



User's Manual

Theory

Release 10.1

DIANA FEA BV

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

Edited by: Jonna Manie and Gerd-Jan Schreppers

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	ix
List of Tables	xv
Release Quality Process	xvii
Preface	xix
Glossary of Symbols	xxi
I Introduction	1
1 DIANA	3
2 Preprocessing	5
3 Analysis Modules	7
4 Postprocessing	11
5 Program Structure	13
II General Finite Element Analysis	15
6 General Concept of Linear Structural Finite Element Analysis	17
7 Library of Finite Elements	21
8 Loading and Constraints	115
9 Analysis Procedures	131
III Linear Dynamic and Nonlinear Finite Element Analysis	189
10 Linear Dynamic Analysis	191
11 Nonlinear Analysis	215
Bibliography	259
Index	265

Contents

List of Figures	ix
List of Tables	xv
Release Quality Process	xvii
Preface	xix
Glossary of Symbols	xxi
I Introduction	1
1 DIANA	3
2 Preprocessing	5
3 Analysis Modules	7
3.1 Linear Static Analysis	7
3.2 Reinforcement Grid Design Checking	7
3.3 Stiffness Adaptation Analysis	7
3.4 Nonlinear Analysis	7
3.5 Dynamic Analysis	8
3.6 Strength Reduction Analysis	8
3.7 Euler Stability Analysis	8
3.8 Potential Flow Analysis	8
3.9 Coupled Flow-stress Analysis	8
3.10 Phased Analysis	9
4 Postprocessing	11
5 Program Structure	13
II General Finite Element Analysis	15
6 General Concept of Linear Structural Finite Element Analysis	17
6.1 Global Formulation	17
7 Library of Finite Elements	21
7.1 Two-dimensional Body Elements	21
7.1.1 Element Axes	22
7.1.2 Displacements and Rotations	23
7.1.3 Strains and Stresses	23
7.1.4 Element Types	25
7.1.5 Integration Schemes and Assumed Strain Options	33
7.1.6 Thickness Definition for Plane Stress Elements	34
7.1.7 Linear Elastic Material Properties	34

7.2	Three-dimensional Bodies or Solids	35
7.2.1	Element Axes	35
7.2.2	Displacements	36
7.2.3	Strains and Stresses	36
7.2.4	Element Types	37
7.2.5	Integration Schemes and Assumed Strain Options	44
7.2.6	Linear Elastic Material Properties	44
7.3	Plates and Shells	45
7.3.1	Element Axes	48
7.3.2	Displacements and Rotations	48
7.3.3	Strains and Stresses	49
7.3.4	Element Types	51
7.3.5	Integration Schemes	65
7.3.6	Thickness and Shape Definition	65
7.3.7	Shear Reduction and Geometry Factors	66
7.3.8	Eccentric Connections for Curved Shell Elements	66
7.3.9	Linear Elastic Material Properties	66
7.4	Trusses and Cables	67
7.4.1	Element Axes	68
7.4.2	Displacements	68
7.4.3	Strains and Stresses	68
7.4.4	Element Types	69
7.4.5	Cross-section	71
7.4.6	Linear Elastic Material Properties	71
7.5	Beams	71
7.5.1	Element Axes	72
7.5.2	Strains and Stresses	73
7.5.3	Element Types	75
7.5.4	Cross-section Definition	80
7.5.5	Integration Schemes	81
7.5.6	Hinging and Sliding Connections	81
7.5.7	Eccentric Connections	81
7.5.8	Linear Elastic Material Properties	82
7.6	Structural Interface Elements	82
7.6.1	Element Axes	83
7.6.2	Displacements, Relative Displacements, and Traction	83
7.6.3	Element Types	85
7.6.4	Integration Schemes	90
7.6.5	Elastic Material Properties	90
7.7	Composed Elements	90
7.7.1	Element Axes	91
7.7.2	Forces and Bending Moments for Composed Line Elements	92
7.7.3	Distributed Forces and Distributed Bending Moments	92
7.7.4	Element Types	93
7.7.5	Thickness of Composed Line Elements	95
7.7.6	Thickness and Shape of Composed Surface Elements	95
7.8	Spring and Mass Elements	95
7.8.1	Element Types	98
7.9	Reinforcements	102
7.9.1	Embedded Reinforcements	102
7.9.2	Bond-Slip Reinforcements and Pile Foundations	110
8	Loading and Constraints	115
8.1	Nodal Variables	115
8.2	Constraints and Supports	116
8.3	Linear Dependencies or Tyings	116
8.3.1	Equalities	117
8.3.2	Interconnections	117

8.3.3	Eccentricities	118
8.3.4	General Connections	119
8.3.5	Automatic Tying	119
8.4	Load Cases	120
8.4.1	Nodal Loads	121
8.4.2	Weight Loads	121
8.4.3	Equivalent Acceleration Loads	121
8.4.4	Centrifugal Loads	121
8.4.5	Fixed Displacement or Deformation Loads	121
8.4.6	Mobile Loads	122
8.4.7	Wind and Water Loads	125
8.4.8	Influence Loads	126
8.4.9	Element Point Loads	126
8.4.10	Element Line Loads	126
8.4.11	Element Face Loads	127
8.4.12	Element Volume Loads	127
8.4.13	Initial Stress Loads	127
8.4.14	Temperature and Concentration Loads	128
8.4.15	Hydrostatic Pressure Loads	129
8.4.16	Pore-pressure and Hydraulic Head Loads	129
8.4.17	Load Sets	129
9	Analysis Procedures	131
9.1	Linear Static Analysis	131
9.1.1	Model Evaluation	131
9.1.2	Solve the System of Equations	133
9.1.3	Output	133
9.2	Eigenvalue Analysis	145
9.2.1	Model Evaluation	145
9.2.2	Type of Eigenvalue Problem	146
9.2.3	Execute Eigenvalue Analysis	147
9.2.4	Calculate Rayleigh Damping Coefficients	147
9.2.5	Output	148
9.2.6	Generalized Mass	149
9.3	Buckling Analysis	151
9.3.1	Applicability of Elements	152
9.3.2	Euler Stability Analysis	153
9.3.3	Perturbation Analysis	157
9.3.4	Continuation Analysis	160
9.3.5	Effect of imperfection	160
9.4	Reinforcement Grid Design Checking Analysis	163
9.4.1	Specific Model Properties	165
9.4.2	Reinforcement Grid Design Check	166
9.4.3	Eurocode 2 EN 1992-1-1 Definitions	166
9.4.4	Model Evaluation	167
9.4.5	Define Load Combinations	167
9.4.6	Define Load Envelopes	167
9.4.7	Output Results	167
9.5	Stiffness Adaptation Analysis	170
9.5.1	Input for Stiffness Adaptation Analysis	173
9.5.2	Performing a Stiffness Adaptation Analysis	174
9.5.3	Model Evaluation	174
9.5.4	Solve the System of Equations	174
9.5.5	Execute Steps	175
9.5.6	Output Results	175
9.6	Phased Construction Analysis	175
9.6.1	Model Changes	180
9.7	Solving Linear Sets of Equations	180

9.7.1	Direct Solution Methods	181
9.7.2	Iterative Solution Methods	182
9.7.3	Some Remarks on Practical Use	185
9.7.4	Substructuring	186

III Linear Dynamic and Nonlinear Finite Element Analysis 189

10	Linear Dynamic Analysis	191
10.1	Mass and Damping	191
10.1.1	Mass	191
10.1.2	Damping	192
10.2	Modal Pushover Analysis	193
10.3	Frequency Response Analysis	193
10.3.1	Mode Superposition	194
10.3.2	Direct Solution	194
10.3.3	Analysis Results	194
10.4	Response Spectrum Analysis	195
10.5	Transient Dynamic Analysis	196
10.5.1	Newmark	197
10.5.2	Euler Backward	198
10.5.3	Hilber–Hughes–Taylor	198
10.5.4	Wilson	198
10.5.5	Runge–Kutta	199
10.5.6	Fluid–Structure Interaction Analysis	200
10.5.7	Simplification for Fixed Fluid Boundaries	203
10.6	Specific Elements	204
10.6.1	Boundary Surface Elements	204
10.6.2	Fluid–Structure Interface Elements	209
11	Nonlinear Analysis	215
11.1	Geometrical Nonlinear Analysis	215
11.1.1	Large Displacements, Rotations and Strains	215
11.1.2	Non-conservative Loads	219
11.1.3	Specific Elements	220
11.2	Physical Nonlinear Analysis	224
11.2.1	Nonlinear Elasticity	224
11.2.2	Hyperelasticity	228
11.2.3	Plasticity	229
11.3	Nonlinear Solution Procedures	242
11.3.1	Iterative Procedures	243
11.3.2	Continuation	247
11.3.3	Line Search	248
11.3.4	Convergence Criteria	248
11.3.5	Stepping Schemes and Arc-length Method	250
11.4	User-supplied Subroutines	256

Bibliography 259

Index 265

List of Figures

7.1	Plane stress elements, characteristics	21
7.2	Plane strain elements, characteristics	22
7.3	Axisymmetric elements, characteristics	22
7.4	Axes for two-dimensional body elements	22
7.5	Displacements of two-dimensional body elements	23
7.6	Deformation	23
7.7	Stresses	23
7.8	Deformation	24
7.9	Deformation	24
7.10	Cauchy stresses	25
7.11	T6MEM	25
7.12	Q8MEM	26
7.13	CT12M	26
7.14	CQ16M	26
7.15	CQ18M	27
7.16	T9MEM	27
7.17	Q12ME	28
7.18	T6OME	28
7.19	Q8OME	28
7.20	CT12O	29
7.21	CQ16O	29
7.22	T6EPS	30
7.23	Q8EPS	30
7.24	CT12E	30
7.25	CQ16E	31
7.26	CT30E	31
7.27	T6AXI	32
7.28	Q8AXI	32
7.29	CT12A	32
7.30	CQ16A	33
7.31	CT30A	33
7.32	Thickness	34
7.33	Solid elements, characteristics	36
7.34	Axes	36
7.35	Displacements	36
7.36	Deformation	37
7.37	Cauchy stresses	37
7.38	TE12L	38
7.39	PY15L	38
7.40	TP18L	39
7.41	HX24L	39
7.42	CTE30	40
7.43	CPY39	40
7.44	CTP45	41
7.45	CHX60	42
7.46	CTE48	42

7.47	CTP72	43
7.48	CHX96	44
7.49	Plate bending elements, characteristics	45
7.50	Flat shell elements, characteristics	46
7.51	Curved shell elements, characteristics	47
7.52	Axes	48
7.53	Displacements	48
7.54	Translations	49
7.55	Rotations	49
7.56	Planes for Green–Lagrange strains and Cauchy stresses	50
7.57	Cauchy stresses	51
7.58	Generalized moments and forces	51
7.59	T9PLA	52
7.60	Q12PL	52
7.61	CT18P	53
7.62	CQ24P	53
7.63	T15SF	54
7.64	Q20SF	54
7.65	CT30F	55
7.66	CQ40F	55
7.67	T18SF	55
7.68	T18FSH	56
7.69	Q24SF	56
7.70	CT36F	57
7.71	CQ48F	58
7.72	T15SH	58
7.73	Q20SH	59
7.74	CT30S	59
7.75	CQ40S	60
7.76	CT45S	60
7.77	CQ60S	61
7.78	T18SH	61
7.79	Q24SH	62
7.80	CT36S	62
7.81	CQ48S	63
7.82	CT30L	63
7.83	CQ40L	64
7.84	CT36L	64
7.85	CQ48L	65
7.86	Thickness	66
7.87	Predefined shapes	66
7.88	Eccentric connection	67
7.89	Truss elements	68
7.90	Truss elements, basic variables	68
7.91	Truss elements, stresses	68
7.92	L2TRU	69
7.93	L4TRU	69
7.94	L6TRU	70
7.95	CL6TR	70
7.96	CL9TR	70
7.97	Beam elements, characteristics	71
7.98	Axes for two-dimensional beams	72
7.99	Axes for three-dimensional beams	72
7.100	Displacements for class-I and class-II beams	73
7.101	Displacements for class-III beams	74
7.102	Deformation for three-dimensional beams	74
7.103	Moments and forces for three-dimensional class-I & -II beams	75
7.104	Cauchy stresses for three-dimensional beams	75

7.105L6BEN	76
7.106L12BE	76
7.107L7BEN	77
7.108L13BE	77
7.109L6BEA	78
7.110CL9BE	78
7.111L12BEA	79
7.112CL18B	79
7.113Predefined cross-sections	80
7.114Arbitrary cross-sections	80
7.115Eccentric connection class-I beams	81
7.116Normal force load on eccentric beam	82
7.117Variables of two-dimensional nodal interfaces	83
7.118Variables of two-dimensional interfaces	84
7.119Variables for structural interfaces to shells	84
7.120Variables of the line–solid interface elements	84
7.121Variables of three-dimensional nodal interfaces	85
7.122Variables of three-dimensional surface interfaces	85
7.123N4IF	85
7.124N6IF	86
7.125L8IF	86
7.126CL12I	86
7.127T18IF	87
7.128Q24IF	87
7.129CT36I	87
7.130CQ48I	88
7.131L16IF	88
7.132L20IF	88
7.133CL24I	89
7.134CL32I	89
7.135L12IF	89
7.136CL18I	90
7.137Axes	91
7.138Moments and forces for composed line elements	92
7.139Generalized moments and forces	92
7.140L2CMP	93
7.141CL3CM	93
7.142T3CMP	93
7.143Q4CMP	94
7.144CT6CM	94
7.145CQ8CM	94
7.146CT9CM	94
7.147CQ12C	95
7.148Predefined shapes	95
7.149Discrete springs	96
7.150Force–elongation diagram for spring elements	96
7.151Local element axes for matrix spring elements	97
7.152Spring stiffnesses depending on axial force (two-dimensional)	97
7.153Input and application of nonlinear stiffness K_{ϕ_z} (two dimensional)	98
7.154SP1TR	98
7.155SP2TR	99
7.156SP1RO	99
7.157SP2RO	99
7.158N6SPR	99
7.159N12SPR	100
7.160SP6BA	100
7.161SP12BA	101
7.162PT3T	101

7.163PT3RO	102
7.164Reinforcement bar	103
7.165Reinforcement grid	104
7.166Bar reinforcement in beam element	105
7.167Bar reinforcement in plane stress element	105
7.168Bar in plane strain element	106
7.169Bar in axisymmetric element	106
7.170Bar in curved shell element	106
7.171Bar in solid element	107
7.172Grid reinforcement in plane stress element	107
7.173Grid reinforcement in plane strain element	107
7.174Grid reinforcement in axisymmetric element	108
7.175GRID section in curved shell elements (example)	108
7.176Grid reinforcement with eccentricities in curved shell element	109
7.177Grid reinforcement in solid element	109
7.178Triangularization of grid reinforcement in solid element	109
7.179Axes for reinforcements in shell or solid elements	112
8.1 Tying degrees of freedom	116
8.2 Multi-point tying	117
8.3 Equality tyings	117
8.4 Mesh refinement	118
8.5 Multi-point interconnection	118
8.6 Eccentricity in two-dimensional model	118
8.7 Multi-point eccentricity	119
8.8 General connections in two-dimensional space	119
8.9 Multi-point general connection	119
8.10 Automatic tying of connected elements	120
8.11 Automatic tying of loose elements	121
8.12 Mobile load definition	122
8.13 Ultimate truck positions	123
8.14 Mobile truck and distributed load	123
8.15 Lanes on a carriage way applied for VOSB code	124
8.16 Influence line in transverse direction	124
8.17 Lanes on a carriage way applied for ENV 1991-3 code	124
8.18 Mobile truck and distributed load according to the ENV 1991-3 code	125
8.19 Line load examples	127
8.20 Face load examples	127
8.21 Temperature load examples for two-node beam	128
8.22 Temperature load	129
9.1 Element shapes	131
9.2 Reinforcement directions	138
9.3 Reinforcement moments \perp reinforcement	138
9.4 Forces in reinforcement directions	139
9.5 Shear stresses	140
9.6 Internal beam arm for bending only	140
9.7 Internal beam arm for bending plus compressive force	141
9.8 Internal beam arm for bending plus large tension force	141
9.9 Static (inner) and dynamic (outer) concrete biaxial failure envelope	143
9.10 Arch, loading and deformation	155
9.11 Arch, equilibrium node 3 deformed state	155
9.12 Shear stresses	164
9.13 Uniaxial stress-strain curve with stress and stiffness reduction	171
9.14 Flowchart of analysis sequence for stiffness adaptation analysis	172
9.15 Predefined tension softening for stiffness adaptation analysis	173
9.16 Predefined compression behaviour for stiffness adaptation analysis	174
9.17 Phases with unchanged model	175

9.18 Increasing load F	177
9.19 Phases with unchanged model and initial stress	177
9.20 Increasing load F and initial stress σ^0	179
9.21 Addition of an element	180
9.22 Krylov-based iterative solution methods.	183
10.1 Linear acceleration	198
10.2 Fluid–structure interaction	200
10.3 L4TM	205
10.4 CL6TM	206
10.5 L6TM	206
10.6 CL9TM	207
10.7 T9TM	207
10.8 Q12TM	207
10.9 CT18TM	208
10.10CQ24TM	209
10.11Variables of two-dimensional fluid–structural interfaces	210
10.12Variables of three-dimensional fluid–structural interfaces	210
10.13BL4S2	210
10.14BCL6S3	211
10.15BCL6S2	211
10.16BQ12S4	211
10.17BQ24S8	212
10.18BQ24S4	212
10.19BT9S3	213
10.20BT18S6	213
10.21BT18S3	214
11.1 Large displacements	215
11.2 Coordinate Frames	216
11.3 Coulomb friction	220
11.4 Two-dimensional contact algorithm	221
11.5 Three-dimensional contact algorithm	221
11.6 Stresses for contact elements	221
11.7 Contact behaviour	222
11.8 L4CT	222
11.9 CL6CT	222
11.10T9CT	223
11.11Q12CT	223
11.12CT18C	223
11.13CQ24C	224
11.14Hypoelastic behaviour of granular materials	224
11.15Jardine parameters in stiffness–log(strain) diagram	226
11.16Return-mapping algorithm	232
11.17Tresca and Von Mises yield condition (in π -and meridional plane)	233
11.18Derivation of hardening diagram for Tresca	234
11.19Derivation of hardening diagram for Von Mises	236
11.20Mohr–Coulomb and Drucker–Prager yield condition	237
11.21Derivation of hardening diagram for Mohr–Coulomb	238
11.22Tension cut-off surface for Mohr–Coulomb plasticity model	238
11.23Derivation of hardening diagram for Drucker–Prager	240
11.24Biaxial strength of plain concrete, Kupfer and Gerstle	241
11.25Iteration process	243
11.26Regular Newton–Raphson iteration	244
11.27Modified Newton–Raphson iteration	245
11.28Quasi-Newton iteration	245
11.29Linear Stiffness iteration	247
11.30Continuation method (Linear Stiffness)	248

11.31Line Search iteration	249
11.32Norm items	249
11.33Load and displacement control	251
11.34Arc-length control	252
11.35Work increment	255

List of Tables

7.1	LINEAR ORTHOTROPIC ELASTICITY	35
7.2	LINEAR ORTHOTROPIC ELASTICITY	45
7.3	LINEAR ORTHOTROPIC ELASTICITY	67
7.4	ELEMENTS FOR BOND-SLIP REINFORCEMENTS IN PLANE STRESS	111
7.5	ELEMENTS FOR BOND-SLIP REINFORCEMENTS IN CURVED SHELL	111
7.6	ELEMENTS FOR BOND-SLIP REINFORCEMENTS IN SOLID	112
8.1	MOBILE LOADS ACCORDING TO THE VOSB/VBB CODE	123
8.2	LANES AND CARRIAGE WAY ACCORDING TO THE ENV 1991-3 CODE . .	125
8.3	LOAD CLASSES ACCORDING TO THE ENV 1991-3 CODE	125
9.1	SHAPE PARAMETERS	132
9.2	STABILITY ANALYSIS FOR STRUCTURAL ELEMENTS	152
9.3	NATIONAL ANNEX PARAMETERS	166
11.1	STRESS AND STRAIN VECTORS IN USER-SUPPLIED SUBROUTINES	258

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

Cautionary note. Throughout this manual, it will be assumed that the reader has a basic understanding of applied mechanics and the Finite Element Method in general. Very informative introductions are the “Guidelines to Finite Element Practice” [53] and the book “A Finite Element Primer” [54], both published by NAFEMS. Also some experience with use of computers and computer programs is assumed.

