imperfection pattern, [302](#)
 Imperfection, random, [303](#)
 Imperfections, [296](#), [302](#), [620](#)
 Imperfections, user-specified, [621](#)
 Implicit time integration, phased analysis, [416](#)
 Implicit time integration, potential flow analysis, [628](#)
 Implicit time integration, structural dynamics, [593](#)
 IMPORT input, Eurocode 8 EN 1998-1, [114](#), [116](#)
 IMPORT input, frequency domain analysis, [112](#)
 IMPORT input, NPR 9998:2015, [117](#), [118](#)
 IMPORT input, phased analysis, [403](#)
 IMPORT input, transient analysis, [110](#)
 IMPORT input, transient potential flow, [319](#)
 Import of deformations, [402](#)
 IMPORT option, nodal results superposition, [406](#)
 Impulsive loading, [586](#)
 IN input, [4](#)
 INCFS command, [282](#)
 Inch, length unit, [4](#)
 Incompatible modes, [385](#)
 Incomplete LU-decomposition preconditioning, *see* ILU preconditioning
 INCREM option, displacements, [126](#)
 INCREM option, nonlinear stop criterion, [241](#)
 INCREM option, output selection, [57](#)
 Incremental displacement vector length, transient analysis, [127](#)
 Incremental procedures, [237](#), [560](#)
 INDEX parameter, [508](#), [509](#)

- Indirect Displacement control, [227](#), [562](#)
- Induced earthquakes, [117](#), [118](#)
- INERTI option, nonlinear analysis, [271](#), [273](#)
- INERTI option, transient analysis, [139](#), [141](#)
- Inertia forces, element, [141](#), [273](#)
- Inertia forces, nodal element, [139](#), [271](#)
- INFLUE subtable of 'LOADS', [46](#)
- Influence field, [46](#)
- INFORM command, [49](#)
- INIFS command, [282](#)
- INISIZ parameter, engineering liquefaction analysis, [289](#)
- INISIZ parameter, engineering liquefaction analysis analysis, [289](#)
- INISIZ parameter, nonlinear load step, [226](#), [227](#)
- INISIZ parameter, nonlinear time step, [232](#)
- INISIZ parameter, strength reduction analysis, [284](#)
- INITIA command, FILOS file, [49](#)
- INITIA command, nonlinear analysis, [221](#)
- INITIA command, structural nonlinear analysis, [220](#)
- INITIA command, transient groundwater flow, [357](#)
- INITIA command, transient heat flow, [339](#)
- INITIA option, output selection, [57](#)
- Initial conditions, nonlinear analysis, [221](#), [223](#)
- Initial conditions, phased analysis, [220](#)
- Initial degree of reaction, [340](#)
- Initial displacement, stability analysis, [296](#)
- Initial displacement, transient analysis, [107](#)
- Initial equilibrium, [391](#)
- Initial equivalent age, [340](#)
- Initial forces, linear static analysis, [86](#)
- Initial potential field, [316](#)
- Initial state reference, [219](#), [222](#), [244](#)
- Initial state, structural nonlinear analysis, [220](#)
- Initial stress, linear static analysis, [85](#)
- Initial stress, phased analysis, [411](#)
- Initial tractions, [87](#)
- Initial velocity, [107](#)
- Initialization of FILOS file, [48](#)
- Initialization subroutines, [663](#)
- 'INIVAR' table, stability analysis, [296](#)
- 'INIVAR' table, potential flow analysis, [316](#)
- 'INIVAR' table, random imperfection pattern, [303](#)
- 'INIVAR' table, transient analysis, [106](#)
- 'INIVAR' table, user-specified imperfection pattern, [303](#)
- Inner-products subroutines, [658](#)
- *INPUT command, [50](#)
- *INPUT command, phased analysis, [402](#), [417](#)
- Input commands, [50](#)
- Input data, dynamic analysis, [103](#)
- Input data, finite element model, [3](#)
- Input data, model codes, [3](#)
- Input data, nonlinear analysis, [209](#)
- Input data, parameter estimation, [493](#)
- Input data, potential flow analysis, [315](#)
- Input data, steel profiles, [3](#)
- Input data, structural analysis, [15](#)
- Input file, [3](#)
- INPUT option, direct response analysis, [151](#)
- INPUT option, eigenvalue analysis, [430](#), [431](#)
- INPUT option, HFTD analysis, [191](#)
- INPUT option, initial displacements, [222](#)
- INPUT option, initial stresses, [222](#)
- INPUT option, initial velocities, [222](#)
- INPUT option, steady-state groundwater flow, [354](#)
- INPUT option, steady-state heat flow, [336](#)
- INPUT option, transient groundwater flow, [357](#)
- INPUT option, transient heat flow, [339](#)
- Input reading, [50](#)
- Instability, spatial, [636](#)
- Insulation, [627](#)
- Insulation, staggered analysis, [378](#)
- Integration point coordinates, tabular output, [58](#)
- Integration points, output selection, [58](#), [60](#), [62](#)
- INTER1 subroutine, [666](#)
- INTER2 subroutine, [666](#)
- Interaction analysis, [385](#), [631](#)
- Interconnection tying, [21](#), [324](#)
- INTERF command, [215](#)
- INTERF option, [53](#)
- Interface elements, mass density, [93](#), [275](#)
- Interface elements, mixture analysis, [386](#)
- Interface elements, stability analysis, [298](#)
- INTERN option, r.h.s. vectors, [89](#)
- Internal beam arm, [579](#)
- Internal element forces, [91](#)
- Internal element forces, nonlinear analysis, [273](#)
- Internal element forces, response spectrum analysis, [186](#)
- Internal element forces, transient analysis, [141](#)
- Internal force vector, [89](#), [532](#)
- Internal forces, element, [141](#), [273](#)
- Internal forces, nodal element, [139](#), [271](#)
- Internal nodal element forces, linear static analysis, [90](#)
- Internal nodal element forces, nonlinear analysis, [272](#)
- Internal nodal element forces, response spectrum analysis, [186](#)
- Internal nodal element forces, transient analysis, [140](#)
- Internal nodal reinforcement forces, linear static analysis, [90](#)
- Internal nodal reinforcement forces, nonlinear analysis, [272](#)
- Internal nodal reinforcement forces, response spectrum analysis, [186](#)
- Internal nodal reinforcement forces, transient analysis, [140](#)
- Internal nodal total forces, linear static analysis, [90](#)
- Internal nodal total forces, nonlinear analysis, [272](#)
- Internal nodal total forces, response spectrum analysis, [185](#)
- Internal nodal total forces, transient analysis, [140](#)
- Interpolation functions, [531](#), [533](#)

Interpolation matrix, [533](#)
 Interpolation subroutines, [666](#)
 INTPNT command, [60](#), [62](#)
 INTPNT option, [58](#), [519](#)
 INTTMP command, [343](#), [346](#)
 INV2 subroutine, [653](#)
 INV3 subroutine, [654](#)
 INVARI option, [57](#)
 Inversion subroutines, [653](#)
 INVMTX subroutine, [654](#)
 INVPGB subroutine, [654](#)
 IrD output, [127](#)
 Iron, elasticity modulus, [4](#)
 Iron, thermal capacity, [4](#)
 Iron, thermal conductivity, [4](#)
 ISET subroutine, [664](#)
 Isoparametric elements, nonlinear analysis, [209](#)
 Isoparametric elements, stability analysis, [296](#)
 ITEM input, [499](#)
 Items, restoring, [49](#)
 Items, saving, [49](#)
 ITERAT command, engineering creep analysis, [291](#)
 ITERAT command, engineering liquefaction analysis, [287](#), [289](#)
 ITERAT command, heat transfer, [334](#), [335](#)
 ITERAT command, nonlinear analysis, [219](#), [235](#)
 ITERAT command, nonlinear initial state, [224](#)
 ITERAT command, nonlinear load step, [225](#)
 ITERAT command, nonlinear time step, [231](#)
 ITERAT command, parameter estimation, [509](#)
 ITERAT command, solution procedures, [421](#), [423](#)
 ITERAT command, steady-state groundwater flow, [352](#), [353](#)
 ITERAT command, stiffness adaptation analysis, [482](#)
 ITERAT command, strength reduction analysis, [282–284](#)
 ITERAT command, transient analysis, [124](#), [125](#)
 ITERAT command, transient groundwater flow, [359](#)
 ITERAT command, heat transfer, [342](#)
 ITERAT option, nonlinear analysis logging, [243](#)
 ITERAT option, nonlinear load step, [226](#)
 ITERAT option, nonlinear time step, [232](#)
 ITERAT option, transient heat flow, [342](#)
 Iterations, eigenvalue analysis, [433](#), [434](#)
 Iterative solution procedure, linear analysis, [423](#), [546](#)
 Iterative solution procedure, nonlinear analysis, [235](#), [552](#)
 Iterative solution procedure, potential flow analysis, [629](#)
 Iterative solution procedure, strength reduction analysis, [284](#)
 IVpr output, [395](#)

J

Jacobi preconditioning, [423](#), [548](#)
 JANFOR option, linear static analysis, [79](#), [84](#)
 JANFOR option, nonlinear analysis, [250](#), [257](#)
 JANMOM option, linear static analysis, [79](#), [84](#)

JANMOM option, nonlinear analysis, [250](#), [257](#)
 Janssen model, output, [79](#), [84](#), [250](#), [257](#)
 Jaumann rate, [568](#)
 Job logging, nonlinear analysis, [243](#)
 JSCE 2012, Damage Index, [262](#)
 JSCE 2012, normalized cumulative energy, [262](#)
 JSCE tension softening, stiffness adaptation analysis, [472](#)
 JSCE tension stiffening, stiffness adaptation analysis, [473](#)
 JSCE, result averaging over a region, [61](#)
 JSCESO input, stiffness adaptation analysis, [472](#)
 JSCETS input, stiffness adaptation analysis, [473](#)

K

K output, fracture mechanics, [95](#), [278](#)
 K output, frequency response analysis, [161](#)
 K output, linear static analysis, [80](#)
 K output, nonlinear analysis, [252](#)
 K output, response spectrum analysis, [179](#)
 K output, transient analysis, [133](#)
 DESSPC input, design spectrum, [118](#)
 DESSPC input, elastic response spectrum, [117](#)
 Kalman filtering, [490](#)
 KAPSIG input, stiffness adaptation analysis, [469](#)
 KELVIN input, units, [4](#)
 Kelvin, temperature unit, [4](#)
 KG input, [4](#)
 KGF input, [4](#)
 Kilo-newton, force unit, [4](#)
 Kilo-pond, force unit, [4](#)
 Kilo-pound, force unit, [4](#)
 Kilogram, mass unit, [4](#)
 Kilogram-force, force unit, [4](#)
 Kilometer, length unit, [4](#)
 Kilopound, mass unit, [4](#)
 KINETI option, [137](#), [263](#)
 Kinetic energy, specific, [137](#), [263](#)
 KLB input, [4](#)
 KLBF input, [4](#)
 KM input, [4](#)
 KN input, [4](#)
 KOTSOV option, [266](#)
 Kotsovos concrete model, Damage Index, [262](#)
 Kotsovos concrete model, elastic energy, [266](#)
 Kotsovos concrete model, normalized cumulative energy, [262](#)
 Kotsovos concrete model, plastic energy, [266](#)
 Kotsovos concrete model, status output, [266](#)
 Kotsovos concrete model, total strain energy, [266](#)
 Kozeny–Karman modified permeability, [633](#)
 KP input, [4](#)
 Krylov subspace, [546](#)
 Krylov subspace, Conjugate Gradient, [547](#)
 Krylov subspace, Generalized Minimal Residual, [547](#)
 KTHORE input, stiffness adaptation analysis, [476](#)

L

Lagrange, [209](#), [216](#), [567](#), [568](#)