Glossary of Symbols

Scalars ¹

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].	OCR	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].

¹SI 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.	$\mathbf{u}^{(e)}$...in element orientation.
S Second Piola–Kirchhoff stress tensor.	$\mathbf{u}^{(s)}$...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
$\tilde{\mathbf{M}}$ Added ...	$\Im u$ Imaginary part of ...
	$\Re u$ Real part of ...
	φu Phase angle of ...

Part I

Introduction

Chapter 1

DIANA

DIANA is a general purpose finite element code, based on the Displacement Method (DIANA is an acronym for DIplacement method ANalyser). It has been under development at Nederlandse Organisatie voor toegepast-natuurwetenschappelijk onderzoek (TNO) since 1972. In the beginning of 2003 a new organisation around DIANA was founded: TNO DIANA BV. As of 1 July 2016, TNO DIANA BV changed its name to DIANA FEA BV. After 14 years trading as TNO DIANA BV and the worldwide success of the flagship product DIANA, the decision was taken to change the name to reflect the name of the product and place more emphasis on DIANA rather than its ties to TNO (the Nederlandse Organisatie voor Toegepast Natuurwetenschappelijk Onderzoek). DIANA FEA BV remains a TNO Company, but stands alone as a software development company focusing on finite element analysis for the building industry of the future. DIANA FEA BV is the developer and distributor of the DIANA Finite Element Code. This chapter is a general introduction to the use of the DIANA Finite Element Code.

DIANA is a multi-purpose finite element program (three-dimensional and nonlinear) with extensive material, element and procedure libraries and dedicated pre- and post-processors for defining the finite element model and viewing and processing results in a graphical interactive environment. Developed by civil engineers from a civil engineering perspective, DIANA's most appealing capabilities are in the fields of concrete and soil. Worldwide, engineering consultants apply DIANA to their work on bridge design, dams, offshore platforms, road and rail design, tunneling and oil & gas applications. Furthermore, DIANA is extensively used for research and analysis purposes at technical universities on every continent.

Civil and mechanical and other engineering problems can be solved with the DIANA program. Standard DIANA application work includes: design and assessment of reinforced concrete, composite and steel structures, simulation of the process of excavations, tunneling, construction of buildings and structures, prediction and quantification of force transmission and deformation of all kind of systems. A wide range of material aspects such as cracking of concrete, plastic yielding of steel, creep and shrinkage, aging and ambient influences, can be considered. In geotechnical applications the groundwater flow conditions are considered. For the casting of concrete the heat generation and maturing of the young hardening concrete can be simulated. Both linear and nonlinear dynamic analysis can be performed as well as fluid-structure interaction.

DIANA offers a great variety of elements [§ 7 p. 21], such as cables, beams (straight and curved), solids, plane stress (membranes), axisymmetric and plane strain elements, plates, shells, springs, contact and interface elements (gap). All these elements may be combined in a particular finite element model. Moreover, special elements may be used to model embedded reinforcement in concrete structures: bars, grids and prestressed cables. Embedded reinforcements may also be used for bond-slip analysis or for pile foundations. To model these reinforcements DIANA has a built-in preprocessor in which reinforcement can be defined globally and independent from the element mesh. [Vol. *Element Library*] gives a complete overview of the available element types.

DIANA offers a variety of advantages over other commercially available FEM software. One of the most notable benefits is its power in the field of concrete and soil, where

excellent material models are available, developed by researchers in the Netherlands since the early 1970s. Notably are the models for smeared and discrete cracking, and for reduction of prestress due to special effects. DIANA also offers unique material models for soil and rock combining compaction and shear failure. In addition, DIANA can do various types of dynamic analysis important in earthquake engineering. Over the years several specific functions have been developed for our customers for bridge design, failure of reinforced concrete, dam analysis, oil & gas applications and others.

Chapter 2

Preprocessing

A wide range of modelling capabilities is available. A common way of modelling is creating a finite element mesh in a defined geometry. A geometry definition can be imported as an externally defined model, or can be defined in the DIANA preprocessor. Intersections of lines and surfaces and Boolean combinations of bodies are supported. Automatic meshing procedures generate a finite element mesh in a geometry definition. Properties, such as materials, loadings, and physical properties are assigned to geometry definitions or element sets. Finite elements can be defined by extrusion, sweeping and copying of existing meshes. Also externally defined finite element meshes can be used.

Chapter 3

Analysis Modules

With DIANA you can choose from a wide range of analysis types, all extensively described in [Vol. *Analysis Procedures*]. Here we give a short overview.

3.1 Linear Static Analysis

The Linear Static module provides a solid base for the DIANA finite element program. We mention some of the most important features. Linear constraints (tyings) can be specified to model linear dependencies between degrees of freedom of the system of equations (displacements, rotations, temperatures, etc.). Moving loads can be applied to determine influence lines and fields for critical result items. Fatigue failure analysis can be performed using standard Wöhler diagrams.

3.2 Reinforcement Grid Design Checking

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 Reinforcement Grid Design Checking 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.

3.3 Stiffness Adaptation Analysis

As an alternative for a full nonlinear analysis for calculating load distributions, deformations, crack patterns, and crack openings in constructions with nonlinear materials, such as reinforced concrete, a stiffness adaptation analysis 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.

3.4 Nonlinear Analysis

DIANA's strongest points lie in its nonlinear capabilities. For physical nonlinear analysis various material models are available including plasticity, viscoplasticity, cracking, viscoelasticity, creep, hyperelasticity, liquefaction of soil and many more. Time dependent development of temperature, concentration and maturity can be specified.

For geometrical nonlinear analysis the Total and Updated Lagrange methods are available. Moreover, contact analysis can be performed to check whether contact occurs in user-specified possible contact zones in the model.

Load can be applied in load steps or time steps. Stress initialisation functions for geotechnical analysis, or prestressing of structures, or for construction stage analysis in which parts of the model are in subsequent steps removed or added, are available.

Dynamic effects can be considered in combination with either or both physical and geometrical nonlinear aspects.

3.5 Dynamic Analysis

All appropriate types of dynamic structural and fluid-structure interaction analysis may be performed with DIANA: steady-state harmonic modal and direct frequency response analysis, Response Spectrum analysis, hybrid frequency time domain analysis, and linear and nonlinear transient analysis.

3.6 Strength Reduction Analysis

In DIANA strength reduction method is implemented as a separate module. The main output of this type of analysis is the factor of safety. 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 Mohr-Coulomb and Drucker-Prager models are considered. Future extension may include Modified Mohr-Coulomb, Hoek-Brown, and Coulomb friction (for interface elements) models.

3.7 Euler Stability Analysis

Euler stability analysis gives information about 'linearized stability' of a structure and provides a relatively simple and effective method to get a fair impression of a structure's buckling modes. The Euler stability analysis may be followed by a perturbation analysis to investigate the postbuckling behaviour. The postbuckling displacement field is solved by applying a continuation analysis using a stepwise generalized Newton-Raphson scheme.

3.8 Potential Flow Analysis

A potential flow analysis may be employed to solve general single potential convection-diffusion problems. It can be used in the following application fields: heat flow, detailed and regional groundwater flow, beam cross-section analysis, and Reynolds flow or lubrication. The heat flow module includes special features to perform advanced potential flow analysis. For instance, hydration heat and cooling pipe elements can be used to study the thermal behaviour of cement based materials at early ages, or radiation effects. The solidification and evaporation process within a liquid can also be modeled. Groundwater flow analysis also benefits from advanced features such as the modelling of seepage faces or study of the contamination transport of a pollutant within a soil.

3.9 Coupled Flow-stress Analysis

In coupled flow-stress analysis the interaction may be two- or one-directional. You may use a mixture analysis with mixture elements for two-directional interaction problems, for example a geotechnical transient consolidation analysis. A staggered analysis can be performed to solve one-directional interaction problems like geotechnical (static) stability analysis or structural analysis with thermal load.

3.10 Phased Analysis

DIANA enables modelling of phased construction. It determines the effects of construction history and shows the critical construction stages. Phased analysis can be performed on a structural level and on a potential flow level.

Chapter 4

Postprocessing

A very wide range of results can be calculated, printed in tables, processed in graphs or viewed in contour plots and other view modes. Some examples of specific postprocessing functions are:

- Envelopes over load cases in a linear analysis or time steps in a nonlinear analysis can be calculated or in transient flow analysis.
- Calculation of influence fields
- User defined combination of different result components in tabular output
- Crack visualization
- Output of crack status or plasticity status or status output of used materials
- User-supplied routines for derived results
- Combination of load cases.

Next to tabular output graphs results can be projected in the finite element mesh as a colour contour, as symbols, as vectors, as a deformed shape and as diagrams along line elements. Result values in selected points can be labeled. Cross-sections through a three-dimensional model and diagrams along selected lines can be displayed.

Chapter 5

Program Structure

For a long time the development of DIANA was focusing on analysis functionality while we relied on external software for pre- and postprocessing. Traditionally, DIANA ran only in batch mode, reading text files, in which the finite element model and analysis specifications are defined, and writing text files with tables of results. For performing a finite element analysis efficiently the preprocessor functionality (for geometry definition, mesh generation and property definitions) and postprocessing functionality (for visualization and processing) have become more important. For these functions graphical interactive programs, such as *i*DIANA and *FX*⁺ for DIANA, are offered.

With DIANA 9.5 a renewed MESH EDITOR program has been made available. This graphical interactive application can import the traditional DIANA text files for model and analysis definition as well as the models defined in external formats such as Nastran. Finite Element models defined in *i*DIANA and *FX*⁺ for DIANA can directly be transferred to the MESH EDITOR. In this application the finite element mesh and properties are viewed and properties, such as groups, materials, loadings, boundary conditions, and analysis settings can be inspected and modified. In the same environment the DIANA finite element analysis can be run and basic functions for viewing results are available. For advanced postprocessing *i*DIANA and *FX*⁺ for DIANA could be used.

With DIANA 10.0 the MESH EDITOR has been renamed to Diana Interactive Environment (DianaIE) and has been extended with functionality for modelling the geometry with properties and generation of the finite element meshes. Further, the post-processing capabilities have been extended with visualisation of crack results and tabulated output with copy and paste functionality to easily import and/or export to programs like Microsoft Excel. In upcoming DIANA versions the Diana Interactive Environment will develop into a full functionality preprocessing, analysis and postprocessing program.

The user can run DIANA programs in different modes:

- As a graphical interactive program. The Diana Interactive Environment (DianaIE) can be used to define the geometry with properties, generate the finite element mesh, run the analysis, and view the results without requiring any external pre- and/or postprocessor.
- As a graphical interactive program in combination with an external pre- and/or postprocessor(s). The programs *i*DIANA or *FX*⁺ for DIANA can be used to define the finite element mesh with properties and this model can be directly transferred to the Diana Interactive Environment (DianaIE), in which property definition can be completed, and analysis is done. Results can be viewed in the Diana Interactive Environment (DianaIE) or in *i*DIANA or *FX*⁺ for DIANA.
- As a batch program. Other programs or tools can be used to define text files in which the finite element model and analysis specifications are defined. The analysis can be performed by typing 'diana' command in a DIANA command box or in a Linux shell. Then model and analysis specification text files will be read, the DIANA analysis will be performed and result files will be written in text or binary format.
- In a combination of batch and graphical interactive mode.

iDIANA generates a DIANA model text file (*.dat) in which the finite element mesh with properties (groups, materials, properties, supports, loadings, etc.) are defined. DIANA analysis modules generate a binary FEMVIEW file (*.v72) with results and mesh. Because the FEMVIEW file contains both mesh and results, also models which are not defined in iDIANA can be visualized.

FX⁺ for DIANA generates a model text file (*.fxd) in which the finite element mesh with properties (groups, materials, properties, supports, loadings, etc.) are defined. DIANA analysis modules generate a binary FX⁺ for DIANA result file (*.dpb) with only results. Because the FX⁺ for DIANA result file does not contain the mesh, only results for models that have been generated in FX⁺ for DIANA can be viewed in FX⁺ for DIANA.

Analysis control data, such as type of analysis (Linear elastic analysis, or construction stage analysis) and analysis specifications (type of equation solver, solutions procedures, result specification) are defined graphical interactively in the Diana Interactive Environment (DianaIE) or in a text file (*.dcf). This text file can be imported and exported by DianaIE.

DIANA analysis modules can also be run directly from a Nastran model file (*.nas). Only mesh data and properties in the Nastran file are supported, analysis control is not supported. For restrictions to NASTRAN files see Volume *Getting Started*.

All DIANA analysis modules have data communication with a central database, the FILOS file (*.ff). This file allows the user to restart analysis, or to run the analysis in different runs, and to inspect results and status of the model and analysis in between different runs. Results for postprocessing in the Diana Interactive Environment (DianaIE) are written by the analysis modules in a binary result file (*.dnb). Results can be imported and viewed in the Diana Interactive Environment (DianaIE) when the analysis runs in batch mode. Only the corresponding model file (*.dat, *.nas, *.fxd) are needed and not the FILOS (*.ff) files for this.

All settings (mesh, properties, analysis control and graphical settings) in the Diana Interactive Environment (DianaIE) can be saved in and retrieved from a binary project file (*.dpf). All operations in DianaIE are logged as python commands in a text file (*.py). The Diana Interactive Environment (DianaIE) can be used in graphical interactive mode or by typing commands or by reading files with sequences of python commands and operations.

Part II

General Finite Element Analysis

Chapter 6

General Concept of Linear Structural Finite Element Analysis

A finite element model consists of one or more finite elements, their properties, boundary conditions, loadings and the definition of analysis procedures. In this chapter the basic functions for structural analysis will be outlined. First we will summarize the general concepts of a linear structural finite element analysis.

6.1 Global Formulation

For linear elastic problems the system of equations to be solved is

$$\mathbf{K}\mathbf{u} = \mathbf{f} \quad (6.1)$$

where \mathbf{K} is the *system stiffness matrix*, \mathbf{u} is a vector of the unknown nodal *variables* such as displacements and rotations and \mathbf{f} is the vector of the *nodal forces* corresponding with the nodal variables \mathbf{u} . This chapter describes the composition of the system stiffness matrix, the solution of the variables and the calculation of the element results as strains and stresses.

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.

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

$$\mathbf{u}_c(x, y, z) = \mathbf{N}\mathbf{u}_e \quad (6.2)$$

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

$$\mathbf{u}_e = \mathbf{T}_e\mathbf{u} \quad (6.3)$$

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.

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.

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

$$\mathbf{u} = \bar{\mathbf{u}} \quad \text{on} \quad S_u \quad (6.4)$$

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

$$\mathbf{u}_c(x, y, z) \approx \tilde{\mathbf{u}}(x, y, z) = \mathbf{N}(x, y, z)\mathbf{u} \quad (6.5)$$

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.

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

$$\boldsymbol{\varepsilon} = \mathbf{B}\mathbf{u}_e \quad (6.6)$$

The relation between strains and stresses, including initial strains and initial stresses, can be written for an element as

$$\boldsymbol{\sigma} = \mathbf{D}(\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_0) + \boldsymbol{\sigma}_0 \quad (6.7)$$

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

$$\boldsymbol{\varepsilon}_l = \mathbf{T}_\varepsilon \boldsymbol{\varepsilon} \quad (6.8)$$

where \mathbf{T}_ε is the strain transformation matrix. With Eq. (6.6) the local strain vector $\boldsymbol{\varepsilon}_l$ can now be related directly to the local element degrees of freedom vector \mathbf{u}_e by

$$\boldsymbol{\varepsilon}_l = \mathbf{T}_\varepsilon \mathbf{B}\mathbf{u}_e = \mathbf{B}_l \mathbf{u}_e \quad (6.9)$$

In a structural problem the governing equilibrium equations can be written as

$$\begin{aligned} \mathbf{L}^T \boldsymbol{\sigma} + \mathbf{g} &= 0 & \text{on} & V \\ \mathbf{L}_n^T \boldsymbol{\sigma} &= \mathbf{t} & \text{on} & S_t \end{aligned} \quad (6.10)$$

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.

A simpler way of introducing the equilibrium relations of Eq. (6.10) 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

$$\int_V \delta \boldsymbol{\varepsilon}^T \boldsymbol{\sigma} dV = \int_V \delta \mathbf{u}^T \mathbf{g} dV + \int_{S_t} \delta \mathbf{u}^T \mathbf{t} dS_t \quad (6.11)$$

where $\delta \boldsymbol{\varepsilon}$ are the virtual strains which correspond to the virtual displacements $\delta \mathbf{u}$.

Now the structure has been idealized as an assemblage of elements, the integral form of the virtual work Eq. (6.11) 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

$$\sum_{e=1}^{n_e} \int_{V_e} \delta \boldsymbol{\varepsilon}^T \boldsymbol{\sigma} dV = \sum_{e=1}^{n_e} \int_{V_e} \mathbf{u}^T \mathbf{g}_e dV + \sum_{e=1}^{n_e} \int_{S_e} \mathbf{u}^T \mathbf{t}_e dS \quad (6.12)$$

Substituting Eqs. (6.5) and (6.6) into Eq. (6.11) gives

$$\delta \mathbf{u}^T \int_V \mathbf{B}^T \boldsymbol{\sigma} dV = \delta \mathbf{u}^T \left(\int_V \mathbf{N}^T \mathbf{g} dV + \int_{S_t} \mathbf{N}^T \mathbf{t} dS_t \right) = \delta \mathbf{u}^T \mathbf{r} \quad (6.13)$$

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. (6.13) should be satisfied for any \mathbf{u} so that

$$\int_V \mathbf{B}^T \boldsymbol{\sigma} dV = \mathbf{r} \quad (6.14)$$

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. (6.7) and (6.6), the left hand side of Eq. (6.14) can be written as

$$\int_V \mathbf{B}^T \boldsymbol{\sigma} dV = \left(\int_V \mathbf{B}^T \mathbf{D} \mathbf{B} dV \right) \mathbf{u} - \int_V \mathbf{B}^T \mathbf{D} \boldsymbol{\varepsilon}_0 dV + \int_V \mathbf{B}^T \boldsymbol{\sigma}_0 dV = \mathbf{r} \quad (6.15)$$

Going back to Eq. (6.12) and using the piecewise approximation for the displacements Eq. (6.2) and the discrete strain–displacement relation Eq. (6.6), the virtual work equation is now obtained by

$$\sum_{e=1}^{n_e} \delta \mathbf{u}_e^T \int_{V_e} \mathbf{B}^T \boldsymbol{\sigma} dV = \sum_{e=1}^{n_e} \delta \mathbf{u}_e^T \int_{V_e} \mathbf{N}^T \mathbf{g}_e dV + \sum_{e=1}^{n_e} \delta \mathbf{u}_e^T \int_{S_e} \mathbf{N}^T \mathbf{t}_e dS \quad (6.16)$$

Substitution of the stress–strain relation Eq. (6.7) in case of linear elastic behaviour yields

$$\begin{aligned} \sum_{e=1}^{n_e} \delta \mathbf{u}_e^T \left(\int_{V_e} \mathbf{B}^T \mathbf{D} \mathbf{B} dV \right) \mathbf{u}_e = \\ \sum_{e=1}^{n_e} \delta \mathbf{u}_e^T \int_{V_e} \mathbf{N}^T \mathbf{g}_e dV + \sum_{e=1}^{n_e} \delta \mathbf{u}_e^T \int_{S_{te}} \mathbf{N}^T \mathbf{t}_e dS + \\ \sum_{e=1}^{n_e} \delta \mathbf{u}_e^T \left(\int_{V_e} \mathbf{B}^T \mathbf{D} \boldsymbol{\varepsilon}_0 dV - \int_{V_e} \mathbf{B}^T \boldsymbol{\sigma}_0 dV \right) \end{aligned} \quad (6.17)$$

Combining the expression for \mathbf{r} in Eqs. (6.13) and (6.15) we obtain

$$\mathbf{K} \mathbf{u} = \mathbf{f} \quad (6.18)$$

where

$$\mathbf{K} = \sum_{e=1}^{n_e} \mathbf{T}_e^T \mathbf{K}_e \mathbf{T}_e \quad (6.19)$$

is the system stiffness matrix, and \mathbf{f} is the right hand side vector defined by

$$\mathbf{f} = \mathbf{f}_g + \mathbf{f}_t + \mathbf{f}_{\varepsilon_0} - \mathbf{f}_{\sigma_0} + \mathbf{f}_c \quad (6.20)$$

with

$$\mathbf{f}_g = \int_V \mathbf{N}^T \mathbf{g} dV \quad \text{the contribution of the body forces.}$$

$$\mathbf{f}_t = \int_{S_t} \mathbf{N}^T \mathbf{t} dS_t \quad \text{the contribution of the surface tractions.}$$

$\mathbf{f}_{\varepsilon_0} = \int_V \mathbf{B}^T \mathbf{D} \boldsymbol{\varepsilon}_0 \, dV$ the effect of the initial strains.

$\mathbf{f}_{\sigma_0} = \int_V \mathbf{B}^T \boldsymbol{\sigma}_0 \, dV$ the effect of the initial stresses.

\mathbf{f}_c the contribution of the concentrated nodal forces.

This provides a set of linear simultaneous equations which can be solved in a direct or indirect way:

$$\mathbf{u} = \mathbf{K}^{-1} \mathbf{f} \quad (6.21)$$

The method of solution is outlined in § 9.7.

Chapter 7

Library of Finite Elements

This part provides an overview of the most popular elements for structural analysis.

7.1 Two-dimensional Body Elements

In a two-dimensional model the elements are defined in one flat surface and deformations and loadings are restricted to act in the plane of this surface. Three different configurations of a two-dimensional model can be distinguished:

- Plane stress (membrane) elements

These elements can be used for modelling walls or panels. They must be *plane*, i.e., the coordinates of the element nodes must be in one flat plane, the xy plane of the element [Fig. 7.1]. This plane may have any arbitrary orientation in the three-dimensional coordinate system XYZ . The elements must be thin, i.e., the thickness t must be small in relation to the dimensions b in the plane of the element. The thickness t of these elements is defined by the user. Loading F must act in the plane of the element. In these elements there are only stresses in the plane of the surface, and out-of-plane strains are considered as result of the Poisson effect.

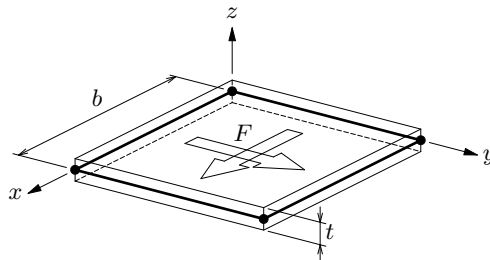


Figure 7.1: Plane stress elements, characteristics

- Plane strain elements

These elements can be used for a modelling a cross-section of (infinitely) long structure, such as a tunnel, a pile-sheet wall or a dam. They must be positioned in the model global XY plane, i.e., the Z coordinate of the element nodes must be zero [Fig. 7.2]. Loading F must act in the plane of the element. Plane strain elements are characterized by the fact that their thickness t is equal to unity and that the strain components perpendicular to the element face are zero: $\varepsilon_{zz} = 0$. The Poisson effect will result in out-of-plane stresses.

- Axisymmetric elements

These elements describe a solid ring. The two-dimensional elements are swept around the global Y axis into a solid ring [Fig. 7.3]. These elements may be used for an axisymmetric structure such as a cylindrical tank, cooling towers, tubes, sockets

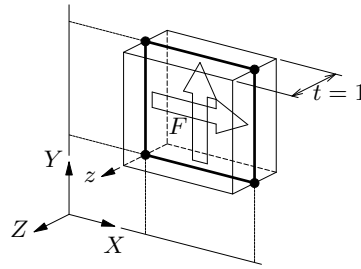


Figure 7.2: Plane strain elements, characteristics

or a cylindrical excavation pit. They must be positioned in the global XY plane, i.e., the Z coordinate of the element nodes must be zero. DIANA considers the Y axis as axis of rotational symmetry, therefore each element models a ring. Loading F must act in the plane of the element. In these elements there are only strains in the plane of the element. Circumferential stress is considered and calculated from the Poisson effect and circumferential strain resulting from radial displacements. The circumference of the solid ring is defined by the distance of the element to the axis of rotation which is the global Y axis.

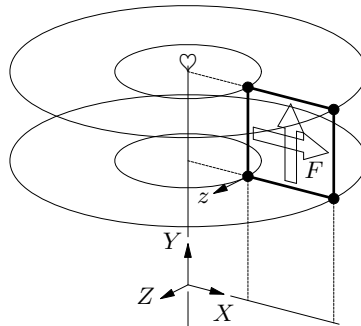


Figure 7.3: Axisymmetric elements, characteristics

7.1.1 Element Axes

For two-dimensional body elements no specific user input is required to set up the element axes. By default, the local element x axis points from the first to the second node of the

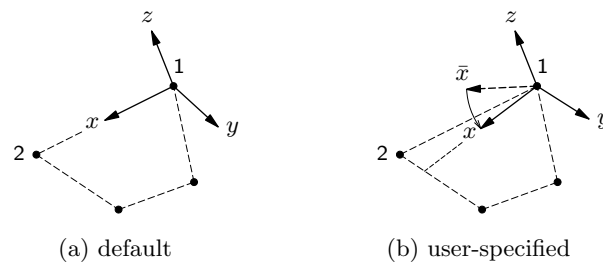


Figure 7.4: Axes for two-dimensional body elements

flat plane stress element. The z axis is always perpendicular to the element plane and $y \perp zx$ according to the right-hand-rule [Fig. 7.4a].

Note that the y axis only points along an element edge if the two edges adjacent to the first node are perpendicular. Note also that the default element axes are fully independent of the global model XYZ axes.

If you prefer an x axis other than default, for instance to get the same axes directions for various elements, then you must specify an \bar{x} axis which DIANA uses to set up the real x axis [Fig. 7.4b]. First the z axis is put perpendicular to the element plane. Then $y \perp z\bar{x}$ is created and finally $x \perp yz$.

7.1.2 Displacements and Rotations

The basic variables of the two-dimensional body elements are the translations of the nodes: u_x and u_y in element xy direction [Fig. 7.5a]. DIANA offers some special plane stress

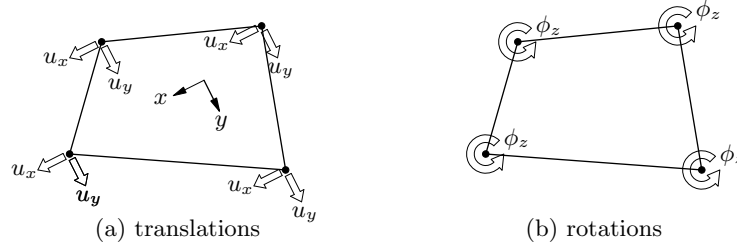


Figure 7.5: Displacements of two-dimensional body elements

elements with the rotations ϕ_z , the so-called *drilling* rotation, as additional variables [Fig. 7.5b].

7.1.3 Strains and Stresses

- Plane stress (membrane) elements

The displacement field yield the deformations du_x and du_y of an infinitesimal part dx dy of the element [Fig. 7.6]. From these deformations, DIANA derives the Green–

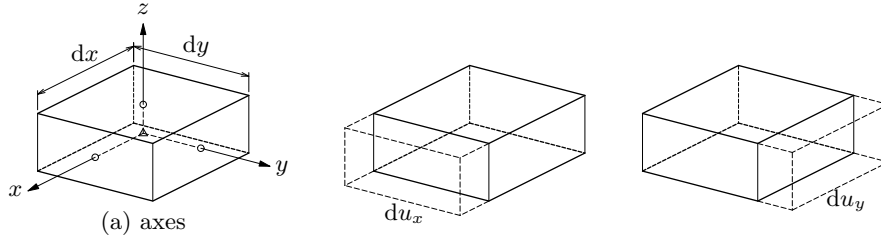


Figure 7.6: Deformation

Lagrange strains of Eq. (7.1).

$$\boldsymbol{\varepsilon} = \begin{Bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{zz} \\ \gamma_{xy} \end{Bmatrix} \quad (7.1)$$

with for isotropic elements

$$\varepsilon_{xx} = \frac{\partial u_x}{\partial x} \quad \varepsilon_{yy} = \frac{\partial u_y}{\partial y} \quad \varepsilon_{zz} = \frac{\nu(\varepsilon_{xx} + \varepsilon_{yy})}{1 - \nu} \quad \gamma_{xy} = \frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x} \quad (7.2)$$

For orthotropic elements DIANA fills in a ‘dummy’ $\varepsilon_{zz} = 0$. The sign convention for strains is that an elongation yields a positive strain.

DIANA can calculate and output two types of stresses for plane stress elements: Cauchy stresses and generalized forces [Fig. 7.7].

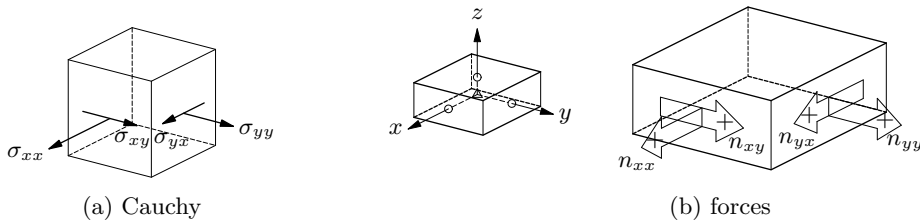


Figure 7.7: Stresses

From the basic strains of Eq. (7.2) DIANA derives the following Cauchy stresses.

$$\boldsymbol{\sigma} = \begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} = 0 \\ \sigma_{xy} = \sigma_{yx} \end{Bmatrix} \quad (7.3)$$

Figure 7.7a shows these stresses on a unit cube in their positive direction. Note that tension stress is positive.

By integration over the thickness, DIANA can calculate the normal and shear forces of Eq. (7.4).

$$\mathbf{n} = \begin{Bmatrix} n_{xx} \\ n_{yy} \\ n_{xy} = n_{yx} \end{Bmatrix} \quad (7.4)$$

Figure 7.7b shows these forces on the infinitesimal part [Fig. 7.6a p. 23] in their positive direction (tension).

- Plane strain and axisymmetric elements

The displacements in the nodes yield the deformations du_x and du_y of an infinitesimal part $dx \, dy$ of the plane strain element [Fig. 7.8].

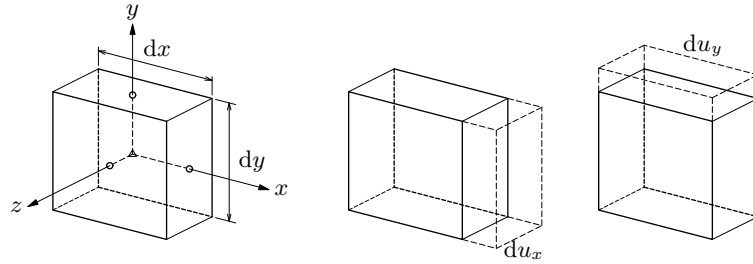


Figure 7.8: Deformation

The displacements in the nodes yield the deformations du_x and du_y of an infinitesimal part $dx \, dy$ with unit thickness in tangential direction t of the axisymmetric element [Fig. 7.9].

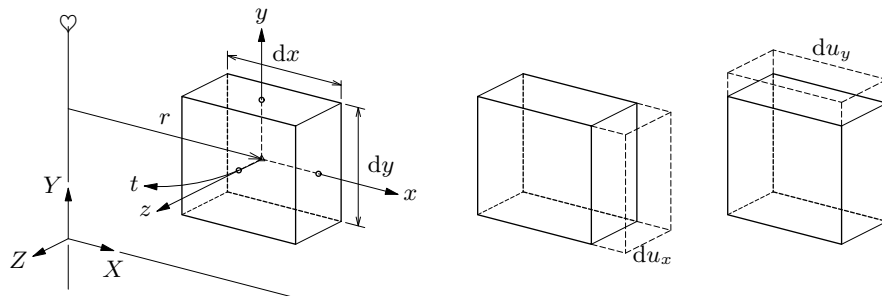


Figure 7.9: Deformation

From these deformations, DIANA derives the Green-Lagrange strains in the local xyz directions of Eq. (7.5).

$$\boldsymbol{\varepsilon} = \begin{Bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{zz} \\ \gamma_{xy} \end{Bmatrix} \quad (7.5)$$

with

$$\varepsilon_{xx} = \frac{\partial u_x}{\partial x} \quad \varepsilon_{yy} = \frac{\partial u_y}{\partial y} \quad \varepsilon_{zz} = 0 \quad \gamma_{xy} = \frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x} \quad (7.6)$$

From the basic strains of Eq. (7.6) DIANA derives the Cauchy stresses of Eq. (7.7).

$$\boldsymbol{\sigma} = \begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{xy} = \sigma_{yx} \end{Bmatrix} \quad (7.7)$$

Figure 7.10 shows these stresses on a unit cube in their positive direction. Note that tension stress is positive.

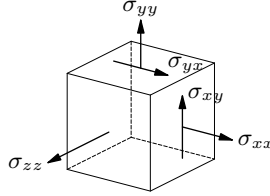


Figure 7.10: Cauchy stresses

7.1.4 Element Types

Regular two-dimensional structural element types for bodies in DIANA can be triangles or quadrilaterals. The elements may have linear or quadratic interpolation functions for the displacement field. For elements with linear interpolation functions there are only two nodes at each edge, whereas for elements with quadratic interpolation functions there are three nodes at each edge. There is also a quadrilateral element with nine nodes and full quadratic interpolation functions and a triangular element with fourth order interpolation functions for the displacement field. The last two elements have internal nodes. The next paragraphs [§ 7.1.4.1 to § 7.1.4.21] give an overview of the different element types where the numbers display the relative node numbers for the element connectivity. In DIANA the relative node numbers are always defined counterclockwise. The isoparametric coordinates ξ and η are also displayed for the different element shapes.