- LAMBDA option, linear static analysis, [93](#)
- LAMBDA option, nonlinear analysis, [276](#)
- LAMBDA output, linear static analysis, [95](#)
- LAMBDA output, nonlinear analysis, [278](#)
- LAMBDA parameter, [309](#)
- Large deformations, [619](#)
- Lateral pressure ratio, [245](#)
- LAYOUT command, [55](#), [65](#)
- LAYOUT command, cross-section analysis, [369](#)
- LAYOUT command, groundwater flow, [361](#)
- LAYOUT command, heat flow, [343](#)
- LAYOUT command, HFTD analysis, [197](#)
- LAYOUT command, lubrication analysis, [373](#)
- LAYOUT command, nonlinear analysis output, [247](#)
- Layout of tabular output, [65](#)
- LB input, [4](#)
- LBF input, [4](#)
- LBOUND input, [116](#)
- LDU decomposition, [539](#)
- LEFM, *see* Fracture mechanics
- LEFM, energy release rate, [95](#), [278](#)
- LEFM, stress intensity factor, [95](#), [278](#)
- LENGTH input, [4](#)
- Length units, [4](#)
- LEVELS option, contour plot, [520](#)
- Libraries of subroutines, [648](#)
- Line Search, [236](#), [237](#), [557](#), [571](#)
- Linear analysis, parameter estimation, [517](#)
- Linear analysis, static, [71](#)
- Linear buckling, *see* Euler buckling
- LINEAR command, nonlinear analysis, [237](#)
- LINEAR command, steady-state groundwater flow, [352](#)
- LINEAR command, steady-state heat flow, [333](#)
- LINEAR command, transient groundwater flow, [357](#)
- LINEAR command, transient heat flow, [339](#)
- Linear constraints, *see* Tying
- Linear Elastic Fracture Mechanics, *see* Fracture mechanics
- LINEAR input, stiffness adaptation analysis, [472](#)
- LINEAR option, eigenvalue analysis, [429](#), [431](#)
- LINEAR option, nonlinear analysis, [237](#)
- Linear Stiffness iteration, [237](#), [556](#)
- Linear structural analysis, transient, [121](#)
- Linearized elements, status output, [267](#)
- LINELM option, [267](#)
- LINELM output, [267](#)
- LINEPS input, stiffness adaptation analysis, [472](#)
- LINESE command, [236](#), [237](#)
- LINESE command, strength reduction analysis, [285](#)
- LINHAR input, stiffness adaptation analysis, [476](#)
- LINPAG command, [65](#)
- *LINSTA command, [71](#)
- LINSTA command, [508](#)
- LINSTR command, [212](#)
- LIQUEF command, [235](#)
- Liquefaction of soil, [235](#)
- Liquefaction of soil, mixture analysis, [390](#)
- Liquefaction status, [265](#)
- LMOVE subroutine, [665](#)
- Load, [31](#), [536](#)
- Load case, [32](#)
- Load combinations, [45](#)
- Load combinations, design checking, [453](#)
- LOAD command, initial conditions, [220](#)
- LOAD command, mixture analysis, [392](#)
- LOAD command, nonlinear load step, [219](#), [224](#)
- LOAD command, nonlinear stop criterion, [240](#), [241](#)
- LOAD command, response spectrum analysis, [173](#)
- LOAD command, stiffness adaptation analysis, [482](#)
- LOAD command, transient analysis, [124](#)
- LOAD command, transient load step, [124](#)
- Load control, [560](#)
- Load cycles to failure, [98](#)
- LOAD input, Eurocode 8 EN 1998-1, [111](#)
- LOAD input, frequency-load diagram, [111](#)
- LOAD input, NPR 9998:2015, [111](#)
- LOAD input, parameter estimation, [499](#)
- LOAD input, time-load diagram, [108](#)
- LOAD parameter, [432](#)
- LOAD parameter, buckling analysis, [301](#)
- LOAD parameter, direct response analysis, [151](#)
- LOAD parameter, eigenvalue analysis, [430](#), [431](#)
- LOAD parameter, HFTD analysis, [191](#)
- LOAD parameter, loads optimization, [515](#)
- LOAD parameter, nonlinear initial state, [222](#)
- LOAD parameter, parameter estimation, [508](#)
- Load set, [45](#)
- Load set, selection, [75](#)
- Load steps, mixture analysis, [392](#)
- Load steps, nonlinear analysis, [219](#), [224](#)
- Load steps, transient analysis, [124](#), [124](#)
- Load vector, output, [89](#)
- Load vector, setup, [536](#)
- Load, import of deformations, [402](#)
- LOADFA parameter, [202](#), [306](#)
- Loading, nonlinear stop criterion, [241](#)
- LOADNR parameter, initial nonlinear conditions, [220](#)
- LOADNR parameter, nonlinear load step, [224](#)
- LOADNR parameter, transient analysis, [124](#)
- LOADS command, design checking, [453](#)
- LOADS command, fatigue failure, [98](#)
- LOADS command, HFTD analysis, [190](#)
- LOADS command, linear static analysis, [72](#), [75](#), [536](#)
- LOADS command, loads optimization, [514](#), [515](#)
- LOADS command, nonlinear analysis, [212](#), [216](#)
- LOADS command, stiffness adaptation analysis, [481](#)
- LOADS command, transient analysis, [123](#)
- Loads optimization, [513](#)
- 'LOADS' table, [31](#)
- 'LOADS' table, mixture analysis, [388](#), [389](#)
- 'LOADS' table, phased analysis, [408](#)
- 'LOADS' table, pushover analysis, [105](#)
- 'LOADS' table, staggered analysis, [377](#), [378](#)
- 'LOADS' table, transient analysis, [104](#)
- Loads, spatial functions, [12](#)
- LOALEN input, [42](#)
- LOARED input, [42](#)

LOCAL option, [57](#)
 Lode angle, [576](#)
 Lode output, linear static analysis, [85](#)
 Lode output, nonlinear analysis, [257](#)
 Lode output, response spectrum analysis, [182](#)
 Lode output, transient analysis, [135](#)
 LODFAC input, [39](#)
 LODFAC input, Dutch Code, [42](#)
 LODFAC input, European Code, [44](#)
 LODSET input, [403](#)
 LODSUP command, [173](#)
 LOG command, [47](#)
 LOGGIN command, engineering creep analysis, [291](#)
 LOGGIN command, engineering liquefaction analysis, [287](#)
 LOGGIN command, HFTD analysis, [194](#), [195](#)
 LOGGIN command, nonlinear analysis, [219](#), [243](#)
 LOGGIN command, strength reduction analysis, [282](#)
 Logging a job, [47](#)
 Logging a job, HFTD analysis, [194](#), [195](#)
 Logging a job, nonlinear analysis, [219](#), [243](#)
 Logging a job, parameter estimation, [509](#)
 Long term loading, [586](#)
 Long ton, mass unit, [4](#)
 LORRY input, [38](#)
 LSET subroutine, [664](#)
 LTN input, [4](#)
 Lubrication, [371](#)
 Lumped mass, [542](#)
 LUMPED option, direct response analysis, [151](#)
 LUMPED option, eigenvalue analysis, [430](#)
 LUMPED option, HFTD analysis, [191](#)
 LUMPED option, nonlinear analysis, [218](#)
 LUMPED option, transient groundwater flow, [356](#)
 LUMPED option, transient heat flow, [338](#)
 Lumping, [585](#)

M

M input, [4](#)
 M output, frequency response analysis, [165](#)
 M output, linear static analysis, [86](#)
 M output, nonlinear analysis, [259](#)
 M output, response spectrum analysis, [183](#)
 M output, transient analysis, [136](#)
 M0 output, [86](#)
 MA output, [270](#)
 Machine parameters, [665](#)
 Maekawa–Fukuura concrete model, Damage Index, [262](#)
 Maekawa–Fukuura concrete model, normalized cumulative energy, [262](#)
 MASS command, direct response analysis, [151](#)
 MASS command, eigenvalue analysis, [430](#)
 MASS command, HFTD analysis, [191](#)
 MASS command, modal response analysis, [146](#)
 MASS command, nonlinear analysis, [218](#)
 MASS command, nonlinear vibration response analysis, [201](#)
 Mass conservation, [633](#), [637](#)
 Mass density, dynamic analysis, [103](#)

Mass elements, stability analysis, [298](#)
 MASS input, [4](#)
 Mass matrices, [218](#)
 Mass units, [4](#)
 Mass, added, [599](#)
 Mass, dynamic analysis, [103](#)
 Master node, [16](#), [320](#)
 MATERI input, [499](#)
 'MATERI' table, phased analysis, [418](#)
 Material codes, *see* Model codes
 Material points, [495](#)
 Material properties, phased analysis, [406](#)
 MATPRI subroutine, [662](#)
 MATRIX command, cross-section analysis, [366](#)
 MATRIX command, design checking, [453](#)
 MATRIX command, direct response analysis, [150](#), [151](#)
 MATRIX command, HFTD analysis, [190](#)
 MATRIX command, linear static analysis, [72](#), [536](#)
 MATRIX command, loads optimization, [514](#)
 MATRIX command, lubrication analysis, [372](#)
 MATRIX command, nonlinear analysis, [212](#)
 MATRIX command, steady-state groundwater flow, [351](#)
 MATRIX command, steady-state heat flow, [333](#)
 MATRIX command, stiffness adaptation analysis, [481](#)
 MATRIX command, transient analysis, [123](#)
 MATRIX command, transient groundwater flow, [356](#)
 MATRIX command, transient heat flow, [338](#)
 MATRIX option, [439](#)
 Matrix subroutines, [648](#)
 MATURI command, HFTD analysis, [198](#)
 MATURI command, nonlinear analysis, [215](#), [269](#)
 'MATURI' table, staggered analysis, [378](#)
 Maturity, [215](#), [269](#)
 Maturity, staggered analysis, [382](#)
 MAX command, [302](#), [303](#)
 MAX option, HFTD analysis, [198](#)
 MAX option, linear static analysis, [75](#)
 MAX option, nonlinear analysis, [248](#)
 MAX option, steady-state groundwater flow, [360](#)
 MAX option, steady-state heat flow, [343](#)
 MAX option, transient heat flow, [344](#), [361](#)
 MAX output, linear static analysis, [85](#)
 MAXAMP parameter, [202](#), [306](#)
 MAXFRE command, [194](#)
 Maximum aggregate size, stiffness adaptation analysis, [473](#)
 Maximum criterion, composites, [84](#)
 Maximum shear stress, linear static analysis, [82](#), [85](#)
 Maximum shear stress, nonlinear analysis, [254](#), [258](#)
 Maximum shear stress, response spectrum analysis, [181](#), [182](#)
 Maximum shear stress, transient analysis, [134](#), [136](#)
 Maximum values, steady-state groundwater flow, [360](#)
 Maximum values, steady-state heat flow, [343](#)

- Maximum values, transient groundwater flow, [361](#)
- Maximum values, transient heat flow, [344](#)
- MAXITE parameter, eigenvalue analysis, [433](#), [434](#)
- MAXITE parameter, heat transfer, [335](#)
- MAXITE parameter, HFTD analysis, [195](#)
- MAXITE parameter, iterative solution, [423](#)
- MAXITE parameter, nonlinear analysis, [236](#), [565](#)
- MAXITE parameter, parameter estimation, [509](#)
- MAXITE parameter, steady-state groundwater flow, [353](#)
- MAXITE parameter, stiffness adaptation analysis, [482](#)
- MAXITE parameter, strength reduction analysis, [285](#)
- MAXITE parameter, transient groundwater flow, [359](#)
- MAXITE parameter, transient heat flow, [342](#)
- MAXITE parameter, transient iteration, [125](#)
- MAXLS parameter, [237](#)
- MAXSHR option, linear static analysis, [82](#), [93](#)
- MAXSHR option, nonlinear analysis, [254](#), [276](#)
- MAXSHR option, response spectrum analysis, [181](#)
- MAXSHR option, transient analysis, [134](#)
- MAXSHR output, linear static analysis, [95](#)
- MAXSHR output, nonlinear analysis, [278](#)
- MAXSIZ parameter, engineering liquefaction analysis, [289](#), [290](#)
- MAXSIZ parameter, nonlinear load step, [224](#)
- MAXSIZ parameter, nonlinear time step, [231](#)
- MAXSIZ parameter, strength reduction analysis, [284](#)
- MAXSIZ parameter, strength reduction method, [284](#)
- MAXSTP parameter, nonlinear load step, [227](#)
- MAXSTP parameter, nonlinear time step, [234](#)
- MAXSTP parameter, strength reduction analysis, [284](#)
- MAXTHR command, eigenvalue analysis, [433](#), [434](#)
- MAXTHR command, iterative solver, [423](#)
- MAXTHR command, parallel direct sparse solver, [422](#)
- Maxwell Chain model, parameter estimation, [499](#)
- MB output, frequency response analysis, [166](#)
- MB output, linear static analysis, [89](#)
- MB output, nonlinear analysis, [271](#)
- MB output, response spectrum analysis, [184](#)
- MB output, transient analysis, [139](#)
- MC1990 input, stiffness adaptation analysis, [473](#), [478](#)
- MC2010 input, stiffness adaptation analysis, [473](#), [478](#)
- MD output, [140](#), [272](#)
- ME output, [89](#)
- MED output, nonlinear analysis, [273](#)
- MED output, transient analysis, [141](#)
- MEE output, linear static analysis, [91](#)
- MEE output, nonlinear analysis, [273](#)
- MEE output, response spectrum analysis, [187](#)
- MEE output, transient analysis, [141](#)
- MEM output, nonlinear analysis, [273](#)
- MEM output, transient analysis, [141](#)
- memore example, [517](#)
- MER output, linear static analysis, [91](#)
- MER output, nonlinear analysis, [273](#)
- MER output, response spectrum analysis, [187](#)
- MER output, transient analysis, [141](#)
- Mesh adaptation, [640](#)
- MESH option, display, [520](#)
- Mesh quality, [51](#)
- Mesh refinement, [21](#), [324](#)
- MET output, linear static analysis, [91](#)
- MET output, nonlinear analysis, [273](#)
- MET output, response spectrum analysis, [187](#)
- MET output, transient analysis, [141](#)
- Meter, length unit, [4](#)
- METHOD command, eigenvalue analysis, [432](#)
- METHOD command, heat transfer, [335](#)
- METHOD command, nonlinear analysis, [217](#), [236](#), [236](#)
- METHOD command, parameter estimation, [509](#)
- METHOD command, steady-state groundwater flow, [353](#)
- METHOD command, transient groundwater flow, [359](#)
- METHOD command, transient heat flow, [342](#)
- METHOD parameter, strength reduction analysis, [285](#)
- METIS, [547](#)
- Metis reordering, [541](#)
- Metric-ton, force unit, [4](#)
- MF parameter, [47](#)
- MI input, [4](#)
- MI output, [89](#)
- Mile, length unit, [4](#)
- Millimeter, length unit, [4](#)
- MIN input, [4](#)
- MIN option, HFTD analysis, [198](#)
- MIN option, linear static analysis, [75](#)
- MIN option, nonlinear analysis, [248](#)
- MIN option, steady-state groundwater flow, [360](#)
- MIN option, steady-state heat flow, [343](#)
- MIN option, transient groundwater flow, [361](#)
- MIN option, transient heat flow, [344](#)
- MINFRE command, [194](#)
- Minimum values, steady-state groundwater flow, [360](#)
- Minimum values, steady-state heat flow, [343](#)
- Minimum values, transient groundwater flow, [361](#)
- Minimum values, transient heat flow, [344](#)
- MINSIZ parameter, engineering liquefaction analysis, [289](#), [290](#)
- MINSIZ parameter, nonlinear load step, [224](#)
- MINSIZ parameter, nonlinear time step, [231](#)
- MINSIZ parameter, strength reduction analysis, [284](#)
- MINSIZ parameter, strength reduction method, [284](#)
- Minute, time unit, [4](#)
- MITERA parameter, creep, [214](#)
- MITERA parameter, plasticity, [214](#)
- Mixed boundary condition, [627](#)
- MIXTUR input, [386](#)
- Mixture analysis, [235](#), [348](#), [377](#), [378](#), [385](#), [631](#)

- Mixture elements, [209](#), [378](#), [385](#), [386](#), [636](#)
- MJ output, linear static analysis, [87](#)
- MJ output, nonlinear analysis, [260](#)
- MM input, [4](#)
- MM output, [140](#), [272](#)
- MNE output, linear static analysis, [90](#)
- MNE output, nonlinear analysis, [272](#)
- MNE output, response spectrum analysis, [186](#)
- MNE output, transient analysis, [140](#)
- MNR output, linear static analysis, [90](#)
- MNR output, nonlinear analysis, [272](#)
- MNR output, response spectrum analysis, [186](#)
- MNR output, transient analysis, [140](#)
- MNT output, linear static analysis, [90](#)
- MNT output, nonlinear analysis, [272](#)
- MNT output, response spectrum analysis, [186](#)
- MNT output, transient analysis, [140](#)
- Mobile load, [36](#)
- Mobile load, output of extreme values, [76](#)
- MOBILE subtable of 'LOADS', [37](#)
- *MODAL command, [144](#)
- Modal damping factor, [438](#), [544](#)
- Modal mass, [438](#), [543](#)
- Modal pushover analysis, [104](#)
- Modal response analysis, [144](#)
- Modal result combinations, [169](#), [172](#), [591](#)
- Mode acceleration, [148](#)
- MODE command, [302](#)
- MODE input, pushover analysis, [106](#)
- Mode superposition, frequency response analysis, [589](#)
- Mode superposition, Response Spectrum analysis, [591](#)
- Model codes, [3](#)
- MODEL command, transient heat flow, [337](#)
- MODEL command, buckling analysis, [299](#)
- MODEL command, cross-section analysis, [366](#)
- MODEL command, design checking, [452](#)
- MODEL command, direct response analysis, [150](#)
- MODEL command, eigenvalue analysis, [428](#)
- MODEL command, HFTD analysis, [190](#)
- MODEL command, linear static analysis, [71](#), [72](#)
- MODEL command, loads optimization, [514](#)
- MODEL command, lubrication analysis, [372](#)
- MODEL command, modal response analysis, [145](#)
- MODEL command, nonlinear analysis, [211](#)
- MODEL command, nonlinear structural analysis, [211](#)
- MODEL command, response spectrum analysis, [170](#)
- MODEL command, steady-state groundwater flow, [350](#)
- MODEL command, steady-state heat flow, [333](#)
- MODEL command, stiffness adaptation analysis, [481](#)
- MODEL command, transient analysis, [121](#), [122](#)
- MODEL command, transient groundwater flow, [355](#)
- MODEL command, vibration analysis, [200](#)
- Model equivalent acceleration load, [34](#)
- Model information, [6](#)
- MODEL OFF command, [67](#)
- 'MODEL' table, [6](#)
- MODES command, buckling analysis, [304](#)
- MODES command, eigenvalue analysis, [436](#), [439](#)
- MODES command, HFTD analysis, [193](#)
- MODES command, modal response analysis, [147](#), [148](#)
- MODES command, nonlinear vibration analysis, [202](#)
- MODES command, postbuckling analysis, [306](#)
- MODES command, response spectrum analysis, [172](#)
- MODIFI option, heat transfer, [335](#)
- MODIFI option, nonlinear analysis, [237](#)
- MODIFI option, parameter estimation, [509](#)
- MODIFI option, steady-state groundwater flow, [353](#)
- MODIFI option, transient groundwater flow, [359](#)
- MODIFI option, transient heat flow, [342](#)
- Modified elasticity, stress state update, [213](#)
- Modified Mohr–Coulomb plasticity, minimum sub-step size, [214](#)
- Modified Newton–Raphson iteration, flow analysis, [629](#)
- Modified Newton–Raphson iteration, heat transfer, [335](#)
- Modified Newton–Raphson iteration, nonlinear analysis, [237](#), [554](#)
- Modified Newton–Raphson iteration, steady-state groundwater flow, [353](#)
- Modified Newton–Raphson iteration, transient groundwater flow, [359](#)
- Modified Newton–Raphson iteration, transient heat flow, [342](#)
- MODSUP command, [172](#)
- Mohr–Coulomb plasticity, parameter estimation, [500](#)
- Mohr–Coulomb failure criterion, [582](#)
- Mohr–Coulomb failure criterion, linear static analysis, [83](#), [87](#)
- Mohr–Coulomb failure criterion, nonlinear analysis, [256](#), [268](#)
- Mohr–Coulomb failure criterion, transient analysis, [135](#)
- MOHRCO option, linear static analysis, [83](#), [87](#)
- MOHRCO option, nonlinear analysis, [256](#), [268](#)
- MOHRCO option, transient analysis, [135](#)
- Moisture capacity, [633](#)
- MOMENT input, [33](#), [46](#)
- MOMENT option, [57](#)
- Momentum conservation, [633](#)
- MR output, linear static analysis, [89](#)
- MR output, nonlinear analysis, [271](#)
- MR output, response spectrum analysis, [185](#)
- MR output, transient analysis, [139](#)
- MSC, *see* Mobilized Shear Capacity
- MT input, [4](#)
- MT output, nonlinear analysis, [270](#)
- MT output, transient analysis, [138](#)
- Multi-directional fixed crack model, Damage Index262
- Multi-point tyings, [17](#), [321](#)
- Multiplication subroutines, [649](#)
- MULTLN input, stiffness adaptation analysis, [472](#), [476](#)

MW parameter, [47](#)
 MX parameter, [367](#)
 MXX command, [454](#)
 MXY command, [454](#)
 MYY command, [454](#)

N

N input, [4](#)
 N output, frequency response analysis, [164](#)
 N output, linear static analysis, [86](#)
 N output, nonlinear analysis, [259](#)
 N output, response spectrum analysis, [182](#)
 N output, transient analysis, [136](#)
 NO output, [86](#)
 NAME command, design checking, [453](#), [454](#)
 NAME input, mixture analysis, [388](#)
 NAME input, parameter estimation, [499](#)
 NAME input, potential flow analysis, [317](#)
 NAME input, spatial functions, [12](#)
 NAME input, structural analysis, [15](#), [32](#)
 Natural boundary condition, [627](#)
 NCLOAD command, [216](#)
 NCRACK output, [266](#)
 NDIANA output device, [56](#), [63](#)
 Negative pivots for (un)loading, [228](#), [564](#)
 Neumann, [627](#)
 NEWMAR command, [217](#)
 Newmark time integration, [217](#), [593](#), [635](#)
 NEWREF option, [239](#)
 NEWTON option, [237](#)
 Newton, force unit, [4](#)
 Newton–Raphson iteration, flow analysis, [629](#)
 Newton–Raphson iteration, heat transfer, [335](#)
 Newton–Raphson iteration, nonlinear analysis, [228](#), [237](#), [540](#), [553](#)
 Newton–Raphson iteration, steady-state groundwater flow, [353](#)
 Newton–Raphson iteration, transient groundwater flow, [359](#)
 Newton–Raphson iteration, transient heat flow, [342](#)
 NF output, [99](#)
 NFATIG command, [98](#)
 NITERA parameter, engineering liquefaction analysis, [289](#)
 NITERA parameter, nonlinear load step, [226](#)
 NITERA parameter, nonlinear time step, [232](#)
 NITERA parameter, strength reduction method, [284](#)
 NJ output, linear static analysis, [87](#)
 NJ output, nonlinear analysis, [260](#)
 NLPREB command, [216](#), [301](#)
 NLVIBR option, [205](#)
 NMODES parameter, eigenvalue analysis, [433](#)
 NMODES parameter, FEAST method based eigenvalue analysis, [434](#)
 No-shear function, stiffness adaptation analysis, [479](#)
 No-tension function, stiffness adaptation analysis, [479](#)
 NOAXES option, [58](#)
 NOBOND input, [235](#)
 NOBOND option, [58](#)
 NOBOND option, linear static analysis, [84](#)
 NOBOND option, nonlinear analysis, [257](#)
 NOCOOR option, [58](#)
 Nodal element damping forces, nonlinear analysis, [272](#)
 Nodal element damping forces, transient analysis, [140](#)
 Nodal element forces, linear static analysis, [89](#)
 Nodal element forces, nonlinear analysis, [271](#)
 Nodal element forces, response spectrum analysis, [185](#)
 Nodal element forces, transient analysis, [139](#)
 Nodal element inertia forces, nonlinear analysis, [272](#)
 Nodal element inertia forces, transient analysis, [140](#)
 Nodal flows, *see* Flows
 Nodal load, [33](#), [583](#)
 Nodal loads, spatial functions, [12](#)
 NODAL option, results selection, [520](#)
 Nodal results superposition, [406](#)
 NODAL subtable of 'BOUNDARY', [318](#)
 NODAL subtable of 'LOADS', [33](#)
 NODAL subtable of 'LOADS', mixture analysis, [389](#)
 Node coordinates, [11](#)
 NODE input, loads optimization, [513](#)
 NODE input, parameter estimation, [496](#)
 Node numbers, input, [11](#)
 NODES command, CMOD Arc-length control, [230](#)
 NODES command, output selection, [59](#), [60](#)
 NODES command, output selection, [60](#)
 NODES option, automatic tying, [54](#)
 NODES option, batch output, [58](#)
 NODES option, nonlinear output, [269](#)
 NODES subtable of 'GROUPS', [10](#)
 Nodes, boundary conditions, [317](#)
 Nodes, fixed potential, [315](#)
 Nodes, initial potential, [316](#)
 NODFOR command, HFTD analysis, [198](#)
 NODFOR command, linear static analysis, [90](#)
 NODFOR command, nonlinear analysis, [271](#)
 NODFOR command, nonlinear stop criterion, [242](#)
 NODFOR command, response spectrum analysis, [185](#)
 NODFOR command, stiffness adaptation analysis, [483](#)
 NODFOR command, transient analysis, [139](#)
 NOGRAV input, [106](#)
 NOLOG command, [47](#)
 Nonconservative loading, [216](#), [566](#), [570](#)
 NONE option, mobile load, [38](#)
 NONE option, reinforcement selection, [62](#)
 *NONLIN command, [211](#)
 NONLIN command, [508](#)
 NONLIN command, eigenvalue analysis, [430](#), [431](#)
 NONLIN command, steady-state groundwater flow, [352](#)
 NONLIN command, steady-state heat flow, [334](#)
 NONLIN command, transient groundwater flow, [357](#), [358](#), [359](#)