7.1.4.1 T6MEM – triangle, 3 nodes

The T6MEM element [Fig. 7.11] is a three-node triangular isoparametric plane stress

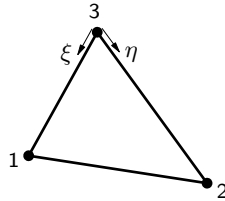


Figure 7.11: T6MEM

element. It is based on linear interpolation and area integration. The polynomial for the displacements u_x and u_y can be expressed as

$$u_i(\xi, \eta) = a_0 + a_1\xi + a_2\eta \quad (7.8)$$

Typically, this polynomial yields strains which are constant over the element area. By default DIANA applies 1-point integration, 3- and 4-point are suitable options. Schemes higher than 4-point are unsuitable. [$n_{lc} = 1$]

7.1.4.2 Q8MEM – quadrilateral, 4 nodes

The Q8MEM element [Fig. 7.12] is a four-node quadrilateral isoparametric plane stress

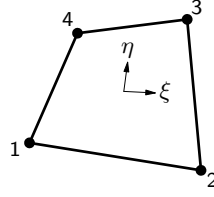


Figure 7.12: Q8MEM

element. It is based on linear interpolation and Gauss integration. The polynomial for the displacements u_x and u_y can be expressed as

$$u_i(\xi, \eta) = a_0 + a_1\xi + a_2\eta + a_3\xi\eta \quad (7.9)$$

Typically, this polynomial yields a strain ε_{xx} which is constant in x direction and varies linearly in y direction and a strain ε_{yy} which is constant in y direction and varies linearly in x direction. For constant shear, which is default, the Q8MEM element yields a constant shear strain γ_{xy} over the element area. By default DIANA applies 2×2 integration, 1×1 is a suitable option for which DIANA applies a stabilization procedure to avoid zero-energy modes. Schemes higher than 2×2 are unsuitable.

$[n_\xi = 2, n_\eta = 2]$

7.1.4.3 CT12M – triangle, 6 nodes

The CT12M element [Fig. 7.13] is a six-node triangular isoparametric plane stress element.

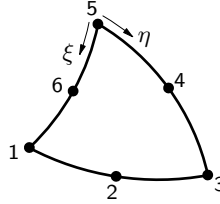


Figure 7.13: CT12M

It is based on quadratic interpolation and area integration. The polynomial for the displacements u_x and u_y can be expressed as

$$u_i(\xi, \eta) = a_0 + a_1\xi + a_2\eta + a_3\xi\eta + a_4\xi^2 + a_5\eta^2 \quad (7.10)$$

Typically, this polynomial yields an approximately linear strain variation in x and y direction. By default DIANA applies 3-point integration. Optional schemes are available, but these are unsuitable because they may cause an oscillating strain distribution.

$[n_{lc} = 3]$

7.1.4.4 CQ16M – quadrilateral, 8 nodes

The CQ16M element [Fig. 7.14] is an eight-node quadrilateral isoparametric plane stress

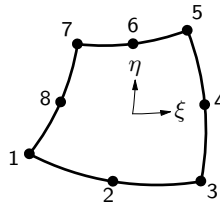


Figure 7.14: CQ16M

element. It is based on quadratic interpolation and Gauss integration. The polynomial for the displacements u_x and u_y can be expressed as

$$u_i(\xi, \eta) = a_0 + a_1\xi + a_2\eta + a_3\xi\eta + a_4\xi^2 + a_5\eta^2 + a_6\xi^2\eta + a_7\xi\eta^2 \quad (7.11)$$

Typically, this polynomial yields a strain ε_{xx} which varies linearly in x direction and quadratically in y direction. The strain ε_{yy} varies linearly in y direction and quadratically in x direction. The shear strain γ_{xy} varies quadratically in both directions. By default DIANA applies 2×2 integration which yields optimal stress points, 3×3 is a suitable option. Schemes higher than 3×3 are unsuitable. $[n_\xi = 2, n_\eta = 2]$

7.1.4.5 CQ18M – quadrilateral, 9 nodes

The CQ18M element [Fig. 7.15] is a nine-node quadrilateral isoparametric plane stress

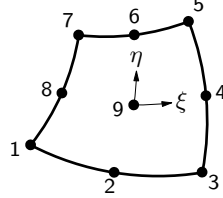


Figure 7.15: CQ18M

element. It is based on quadratic Lagrange interpolation and Gauss integration. The polynomial for the displacements u_x and u_y can be expressed as

$$u_i(\xi, \eta) = a_0 + a_1\xi + a_2\eta + a_3\xi\eta + a_4\xi^2 + a_5\eta^2 + a_6\xi^2\eta + a_7\xi\eta^2 + a_8\xi^2\eta^2 \quad (7.12)$$

Typically, this polynomial yields a strain ε_{xx} which varies linearly in x direction and quadratically in y direction. The strain ε_{yy} varies linearly in y direction and quadratically in x direction. The shear strain γ_{xy} varies quadratically in both directions. By default DIANA applies 3×3 integration, 2×2 is a suitable option. Schemes higher than 3×3 are unsuitable. $[n_\xi = 3, n_\eta = 3]$

7.1.4.6 T9MEM – triangle, 3 nodes

The T9MEM element [Fig. 7.16] is a three-node triangular isoparametric plane stress

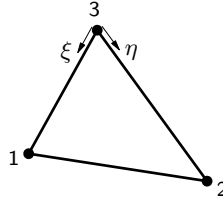


Figure 7.16: T9MEM

element with drilling rotation. In each node there are two translational degrees of freedom defined in the element plane, u_x and u_y , and one rotational degree of freedom around the axis normal to the element plane, ϕ_z . The geometry is interpolated by a linear function and the displacements by linear and hierarchical quadratic functions. The polynomial for the translations u_x and u_y can be expressed as

$$u_i(\xi, \eta) = a_0 + a_1\xi + a_2\eta + h_3\xi\eta + h_4\xi^2 + h_5\eta^2 \quad (7.13)$$

Typically, this polynomial yields a strain which varies linearly over the element area. By default DIANA applies 3-point integration, 1- and 4-point are suitable options. Schemes higher than 4-point are unsuitable. $[n_{lc} = 3]$

7.1.4.7 Q12ME – quadrilateral, 4 nodes

The Q12ME element [Fig. 7.17] is a four-node quadrilateral isoparametric plane stress element with drilling rotation. In each node there are two translational degrees of freedom

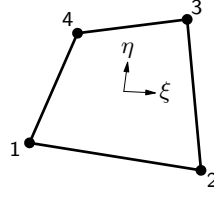


Figure 7.17: Q12ME

defined in the element plane, u_x and u_y , and one rotational degree of freedom around the axis normal to the element plane, ϕ_z . It is based on linear interpolation and area integration. The polynomial for the translations u_x and u_y can be expressed as

$$u_i(\xi, \eta) = a_0 + a_1\xi + a_2\eta + a_3\xi\eta + h_4\xi^2 + h_5\eta^2 + h_6\xi^2\eta + h_7\xi\eta^2 \quad (7.14)$$

Typically, this polynomial yields a strain which varies linearly over the element area. By default DIANA applies 2×2 integration. Optional schemes are available. However, schemes other than 2×2 are unsuitable.

$[n_\xi = 2, n_\eta = 2]$

7.1.4.8 T6OME – triangle, 3 nodes

The T6OME element [Fig. 7.18] is a three-node triangular isoparametric plane stress ele-

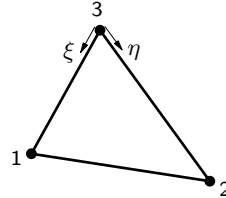


Figure 7.18: T6OME

ment with orthotropic thickness. It is based on linear interpolation and area integration. The polynomial for the displacements u_x and u_y can be expressed as

$$u_i(\xi, \eta) = a_0 + a_1\xi + a_2\eta \quad (7.15)$$

Typically, this polynomial yields a strain which is constant over the element area. By default DIANA applies 1-point integration, 3- and 4-point are suitable options. Schemes higher than 4-point are unsuitable.

$[n_{ic} = 1]$

7.1.4.9 Q8OME – quadrilateral, 4 nodes

The Q8OME element [Fig. 7.19] is a four-node quadrilateral isoparametric plane stress ele-

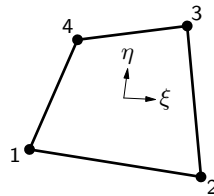


Figure 7.19: Q8OME

ment with orthotropic thickness. It is based on linear interpolation and Gauss integration. The polynomial for the displacements u_x and u_y can be expressed as

$$u_i(\xi, \eta) = a_0 + a_1\xi + a_2\eta + a_3\xi\eta \quad (7.16)$$

Typically, this polynomial yields a strain ε_{xx} which is constant in x direction and varies linearly in y direction. The strain ε_{yy} is constant in y direction and varies linearly in

$[n_\xi = 2, n_\eta = 2]$ x direction. The shear strain γ_{xy} varies linearly in both directions. By default DIANA applies 2×2 integration, 1×1 and 3×3 are suitable options. For the 1×1 scheme, DIANA applies a stabilization procedure to avoid zero-energy modes. Schemes higher than 3×3 are unsuitable.

7.1.4.10 CT12O – triangle, 6 nodes

The CT12O element [Fig. 7.20] is a six-node triangular isoparametric plane stress element

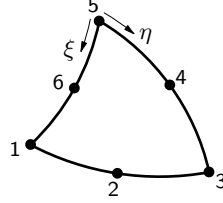


Figure 7.20: CT12O

with isotropic thickness. It is based on quadratic interpolation and area integration. The polynomial for the displacements u_x and u_y can be expressed as

$$u_i(\xi, \eta) = a_0 + a_1\xi + a_2\eta + a_3\xi\eta + a_4\xi^2 + a_5\eta^2 \quad (7.17)$$

Typically, this polynomial yields an approximately linear strain variation in x and y direction. By default DIANA applies 3-point integration, 4-point is a suitable option. $[n_{lc} = 3]$ Schemes higher than 4-point are unsuitable.

7.1.4.11 CQ16O – quadrilateral, 8 nodes

The CQ16O element [Fig. 7.21] is an eight-node quadrilateral isoparametric plane stress

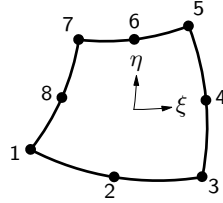


Figure 7.21: CQ16O

element with isotropic thickness. It is based on quadratic interpolation and Gauss integration. The polynomial for the displacements u_x and u_y can be expressed as

$$u_i(\xi, \eta) = a_0 + a_1\xi + a_2\eta + a_3\xi\eta + a_4\xi^2 + a_5\eta^2 + a_6\xi^2\eta + a_7\xi\eta^2 \quad (7.18)$$

Typically, this polynomial yields a strain ε_{xx} which varies linearly in x direction and quadratically in y direction. The strain ε_{yy} varies linearly in y direction and quadratically in x direction. The shear strain γ_{xy} varies quadratically in both directions. By default DIANA applies 2×2 integration which yields optimal stress points, 3×3 is a suitable option. $[n_\xi = 2, n_\eta = 2]$ Schemes higher than 3×3 are unsuitable.

7.1.4.12 T6EPS – triangle, 3 nodes

The T6EPS element [Fig. 7.22] is a three-node triangular isoparametric plane strain element. It is based on linear interpolation and area integration. The polynomial for the displacements u_X and u_Y can be expressed as

$$u_i(\xi, \eta) = a_0 + a_1\xi + a_2\eta \quad (7.19)$$

Typically, this polynomial yields a strain which is constant over the element area. By default DIANA applies a 1-point integration scheme, 3- and 4-point are suitable options. $[n_{lc} = 1]$ Schemes higher than 4-point are unsuitable.

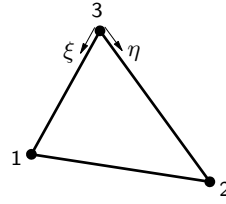


Figure 7.22: T6EPS

7.1.4.13 Q8EPS – quadrilateral, 4 nodes

The Q8EPS element [Fig. 7.23] is a four-node quadrilateral isoparametric plane strain

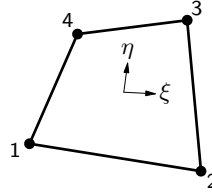


Figure 7.23: Q8EPS

element. It is based on linear interpolation and Gauss integration. The polynomial for the displacements u_X and u_Y can be expressed as

$$u_i(\xi, \eta) = a_0 + a_1\xi + a_2\eta + a_3\xi\eta \quad (7.20)$$

For constant shear, which is default, this polynomial yields a strain ε_{xx} which is constant in x direction and varies linearly in y direction and a strain ε_{yy} which is constant in y direction and varies linearly in x direction. The shear strain γ_{xy} is constant over the element area. By default DIANA applies a 2×2 integration scheme, 1×1 and 3×3 are suitable options. For the 1×1 scheme, DIANA applies a stabilization procedure to avoid zero-energy modes. Schemes higher than 3×3 are unsuitable.

$[n_\xi = 2, n_\eta = 2]$

7.1.4.14 CT12E – triangle, 6 nodes

The CT12E element [Fig. 7.24] is a six-node triangular isoparametric plane strain element.

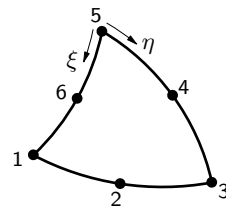


Figure 7.24: CT12E

It is based on quadratic interpolation and area integration. The polynomial for the displacements u_X and u_Y can be expressed as

$$u_i(\xi, \eta) = a_0 + a_1\xi + a_2\eta + a_3\xi\eta + a_4\xi^2 + a_5\eta^2 \quad (7.21)$$

Typically, this polynomial yields an approximately linear strain variation in x and y direction. By default DIANA applies a 3-point integration scheme, 1- and 4-point are suitable options. Schemes higher than 4-point are unsuitable.

$[n_{lc} = 3]$

7.1.4.15 CQ16E – quadrilateral, 8 nodes

The CQ16E element [Fig. 7.25] is an eight-node quadrilateral isoparametric plane strain

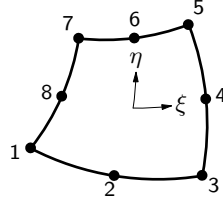


Figure 7.25: CQ16E

element. It is based on quadratic interpolation and Gauss integration. The polynomial for the displacements u_X and u_Y can be expressed as

$$u_i(\xi, \eta) = a_0 + a_1\xi + a_2\eta + a_3\xi\eta + a_4\xi^2 + a_5\eta^2 + a_6\xi^2\eta + a_7\xi\eta^2 \quad (7.22)$$

Typically, this polynomial yields a strain ε_{xx} which varies linearly in x direction and quadratically in y direction. The strain ε_{yy} varies linearly in y direction and quadratically in x direction. The shear strain γ_{xy} varies quadratically in both directions. By default DIANA applies a 2×2 integration scheme, which yields optimal stress points, 3×3 is a suitable option. Schemes higher than 3×3 are unsuitable. [$n_\xi = 2, n_\eta = 2$]

7.1.4.16 CT30E – triangle, 15 nodes

The CT30E element [Fig. 7.26] is a fifteen-node triangular isoparametric plane strain

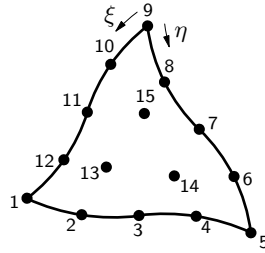


Figure 7.26: CT30E

element. It is based on fourth-order interpolation and area integration and particularly useful in nonlinear analysis to determine collapse loads of soil structures [Vol. *Analysis Procedures*]. The polynomial for the displacements u_X and u_Y can be expressed as

$$u_i(\xi, \eta) = a_0 + a_1\xi + a_2\eta + a_3\xi\eta + a_4\xi^2 + a_5\eta^2 + a_6\xi^2\eta + a_7\xi\eta^2 + a_8\xi^3 + a_9\eta^3 + \dots + a_{13}\xi^4 + a_{14}\eta^4 \quad (7.23)$$

Typically, this polynomial yields a strain ε_{xx} which varies third-order in x direction and fourth-order in y direction. The strain ε_{yy} varies third-order in y direction and third-order in x direction. The shear strain γ_{xy} varies fourth-order in both directions. By default DIANA applies a 12-point integration scheme, 7-point is a suitable option. [$n_{lc} = 12$]

7.1.4.17 T6AXI – triangle, 3 nodes

The T6AXI element [Fig. 7.27] is a three-node isoparametric axisymmetric solid ring element with a triangular cross-section. It is based on linear interpolation and area integration. The polynomial for the displacements u_X and u_Y can be expressed as

$$u_i(\xi, \eta) = a_0 + a_1\xi + a_2\eta \quad (7.24)$$

Typically, this polynomial yields a strain which is constant over the element area. By default DIANA applies a 1-point integration scheme, 3- and 4-point are suitable options. Schemes higher than 4-point are unsuitable. [$n_{lc} = 1$]

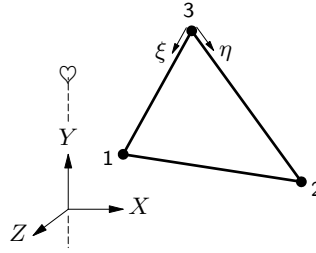


Figure 7.27: T6AXI

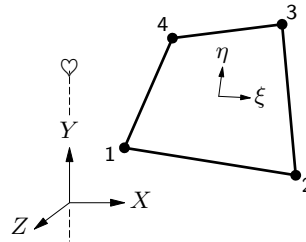


Figure 7.28: Q8AXI

7.1.4.18 Q8AXI – quadrilateral, 4 nodes

The Q8AXI element [Fig. 7.28] is a four-node isoparametric axisymmetric solid ring element with a quadrilateral cross-section. It is based on linear interpolation and Gauss integration. The polynomial for the displacements u_X and u_Y can be expressed as

$$u_i(\xi, \eta) = a_0 + a_1\xi + a_2\eta + a_3\xi\eta \quad (7.25)$$

Typically, this polynomial yields a strain ε_{xx} which is constant in x direction and varies linearly in y direction and a strain ε_{yy} which is constant in y direction and varies linearly in x direction. For the default ‘constant shear’ option, the shear strain γ_{xy} is constant over the element area. By default DIANA applies a 2×2 integration scheme. The 1×1 scheme is a suitable option for which DIANA applies a stabilization procedure to avoid zero-energy modes. Schemes higher than 2×2 are unsuitable.

$[n_\xi = 2, n_\eta = 2]$

7.1.4.19 CT12A – triangle, 6 nodes

The CT12A element [Fig. 7.29] is a six-node isoparametric axisymmetric solid ring element

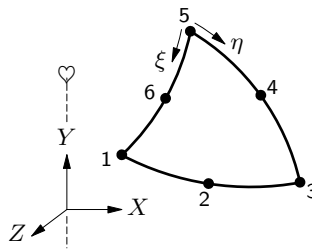


Figure 7.29: CT12A

with a triangular cross-section. It is based on quadratic interpolation and area integration. The polynomial for the displacements u_X and u_Y can be expressed as

$$u_i(\xi, \eta) = a_0 + a_1\xi + a_2\eta + a_3\xi\eta + a_4\xi^2 + a_5\eta^2 \quad (7.26)$$

Typically, this polynomial yields an approximately linear strain variation in x and y direction. By default DIANA applies a 4-point integration scheme, 1- and 3-point are suitable options.