NONLIN command, transient heat flow, [339](#), [341](#), [341](#)
 NONLIN command, transient heat flow analysis, [340](#)
 NONLIN input, [403](#)
 Nonlinear analysis, [209](#), [551](#)
 Nonlinear analysis, iteration, [540](#), [552](#)
 Nonlinear analysis, mixture, [390](#)
 Nonlinear analysis, potential flow, [629](#)
 Nonlinear analysis, restoring steps, [245](#)
 Nonlinear analysis, saving steps, [245](#)
 Nonlinear analysis, structural, [211](#)
 Nonlinear structural analysis, [211](#)
 Nonlinear transient heat flow analysis, [340](#)
 Nonlinear vibration analysis, [199](#), [201](#)
 nonlinear vibration analysis, displacements, [206](#)
 NORM command, [204](#), [309](#)
 NORM option, [77](#), [127](#), [128](#), [130](#)
 NORM option, response spectrum analysis, [174](#)–[176](#), [184](#)
 NORM subroutine, [659](#)
 NORM2 subroutine, [660](#)
 Normalized cumulative energy, [262](#)
 Norms for convergence, [238](#), [558](#)
 Norms for convergence, HFTD analysis, [194](#)
 Norms subroutines, [659](#)
 NOSHTE input, stiffness adaptation analysis, [479](#)
 Notation convention, [631](#)
 NOTENS input, stiffness adaptation analysis, [479](#)
 NPR 9998:2015, design spectrum, [118](#)
 NPR 9998:2015, elastic response spectrum, [117](#)
 NPR input, design spectrum, [118](#)
 NPR input, elastic response spectrum, [117](#)
 NRMCM option, [263](#)
 NSTEPS parameter, engineering liquefaction analysis, [289](#), [290](#)
 NSTEPS parameter, nonlinear load step, [226](#), [227](#)
 NSTEPS parameter, nonlinear time step, [233](#)
 NSTEPS parameter, nonlinear vibration analysis, [204](#)
 NSTEPS parameter, postbuckling analysis, [309](#)
 NSTEPS parameter, strength reduction analysis, [284](#)
 NTHORE input, stiffness adaptation analysis, [476](#)
 NTRIAL parameter, [433](#), [434](#)
 NULLVC subroutine, [660](#)
 NUMBER command, eigenvalue analysis, [436](#)
 NUMBER command, HFTD analysis, [194](#)
 NUMBER parameter, [148](#)
 NUMSUB parameter, [424](#)
 NUMTHR command, [47](#)
 NXX command, [454](#)
 NXY command, [454](#)
 NYY command, [454](#)

O

OBSERV parameter, [506](#)
 OBSERV subtable of 'TARGET', loads optimization, [513](#)
 OBSERV subtable of 'TARGET', parameter estimation, [493](#), [495](#)

Observables, [495](#), [513](#)
 Observables, estimation report, [510](#)
 Observables, weighting, [496](#)
 OLDREF option, [239](#)
 ONESID input option, [386](#)
 OOPSHR output, [266](#)
 OpenMP, [547](#)
 Ordering, [549](#)
 ORIENT input, Eurocode 8 EN 1998-1, [114](#), [116](#)
 ORIENT input, NPR 9998:2015, [117](#), [118](#)
 Orthotropy, parameter estimation, [500](#), [518](#)
 Ounce, mass unit, [4](#)
 Out-of-balance flows, *see* Residual flows
 Out-of-balance forces, *see* Residual forces
 Out-of-core solution, [422](#)
 OUTCOR command, eigenvalue analysis, [433](#), [434](#)
 OUTCOR command, iterative solution, [423](#)
 OUTER subroutine, [659](#)
 Outer-products subroutines, [658](#)
 Output, [55](#)
 OUTPUT command, [55](#), [56](#)
 OUTPUT command, buckling analysis, [300](#), [303](#)
 OUTPUT command, design checking, [451](#), [454](#)
 OUTPUT command, eigenvalue analysis, [438](#)
 OUTPUT command, engineering creep analysis, [291](#)
 OUTPUT command, engineering liquefaction analysis, [287](#)
 OUTPUT command, fatigue failure, [98](#)
 OUTPUT command, frequency response analysis, [152](#)
 OUTPUT command, groundwater flow, [360](#)
 OUTPUT command, heat flow, [343](#)
 OUTPUT command, HFTD analysis, [192](#)
 OUTPUT command, linear static analysis, [71](#), [73](#)
 OUTPUT command, modal response analysis, [146](#)
 OUTPUT command, nonlinear analysis, [219](#)
 OUTPUT command, nonlinear structural analysis, [211](#)
 OUTPUT command, nonlinear vibration analysis, [201](#), [202](#), [204](#)
 OUTPUT command, parameter estimation, [506](#)
 OUTPUT command, postbuckling analysis, [306](#), [309](#)
 OUTPUT command, response spectrum analysis, [172](#)
 OUTPUT command, steady-state heat flow, [332](#)
 OUTPUT command, stiffness adaptation analysis, [482](#)
 OUTPUT command, strength reduction analysis, [281](#)
 OUTPUT command, tabular output, [64](#)
 OUTPUT command, transient analysis, [121](#), [124](#), [125](#)
 Output components, [58](#)
 Output results, loads optimization, [515](#)
 Output selection, [55](#)
 Overconsolidation, mixture analysis, [378](#)
 OZ input, [4](#)

P

P input, prescribed potential, [318](#)

- P input, units, [4](#)
- P output, [363](#)
- P output, frequency response analysis, [161](#)
- P output, linear static analysis, [80](#), [85](#)
- P output, nonlinear analysis, [252](#), [257](#), [268](#)
- P output, response spectrum analysis, [178](#), [182](#)
- P output, transient analysis, [132](#), [133](#), [135](#)
- P-STRESS option, [520](#)
- PA output, [131](#)
- PAPr output, [395](#)
- PARABO input, stiffness adaptation analysis, [478](#)
- Parallel direct sparse solver, [421](#), [541](#)
- Parallel iterative solver, [423](#)
- Parallel processing, element loops, [47](#)
- PARAM command, HFTD analysis, [198](#)
- PARAM command, linear static analysis, [92](#)
- PARAM command, nonlinear analysis, [274](#)
- PARAM command, parameter estimation, [509](#)
- PARAM command, stiffness adaptation analysis, [483](#)
- PARAM subtable of 'ESTIMA', parameter estimation, [497](#), [499](#)
- Parameter estimation, [487](#), [490](#), [496](#), [517](#)
- Parameter estimation, report, [510](#)
- Parameter weighting, [501](#)
- PARDIS command, [421](#)
- PARDIS command, eigenvalue analysis, [432](#), [433](#)
- PARDISO, [421](#), [541](#)
- *PAREST command, [526](#)
- Partial pivoting, [654](#)
- PARTIC option, [439](#)
- Participation factor, [438](#), [542](#)
- Participation vector, [439](#), [543](#)
- PAXYZ output, transient analysis, [131](#)
- Pc output, [278](#)
- pD output, [78](#)
- PD output, linear static analysis, [78](#)
- PD output, transient analysis, [128](#)
- PDISPX input, [496](#)
- PDISPZ input, [496](#)
- PDL input, [4](#)
- PDpr output, [395](#)
- PDSIPY input, [496](#)
- PDXYZ output, linear static analysis, [78](#)
- pDXYZ output, linear static analysis, [78](#)
- PDXYZ output, transient analysis, [128](#)
- PEAK input, frequency domain analysis, [112](#)
- PEAK input, transient analysis, [110](#)
- PEAK input, transient potential flow, [319](#)
- Peat, [235](#)
- Penalty conduction coefficient, [348](#)
- Penalty formulation, [636](#)
- PERCEN command, [436](#)
- Perfectly matched layers, [213](#)
- Perfectly matched layers, initial conditions, [220](#)
- PERIOD input, complex loads, [113](#), [114](#)
- PERIOD input, direct input, [111](#)
- PERIOD input, external file, [112](#)
- Periodic response, *see* Steady-state
- Permeability, Darcy flow, [633](#)
- Permeability, mixture analysis, [378](#)
- Permeability, staggered analysis, [377](#)
- Perturbation analysis, [201](#), [295](#), [305](#), [545](#), [621](#), [622](#)
- Perturbation analysis, displacements, [203](#), [307](#)
- PH option, [390](#), [394](#)
- *PHASE command, potential flow analysis, [417](#)
- *PHASE command, structural analysis, [404](#)
- PHASE input, frequency domain analysis, [114](#)
- PHASE option, accelerations, [130](#)
- PHASE option, direct response analysis, [151](#)
- PHASE option, displacements, [126](#)
- PHASE option, eigenvalue analysis, [430](#), [431](#)
- PHASE option, HFTD analysis, [191](#)
- PHASE option, output selection, [57](#), [409](#)
- PHASE option, velocities, [128](#)
- Phased acceleration vector length, transient analysis, [131](#)
- Phased analysis, [57](#), [399](#), [496](#)
- Phased analysis, acceleration, [131](#)
- Phased analysis, displacement, [78](#), [128](#), [410](#)
- Phased analysis, flow, [415](#)
- Phased analysis, groundwater flow, [348](#), [349](#)
- Phased analysis, heat flow, [332](#)
- Phased analysis, initial conditions, [220](#)
- Phased analysis, linear, [78](#)
- Phased analysis, mixture, [386](#), [390](#), [391](#)
- Phased analysis, mixture elements, [378](#)
- Phased analysis, nodal results superposition, [406](#)
- Phased analysis, structural, [401](#), [401](#)
- Phased analysis, velocity, [129](#)
- Phased displacement vector length, linear static analysis, [78](#)
- Phased displacement vector length, transient analysis, [128](#)
- Phased velocity vector length, transient analysis, [129](#)
- PHI input, [447](#)
- PHI option, linear static analysis, [93](#)
- PHI option, nonlinear analysis, [276](#)
- PHI output, linear static analysis, [95](#)
- PHI output, nonlinear analysis, [278](#)
- PHYSIC command, nonlinear analysis, [212](#), [213](#)
- PHYSIC command, nonlinear analysis options, [219](#), [234](#)
- PHYSIC command, transient analysis, [123](#)
- Physical properties, specification, [675](#)
- Piola–Kirchhoff stress (2nd), [259](#), [568](#)
- PIOLAK option, [57](#)
- PIVOTS option, [228](#)
- Plane strain elements, fracture mechanics, [640](#)
- Plane strain elements, stability analysis, [297](#)
- Plane stress elements, fracture mechanics, [640](#)
- Plane stress elements, stability analysis, [297](#)
- PLASTI command, [214](#)
- PLASTI command, HFTD analysis logging, [196](#)
- PLASTI command, nonlinear analysis logging, [243](#)
- PLASTI option, status, [260](#)
- PLASTI option, strains, [249](#)
- Plastic energy, Kotsovos concrete model, [266](#)
- Plastic strain, [249](#), [251](#)
- Plastic yield, [575](#)
- Plasticity status, [260](#)
- PLASTN input, stiffness adaptation analysis, [470](#)
- Plate bending elements, stability analysis, [298](#)