$[n_{lc} = 4]$

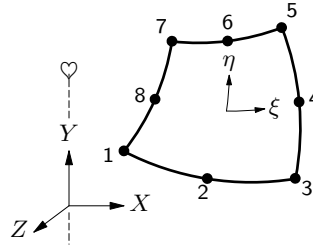


Figure 7.30: CQ16A

7.1.4.20 CQ16A – quadrilateral, 8 nodes

The CQ16A element [Fig. 7.30] is an eight-node isoparametric axisymmetric solid ring element with quadrilateral cross-section. It is based on quadratic interpolation and Gauss integration. The polynomial for the displacements u_X and u_Y can be expressed as

$$u_i(\xi, \eta) = a_0 + a_1\xi + a_2\eta + a_3\xi\eta + a_4\xi^2 + a_5\eta^2 + a_6\xi^2\eta + a_7\xi\eta^2 \quad (7.27)$$

Typically, this polynomial yields a strain ε_{xx} which varies linearly in x direction and quadratically in y direction. The strain ε_{yy} varies linearly in y direction and quadratically in x direction. The shear strain γ_{xy} varies quadratically in both directions. By default DIANA applies a 2×2 integration scheme which yields optimal stress points, 3×3 is a suitable option. Schemes higher than 3×3 are unsuitable. [$n_\xi = 2, n_\eta = 2$]

7.1.4.21 CT30A – triangle, 15 nodes

The CT30A element [Fig. 7.31] is a fifteen-node isoparametric axisymmetric solid ring

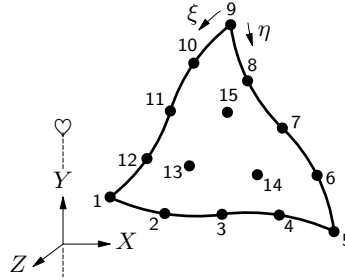


Figure 7.31: CT30A

element with triangular cross-section. It is based on fourth-order interpolation and area integration and particularly useful in nonlinear analysis to determine collapse loads of soil structures [Vol. *Nonlinear Analysis*]. The polynomial for the displacements u_X and u_Y can be expressed as

$$\begin{aligned} u_i(\xi, \eta) = & a_0 + a_1\xi + a_2\eta + a_3\xi\eta + a_4\xi^2 + a_5\eta^2 + \\ & a_6\xi^2\eta + a_7\xi\eta^2 + a_8\xi^3 + a_9\eta^3 + a_{10}\xi^3\eta + \\ & a_{11}\xi^2\eta^2 + a_{12}\xi\eta^3 + a_{13}\xi^4 + a_{14}\eta^4 \end{aligned} \quad (7.28)$$

Typically, this polynomial yields a strain ε_{xx} which varies third-order in x direction and fourth-order in y direction. The strain ε_{yy} varies third-order in y direction and third-order in x direction. The shear strain γ_{xy} varies fourth-order in both directions. By default DIANA applies a 12-point integration scheme, 7-point is a suitable option. [$m_c = 12$]

7.1.5 Integration Schemes and Assumed Strain Options

For all two-dimensional body elements, DIANA performs a numerical integration with an appropriate default scheme. You may choose an alternative scheme via a special data input, depending on the shape of the element.

To enhance the behaviour of linear quadrilaterals in certain situations, assumed strain options may be applied. For more details see Volume *Element Library*.

7.1.6 Thickness Definition for Plane Stress Elements

Isotropic thickness, i.e., the thickness in a point is independent of the direction of the cross-section, of plane stress elements may be uniform or nonuniform. Orthotropic thickness may be input for some special plane stress elements. If you only specify $t1$ then

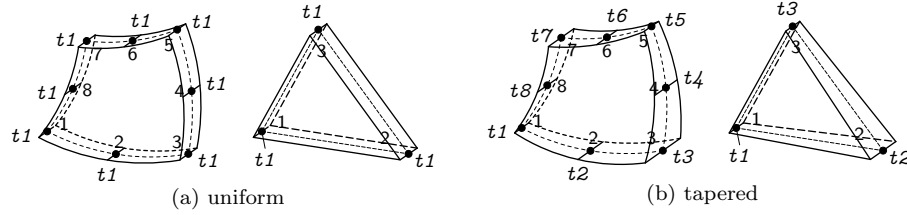


Figure 7.32: Thickness

the thickness is uniform [Fig. 7.32a]. Nonuniform (tapered) thickness varies linearly or quadratically over the element area depending on the order of the element. For non-uniform thickness the thickness is defined in every node of the element [Fig. 7.32b].

The basic variables for plane stress elements with orthotropic thickness are the translations u_x and u_y . The derived variables are the Green–Lagrange strains [Eq. (7.1) p. 23] with $\varepsilon_{zz} = 0$ and the generalized forces [Eq. (7.4) p. 24]. Cauchy stresses cannot be output for these elements. For plane stress elements with an orthotropic thickness, the relation between the in-plane forces n and the strains ε is given by

$$\begin{Bmatrix} n_{xx} \\ n_{yy} \\ n_{xy} \end{Bmatrix} = \begin{bmatrix} c_1 A_{xx} & c_3 A_{\nu} & 0 \\ c_3 A_{\nu} & c_2 A_{yy} & 0 \\ 0 & 0 & c_4 A_{xy} \end{bmatrix} \cdot \begin{Bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{xy} \end{Bmatrix} \quad (7.29)$$

where the factors c are based on the material parameters and the A 's are the characteristic areas for cross-section. The parameters A_{xx} , A_{yy} , A_{xy} , and A_{ν} are direct user input, whereas the parameters c_1 , c_2 , c_3 , and c_4 are derived from the linear elastic material parameters. For an isotropic material model:

$$\begin{aligned} c_1 &= E; \\ c_2 &= E; \\ c_3 &= \nu * E; \\ c_4 &= \frac{(1-\nu)}{2} * E \end{aligned} \quad (7.30)$$

with E the Young's modulus and ν the Poisson's ratio and for an orthotropic material model:

$$\begin{aligned} c_1 &= \frac{E_x}{(1-(E_y/E_x)*\nu_{xy}^2)}; \\ c_2 &= \frac{E_y}{(1-(E_y/E_x)*\nu_{xy}^2)}; \\ c_3 &= \frac{\nu * E_y}{(1-(E_y/E_x)*\nu_{xy}^2)}; \\ c_4 &= G_{xy} \end{aligned} \quad (7.31)$$

with E_x and E_y the Young's modulus in the respective local element directions, ν_{xy} the in-plane Poisson's ratio, and G_{xy} the in-plane shear modulus.

7.1.7 Linear Elastic Material Properties

The material properties to be input for linear elasticity are:

- Young's modulus E , always necessary.
- Poisson's ratio ν , necessary for two-dimensional and three-dimensional elements.
- Shear modulus G , necessary for orthotropic elasticity.
- Thermal expansion coefficient α , necessary to determine a temperature load.
- Concentration expansion coefficient γ , necessary to determine a concentration load.

Table 7.1: LINEAR ORTHOTROPIC ELASTICITY

	pl. stress	pl. strain	axisymm.
Young's modulus	E_x	E_x	E_x
	E_y	E_y	E_y
		E_z	E_z
Poisson's ratio	ν_{xy}	ν_{xy}	ν_{xy}
		ν_{yz}	ν_{yz}
		ν_{xz}	ν_{xz}
Shear modulus	G_{xy}	G_{xy}	G_{xy}
Thermal exp.	α_x	α_x	α_x
	α_y	α_y	α_y
		α_z	α_z
Concentr. exp.	γ_x	γ_x	γ_x
	γ_y	γ_y	γ_y
		γ_z	γ_z

Table 7.1 shows the required input for orthotropic elasticity for the various element families.

For some materials the Young's modulus E may depend on the position of the material in space. A typical example is soil where E may vary with the depth in the soil layer. To model such a dependency, DIANA can apply position dependent characteristics on isotropic elasticity without temperature influence. In this case the user defines the reference Young's modulus E_{ref} , the reference position with the coordinates $(X_{\text{ref}}, Y_{\text{ref}}, Z_{\text{ref}})$, and the gradient of the Young's modulus in the global XYZ directions:

$$\begin{aligned}\partial E / \partial X &= grx \\ \partial E / \partial Y &= gry \\ \partial E / \partial Z &= grz\end{aligned}\tag{7.32}$$

DIANA will calculate the Young's modulus for each element integration point that is located below the reference position via linear interpolation:

$$E(X, Y, Z) = E_{\text{ref}} + (X - X_{\text{ref}}) \frac{\partial E}{\partial X} + (Y - Y_{\text{ref}}) \frac{\partial E}{\partial Y} + (Z - Z_{\text{ref}}) \frac{\partial E}{\partial Z}\tag{7.33}$$

where E_{ref} is the reference Young's modulus. For integration points located above the reference position the reference Young's modulus E_{ref} without gradients will be applied.

7.2 Three-dimensional Bodies or Solids

Solid elements are general purpose elements which are very attractive, because arbitrary geometries can easily be filled with solid elements, without the necessary engineering consideration to reduce real geometries to reduced geometries as necessary for beam, plate and/or shell elements. In earlier years, because of their tendency to produce large systems of equations, these elements are usually applied only when other elements are unsuitable or would produce inaccurate analysis results. Today, because the hardware is very powerful, analysis procedures are much efficient, and cross-section forces and bending moments can be calculated efficiently using composed elements, solid elements can be used efficiently in daily engineering work. Solid elements are characterized by the following properties [Fig. 7.33]: the stress situation is three-dimensional, the loading may be arbitrary, the dimension in three axial directions X , Y and Z are of the same order of magnitude. Typical applications of solid elements are the analysis of voluminous structures like concrete foundations and off-shore structures, thick walls and floors, and soil-structure interaction.

7.2.1 Element Axes

For solid elements DIANA needs no special user input data to set up the element axes. By default, the element x , y and z axes are set up parallel to the global X , Y and Z axes

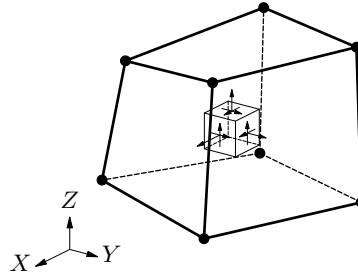


Figure 7.33: Solid elements, characteristics

respectively [Fig. 7.34a]. If you prefer element xyz axes other than default, for instance

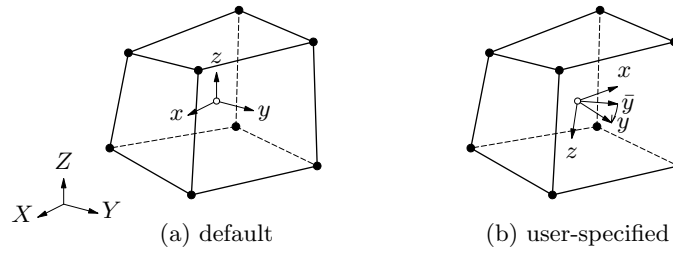


Figure 7.34: Axes

to get the primary stresses in a specific direction, then you must explicitly specify the direction of the x axis. In that case you must also specify an \bar{y} axis which DIANA uses to set up the real y and z axes. [Fig. 7.34b]. First the z axis is put perpendicular to the x and \bar{y} axes. Then $y \perp zx$ is created. The specified direction of the \bar{y} axis may not coincide with the specified x axis.

7.2.2 Displacements

The basic variables in the nodes of solid elements are the translations u_x , u_y and u_z in the local element directions [Fig. 7.35].

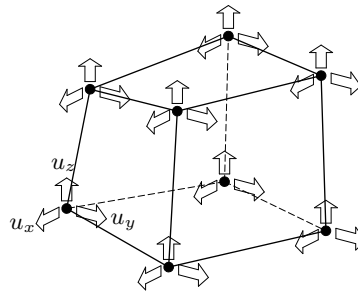


Figure 7.35: Displacements

$$\mathbf{u}_e = \begin{Bmatrix} u_x \\ u_y \\ u_z \end{Bmatrix} \quad (7.34)$$

7.2.3 Strains and Stresses

The displacements in the nodes yield the deformations du_x , du_y and du_z of an infinitesimal part $dx \, dy \, dz$ of the element [Fig. 7.36]. From these deformations, DIANA derives

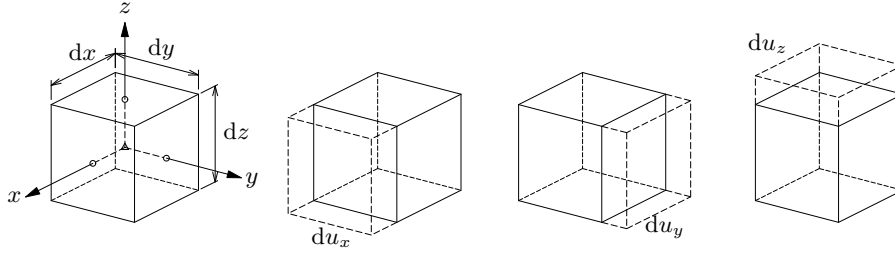


Figure 7.36: Deformation

the Green-Lagrange strains of Eq. (7.35).

$$\boldsymbol{\varepsilon} = \begin{Bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{zz} \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{zx} \end{Bmatrix} \quad (7.35)$$

with

$$\begin{aligned} \varepsilon_{xx} &= \frac{\partial u_x}{\partial x} & \varepsilon_{yy} &= \frac{\partial u_y}{\partial y} & \varepsilon_{zz} &= \frac{\partial u_z}{\partial z} \\ \gamma_{xy} &= \frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x} & \gamma_{yz} &= \frac{\partial u_y}{\partial z} + \frac{\partial u_z}{\partial y} & \gamma_{zx} &= \frac{\partial u_z}{\partial x} + \frac{\partial u_x}{\partial z} \end{aligned} \quad (7.36)$$

These Green-Lagrange strains are derived for all integration points and may be extrapolated to the nodes. The sign convention for strains is that an elongation yields a positive strain.

DIANA can calculate and output Cauchy stresses for all types of solid elements. From the basic strains of Eq. (7.36) DIANA derives the Cauchy stresses of Eq. (7.37) in the integration points.

$$\boldsymbol{\sigma} = \begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{xy} = \sigma_{yx} \\ \sigma_{yz} = \sigma_{zy} \\ \sigma_{zx} = \sigma_{xz} \end{Bmatrix} \quad (7.37)$$

Figure 7.37 shows these stresses on a unit cube in their positive direction. Note that

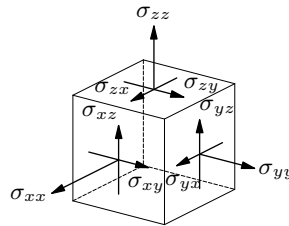


Figure 7.37: Cauchy stresses

tension stress is positive. Forces and bending moments in cross-sections defined with solid elements can be calculated by using composed surface or line elements [§ 7.7 p. 90].

7.2.4 Element Types

Regular three-dimensional structural element types for bodies in DIANA can be tetrahedrons, pentahedrons, or hexahedrons. The elements may have linear, quadratic, or cubic interpolation functions for the displacement field. For elements with linear interpolation functions there are two nodes at each edge, whereas for elements with quadratic

interpolation functions there are three nodes at each edge, and for elements with cubic interpolation functions the number of nodes along an edge is equal to four. The next paragraphs [§ 7.2.4.1 to § 7.2.4.11] give an overview of the different element types where the numbers display the relative node numbers for the element connectivity. In DIANA the relative node numbers are always defined counterclockwise. The isoparametric coordinates ξ , η , and ζ are also displayed for the different element shapes.

7.2.4.1 TE12L – tetrahedron, 3 sides, 4 nodes

The TE12L element [Fig. 7.38] is a four-node, three-side isoparametric solid tetrahedron

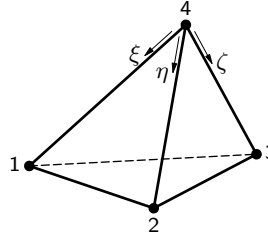


Figure 7.38: TE12L

element. It is based on linear interpolation and numerical integration. The polynomials for the translations u_{xyz} can be expressed as

$$u_i(\xi, \eta, \zeta) = a_0 + a_1\xi + a_2\eta + a_3\zeta \quad (7.38)$$

These polynomials yield a constant strain and stress distribution over the element volume. By default DIANA applies a 1-point integration scheme over the volume, 4- and 5-point are suitable options. Schemes higher than 5-point are not available.