- PML, [213](#)
- PML command, [213](#)
- PML, initial conditions, [220](#)
- PMLX option, linear static analysis, [93](#)
- PMLX option, nonlinear analysis, [276](#)
- PMLY option, linear static analysis, [93](#)
- PMLY option, nonlinear analysis, [276](#)
- PMLZ option, linear static analysis, [94](#)
- PMLZ option, nonlinear analysis, [276](#)
- PMSC output, [484](#)
- PMST output, [484](#)
- POINT input, parameter estimation, [496](#)
- Point source, [318](#)
- POINTS subtable of 'TARGET', [493](#), [495](#)
- POISLT input, [291](#)
- POISON input, parameter estimation, [519](#)
- POISON input, stiffness adaptation analysis, [469](#)
- POISON option, linear static analysis, [93](#)
- POISON option, nonlinear analysis, [275](#)
- POISON output, linear static analysis, [95](#)
- POISON output, nonlinear analysis, [278](#)
- Poisson's ratio, stiffness adaptation analysis, [469](#)
- Polar decomposition, [655](#)
- Pond, force unit, [4](#)
- Pore fluid, [631](#)
- Pore fluid, mixture analysis, [378](#), [385](#)
- Pore fluid, staggered analysis, [382](#)
- PORE option, groundwater flow, [363](#)
- Pore pressure, [632](#), [637](#)
- Pore pressure, groundwater flow, [363](#)
- Pore pressure, mixture analysis, [378](#), [387](#), [394](#)
- Pore pressure, nonlinear analysis, [255](#), [258](#), [268](#)
- Pore pressure, staggered analysis, [382](#)
- POROSI input, staggered analysis, [383](#)
- Porosity, [631](#)
- Porosity, mixture analysis, [378](#), [385](#)
- Porosity, staggered analysis, [383](#)
- Position dependency, loads, [12](#)
- Position vector, [6](#)
- Position vector, Darcy flow, [633](#)
- POSTBU option, [309](#)
- POSTBU option, nonlinear analysis, [222](#)
- Postbuckling analysis, [295](#), [305](#), [621](#), [622](#)
- Postbuckling analysis, displacements, [310](#)
- Postbuckling analysis, load factor, [310](#)
- POTENT command, steady-state groundwater flow, [354](#)
- POTENT command, transient groundwater flow, [357](#)
- POTENT subtable of 'INIVAR', [316](#)
- Potential flow analysis, phased, [416](#)
- Potential flow analysis, staggered, [379](#)
- Potentials, [315](#)
- Pound, mass unit, [4](#)
- Pound-force, force unit, [4](#)
- Poundal, force unit, [4](#)
- PPCAP output, linear static analysis, [87](#)
- PPCAP output, nonlinear analysis, [268](#)
- PR input, mixture analysis, [388](#), [389](#)
- PR output, mixture analysis, [395](#)
- PrA output, [131](#)
- prD output, [78](#)
- PrD output, linear static analysis, [78](#)
- PrD output, transient analysis, [128](#)
- PRECON command, [423](#)
- PRECON option, [275](#)
- Preconditioning, [423](#), [548](#)
- Preconsolidation stress, [275](#)
- PRESCR option, [77](#)
- PRESCR option, output selection, [57](#)
- Prescribed accelerations, [107](#)
- Prescribed accelerations, spatial functions, [12](#)
- Prescribed displacement vector length, linear static analysis, [78](#)
- Prescribed displacements, [35](#)
- Prescribed displacements, output, [78](#)
- Prescribed displacements, phased analysis, [408](#)
- Prescribed displacements, spatial functions, [12](#)
- Prescribed flux, [318](#)
- Prescribed head, [318](#)
- Prescribed potential, [318](#)
- Prescribed temperature, [318](#)
- PRESENT command, [261](#)
- PRESENT command, deformation, [520](#)
- PRESS input, [496](#)
- PRESSU command, groundwater flow, [361](#), [363](#)
- PRESSU command, HFTD analysis, [198](#)
- PRESSU command, initial state reference, [245](#)
- PRESSU command, linear static analysis, [87](#)
- PRESSU command, mixture analysis, [395](#)
- PRESSU command, nonlinear analysis, [215](#), [268](#)
- PRESSU input, [383](#)
- PRESSU option, [264](#)
- PRESSU option, hydraulic head, [361](#)
- PRESSU option, mixture analysis, [390](#), [390](#), [394](#), [396](#)
- PRESSU option, output selection, [57](#)
- Pressure, [215](#), [576](#)
- Pressure energy, specific, [264](#)
- Pressure head, [361](#)
- Pressure, mixture analysis, [394](#)
- Pressures, lubrication analysis, [373](#)
- PREVIO option, [237](#)
- PREVIO parameter, initial conditions, [220](#)
- PRfs output, frequency response analysis, [167](#)
- PRfs output, nonlinear analysis, [142](#), [279](#)
- PRGERR subroutine, [668](#)
- PRIIVC subroutine, [662](#)
- PRIIVL subroutine, [661](#)
- PRILVC subroutine, [663](#)
- PRILVL subroutine, [663](#)
- Primary stress changes, [258](#)
- PRIMAT subroutine, [650](#), [661](#)
- PRINCI option, [57](#)
- Principal strain, [57](#), [574](#)
- Principal stress, [57](#), [575](#)
- Printing subroutines, [660](#)
- PRIVAL subroutine, [660](#)
- PRIVEC subroutine, [661](#)
- Profile library, [3](#)
- PRP output, lubrication analysis, [373](#)
- PRSNOD command, [373](#)
- PrV output, [129](#)
- PS output, linear static analysis, [81](#)
- PS output, nonlinear analysis, [253](#)
- PS output, response spectrum Analysis, [179](#)

Pseudo-acceleration, [591](#)
 PSI option, linear static analysis, [93](#)
 PSI option, nonlinear analysis, [276](#)
 PSI output, linear static analysis, [95](#)
 PSI output, nonlinear analysis, [278](#)
 PSI parameter, [237](#)
 PTE output, heat flow, [344](#), [346](#)
 PUSHOV subtable of 'LOADS', [106](#)
 Pushover analysis, [104](#)
 PV output, [129](#)
 PVpr output, [395](#)
 PVXYZ output, transient analysis, [129](#)

Q

Q input, [318](#)
 Q output, frequency response analysis, [164](#)
 Q output, groundwater flow, [362](#)
 Q output, heat flow, [345](#)
 Q output, linear static analysis, [85](#), [86](#)
 Q output, mixture analysis, [396](#)
 Q output, nonlinear analysis, [257](#), [259](#)
 Q output, response spectrum analysis, [182](#)
 Q output, transient analysis, [135](#), [136](#)
 Q0 output, [86](#)
 Q8MEM element, parameter estimation, [518](#)
 QBpr output, mixture analysis, [396](#)
 QJ output, linear static analysis, [87](#)
 QJ output, nonlinear analysis, [260](#)
 QMATRI subtable of 'ESTIMA', [497](#), [501](#)
 QRpr output, mixture analysis, [396](#)
 QT output, [86](#), [182](#)
 QTpr output, mixture analysis, [396](#)
 Quarterpoint position, [640](#)
 Quasi-Newton iteration, [237](#), [554](#)
 QXZ command, [454](#)
 QY parameter, [367](#)
 QYZ command, [454](#)
 QZ parameter, [367](#)

R

R1 parameter, [60](#)
 R2 parameter, [60](#)
 RAB subroutine, [649](#)
 RABT subroutine, [651](#)
 RAD input, [4](#)
 Radian, angle unit, [4](#)
 Radiation, boundary elements, [330](#)
 RAJSCE command, [61](#)
 RANDOM command, [303](#)
 Random generator, [665](#)
 Random imperfection, [296](#), [303](#), [620](#)
 Rankine–Hill plasticity, minimum sub-step size, [214](#)
 RATB subroutine, [652](#)
 RATBA subroutine, [653](#)
 RATIO parameter, [53](#)
 RAYLEI command, eigenvalue analysis, [435](#)
 RAYLEI input, [6](#)
 RAYLEI option, [218](#)
 Rayleigh damping, [586](#)
 Rayleigh damping, coefficients, [435](#), [544](#)

Rayleigh damping, modal damping factor, [544](#)
 Rayleigh damping, model, [6](#)
 REACTI command, [343](#), [346](#)
 REACTI command, HFTD analysis logging, [196](#)
 REACTI command, nonlinear analysis logging, [243](#)
 REACTI option, forces, [88](#)
 REACTI option, groundwater flow, [363](#)
 REACTI option, heat transfer, [345](#)
 REACTI option, output selection, [57](#)
 REACTI option, response spectrum analysis, [184](#)
 Reaction flows, groundwater flow, [363](#)
 Reaction flows, heat transfer, [345](#)
 Reaction force vector length, response spectrum analysis, [184](#)
 Reaction forces, [584](#)
 Reaction forces, initial conditions, [220](#)
 Reaction forces, linear static analysis, [89](#)
 Reaction forces, nonlinear analysis, [270](#)
 Reaction forces, response spectrum analysis, [184](#)
 Reaction forces, transient analysis, [139](#)
 READ command, [50](#)
 READ command, phased analysis, [402](#)
 REAFOR command, structural nonlinear analysis, [220](#)
 REAL input, frequency domain analysis, [113](#)
 REAXES command, [60](#)
 REAXES option, [57](#)
 REDUCE command, HFTD analysis, [193](#)
 REDUCE command, modal response analysis, [147](#)
 REDUCE command, nonlinear vibration analysis, [201](#)
 REDUCE command, postbuckling analysis, [305](#)
 *REDUCT command, [281](#)
 Reduction factor output, [275](#)
 REFERE command, nonlinear analysis, [219](#), [244](#)
 Reference norm, [558](#)
 Reference point, total head, [6](#)
 REFGLB option, [253](#)
 REFHEA input, [6](#)
 Refinement of mesh, [21](#), [324](#)
 REFINV option, [253](#)
 REGION input, design spectrum, [116](#)
 REGION input, elastic response spectrum, [114](#)
 Regional groundwater flow, [349](#)
 REGULA option, heat transfer, [335](#)
 REGULA option, nonlinear analysis, [237](#)
 REGULA option, steady-state groundwater flow, [353](#)
 REGULA option, transient groundwater flow, [359](#)
 REGULA option, transient heat flow, [342](#)
 Regular Newton–Raphson iteration, [553](#)
 Regular Newton–Raphson iteration, flow analysis, [629](#)
 Regular Newton–Raphson iteration, heat transfer, [335](#)
 Regular Newton–Raphson iteration, nonlinear analysis, [214](#), [215](#), [237](#)
 Regular Newton–Raphson iteration, steady-state groundwater flow, [353](#)
 Regular Newton–Raphson iteration, transient groundwater flow, [359](#)

- Regular Newton–Raphson iteration, transient heat flow, [342](#)
- REINFO command, element evaluation, [53](#)
- REINFO command, output selection, [59](#), [62](#)
- REINFO command, phased analysis, [405](#)
- REINFO option, bonding, [235](#)
- REINFO option, linear static analysis, [90](#), [91](#)
- REINFO option, moments and forces, [57](#)
- REINFO option, nonlinear analysis, [271](#), [273](#)
- REINFO option, response spectrum analysis, [185](#), [187](#)
- REINFO option, slip, [235](#)
- REINFO option, transient analysis, [139](#), [141](#)
- REINFO subtable of 'GROUPS', [9](#)
- REINFO subtable of 'LOADS', [32](#)
- Reinforcement bar, shear stress, [87](#), [581](#)
- Reinforcement bar, stress gradient, [87](#), [581](#)
- Reinforcement evaluation, [53](#)
- Reinforcement forces, [57](#), [576](#)
- Reinforcement forces, combined, [457](#)
- Reinforcement forces, linear static analysis, [86](#)
- Reinforcement forces, response spectrum analysis, [182](#)
- Reinforcement forces, transient analysis, [136](#)
- Reinforcement grid design checking, [445](#)
- Reinforcement moments, [57](#), [576](#)
- Reinforcement moments, linear static analysis, [86](#)
- Reinforcement moments, response spectrum analysis, [183](#)
- Reinforcement moments, transient analysis, [136](#)
- Reinforcement, output selection, [62](#)
- Reinforcement, phased analysis, [408](#)
- Reinforcement, stability analysis, [298](#)
- RELATI option, [58](#)
- RELATI option, frequency response analysis, [153](#)
- RELATI option, linear static analysis, [77](#)
- RELATI option, transient analysis, [127](#), [128](#), [130](#)
- Relative acceleration, [58](#)
- Relative acceleration, frequency response analysis, [157](#)
- Relative acceleration, transient analysis, [130](#)
- Relative displacement, [58](#)
- Relative displacement, frequency response analysis, [153](#), [162](#)
- Relative displacement, linear static analysis, [77](#), [80](#)
- Relative displacement, nonlinear analysis, [253](#)
- Relative displacement, response spectrum analysis, [179](#)
- Relative displacement, transient analysis, [127](#), [133](#)
- Relative velocity, [58](#)
- Relative velocity, frequency response analysis, [155](#)
- Relative velocity, transient analysis, [128](#)
- RELBAC command, [218](#)
- REMAKE command, [50](#)
- Remaking input, [50](#)
- Reordering, *see* Ordering
- REPORT command, HFTD analysis, [195](#)
- REPORT command, nonlinear analysis, [243](#)
- RESFOR command, [405](#)
- RESIDU command, [239](#), [509](#)
- RESIDU option, forces, [88](#)
- RESIDU option, groundwater flow, [363](#)
- RESIDU option, heat transfer, [345](#)
- RESIDU option, output selection, [57](#)
- RESIDU option, response spectrum analysis, [184](#)
- Residual flows, groundwater flow, [363](#)
- Residual flows, heat transfer, [345](#)
- Residual force vector length, response spectrum analysis, [185](#)
- Residual forces, [584](#)
- Residual forces, linear static analysis, [89](#)
- Residual forces, nonlinear analysis, [270](#), [271](#), [552](#)
- Residual forces, response spectrum analysis, [185](#)
- Residual forces, transient analysis, [139](#)
- Residual norm, [239](#), [559](#)
- Residuals, [510](#)
- RESIDX input, loads optimization, [513](#)
- RESIDX input, parameter estimation, [496](#)
- RESIDY input, loads optimization, [513](#)
- RESIDY input, parameter estimation, [496](#)
- RESIDZ input, loads optimization, [513](#)
- RESIDZ input, parameter estimation, [496](#)
- Resonance, [149](#)
- RESPON command, HFTD analysis, [192](#)
- RESPON command, modal response analysis, [146](#)
- RESPON command, response spectrum analysis, [171](#)
- Response analysis, frequency, [143](#)
- Response analysis, hybrid frequency time domain, [189](#)
- Response analysis, spectrum, [169](#)
- Response Spectrum Analysis, [590](#)
- Response spectrum analysis, [169](#)
- Response Spectrum Analysis, modal result combinations, [591](#)
- response spectrum analysis, modal result combinations, [172](#)
- Response Spectrum Method, *see* Response Spectrum analysis
- RESTOR command, FILOS file, [49](#)
- RESTOR command, nonlinear analysis, [246](#)
- RESTOR command, nonlinear initial state, [223](#)
- RESTOR command, nonlinear load step, [225](#)
- RESTOR command, nonlinear time step, [231](#)
- RESTOR command, transient analysis, [124](#)
- RESTOR command, transient time step, [125](#)
- Restore FILOS file, [49](#)
- Restore step in nonlinear analysis, [246](#)
- Restoring items, FILOS file, [49](#)
- Restoring steps, nonlinear analysis, [245](#)
- Result averaging over a region, [61](#)
- RESULT command, [242](#)
- RESULT command, design checking, [454](#)
- RESULT command, nonlinear stop criterion, [240](#)
- RESULT command, strength reduction analysis, [285](#)
- RESULT option, [65](#)
- RESULTS command, [520](#)
- RETSTR command, [213](#)
- *REYNOL command, [372](#)