[$n_{ic} = 1$]

7.2.4.2 PY15L – pyramid, 5 nodes

The PY15L element [Fig. 7.39] is a five-node isoparametric solid pyramid element as de-

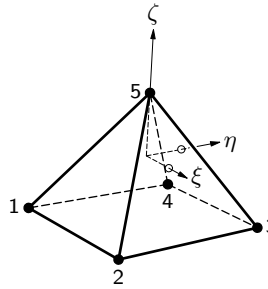


Figure 7.39: PY15L

scribed by the University of Colorado [79]. It is based on linear interpolation and numerical integration. The polynomials for the translations u_{xyz} are based on a isoparametric eight-node brick element where nodes 5 to 8 of the brick element are collapsed to apex 5 of the pyramid element. The polynomials can be expressed as

$$\begin{aligned} u_1(\xi, \eta, \zeta) &= \frac{1}{8} (1 - \xi) (1 - \eta) (1 - \zeta) \\ u_2(\xi, \eta, \zeta) &= \frac{1}{8} (1 + \xi) (1 - \eta) (1 - \zeta) \\ u_3(\xi, \eta, \zeta) &= \frac{1}{8} (1 + \xi) (1 + \eta) (1 - \zeta) \\ u_4(\xi, \eta, \zeta) &= \frac{1}{8} (1 - \xi) (1 + \eta) (1 - \zeta) \\ u_5(\xi, \eta, \zeta) &= \frac{1}{2} (1 + \zeta) \end{aligned} \quad (7.39)$$

Typically, a pyramid element approximates the following strain and stress distribution over the element volume. The strain ε_{xx} and stress σ_{xx} are constant in x direction and vary linearly in y and z direction. The strain ε_{yy} and stress σ_{yy} are constant in y direction and vary linearly in x and z direction. The strain ε_{zz} and stress σ_{zz} are constant in z direction and vary linearly in x and y direction. By default DIANA applies a 5-point integration scheme over the volume, a 1-point integration scheme is a suitable option. [$n_{lc} = 5$]

7.2.4.3 TP18L – wedge, 6 nodes

The TP18L element [Fig. 7.40] is a six-node isoparametric solid wedge element. It is based

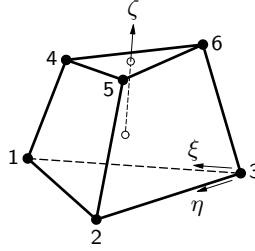


Figure 7.40: TP18L

on linear area interpolation in the triangular domain and a linear isoparametric interpolation in the ζ direction. The polynomials for the translations u_{xyz} can be expressed as

$$u_i(\xi, \eta, \zeta) = a_0 + a_1\xi + a_2\eta + a_3\zeta + a_4\xi\zeta + a_5\eta\zeta \quad (7.40)$$

These polynomials yield a constant strain and stress distribution over the element volume. By default DIANA applies a 1-point integration scheme in the triangular domain and 2-point in the ζ direction. Schemes higher than 1×2 are unsuitable. [$n_{lc} = 1$]
[$n_\zeta = 2$]

7.2.4.4 HX24L – brick, 8 nodes

The HX24L element [Fig. 7.41] is an eight-node isoparametric solid brick element. It is

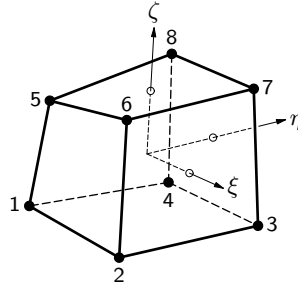


Figure 7.41: HX24L

based on linear interpolation and Gauss integration. The polynomials for the translations u_{xyz} can be expressed as

$$u_i(\xi, \eta, \zeta) = a_0 + a_1\xi + a_2\eta + a_3\zeta + a_4\xi\eta + a_5\eta\zeta + a_6\zeta\xi + a_7\xi\eta\zeta \quad (7.41)$$

Typically, a rectangular brick element approximates the following strain and stress distribution over the element volume. The strain ε_{xx} and stress σ_{xx} are constant in x direction and vary linearly in y and z direction. The strain ε_{yy} and stress σ_{yy} are constant in y direction and vary linearly in x and z direction. The strain ε_{zz} and stress σ_{zz} are constant in z direction and vary linearly in x and y direction. If bending behaviour is dominant the use of incompatible bubble displacement modes is advised. By default DIANA applies a $2 \times 2 \times 2$ integration scheme, $1 \times 1 \times 1$ is a suitable option if all assumed strain options are suppressed. The $1 \times 1 \times 1$ integration scheme is based on uniform strain assumption with orthogonal hourglass control according to Flanagan and Belytschko [26]. Schemes higher than $2 \times 2 \times 2$ are unsuitable. [$n_\xi = 2, n_\eta = 2, n_\zeta = 2$]

7.2.4.5 CTE30 – tetrahedron, 3 sides, 10 nodes

The CTE30 element [Fig. 7.42] is a ten-node, three-side isoparametric solid tetrahedron el-

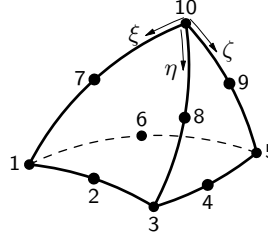


Figure 7.42: CTE30

ement. It is based on quadratic interpolation and numerical integration. The polynomials for the translations u_{xyz} can be expressed as

$$u_i(\xi, \eta, \zeta) = a_0 + a_1\xi + a_2\eta + a_3\zeta + a_4\xi\eta + a_5\eta\zeta + a_6\xi\eta + a_7\xi^2 + a_8\eta^2 + a_9\zeta^2 \quad (7.42)$$

Typically, these polynomials yield a linearly varying strain and stress distribution over the element volume. By default DIANA applies a 4-point integration scheme over the volume, 1- and 5-point are suitable options. Schemes higher than 5-point are not available.

[$n_{ic} = 4$]

7.2.4.6 CPY39 – pyramid, 13 nodes

The CPY39 element [Fig. 7.43] is a thirteen-node isoparametric solid pyramid element

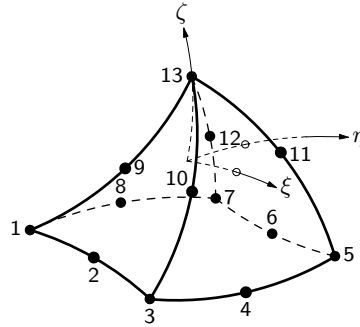


Figure 7.43: CPY39

as described by the University of Colorado [79]. It is based on quadratic interpolation and integration. The polynomials for the translations u_{xyz} are based on a isoparametric twenty-node brick element where nodes 13 to 20 of the brick element are collapsed to apex 13 of the pyramid element. Face bubble functions are added to assure compatibility over the apex faces when attached to a ten-node tetrahedron element. The polynomials can

be expressed as

$$\begin{aligned}
u_1(\xi, \eta, \zeta) &= -\frac{1}{16} (1 - \xi) (1 - \eta) (1 - \zeta) (4 + 3\xi + 3\eta + 2\xi\eta + 2\zeta + \xi\zeta + \eta\zeta + 2\xi\eta\zeta) \\
u_2(\xi, \eta, \zeta) &= \frac{1}{8} (1 - \xi^2) (1 - \eta) (1 - \zeta) (2 + \eta + \eta\zeta) \\
u_3(\xi, \eta, \zeta) &= -\frac{1}{16} (1 + \xi) (1 - \eta) (1 - \zeta) (4 - 3\xi + 3\eta - 2\xi\eta + 2\zeta - \xi\zeta + \eta\zeta - 2\xi\eta\zeta) \\
u_4(\xi, \eta, \zeta) &= \frac{1}{8} (1 + \xi) (1 - \eta^2) (1 - \zeta) (2 - \xi - \xi\zeta) \\
u_5(\xi, \eta, \zeta) &= -\frac{1}{16} (1 + \xi) (1 + \eta) (1 - \zeta) (4 - 3\xi - 3\eta + 2\xi\eta + 2\zeta - \xi\zeta - \eta\zeta + 2\xi\eta\zeta) \\
u_6(\xi, \eta, \zeta) &= \frac{1}{8} (1 - \xi^2) (1 + \eta) (1 - \zeta) (2 - \eta - \eta\zeta) \\
u_7(\xi, \eta, \zeta) &= -\frac{1}{16} (1 - \xi) (1 + \eta) (1 - \zeta) (4 + 3\xi - 3\eta - 2\xi\eta + 2\zeta + \xi\zeta - \eta\zeta - 2\xi\eta\zeta) \\
u_8(\xi, \eta, \zeta) &= \frac{1}{8} (1 - \xi) (1 - \eta^2) (1 - \zeta) (2 + \xi + \xi\zeta) \\
u_9(\xi, \eta, \zeta) &= \frac{1}{4} (1 - \xi) (1 - \eta) (1 - \zeta^2) \\
u_{10}(\xi, \eta, \zeta) &= \frac{1}{4} (1 + \xi) (1 - \eta) (1 - \zeta^2) \\
u_{11}(\xi, \eta, \zeta) &= \frac{1}{4} (1 + \xi) (1 + \eta) (1 - \zeta^2) \\
u_{12}(\xi, \eta, \zeta) &= \frac{1}{4} (1 - \xi) (1 + \eta) (1 - \zeta^2) \\
u_{13}(\xi, \eta, \zeta) &= \frac{1}{2} \zeta (1 + \zeta)
\end{aligned} \tag{7.43}$$

Typically, a thirteen-node pyramid element approximates the following strain and stress distribution over the element volume. The strain ε_{xx} and stress σ_{xx} vary linearly in x direction and quadratically in y and z direction. The strain ε_{yy} and stress σ_{yy} vary linearly in y direction and quadratically in x and z direction. The strain ε_{zz} and stress σ_{zz} vary linearly in z direction and quadratically in x and y direction. DIANA always applies a 13-point integration scheme over the volume.

[$n_{lc} = 13$]

7.2.4.7 CTP45 – wedge, 15 nodes

The CTP45 element [Fig. 7.44] is a fifteen-node isoparametric solid wedge element. It

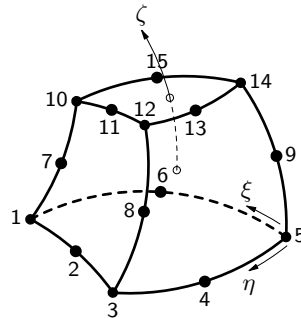


Figure 7.44: CTP45

is based on quadratic interpolation and numerical integration. The polynomials for the translations u_{xyz} can be expressed as

$$\begin{aligned}
u_i(\xi, \eta, \zeta) &= a_0 + a_1\xi + a_2\eta + a_3\zeta + a_4\xi\eta + a_5\eta\zeta + \\
&\quad a_6\xi\zeta + a_7\xi^2 + a_8\eta^2 + a_9\zeta^2 + a_{10}\xi\eta\zeta + \\
&\quad a_{11}\xi^2\zeta + a_{12}\eta^2\zeta + a_{13}\xi\zeta^2 + a_{14}\eta\zeta^2
\end{aligned} \tag{7.44}$$

These polynomials yield a strain and stress distribution which varies approximately linearly over the element volume. By default DIANA applies a 4-point integration scheme in the triangular domain and a 2-point scheme in the ζ direction. In a small element patch (\pm two elements), the 3×2 scheme may lead to zero-energy modes and a too soft behaviour. It is sufficient to apply a 4-point scheme in the triangular domain and a 3-point scheme in ζ direction, schemes of higher order are unsuitable.

7.2.4.8 CHX60 – brick, 20 nodes

The CHX60 element [Fig. 7.45] is a twenty-node isoparametric solid brick element. It is

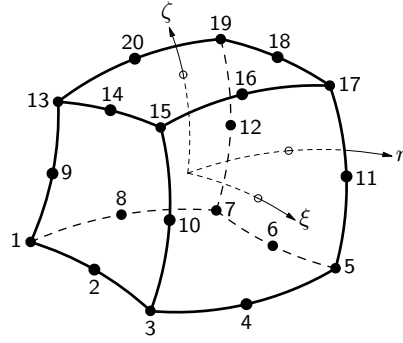


Figure 7.45: CHX60

based on quadratic interpolation and Gauss integration. The polynomials for the translations u_{xyz} can be expressed as

$$\begin{aligned}
 u_i(\xi, \eta, \zeta) = & a_0 + a_1\xi + a_2\eta + a_3\zeta + a_4\xi\eta + a_5\eta\zeta + \\
 & a_6\xi\zeta + a_7\xi^2 + a_8\eta^2 + a_9\zeta^2 + a_{10}\xi\eta\zeta + \\
 & a_{11}\xi^2\eta + a_{12}\xi^2\zeta + a_{13}\xi\eta^2 + a_{14}\xi\zeta^2 + a_{15}\eta^2\zeta + \\
 & a_{16}\eta\zeta^2 + a_{17}\xi^2\eta\zeta + a_{18}\xi\eta^2\zeta + a_{19}\xi\eta\zeta^2
 \end{aligned} \tag{7.45}$$

Typically, a rectangular brick element approximates the following strain and stress distribution over the element volume. The strain ε_{xx} and stress σ_{xx} vary linearly in x direction and quadratically in y and z direction. The strain ε_{yy} and stress σ_{yy} vary linearly in y direction and quadratically in x and z direction. The strain ε_{zz} and stress σ_{zz} vary linearly in z direction and quadratically in x and y direction. By default DIANA applies a $3 \times 3 \times 3$ integration scheme. A suitable option in a patch of more than one element is $2 \times 2 \times 2$ which yields optimal stress points. Schemes lower than $2 \times 2 \times 2$ or higher than $3 \times 3 \times 3$ are unsuitable.

7.2.4.9 CTE48 – tetrahedron, 3 sides, 16 nodes

The CTE48 element [Fig. 7.46] is a sixteen-node, three-side isoparametric solid tetrahe-

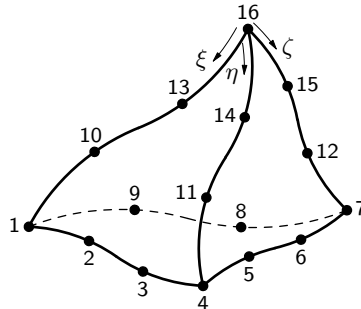


Figure 7.46: CTE48

dron element. It is based on third-order interpolation and numerical integration. The

polynomials for the translations u_{xyz} can be expressed as

$$\begin{aligned}
 u_i(\xi, \eta, \zeta) = & a_0 + a_1\xi + a_2\eta + a_3\zeta + a_4\xi\eta + a_5\eta\zeta + \\
 & a_6\xi\zeta + a_7\xi^2 + a_8\eta^2 + a_9\zeta^2 + a_{10}\xi^2\eta + \\
 & a_{11}\xi^2\zeta + a_{12}\xi\eta^2 + a_{13}\xi\zeta^2 + a_{14}\eta^2\zeta + a_{15}\eta\zeta^2
 \end{aligned} \quad (7.46)$$

Typically, these polynomials approximate the following strain and stress distribution. The strain ε_{xx} and stress σ_{xx} vary linearly in x direction and quadratically in y and z direction. The strain ε_{yy} and stress σ_{yy} vary linearly in y direction and quadratically in x and z direction. The strain ε_{zz} and stress σ_{zz} vary linearly in z direction and quadratically in x and y direction. By default DIANA applies a 64-point integration scheme over the volume. Other schemes are unsuitable. [$n_{lc} = 64$]

7.2.4.10 CTP72 – wedge, 24 nodes

The CTP72 element [Fig. 7.47] is a twenty-four-node isoparametric solid wedge element.

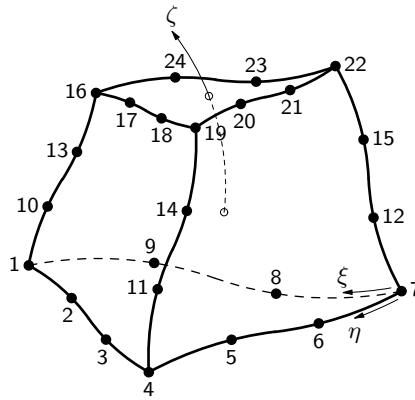


Figure 7.47: CTP72

It is based on third-order interpolation and numerical integration. The polynomials for the translations u_{xyz} can be expressed as

$$\begin{aligned}
 u_i(\xi, \eta, \zeta) = & a_0 + a_1\xi + a_2\eta + a_3\zeta + a_4\xi\eta + a_5\eta\zeta + \\
 & a_6\xi\zeta + a_7\xi^2 + a_8\eta^2 + a_9\zeta^2 + a_{10}\xi^2\eta + \\
 & a_{11}\xi^2\zeta + a_{12}\xi\eta^2 + a_{13}\xi\zeta^2 + a_{14}\eta^2\zeta + a_{15}\eta\zeta^2 + \\
 & a_{16}\xi\eta\zeta + a_{17}\xi^2\eta^2 + a_{18}\xi^2\eta\zeta + a_{19}\xi\eta^2\zeta + a_{20}\xi^2\eta^2\zeta + \\
 & a_{21}\xi\zeta^3 + a_{22}\eta\zeta^3 + a_{23}\zeta^3
 \end{aligned} \quad (7.47)$$

Typically, these polynomials approximate the following strain and stress distribution. The strain ε_{xx} and stress σ_{xx} vary linearly in x direction and quadratically in y and z direction. The strain ε_{yy} and stress σ_{yy} vary linearly in y direction and quadratically in x and z direction. The strain ε_{zz} and stress σ_{zz} vary linearly in z direction and quadratically in x and y direction. By default DIANA applies a 9-point integration scheme in the triangular domain and a 4-point scheme in the ζ direction. Options are 6- and 7-point in the triangular domain, and 3-point in the ζ direction. Other schemes are unsuitable. [$n_{lc} = 9$]
[$n_{\zeta} = 4$]

7.2.4.11 CHX96 – brick, 32 nodes

The CHX96 element [Fig. 7.48] is a thirty-two-node isoparametric solid brick element. It is based on third-order interpolation and Gauss integration. The polynomials for the

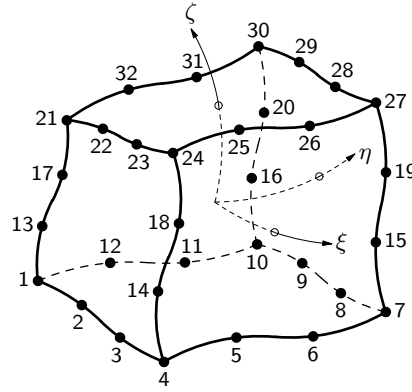


Figure 7.48: CHX96

translations u_{xyz} can be expressed as

$$\begin{aligned}
 u_i(\xi, \eta, \zeta) = & a_0 + a_1\xi + a_2\eta + a_3\zeta + a_4\xi\eta + a_5\eta\zeta + \\
 & a_6\xi\zeta + a_7\xi^2 + a_8\eta^2 + a_9\zeta^2 + a_{10}\xi^3 + \\
 & a_{11}\xi^2\eta + a_{12}\xi\eta^2 + a_{13}\eta^3 + a_{14}\eta^2\zeta + a_{15}\eta\zeta^2 + \\
 & a_{16}\zeta^3 + a_{17}\xi\zeta^2 + a_{18}\xi^2\zeta + a_{19}\xi\eta\zeta + a_{20}\xi^3\eta + \\
 & a_{21}\xi\eta^3 + a_{22}\eta^3\zeta + a_{23}\eta\zeta^3 + a_{24}\xi\zeta^3 + a_{25}\xi^3\zeta + \\
 & a_{26}\xi^2\eta\zeta + a_{27}\xi\eta^2\zeta + a_{28}\xi\eta\zeta^2 + a_{29}\xi^3\eta\zeta + a_{30}\xi\eta^3\zeta + \\
 & a_{31}\xi\eta\zeta^3
 \end{aligned} \tag{7.48}$$

Typically, a rectangular brick element approximates the following strain and stress distribution over the element volume. The strain ε_{xx} and stress σ_{xx} vary linearly in x direction and quadratically in y and z direction. The strain ε_{yy} and stress σ_{yy} vary linearly in y direction and quadratically in x and z direction. The strain ε_{zz} and stress σ_{zz} vary linearly in z direction and quadratically in x and y direction. By default DIANA applies a $4 \times 4 \times 4$ integration scheme. A suitable option is $3 \times 3 \times 3$ which yields optimal stress points and is sufficient. Schemes lower than $3 \times 3 \times 3$ or higher than $4 \times 4 \times 4$ are unsuitable.