Rigidity matrix, [534](#)
 RMATRI subtable of 'TARGET', [493](#), [496](#)
 RMOVE subroutine, [664](#)
 RO input, dynamic analysis, [108](#)
 RO input, stability analysis, [296](#)
 RO input, structural analysis, [35](#)
 Robin, [627](#)
 Rock, [631](#)
 ROTATI option, Arc-length control, [229](#)
 ROTATI option, direct response analysis, [151](#)
 ROTATI option, eigenvalue analysis, [430](#)
 ROTATI option, HFTD analysis, [191](#)
 ROTATI option, nonlinear analysis, [218](#)
 ROTATI option, output selection, [57](#)
 ROTATX input, loads optimization, [513](#)
 ROTATX input, parameter estimation, [496](#)
 ROTATY input, loads optimization, [513](#)
 ROTATY input, parameter estimation, [496](#)
 ROTATZ input, loads optimization, [513](#)
 ROTATZ input, parameter estimation, [496](#)
 RTS option, parameter output, [274](#)
 RSET subroutine, [663](#)
 Rubber vulcanization, staggered analysis, [382](#)
 Runge-Kutta time integration, [594](#)

S

S output, cross-section analysis, [369](#)
 S output, frequency response analysis, [163](#)
 S output, linear static analysis, [85](#)
 S output, mixture analysis, [396](#)
 S output, nonlinear analysis, [257](#)
 S output, response spectrum analysis, [181](#)
 S output, transient analysis, [135](#)
 SO output, [85](#)
 SA option, [118](#)
 SAB subroutine, [649](#)
 SAFETY command, [282](#)
 Safety factor output, [275](#)
 Safety factors, linear static analysis, [83](#)
 Safety factors, nonlinear analysis, [256](#)
 Safety factors, transient analysis, [134](#)
 SATB subroutine, [652](#)
 SATBA subroutine, [653](#)
 SATURA command, [396](#)
 SATURA input, staggered analysis, [383](#)
 SATURA input, stiffness adaptation analysis, [477](#)
 Saturation, [631](#)
 Saturation, mixture analysis, [378](#), [396](#)
 Saturation, staggered analysis, [383](#)
 SAVE command, FILOS file, [49](#)
 SAVE command, nonlinear analysis, [246](#)
 SAVE command, nonlinear initial state, [224](#)
 SAVE command, nonlinear load step, [225](#)
 SAVE command, nonlinear time step, [231](#)
 SAVE command, transient analysis, [124](#)
 SAVE command, transient groundwater flow, [358](#)
 SAVE command, transient heat flow, [341](#)
 SAVE command, transient time step, [125](#)
 Save, FILOS file, [49](#)
 Saving items, FILOS file, [49](#)
 Saving steps, nonlinear analysis, [245](#)
 SCALE input, frequency domain analysis, [112](#)

SCALE input, transient analysis, [110](#)
 SCALE input, transient potential flow, [319](#)
 Scaling subroutines, [657](#)
 Schur Complement, [549](#)
 Schwarz domain decomposition, [547](#)
 SD option, [118](#)
 SDIRK2 command, [218](#), [234](#)
 SDIRK2 method, [234](#), [565](#), [595](#)
 SE output, nonlinear analysis, [257](#)
 SEAREA option, nonlinear analysis, [275](#)
 SEAREA option, stiffness adaptation analysis, [484](#)
 SEC input, [4](#)
 Secant method, *see* Quasi-Newton
 SECANT option, cracking, [215](#)
 SECANT option, nonlinear analysis, [237](#)
 SECMD1 option, [203](#)
 SECMD2 option, [203](#)
 SECMOD option, [306](#)
 SECOND option, [214](#)
 Second order displacements, [203](#), [307](#), [618](#), [621](#)
 Second, time unit, [4](#)
 Seepage face, [348](#)
 SEeq output, nonlinear analysis, [257](#)
 SEer output, [258](#)
 Seismic Moment, [81](#), [179](#), [253](#), [574](#)
 SELECT command, automatic tying, [54](#)
 SELECT command, buckling analysis, [304](#)
 SELECT command, cross-section analysis, [369](#)
 SELECT command, design checking, [454](#), [456](#)
 SELECT command, eigenvalue analysis, [439](#)
 SELECT command, fatigue failure, [98](#)
 SELECT command, frequency response analysis, [153](#)
 SELECT command, groundwater flow, [360](#)
 SELECT command, heat flow, [343](#)
 SELECT command, HFTD analysis, [197](#)
 SELECT command, linear static analysis, [74](#)
 SELECT command, lubrication analysis, [373](#)
 SELECT command, nonlinear analysis, [246](#)
 SELECT command, nonlinear stop criterion, [242](#)
 SELECT command, nonlinear vibration analysis, [202](#), [204](#)
 SELECT command, output selection, [55](#), [59](#)
 SELECT command, postbuckling analysis, [306](#), [309](#)
 SELECT command, response spectrum analysis, [173](#)
 SELECT command, stiffness adaptation analysis, [482](#)
 SELECT command, transient analysis, [126](#)
 Seq output, frequency response analysis, [164](#)
 Seq output, linear static analysis, [85](#)
 Seq output, nonlinear analysis, [257](#)
 Seq output, response spectrum analysis, [182](#)
 Seq output, transient analysis, [135](#)
 Sequential minimum variance, [490](#)
 Service libraries, [648](#)
 Serviceability Limit State, [453](#), [454](#), [461](#)
 SG output, [87](#)
 Shape functions, [531](#), [533](#)
 SHAPE option, deformation, [520](#)
 SHAPE parameter, [52](#)
 Shape test, [52](#)

- Shear capacity, [581](#)
- Shear capacity Coulomb friction, [583](#)
- Shear capacity Coulomb friction, linear static analysis, [83](#), [85](#)
- Shear capacity Coulomb friction, nonlinear analysis, [256](#), [258](#)
- Shear capacity Hoek-Brown, [582](#)
- Shear capacity Hoek-Brown, linear static analysis, [83](#), [85](#)
- Shear capacity Hoek-Brown, nonlinear analysis, [256](#), [258](#)
- Shear capacity Hoek-Brown, transient analysis, [135](#), [136](#)
- Shear capacity Mohr-Coulomb, [582](#)
- Shear capacity Mohr-Coulomb, linear static analysis, [83](#), [85](#)
- Shear capacity Mohr-Coulomb, nonlinear analysis, [256](#), [258](#)
- Shear capacity Mohr-Coulomb, transient analysis, [135](#), [136](#)
- SHEAR command, [369](#)
- Shear modulus, [632](#)
- SHEAR option, [87](#)
- Shear stress, linear static analysis, [82](#), [85](#)
- Shear stress, nonlinear analysis, [254](#), [258](#)
- Shear stress, reinforcement bar, [87](#), [581](#)
- Shear stress, response spectrum analysis, [181](#), [182](#)
- Shear stress, transient analysis, [134](#), [136](#)
- Shell elements, automatic tying, [28](#)
- Shell elements, tying to solid, [29](#), [30](#)
- Shift eigenvalues, [433](#)
- SHIFT parameter, eigenvalue analysis, [434](#), [435](#)
- Shift, eigenfrequencies, [544](#)
- Shift, eigenvalues, [434](#), [435](#), [544](#)
- Shifting eigenvalues, [425](#)
- SHRCAP output, linear static analysis, [85](#)
- SHRCAP output, nonlinear analysis, [258](#)
- SHRCAP output, transient analysis, [136](#)
- SHRINK command, [215](#)
- SHRINK option, strain output, [250](#)
- Shrinkage, [215](#)
- Shrinkage strain output, [250](#)
- Shrinkage, staggered analysis, [382](#)
- SHRRED input, [287](#)
- SI units, [3](#), [4](#)
- SIDE1 command, [230](#)
- SIDE2 command, [230](#)
- Sign convention, [631](#)
- SIGN option, nonlinear step size, [228](#)
- SIGN option, nonlinear stop criterion, [241](#)
- Simple soil, [215](#)
- SIMULT command, [239](#)
- Single Program, Multiple Data, [547](#)
- Single-point tyings, [16](#), [321](#)
- SINPHI input, parameter estimation, [500](#)
- SIZE parameter, nonlinear load step, [227](#)
- SIZE parameter, nonlinear time step, [234](#)
- SIZE parameter, nonlinear vibration analysis, [204](#)
- SIZE parameter, postbuckling analysis, [309](#)
- SIZE parameter, strength reduction analysis, [284](#)
- SIZES command, nonlinear load step, [225](#)
- SIZES command, nonlinear time step, [232](#)
- SIZES command, strength reduction analysis, [283](#)
- SIZES command, transient groundwater flow, [358](#)
- SIZES command, transient heat flow, [341](#)
- Sk output, [258](#)
- SKIP input, frequency domain analysis, [112](#)
- SKIP input, transient analysis, [110](#)
- SKIP input, transient potential flow, [319](#)
- Slave node, [16](#), [320](#)
- SLIP command, [235](#)
- Slope stability, [281](#), [611](#)
- SMOOTH option, [58](#)
- SMOOTH option, linear static analysis, [84](#)
- Smoothing, strain output, [58](#)
- Smoothing, strain output, linear static analysis, [84](#)
- Smoothing, stress output, [58](#)
- Smoothing, stress output, linear static analysis, [84](#)
- Snap-back behaviour, [215](#), [227](#), [561](#)
- Snap-through behaviour, [227](#), [561](#), [563](#), [564](#)
- Softening, [545](#)
- Soil, [385](#), [631](#)
- SOIL command, [215](#)
- Soil, drained/undrained, [235](#), [268](#), [636](#)
- Soil, pore fluid, [382](#)
- Soil-structure interaction, [385](#)
- Solid elements, automatic tying, [28](#)
- Solid elements, fracture mechanics, [641](#)
- Solid elements, stability analysis, [298](#)
- Solidification, [331](#)
- Solution methods, [421](#), [539](#)
- Solution, iterative procedure, [235](#), [284](#), [551](#), [552](#), [629](#)
- Solution, phased transient analysis, [416](#)
- Solution, transient flow analysis, [628](#)
- SOLVE command, [421](#)
- SOLVE command, cross-section analysis, [367](#)
- SOLVE command, design checking, [452](#)
- SOLVE command, engineering creep analysis, [291](#)
- SOLVE command, engineering liquefaction analysis, [287](#)
- SOLVE command, linear static analysis, [71](#)
- SOLVE command, loads optimization, [515](#)
- SOLVE command, lubrication analysis, [373](#)
- SOLVE command, nonlinear structural analysis, [211](#)
- SOLVE command, steady-state groundwater flow, [352](#)
- SOLVE command, steady-state heat flow, [333](#), [334](#)
- SOLVE command, stiffness adaptation analysis, [481](#)
- SOLVE command, strength reduction analysis, [282](#)
- SOLVE command, transient analysis, [121](#)
- SOLVE command, transient groundwater flow, [357](#), [358](#)
- SOLVE command, transient heat flow, [339](#), [341](#)

- Solver settings, nonlinear structural analysis, [211](#)
- Solver settings, transient structural analysis, [121](#)
- Sommerfeld radiation, [597](#)
- SORTI subroutine, [667](#)
- SORTII subroutine, [668](#)
- Sorting subroutines, [666](#)
- SORTR subroutine, [666](#)
- SORTRI subroutine, [667](#)
- Source code, [645](#)
- SP output, [259](#)
- Space discretization, [634](#)
- SPACE option, [49](#)
- SPACIN input, [447](#)
- Sparse Cholesky based solution method, [422](#)
- Sparse Cholesky solver, [541](#)
- Sparse matrix, [539](#)
- Spatial functions, nodal loads, [12](#)
- Spatial functions, prescribed accelerations, [12](#)
- Spatial functions, prescribed displacements, [12](#)
- SPE output, [259](#)
- SPECIF input, [106](#)
- Specific gravitational energy, nonlinear analysis, [264](#)
- Specific gravitational energy, transient analysis, [137](#)
- Specific kinetic energy, nonlinear analysis, [263](#)
- Specific kinetic energy, transient analysis, [137](#)
- Specific pressure energy, [264](#)
- Specific stress energy, [263](#)
- *SPECTR command, [169](#)
- SPECTR input, Eurocode 8 EN 1998-1, [114](#), [116](#)
- SPECTR input, NPR 9998:2015, [117](#), [118](#)
- Spectral acceleration, [591](#)
- Spectral displacement, [591](#)
- Spectral response analysis, *see* Response Spectrum analysis
- SPHERI command, [228](#)
- Spherical Path, [228](#), [562](#)
- Spline elements, stability analysis, [298](#)
- SPMD, [547](#)
- SPREAD input, [447](#)
- Spring elements, stability analysis, [298](#)
- Spurious modes, [209](#)
- Square-Root-of-Sum-of-Squares rule, *see* SRSS rule
- SRSS option, [172](#)
- SRSS rule, [169](#), [172](#), [591](#)
- SSHR output, [87](#)
- ST output, frequency response analysis, [165](#)
- ST output, linear static analysis, [87](#)
- ST output, nonlinear analysis, [260](#)
- ST output, response spectrum analysis, [183](#)
- ST output, transient analysis, [137](#)
- STABIL command, [300](#)
- Stability analysis, [295](#), [617](#)
- Stability analysis, Euler, [299](#)
- Stability analysis, linearized buckling, [431](#), [437](#), [542](#)
- Stability of soil, [377](#)
- *STADAP command, [481](#)
- STADAP option, status, [483](#)
- Staggered analysis, [313](#), [332](#), [348](#), [377](#), [378](#), [379](#)
- Standard eigenproblem, [430](#), [437](#), [541](#), [542](#)
- standard eigenvalue analysis, [425](#)
- START command, mixture analysis, [391](#)
- START command, nonlinear analysis, [219](#), [255](#)
- START command, nonlinear vibration analysis, [204](#)
- START command, postbuckling analysis, [309](#)
- START command, structural nonlinear analysis, [220](#)
- START command, transient analysis, [106](#), [124](#)
- Starting time, [220](#)
- Static analysis, [71](#)
- STATIC command, response spectrum analysis, [173](#)
- Static concrete biaxial failure envelope, linear static analysis, [83](#)
- Static concrete biaxial failure envelope, nonlinear analysis, [256](#)
- Static concrete biaxial failure envelope, transient analysis, [134](#)
- Static condensation, [549](#)
- STATUS command, [260](#), [261](#), [265](#), [268](#)
- STATUS command, Engineering Masonry model, [266](#)
- STATUS command, HFTD analysis, [197](#)
- STATUS command, Kotsovos concrete model, [266](#)
- STATUS command, linearized elements, [267](#)
- Status output, damage index, [262](#)
- Status output, Engineering Masonry model, [266](#)
- Status output, Kotsovos concrete model, [266](#)
- Status output, linearized elements, [267](#)
- Status output, nonlinear analysis, [260](#)
- Status output, normalized cumulative energy, [262](#)
- Status output, transient analysis, [137](#)
- Statute mile, length unit, [4](#)
- STE output, nonlinear analysis, [260](#)
- Steady-state groundwater flow, [350](#)
- Steady-state heat flow, [332](#)
- Steady-state response, [585](#)
- STEEL input, stiffness adaptation analysis, [470](#)
- Steel profiles, [3](#)
- STEP option, nonlinear analysis logging, [243](#)
- STEP option, transient heat flow, [342](#)
- STEP parameter, [509](#)
- STEPS command, engineering creep analysis, [291](#)
- STEPS command, engineering liquefaction analysis, [287](#), [288](#)
- STEPS command, HFTD analysis, [194](#), [197](#), [198](#)
- STEPS command, initial conditions, [221](#)
- STEPS command, initial nonlinear conditions, [223](#)
- STEPS command, nonlinear analysis output, [247](#), [248](#)
- STEPS command, nonlinear load step, [225](#)
- STEPS command, nonlinear time step, [231](#)
- STEPS command, nonlinear vibration analysis, [204](#)
- STEPS command, nonlinear vibration analysis output, [205](#)
- STEPS command, postbuckling analysis, [309](#)

- STEPS command, postbuckling analysis output, [309](#)
- STEPS command, stiffness adaptation analysis, [482](#)
- STEPS command, strength reduction analysis, [282](#), [283](#)
- STEPS command, transient analysis, [124](#)
- STEPS command, transient analysis output, [126](#)
- STEPS command, transient groundwater flow, [361](#)
- STEPS command, transient heat flow, [343](#), [344](#)
- STEPS command, transient time step, [125](#)
- STEPS option, mixture analysis, [391](#)
- STEPS option, transient heat flow, [341](#)
- Steps, restoring, [245](#)
- Steps, saving, [245](#)
- STFPAR option, [228](#)
- STIFFN command, eigenvalue analysis, [429](#), [431](#)
- Stiffness adaptation analysis, [465](#)
- Stiffness matrix, [536](#)
- STOP command, nonlinear analysis, [219](#), [240](#)
- STOP command, strength reduction analysis, [282](#), [285](#)
- STOP command, transient analysis, [124](#)
- Stop criterion, [240](#), [241](#)
- Stop criterion, Line Search, [237](#)
- Storage equation, [633](#)
- STORAT command, [356](#)
- Storativity, [347](#)
- Storativity matrices, [356](#)
- Strain, [534](#), [573](#)
- STRAIN command, frequency response analysis, [159](#)
- STRAIN command, HFTD analysis, [197](#)
- STRAIN command, linear static analysis, [79](#)
- STRAIN command, nonlinear analysis output, [247](#), [248](#)
- STRAIN command, nonlinear stop criterion, [242](#)
- STRAIN command, response spectrum analysis, [176](#)
- STRAIN command, stiffness adaptation analysis, [482](#)
- STRAIN command, transient analysis, [131](#)
- Strain energy based damping, [438](#), [544](#)
- Strain matrix, [573](#)
- STRAIN option, [235](#)
- Strain, frequency response analysis, [159](#)
- Strain, linear static analysis, [78](#)
- Strain, nonlinear analysis, [248](#)
- Strain, response spectrum analysis, [176](#)
- Strain, transient analysis, [131](#)
- Strain–displacement relation, [532](#)
- Strength reduction analysis, [281](#)
- Strength reduction method, [611](#)
- Stress, [534](#), [575](#)
- Stress amplitude, [97](#)
- Stress analysis, coupled to, groundwater flow, [348](#)
- Stress analysis, coupled to, heat flow, [332](#)
- Stress change invariants, [258](#)
- Stress changes, [258](#)
- STRESS command, [215](#)
- STRESS command, design checking, [457](#)
- STRESS command, direct response analysis, [151](#)
- STRESS command, eigenvalue analysis, [430](#), [431](#)
- STRESS command, frequency response analysis, [162](#)
- STRESS command, HFTD analysis, [191](#), [197](#)
- STRESS command, initial state reference, [245](#)
- STRESS command, initial stresses, [222](#)
- STRESS command, linear static analysis, [81](#), [519](#)
- STRESS command, nonlinear analysis output, [254](#)
- STRESS command, nonlinear stop criterion, [242](#)
- STRESS command, response spectrum analysis, [180](#)
- STRESS command, stiffness adaptation analysis, [482](#)
- STRESS command, transient analysis, [133](#)
- Stress energy, specific, [263](#)
- Stress gradient, reinforcement bar, [87](#), [581](#)
- Stress intensity factor, fracture mechanics, [639](#)
- Stress intensity factor, linear static analysis, [95](#)
- Stress intensity factor, nonlinear analysis, [278](#)
- Stress invariants, [57](#), [576](#)
- Stress invariants, linear static analysis, [85](#)
- Stress invariants, nonlinear analysis, [257](#)
- Stress invariants, response spectrum analysis, [182](#)
- Stress matrix, [575](#)
- STRESS option, [235](#), [264](#)
- Stress stiffening, [566](#)
- Stress, frequency response analysis, [162](#)
- Stress, linear static analysis, [81](#)
- Stress, nonlinear structural analysis, [253](#)
- Stress, response spectrum analysis, [180](#)
- Stress, separation, [632](#)
- Stress, transient analysis, [133](#)
- Stress–strain relation, [532](#)
- Stresses, effective, [254](#)
- Structural analysis, input, [15](#)
- Structural analysis, nonlinear, [211](#)
- Structural analysis, phased, [401](#)
- Structural dynamics, [585](#)
- Structural elements, staggered analysis, [380](#)
- Structural loads, [31](#)
- SUBSTP parameter, [214](#)
- SUBSTR command, [424](#)
- SUBSTR command, iterative solution, [423](#)
- SUBSTR command, parallel direct sparse solver, [422](#)
- SUBSTR command, Sparse Cholesky based solution method, [422](#)
- Substructuring, [424](#), [549](#)
- Superelement, [549](#)
- SUPERP command, [406](#)
- SUPERP input, [499](#)
- SUPERP subtable of 'ESTIMA', [497](#), [500](#)
- Superparameters, [500](#)
- Superparameters, user-supplied, [501](#)
- Superposition subroutines, [656](#)
- 'SUPPOR' table, [15](#)
- 'SUPPOR' table, mixture analysis, [388](#)
- 'SUPPOR' table, phased analysis, [408](#)
- Supports, [15](#)
- Supports, phased analysis, [408](#)

SUPPRE command, [235](#)
 Swelling of soil, [385](#)
 Swelling of soil, mixture analysis, [390](#)
 SYMBOL option, results presentation, [261](#)
 Symbols, glossary of, [xxix](#)
 Symbols, glossary of, design checking, [443](#)
 Symbols, glossary of, parameter estimation, [487](#)

T

T input, prescribed temperature, [318](#)
 T input, units, [4](#)
 TA output, [130](#)
 TABLE option, [50](#)
 TABULA output device, [56](#), [64](#)
 Tabular output, [64](#)
 TANGEN option, [237](#)
 Tangent stiffness, [617](#)
 Tangent stiffness, freeze, [237](#)
 Tangent stiffness, nonlinear analysis, [237](#)
 Tangential stiffness matrix, eigenvalue analysis, [425](#)
 TApr output, [395](#)
 Target data, loads optimization, [513](#)
 Target data, parameter estimation, [493](#)
 'TARGET' table, loads optimization, [513](#)
 'TARGET' table, parameter estimation, [493](#)
 TAUFAC option, linear static analysis, [93](#)
 TAUFAC option, nonlinear analysis, [276](#)
 TAUMAX output, [266](#)
 TAXYZ output, transient analysis, [130](#)
 TD output, [127](#)
 TDpr output, [395](#)
 TDXYZ output, transient analysis, [127](#)
 TE output, [270](#)
 TEMPER command, heat flow, [343](#), [344](#)
 TEMPER command, HFTD analysis, [198](#)
 TEMPER command, nonlinear analysis, [214](#), [269](#)
 TEMPER command, steady-state heat flow, [336](#)
 TEMPER command, transient heat flow, [339](#)
 TEMPER input, [4](#)
 TEMPER option, output selection, [57](#)
 TEMPER option, strain output, [249](#)
 TEMPER subtable of 'INIVAR', [316](#)
 'TEMPER' table, staggered analysis, [378](#)
 Temperature load, staggered analysis, [382](#)
 Temperature stability, [377](#)
 Temperature units, [4](#)
 Temperature, heat flow, [344](#)
 Temperature, nonlinear analysis, [269](#)
 TENCrv input, stiffness adaptation analysis, [469](#), [470](#)
 Tensile strength utilization, nonlinear analysis, [255](#)
 Tensile strenmgth utilization, linear static analysis, [82](#)
 TENSTR input, stiffness adaptation analysis, [472](#), [473](#)
 TENSTR option, linear static analysis, [93](#)
 TENSTR option, nonlinear analysis, [276](#)
 TENSTR output, linear static analysis, [95](#)
 TENSTR output, nonlinear analysis, [278](#)
 TERMIN option, [195](#), [239](#)
 TEXT command, [64](#)
 Textile, [518](#)
 Thermal expansion, staggered analysis, [382](#)
 Thermal strain, [252](#)
 Thermal strain output, nonlinear analysis, [249](#)
 THETA parameter, [217](#)
 THICK option, linear static analysis, [94](#)
 THICK option, nonlinear analysis, [276](#)
 THICK output, linear static analysis, [95](#)
 THICK output, nonlinear analysis, [278](#)
 THOREN input, stiffness adaptation analysis, [476](#)
 Thorenfeldt hardening, stiffness adaptation analysis, [476](#)
 Threshold angle, nonlinear analysis, [215](#)
 TIME command, mixture analysis, [393](#)
 TIME command, nonlinear time step, [219](#), [231](#)
 TIME command, transient time step, [124](#)
 Time dependency, [3](#)
 Time dependency, phased analysis, [418](#)
 Time dependency, staggered analysis, [382](#)
 Time discretization, [635](#)
 Time domain, *see* Transient analysis
 TIME input, units, [4](#)
 Time integration, [592](#)
 Time integration, mixture analysis, [390](#), [390](#)
 Time integration, nonlinear analysis, [217](#)
 Time integration, phased analysis, [416](#)
 Time integration, transient flow analysis, [628](#)
 TIME parameter, initial conditions, [220](#)
 TIME parameter, transient groundwater flow, [357](#)
 TIME parameter, transient heat flow, [339](#)
 Time segment, [194](#)
 Time steps, linear transient analysis, [124](#)
 Time steps, mixture analysis, [393](#)
 Time steps, nonlinear analysis, [231](#)
 Time units, [4](#)
 Time, starting, [220](#)
 Time-load diagram, [108](#)
 'TIMEBO' table, [318](#)
 TIMEDE command, [218](#)
 'TIMELO' table, [108](#)
 'TIMELO' table, mixture analysis, [392](#)
 TIMES input, transient analysis, [109](#)
 TIMES input, transient potential flow, [318](#), [319](#)
 TIMESE command, [194](#)
 Tmax output, linear static analysis, [85](#)
 Tmax output, nonlinear analysis, [258](#)
 Tmax output, response spectrum analysis, [182](#)
 Tmax output, transient analysis, [136](#)
 TOLABT parameter, HFTD analysis, [195](#)
 TOLABT parameter, nonlinear analysis, [239](#)
 TOLCON parameter, heat flow, [335](#)
 TOLCON parameter, HFTD analysis, [195](#)
 TOLCON parameter, nonlinear analysis, [239](#)
 TOLCON parameter, parameter estimation, [509](#)
 TOLCON parameter, steady-state groundwater flow, [353](#)
 TOLCON parameter, transient groundwater flow, [359](#)
 TOLCON parameter, transient heat flow, [342](#)
 TOLDIR parameter, [215](#)
 TOLERA parameter, [433–435](#)
 TOLERA parameter, creep, [214](#)