$[n_\xi = 4, n_\eta = 4,$
 $n_\zeta = 4]$

7.2.5 Integration Schemes and Assumed Strain Options

For all three-dimensional body elements, DIANA performs a numerical integration with an appropriate default scheme. You may choose an alternative scheme via a special data input, depending on the shape of the element. To enhance the behaviour of linear hexahedrons in certain situations, assumed strain options may be applied. For more details see Volume *Element Library*.

7.2.6 Linear Elastic Material Properties

The material properties to be input for linear elasticity are:

- Young's modulus E , always necessary.
- Poisson's ratio ν , always necessary for three-dimensional elements.
- Shear modulus G , necessary for orthotropic elasticity.
- Thermal expansion coefficient α , necessary to determine a temperature load.
- Concentration expansion coefficient γ , necessary to determine a concentration load.

Table 7.2 on the facing page shows the required input for orthotropic elasticity for the three-dimensional body elements.

For some materials the Young's modulus E may depend on the position of the material in space. A typical example is soil where E may vary with the depth in the soil layer.

Table 7.2: LINEAR ORTHOTROPIC ELASTICITY

	solid
Young's modulus	E_x
	E_y
	E_z
Poisson's ratio	ν_{xy}
	ν_{yz}
	ν_{xz}
Shear modulus	G_{xy}
	G_{yz}
	G_{xz}
Thermal exp.	α_x
	α_y
	α_z
Concentr. exp.	γ_x
	γ_y
	γ_z

To model such a dependency, DIANA can apply position dependent characteristics on isotropic elasticity without temperature influence. In this case the user defines the reference Young's modulus E_{ref} , the reference position with the coordinates $(X_{\text{ref}}, Y_{\text{ref}}, Z_{\text{ref}})$, and the gradient of the Young's modulus in the global XYZ directions:

$$\begin{aligned}\partial E / \partial X &= grx \\ \partial E / \partial Y &= gry \\ \partial E / \partial Z &= grz\end{aligned}\tag{7.49}$$

DIANA will calculate the Young's modulus for each element integration point that is located below the reference position via linear interpolation:

$$E(X, Y, Z) = E_{\text{ref}} + (X - X_{\text{ref}}) \frac{\partial E}{\partial X} + (Y - Y_{\text{ref}}) \frac{\partial E}{\partial Y} + (Z - Z_{\text{ref}}) \frac{\partial E}{\partial Z}\tag{7.50}$$

where E_{ref} is the reference Young's modulus. For integration points located above the reference position the reference Young's modulus E_{ref} without gradients will be applied.

7.3 Plates and Shells

Plate or shell structures with a thickness that is relatively small compared to the in-plane dimensions and with out-of-plane loadings, may be defined with plate bending or shell elements. These elements may be located anywhere in three-dimensional space. Three different types of plate and shell elements can be distinguished:

- Plane bending elements

Plate bending elements must fulfill the following conditions with respect to shape and loading [Fig. 7.49]. They must be *plane*, i.e., the coordinates of the element

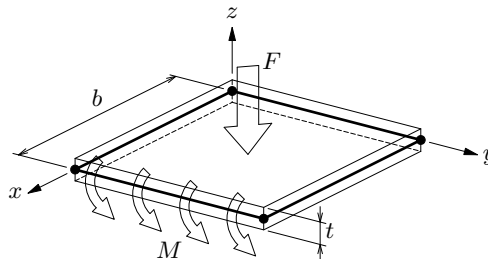


Figure 7.49: Plate bending elements, characteristics

nodes must be in one flat plane, the xy plane of the element. The thickness t must be small in relation to the dimensions b in the plane of the element. Force loading F must act perpendicular to the element plane, moment loading M must act around an axis which is in the element plane.

$$(\sigma_{zz} = 0)$$

Plate bending elements are characterized by the following facts. The direct stress component perpendicular to the face is zero, this means that the plane stress condition is fulfilled. The normals of the element plane remain straight after the deformation, but by definition, they do not have to be perpendicular to the element plane. The displacement perpendicular to the plane does not vary along the thickness.

DIANA offers two classes of plate bending elements: the first based on the Discrete Kirchhoff theory and therefore called *Discrete Kirchhoff* plate elements, the second based on a Mindlin–Reissner theory and simply called *Mindlin* plate elements. Both classes of plate bending elements are numerically integrated.

Typical applications of plate bending elements are the analysis of floors and other two-dimensional structural parts which are not subjected to in-plane forces.

Plate elements implemented according to the Discrete Kirchhoff theory, simply called ‘Kirchhoff plates’, are based on the principle that the condition of zero transverse shear strain is satisfied at some discrete points in the element. The displacement and rotation field is expanded by introducing some shear constraints. The effect of shear deformation is included which makes Kirchhoff plate elements suitable for application in both thin and thick plates.

In the Mindlin–Reissner plate theory the transverse displacements and rotations of the mid surface normals are independent and obtained by employing an isoparametric interpolation respectively from the translations and rotations in the nodes. This technique includes transverse shear deformation. Elements implemented according to this theory are simply called ‘Mindlin plate elements’.

In their standard form these elements are sensitive for shear locking which results in an excessively stiff behaviour. To overcome this difficulty for the linear and quadratic elements, DIANA modifies the transverse shear strain field.

- Flat shell elements

Flat shell elements basically are a combination of plane stress elements and plate bending elements. But unlike the plane stress elements, the basic variables are forces rather than Cauchy stresses. Flat shell elements must fulfill the following conditions with respect to shape and loading [Fig. 7.50]. They must be *plane*, i.e.,

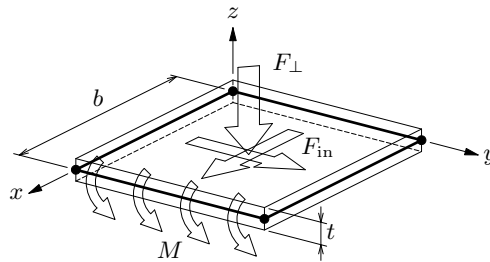


Figure 7.50: Flat shell elements, characteristics

the coordinates of the element nodes must be in one flat plane, the xy plane of the element, otherwise the curved shell elements must be used. They must be *thin*, i.e., the thickness t must be small in relation to the dimensions b in the plane of the element. Force loads F may act in any direction between perpendicular to the plane and in the plane. Moment loads M must act *in the plane* of the element.

Flat shell elements are characterized by the following facts. The normals of the element plane remain straight after the deformation. The displacement perpendicular to the plane does not vary in the direction of the thickness.

The flat shell elements in DIANA basically are combinations of a plane stress element and a plate bending element and there is no coupling between membrane and bending behaviour. Generally, the membrane behaviour is conform its corresponding plane stress element except the primary stresses which are defined in terms of moments and forces rather than Cauchy stresses. The bending behaviour is based on the Mindlin–Reissner theory and is conform its corresponding Mindlin plate bending element. For all flat shell elements the numerical integration is only performed in the reference surface.

DIANA offers two classes of flat shell elements: *regular* elements, elements with *drilling rotation*. The regular elements have three translations and two in-plane rotation degrees of freedom in each node. The elements with drilling rotation have an additional rotation ϕ_z normal to the plane in each node, the so-called ‘drilling rotation’. This drilling rotation may avoid an ill-condition of the global stiffness matrix in some cases.

Typical applications of flat shell elements are the analysis of tunnels and other two-and-a-half dimensional structural parts like box girders.

- Curved shell elements

The curved shell elements in DIANA are based on isoparametric degenerated-solid approach by introducing two shell hypotheses:

Straight-normals — assumes that normals remain straight, but not necessarily normal to the reference surface. Transverse shear deformation is included according to the *Mindlin–Reissner* theory.

Zero-normal-stress — assumes that the normal stress component in the normal direction of a lamina basis is forced to zero: $\sigma_{zzl(\xi,\eta,z)} = 0$. The element tangent plane is spanned by a lamina basis which corresponds to a local Cartesian coordinate system (x_l, y_l) defined at each point of the shell with x_l and y_l tangent to the ξ, η plane and z_l perpendicular to it.

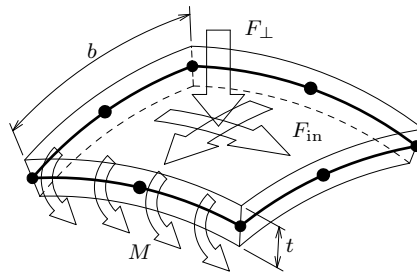


Figure 7.51: Curved shell elements, characteristics

The in-plane lamina strains ε_{xx} , ε_{yy} and γ_{xy} vary linearly in the thickness direction. The transverse shear strains γ_{xz} and γ_{yz} are forced to be constant in the thickness direction. Since the actual transverse shearing stresses and strains vary parabolically over the thickness, the shearing strains are an equivalent constant strain on a corresponding area. A shear correction factor is applied using the condition that a constant transverse shear stress yields approximately the same shear strain energy as the actual shearing stress.

Five degrees of freedom have been defined in each element node: three translations and two rotations. Further characteristics of curved shells are the following [Fig. 7.51]. They must be *thin*, i.e., the thickness t must be small in relation to the dimensions b in the plane of the element. Force loads F may act in any direction between perpendicular to the surface and in the surface. Moment loads M should act around an axis which is in the element face.

Also a set of curved shell elements with drilling rotation is available. These curved shell elements have six degrees of freedom in each element node: three translations and three rotations, because an additional rotation ϕ_z , the drilling rotation, has been

added to the basic variables of the regular curved shell elements. In applications where the elements are nearly co-planar in the nodes, the use of shell elements with drilling rotation is very attractive because they avoid an ill-condition of the assembled global stiffness matrix. Like for regular curved shell elements, the basic variables of regular curved shell elements with drilling rotation are the translations u and the rotations ϕ . The derived variables are the strains, the Cauchy stresses and the generalized moments and forces.

Typical applications of curved shell elements are the analysis of curved structures like shell roofs, storage tanks and ship or aircraft hulls.

7.3.1 Element Axes

For plate bending and shell elements DIANA needs no special user input data to set up the element axes. By default, a local element \bar{x} axis points from the first to the second node of the element [Fig. 7.52a]. DIANA uses the \bar{x} axis to set up xyz axes for rotations in the

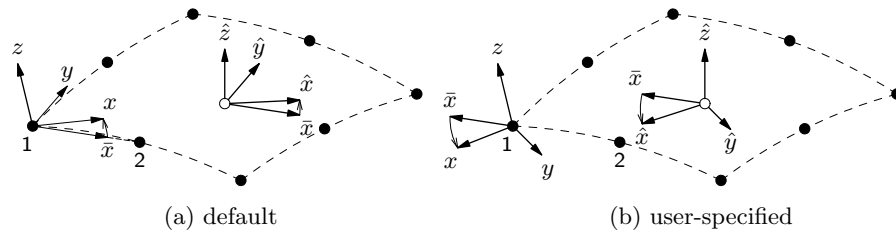


Figure 7.52: Axes

nodes and $\hat{x}\hat{y}\hat{z}$ for strains and stresses locally inside the element. The z axes are always set up perpendicular to the element plane, y is created $\perp z\bar{x}$ and finally $x \perp yz$ according to the right-hand-rule. Note that the default element axes are fully independent of the global model XYZ axes.

If you prefer x axes other than default, for instance to get the same axes directions for various elements, then you must specify an \bar{x} axis which DIANA uses to set up the real x and \hat{x} axes [Fig. 7.52b]. First the z axis is put perpendicular to the element plane. Then $y \perp z\bar{x}$ is created and finally $x \perp yz$.

7.3.2 Displacements and Rotations

The basic variables in the nodes of the curved shell elements are the translations u_X, u_Y

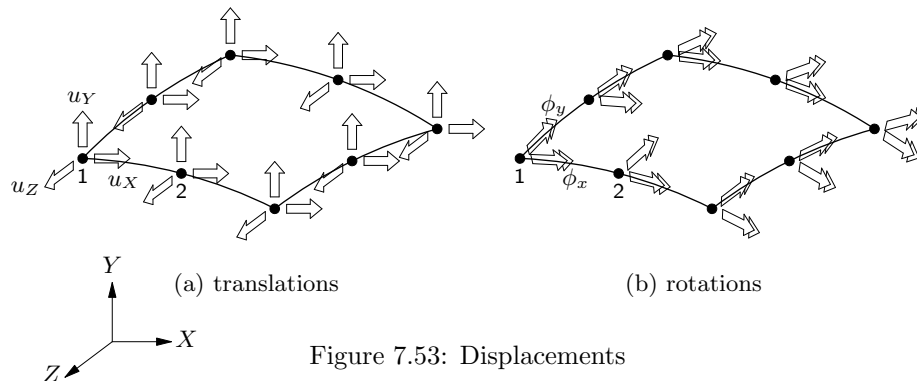


Figure 7.53: Displacements

and u_Z in the global XYZ directions [Fig. 7.53a] and the rotations ϕ_x and ϕ_y respectively around the local $+x$ and $+y$ axes in the tangent plane [Fig. 7.53b]. For plate bending elements the degrees of freedom are:

$$\mathbf{u}_e = \begin{Bmatrix} u_X \\ u_Y \\ u_Z \end{Bmatrix} \quad (7.51)$$

For regular flat and curved shell elements the degrees of freedom are:

$$\mathbf{u}_e = \begin{Bmatrix} u_X \\ u_Y \\ u_Z \\ \phi_x \\ \phi_y \end{Bmatrix} \quad (7.52)$$

For flat and curved shell elements with additional drilling degree of freedom, the degrees of freedom are:

$$\mathbf{u}_e = \begin{Bmatrix} u_X \\ u_Y \\ u_Z \\ \phi_x \\ \phi_y \\ \phi_z \end{Bmatrix} \quad (7.53)$$

7.3.3 Strains and Stresses

The displacements in the nodes yield the deformations du_X , du_Y and du_Z of an infinitesimal part dX dY [Fig. 7.54] and the deformations $d\phi_{\hat{x}}$, $d\phi_{\hat{y}}$ of an infinitesimal part $d\hat{x}$ $d\hat{y}$ [Fig. 7.55]. The sign convention for strains is that an elongation yields a positive strain

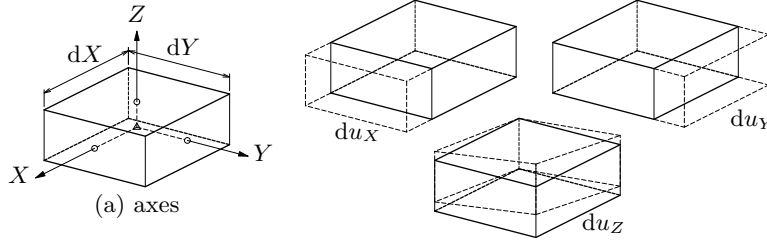


Figure 7.54: Translations

$d\hat{y}$ [Fig. 7.55]. The sign convention for strains is that an elongation yields a positive strain

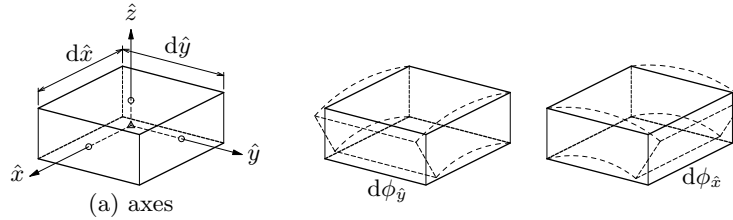


Figure 7.55: Rotations

and that a positive curvature has the convex side in $+z$ direction.

From these deformations, DIANA derives for plate bending elements the ‘generalized’ strains of Eq. (7.54).

$$\boldsymbol{\varepsilon} = \begin{Bmatrix} \kappa_{xx} \\ \kappa_{yy} \\ \kappa_{xy} \\ \Psi_{yz} \\ \Psi_{zx} \end{Bmatrix} \quad (7.54)$$

and for flat shell elements the Green–Lagrange and ‘generalized’ strains of Eq. (7.55)

$$\boldsymbol{\varepsilon} = \begin{Bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{zz} \\ \gamma_{xy} \\ \kappa_{xx} \\ \kappa_{yy} \\ \kappa_{xy} \\ \Psi_{yz} \\ \Psi_{zx} \end{Bmatrix} \quad (7.55)$$

with

$$\begin{aligned} \varepsilon_{xx} &= \frac{\partial u_x}{\partial x} & \varepsilon_{yy} &= \frac{\partial u_y}{\partial y} & \varepsilon_{zz} &= \frac{\nu(\varepsilon_{xx} + \varepsilon_{yy})}{1 - \nu} & \gamma_{xy} &= \frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x} \\ \kappa_{xx} &= \frac{\partial \phi_y}{\partial x} & \kappa_{yy} &= -\frac{\partial \phi_x}{\partial y} & \kappa_{xy} &= \frac{\partial \phi_y}{\partial y} - \frac{\partial \phi_x}{\partial x} \\ \Psi_{yz} &= \frac{\partial u_z}{\partial y} - \phi_x & \Psi_{zx} &= \frac{\partial u_z}{\partial x} + \phi_y \end{aligned} \quad (7.56)$$

The Green–Lagrange strains ε and γ are derived for the upper, mid and lower planes [Fig. 7.56 p. 50].