- TOLERA parameter, d.o.f. coincidence, [54](#)
 TOLERA parameter, plasticity, [214](#)
 TOLERA parameter, solution procedures, [421](#)
 TOLFS command, [282](#)
 TOLTCO parameter, [215](#)
 Ton, mass unit, [4](#)
 TOTAL command, [216](#)
 Total head, reference point, [6](#)
 Total Lagrange, [209](#), [216](#), [568](#)
 TOTAL option, accelerations, [130](#)
 TOTAL option, concentration, [269](#)
 TOTAL option, displacements, [126](#)
 TOTAL option, hydraulic head, [361](#)
 TOTAL option, maturity, [269](#)
 TOTAL option, nonlinear stop criterion, [241](#)
 TOTAL option, nonlinear stress, [254](#)
 TOTAL option, output selection, [57](#)
 TOTAL option, strains, [249](#)
 TOTAL option, temperature, [269](#)
 TOTAL option, velocities, [128](#)
 Total Strain crack model, Damage Index, [262](#)
 Total Strain crack model, normalized cumulative energy, [262](#)
 Total strain energy, Kotsovos concrete model, [266](#)
 Total strain output, nonlinear analysis, [249](#)
 TOTALT parameter, [233](#)
 TOTCRK command, [215](#)
 TOWHAT option, [265](#)
 Towhata-Iai liquefaction, status output, [265](#)
 TR input, dynamic analysis, [108](#)
 TR input, stability analysis, [296](#)
 TR input, structural analysis, [35](#)
 TrA output, [130](#)
 TRACTI option, [57](#)
 Traction, frequency response analysis, [165](#)
 Traction, linear static analysis, [86](#)
 Traction, nonlinear analysis, [260](#)
 Traction, response spectrum analysis, [183](#)
 Traction, transient analysis, [137](#)
 Traffic load, ENV 1991-3 code, [43](#)
 Transformation factors, [438](#), [544](#)
 Transformation, element matrix, [533](#)
 Transformation, stress and strain, [576](#)
 TRANSI command, nonlinear analysis, [212](#)
 TRANSI command, transient analysis, [123](#)
 Transient analysis, input, [106](#)
 Transient analysis, linear structural, [121](#)
 Transient analysis, mixture, [390](#)
 Transient analysis, nonlinear, [209](#), [217](#)
 Transient flow analysis, phased, [416](#)
 Transient flow analysis, staggered, [380](#)
 Transient groundwater flow, [355](#)
 Transient heat flow, [337](#)
 Transient heat flow analysis, nonlinear, [340](#)
 Transient response, [585](#), [592](#)
 TRANSL option, Arc-length control, [229](#)
 TRANSL option, output selection, [57](#)
 TrD output, [127](#)
 Trial vectors, [433](#), [434](#)
 Truck load, [38](#), [40](#)
 Truck load, ENV 1991-3 code, [43](#)
 Truss elements, stability analysis, [297](#)
 TrV output, [129](#)
 TRVINP input, [41](#)
 Tsai-Hill criterion, composites, [84](#)
 Tsai-Wu criterion, composites, [84](#)
 TSAIWU option, [84](#)
 TSAIWU output, [85](#)
 Turbulence, [348](#)
 Turbulence, staggered analysis, [377](#)
 TV output, [129](#)
 TVpr output, [395](#)
 TVXYZ output, transient analysis, [129](#)
 TWOBEA input, [41](#)
 TWOSID input, [386](#)
 Tyings, [16](#), [17](#), [320](#), [321](#), [534](#)
 'TYINGS' table, [16](#), [17](#), [320](#), [321](#)
 'TYINGS' table, mixture analysis, [389](#)
 'TYINGS' table, phased analysis, [408](#)
 Tyings, automatic, [28](#)
 Tyings, mixture analysis, [389](#)
 Tyings, phased analysis, [408](#)
 TYPE command, CMOD Arc-length control, [230](#)
 TYPE command, design checking, [453](#), [454](#)
 TYPE command, eigenvalue analysis, [428](#)
 TYPE command, nonlinear analysis, [212](#)
 TYPE command, nonlinear structural analysis, [211](#)
 TYPE command, phased analysis, [212](#)
 TYPE command, transient analysis, [121](#), [123](#)
 TYPE1 input, design spectrum, [116](#)
 TYPE1 input, elastic response spectrum, [114](#)
 TYPE2 input, design spectrum, [116](#)
 TYPE2 input, elastic response spectrum, [114](#)
- ## U
- UCCOV command, [457](#)
 UCMAX command, [457](#)
 UCMIN command, [456](#)
 UCPHI command, [457](#)
 UCSHR command, [457](#)
 UCSPA command, [457](#)
 Ultimate Limit State, [453–455](#), [459](#)
 Unconfined aquifer, [352](#), [357](#)
 UNCPL parameter, [306](#)
 UNDRAI input, [235](#), [268](#)
 UNITMX subroutine, [656](#)
 Units, [3](#), [382](#)
 'UNITS' table, [4](#)
 'UNITS' table, stiffness adaptation analysis, [476](#)
 Units, angle, [4](#)
 Units, eigenfrequencies, [437](#)
 Units, model codes, [3](#)
 Units, steel profiles, [3](#)
 Units, stiffness adaptation analysis, [476](#)
 Units, syntax, [4](#)
 Unity matrix, [656](#)
 UNLOCK command, [49](#)
 UPDATE command, Arc-length method, [228](#)
 UPDATE command, Lagrange formulation, [216](#)
 Updated Lagrange, [209](#), [216](#), [568](#)
 Updated Normal Plane, [228](#), [562](#)
 US units, [3](#)
 USE command, [646](#)

- USER command, [303](#)
 - USER option, [265](#), [268](#)
 - User-defined derived results, [87](#)
 - User-supplied subroutines, [645](#), [648](#)
 - User-supplied subroutines, general material model, [268](#)
 - User-supplied subroutines, parameter estimation, [501](#)
 - USRBOU input, [499](#)
 - USRBOU user-supplied subroutine, [503](#)
 - USRPAR input, parameter estimation, [500](#)
 - USRPAR user-supplied subroutine, [501](#)
 - USRRBE option, [82](#), [87](#)
 - USRRSH option, [82](#), [87](#)
 - URSTA input, general material model, [268](#)
 - URSTA input, interface elements, [268](#)
 - UV subroutine, [658](#)
 - UVMW subroutine, [657](#)
 - UVPW subroutine, [656](#)
 - UVPWS subroutine, [657](#)
 - UVS subroutine, [657](#)
- ## V
- V output, frequency response analysis, [156](#)
 - V output, response spectrum analysis, [175](#)
 - VBB code, [40](#)
 - VBB input, [38](#), [40](#)
 - Vector plots, [520](#)
 - Vector subroutines, [656](#)
 - VECTORS option, [520](#)
 - VELMOD input, [404](#)
 - VELOCI command, frequency response analysis, [155](#)
 - VELOCI command, HFTD analysis, [197](#)
 - VELOCI command, mixture analysis, [394](#)
 - VELOCI command, nonlinear analysis, [222](#)
 - VELOCI command, nonlinear analysis output, [247](#)
 - VELOCI command, nonlinear stop criterion, [242](#)
 - VELOCI command, response spectrum analysis, [175](#)
 - VELOCI command, transient analysis output, [128](#)
 - VELOCI subtable of 'INIVAR', [107](#)
 - Velocity vector length, response spectrum analysis, [175](#)
 - Velocity vector length, transient analysis, [129](#)
 - Velocity, frequency response analysis, [155](#)
 - Velocity, relative, [58](#)
 - Velocity, response spectrum analysis, [174](#)
 - Velocity, transient analysis, [128](#)
 - VERTIC input, Eurocode 8 EN 1998-1, [114](#), [116](#)
 - VERTIC input, NPR 9998:2015, [117](#), [118](#)
 - *VIBRAT command, [199](#)
 - VIEW command, [520](#)
 - VINPRO subroutine, [658](#)
 - Virtual crack extension, [641](#)
 - Virtual displacements, [532](#)
 - Virtual strain energy, [534](#)
 - VISCOE command, [215](#)
 - Viscoelasticity, [215](#)
 - Viscoelasticity, parameter estimation, [499](#)
 - VISCOP command, [215](#)
 - Viscoplasticity, [215](#)
 - Viscosity, dynamic, [371](#)
 - Viscous damping, *see* Damping, Rayleigh, [586](#)
 - VMISES input, stiffness adaptation analysis, [469](#)
 - VOID option, [250](#), [253](#)
 - VOID output, [253](#)
 - Void ratio, [253](#)
 - VOLUME option, [57](#)
 - Volumetric deformation, [637](#)
 - Volumetric locking, [209](#)
 - Volumetric shrinkage strain, [250](#), [252](#)
 - Volumetric strain, [57](#), [574](#)
 - Volumetric strain, linear static analysis, [80](#)
 - Volumetric strain, response spectrum analysis, [178](#)
 - Volumetric strain, transient analysis, [132](#)
 - Von Mises plasticity, stiffness adaptation analysis, [469](#)
 - VONMIS option, [57](#)
 - VONMISES option, [520](#)
 - Von Mises strain, [573](#)
 - Von Mises strain, frequency response analysis, [161](#)
 - Von Mises strain, linear static analysis, [80](#)
 - Von Mises strain, nonlinear analysis, [251](#)
 - Von Mises strain, output selection, [57](#)
 - Von Mises strain, response spectrum analysis, [178](#)
 - Von Mises strain, transient analysis, [132](#)
 - Von Mises stress, [575](#)
 - Von Mises stress, frequency response analysis, [164](#)
 - Von Mises stress, linear static analysis, [85](#), [520](#)
 - Von Mises stress, nonlinear analysis, [257](#)
 - Von Mises stress, output selection, [57](#)
 - Von Mises stress, response spectrum analysis, [182](#)
 - Von Mises stress, transient analysis, [135](#)
 - VOSB code, [40](#)
 - VOSB input, [38](#), [40](#)
 - Vr output, frequency response analysis, [156](#)
 - VtXYZ output, response spectrum analysis, [175](#)
 - Vulcanization, staggered analysis, [382](#)
- ## W
- Water, bulk modulus, [4](#)
 - Water, dynamic viscosity, [4](#)
 - Water, mass density, [4](#)
 - WEIGHT command, [245](#)
 - Weight load, [33](#), [536](#)
 - Weight load, mixture analysis, [389](#), [390](#)
 - WEIGHT subtable of 'LOADS', [34](#)
 - WEIGHT subtable of 'LOADS', mixture analysis, [390](#)
 - Weighted least squares, [490](#)
 - Weighting, [496](#), [501](#)
 - Westergaard distributed mass, dynamic analysis, [103](#)
 - WILSON command, [217](#)
 - Wilson time integration, [217](#), [593](#), [635](#)
 - WLANE input, [40](#)

Wöhler diagram, [97](#)

X

X input, spatial functions, [12](#)

XAXIS input, parameter estimation, [519](#)

Y

Y input, spatial functions, [12](#)

Yard, length unit, [4](#)

YD input, [4](#)

YEAR input, [4](#)

Year, time unit, [4](#)

YIELD input, stiffness adaptation analysis, [469](#)

YLDSTR input, design checking, [448](#)

YLDSTR input, stiffness adaptation analysis, [469](#)

YLDSTR option, linear static analysis, [93](#)

YLDSTR option, nonlinear analysis, [276](#)

YLDSTR output, linear static analysis, [95](#)

YLDSTR output, nonlinear analysis, [278](#)

YOUNCM input, stiffness adaptation analysis, [478](#)

Young hardening concrete, staggered analysis,
[382](#)

YOUNG input, design checking, [448](#)

YOUNG input, parameter estimation, [519](#)

YOUNG input, stiffness adaptation analysis, [469](#)

YOUNG option, linear static analysis, [93](#)

YOUNG option, nonlinear analysis, [275](#)

YOUNG output, linear static analysis, [95](#)

YOUNG output, nonlinear analysis, [278](#)

Young's modulus, isotropic, [448](#)

YOUNLT input, [291](#)

Z

Z input, spatial functions, [12](#)

ZERO option, [214](#)

ZR input, [447](#)

ZR parameter, [60](#)