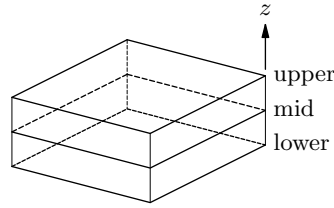


Figure 7.56: Planes for Green–Lagrange strains and Cauchy stresses

From these deformations, DIANA derives for curved shell elements the Green–Lagrange strains in the local $\hat{x}\hat{y}\hat{z}$ axes of Eq. (7.57).

$$\hat{\boldsymbol{\varepsilon}} = \begin{Bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{zz} \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{zx} \end{Bmatrix} = \mathbf{T}_\varepsilon \begin{Bmatrix} \varepsilon_{XX} \\ \varepsilon_{YY} \\ \varepsilon_{ZZ} \\ \gamma_{XY} \\ \gamma_{YZ} \\ \gamma_{ZX} \end{Bmatrix} \quad (7.57)$$

where \mathbf{T}_ε denotes the transformation matrix from global XYZ orientation to local $\hat{x}\hat{y}\hat{z}$ and

$$\begin{aligned} \varepsilon_{XX} &= \frac{\partial u_X}{\partial X} & \varepsilon_{YY} &= \frac{\partial u_Y}{\partial Y} & \varepsilon_{ZZ} &= \frac{\partial u_Z}{\partial Z} \\ \gamma_{XY} &= \frac{\partial u_X}{\partial Y} + \frac{\partial u_Y}{\partial X} & \gamma_{YZ} &= \frac{\partial u_Y}{\partial Z} + \frac{\partial u_Z}{\partial Y} & \gamma_{ZX} &= \frac{\partial u_Z}{\partial X} + \frac{\partial u_X}{\partial Z} \end{aligned} \quad (7.58)$$

These Green–Lagrange strains are derived for all integration points.

Cauchy stresses and generalized stresses such as distributed moments and distributed forces for plate bending and shell elements are defined as in Figure 7.57 and Figure 7.58.

For plate bending elements, DIANA derives from the basic strains of Eq. (7.54) the bending moments \mathbf{m} and forces \mathbf{q} of Eq. (7.59).

$$\mathbf{m} = \begin{Bmatrix} m_{xx} \\ m_{yy} \\ m_{xy} = m_{yx} \end{Bmatrix} \quad \mathbf{q} = \begin{Bmatrix} q_{xz} \\ q_{yz} \end{Bmatrix} \quad (7.59)$$

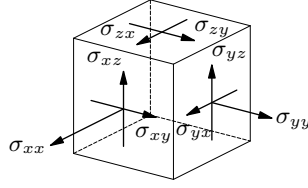


Figure 7.57: Cauchy stresses

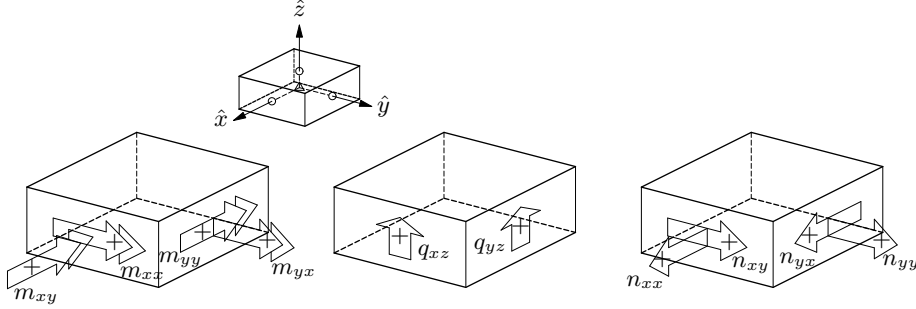


Figure 7.58: Generalized moments and forces

For flat shell elements, DIANA derives from the basic strains of Eq. (7.56) the bending moments \mathbf{m} and forces \mathbf{f} of Eq. (7.60).

$$\mathbf{m} = \begin{Bmatrix} m_{xx} \\ m_{yy} \\ m_{xy} = m_{yx} \end{Bmatrix} \quad \mathbf{f} = \begin{Bmatrix} n_{xx} \\ n_{yy} \\ n_{xy} = n_{yx} \\ q_{xz} \\ q_{yz} \end{Bmatrix} \quad (7.60)$$

From the moments and forces of Eq. (7.59) and Eq. (7.60), DIANA can derive the Cauchy stresses of Eq. (7.61).

$$\boldsymbol{\sigma} = \begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} = 0 \\ \sigma_{xy} = \sigma_{yx} \\ \sigma_{yz} = \sigma_{zy} \\ \sigma_{zx} = \sigma_{xz} \end{Bmatrix} \quad (7.61)$$

DIANA can derive these stresses for the upper, mid and lower planes respectively referring to the extreme $+z$, the $z = 0$ and the extreme $-z$ local coordinates [Fig. 7.56].

For plate bending elements and flat shell elements the element stiffness is defined in terms of cross-section properties such as forces and moments against generalized strains such as elongations and rotations. Therefore, these elements have only one plane of integration points.

However, for curved shell elements the stiffness is defined in terms of stress-strain relations in each integration point of the element. Curved shell elements have more than one planes of integration points. The Cauchy stresses for curved shell elements are similar to Cauchy stress for plate bending and flat shell elements as defined in Eq. (7.61). From the basic stresses of Eq. (7.61) DIANA can derive the bending moments \mathbf{m} and forces \mathbf{f} of Eq. (7.60) for curved shell elements.

7.3.4 Element Types

Plate bending and shell element types in DIANA can be triangles or quadrilaterals. The elements may have a linear, a quadratic or a cubic interpolation functions for the displacement field. For elements with linear interpolation functions there are only two nodes at each edge, whereas for elements with quadratic interpolation functions there are three nodes at each edge and for elements with cubic interpolation function the number of

nodes along an edge is equal to four. The next paragraphs [§ 7.3.4.1 to § 7.3.4.27] give an overview of the different element types where the numbers display the relative node numbers for the element connectivity. In DIANA the relative node numbers are always defined counterclockwise. The isoparametric coordinates ξ , η , and ζ are also displayed for the different element shapes.

For the flat and curved shell elements, versions with and without drilling rotations are available. For some curved shell elements also versions with layered properties are available. For the layered shell elements, the thickness is subdivided in a number of layers. Each layer has its own material properties and is numerically integrated separately. The variables of curved layered shell elements are the same as for the regular curved shell elements: translations u and the rotations ϕ , strains, Cauchy stresses and generalized moments and forces. Typical applications of layered shell elements are laminated structures like sandwich panels.

7.3.4.1 T9PLA – triangle, 3 nodes

The T9PLA element [Fig. 7.59] is a three-node triangular isoparametric plate bending

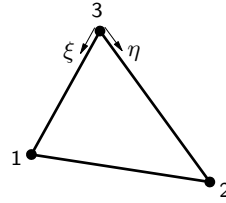


Figure 7.59: T9PLA

element according to the Discrete Kirchhoff theory, adapted to take shear deformation into account. It is based on linear geometry interpolation. The polynomial for the rotations ϕ_ξ and ϕ_η can be expressed as

$$\phi_i(\xi, \eta) = a_0 + a_1\xi + a_2\eta + a_3\xi\eta + a_4\xi^2 + a_5\eta^2 \quad (7.62)$$

Typically, this polynomial yields a curvature κ which varies linearly over the element area. The moments m also vary linearly over the element area. By default DIANA applies a 3-point integration scheme, 1-point is a suitable option. Schemes higher than 3-point are unsuitable.

[$n_{ic} = 3$]

7.3.4.2 Q12PL – quadrilateral, 4 nodes

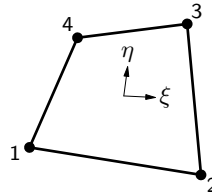


Figure 7.60: Q12PL

The Q12PL element [Fig. 7.60] is a four-node quadrilateral isoparametric plate bending element according to the Mindlin–Reissner theory. It is based on linear interpolation. The polynomials for the displacements u_z and rotations ϕ_x and ϕ_y are

$$\begin{aligned} u_z(\xi, \eta) &= a_0 + a_1\xi + a_2\eta + a_3\xi\eta \\ \phi_i(\xi, \eta) &= b_0 + b_1\xi + b_2\eta + b_3\xi\eta \end{aligned} \quad (7.63)$$

Typically, these polynomials yield the following strain and stress distribution over the element area. The curvature κ_{xx} , the moment m_{xx} and the shear force q_{xz} are constant in x direction and vary linearly in y direction. The curvature κ_{yy} , the moment m_{yy} and the shear force q_{yz} are constant in y direction and vary linearly in x direction. The only possible integration scheme is 2×2 , which DIANA applies by default.

[$n_\xi = 2, n_\eta = 2$]

7.3.4.3 CT18P – triangle, 6 nodes

The CT18P element [Fig. 7.61] is a six-node triangular isoparametric plate bending element

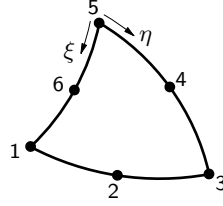


Figure 7.61: CT18P

according to the Mindlin–Reissner theory. It is based on quadratic interpolation and area integration. The polynomials for the displacements u_z and rotations ϕ_x and ϕ_y are

$$\begin{aligned} u_z(\xi, \eta) &= a_0 + a_1\xi + a_2\eta + a_3\xi\eta + a_4\xi^2 + a_5\eta^2 \\ \phi_i(\xi, \eta) &= b_0 + b_1\xi + b_2\eta + b_3\xi\eta + b_4\xi^2 + b_5\eta^2 \end{aligned} \quad (7.64)$$

Typically, these polynomials yield the following strain and stress distribution over the element area. The curvature κ_{xx} , the moment m_{xx} and the shear force q_{xz} vary linearly in x direction and quadratically in y direction. The curvature κ_{yy} , the moment m_{yy} and the shear force q_{yz} vary linearly in y direction and quadratically in x direction. By default DIANA applies a 3-point integration scheme, 1-point is a suitable option. Schemes higher than 3-point are unsuitable. [$n_c = 3$]

7.3.4.4 CQ24P – quadrilateral, 8 nodes

The CQ24P element [Fig. 7.62] is an eight-node quadrilateral isoparametric plate bending

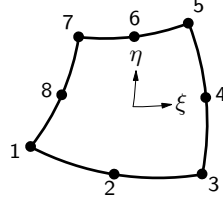


Figure 7.62: CQ24P

element according to the Mindlin–Reissner theory. It is based on quadratic interpolation. The polynomials for the displacements u_z and rotations ϕ_x and ϕ_y are

$$\begin{aligned} u_z(\xi, \eta) &= a_0 + a_1\xi + a_2\eta + a_3\xi\eta + a_4\xi^2 + a_5\eta^2 + a_6\xi^2\eta + a_7\xi\eta^2 \\ \phi_i(\xi, \eta) &= b_0 + b_1\xi + b_2\eta + b_3\xi\eta + b_4\xi^2 + b_5\eta^2 + b_6\xi^2\eta + b_7\xi\eta^2 \end{aligned} \quad (7.65)$$

Typically, a rectangular element yields the following strain and stress distribution over the element area. The curvature κ_{xx} , the moment m_{xx} and the shear force q_{xz} vary linearly in x direction and quadratically in y direction. The curvature κ_{yy} , the moment m_{yy} and the shear force q_{yz} vary linearly in y direction and quadratically in x direction. By default DIANA applies a 2×2 integration scheme which yields optimal stress points, 3×3 is a suitable option. Schemes other than 2×2 or 3×3 are unsuitable. [$n_\xi = 2, n_\eta = 2$]

7.3.4.5 T15SF – triangle, 3 nodes

The T15SF element [Fig. 7.63] is a three-node triangular isoparametric flat shell element. The plate bending is according to the Mindlin–Reissner theory with an adapted transverse shear interpolation. The membrane behaviour is conform the T6MEM plane stress element [§ 7.1.4.1 p. 25]. The geometry and the displacements are interpolated by bi-linear

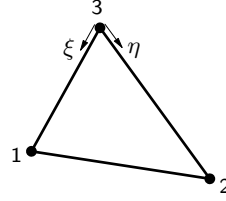


Figure 7.63: T15SF

functions. The integration perpendicular to the element face is direct. The polynomials for the translations u and the rotations ϕ can be expressed as

$$\begin{aligned} u_i(\xi, \eta) &= a_0 + a_1\xi + a_2\eta \\ \phi_i(\xi, \eta) &= b_0 + b_1\xi + b_2\eta \end{aligned} \quad (7.66)$$

These polynomials yield a constant strain and stress over the element area. By default [n_{ic} = 1] DIANA applies a 1-point integration scheme, 3-point is a suitable option. Schemes higher than 3-point are unsuitable.

7.3.4.6 Q20SF – quadrilateral, 4 nodes

The Q20SF element [Fig. 7.64] is a four-node quadrilateral isoparametric flat shell element.

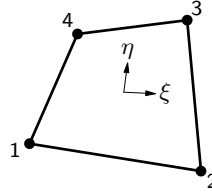


Figure 7.64: Q20SF

The plate bending is according to the Mindlin–Reissner theory with an adapted transverse shear interpolation and is conform the Q12PL plate bending element [§ 7.3.4.2 p. 52]. The membrane behaviour is conform the Q8MEM plane stress element [§ 7.1.4.2 p. 25]. The geometry and the displacements are interpolated by bi-linear functions. The integration perpendicular to the element face is direct. The polynomials for the translations u and the rotations ϕ can be expressed as

$$\begin{aligned} u_i(\xi, \eta) &= a_0 + a_1\xi + a_2\eta + a_3\xi\eta \\ \phi_i(\xi, \eta) &= b_0 + b_1\xi + b_2\eta + b_3\xi\eta \end{aligned} \quad (7.67)$$

Due to these polynomials, a rectangular element yields approximately the following strain and stress distribution along the element area. The strain ε_{xx} , the curvature κ_{xx} , the moment m_{xx} , the membrane force n_{xx} and the shear force q_{xz} are constant in x direction and vary linearly in y direction. The strain ε_{yy} , the curvature κ_{yy} , the moment m_{yy} , the membrane force n_{yy} and the shear force q_{yz} are constant in y direction and vary linearly in x direction. The only possible integration scheme is 2×2 which DIANA applies by default.

7.3.4.7 CT30F – triangle, 6 nodes

The CT30F element [Fig. 7.65] is a six-node triangular isoparametric flat shell element. The plate bending is according to the Mindlin–Reissner theory and behaves conform a standard six-node triangular plate bending element. The membrane behaviour is conform the CT12M plane stress element [§ 7.1.4.3 p. 26]. The geometry and the displacements are interpolated by quadratic functions. The integration perpendicular to the element face is direct. The polynomials for the translations u and the rotations ϕ can be expressed as

$$\begin{aligned} u_i(\xi, \eta) &= a_0 + a_1\xi + a_2\eta + a_3\xi\eta + a_4\xi^2 + a_5\eta^2 \\ \phi_i(\xi, \eta) &= b_0 + b_1\xi + b_2\eta + b_3\xi\eta + b_4\xi^2 + b_5\eta^2 \end{aligned} \quad (7.68)$$

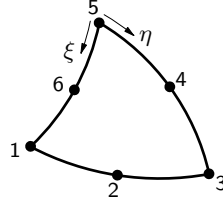


Figure 7.65: CT30F

Due to these polynomials the following strain and stress distribution can be approximated. The strain ε_{xx} , the curvature κ_{xx} , the membrane force n_{xx} , the moment m_{xx} , and the shear force q_{xz} vary linearly in x direction and quadratically in y direction. The strain ε_{yy} , the curvature κ_{yy} , the membrane force n_{yy} , the moment m_{yy} , and the shear force q_{yz} vary linearly in y direction and quadratically in z direction. The only possible integration scheme is 3-point which DIANA applies by default.

[$n_{lc} = 3$]

7.3.4.8 CQ40F – quadrilateral, 8 nodes

The CQ40F element [Fig. 7.66] is an eight-node quadrilateral isoparametric flat shell el-

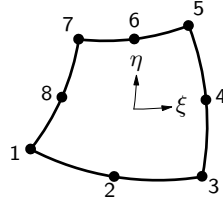


Figure 7.66: CQ40F

ement. The plate bending is according to the Mindlin–Reissner theory and behaves conform a standard eight-node quadrilateral plate bending element. The membrane behaviour is conform the CQ16M plane stress element [§ 7.1.4.4 p. 26]. The geometry and the displacements are interpolated by bi-quadratic functions. The integration perpendicular to the element face is direct. The polynomials for the translations u and the rotations ϕ can be expressed as

$$\begin{aligned} u_i(\xi, \eta) &= a_0 + a_1\xi + a_2\eta + a_3\xi\eta + a_4\xi^2 + a_5\eta^2 + a_6\xi^2\eta + a_7\xi\eta^2 \\ \phi_i(\xi, \eta) &= b_0 + b_1\xi + b_2\eta + b_3\xi\eta + b_4\xi^2 + b_5\eta^2 + b_6\xi^2\eta + b_7\xi\eta^2 \end{aligned} \quad (7.69)$$

Due to these polynomials, a rectangular element yields approximately the following strain and stress distribution along the element area. The strain ε_{xx} , the curvature κ_{xx} , the moment m_{xx} , the membrane force n_{xx} and the shear force q_{xz} vary linearly in x direction and quadratically in y direction. The strain ε_{yy} , the curvature κ_{yy} , the moment m_{yy} , the membrane force n_{yy} and the shear force q_{yz} vary linearly in y direction and quadratically in x direction. The only possible integration scheme is 2×2 which DIANA applies by default.

[$n_\xi = 2, n_\eta = 2$]

7.3.4.9 T18SF – triangle, 3 nodes

The T18SF element [Fig. 7.67] is a three-node triangular isoparametric flat shell element

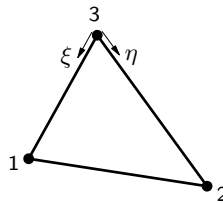


Figure 7.67: T18SF

with drilling rotation. In each node there are three translational degrees of freedom defined, u_x , u_y , and u_z , two rotational degrees of freedom around the axes in the element plane ϕ_x and ϕ_y , and one rotational degree of freedom around the axis normal to the element plane, ϕ_z . The plate bending is based on a Discrete Kirchhoff theory and is conform the T9PLA element [§ 7.3.4.1 p. 52]. The membrane behaviour is identical to the T9MEM plane stress element. The geometry is interpolated by a linear function and the displacements by linear and hierarchical quadratic functions. The polynomials for the translations $u_{i=x,y}$ and the rotations $\phi_{i=x,y}$ can be expressed as

$$\begin{aligned} u_i(\xi, \eta) &= a_0 + a_1\xi + a_2\eta + a_3\xi\eta + a_4\xi^2 + a_5\eta^2 \\ \phi_i(\xi, \eta) &= b_0 + b_1\xi + b_2\eta + b_3\xi\eta + b_4\xi^2 + b_5\eta^2 \end{aligned} \quad (7.70)$$

These polynomials yield a strain and stress distribution which is constant over the element area. By default DIANA applies a 3-point integration scheme, 1-point is a suitable option. Schemes higher than 3-point are unsuitable.

7.3.4.10 T18FSH – triangle, 3 nodes, analytically integrated

The T18FSH element [Fig. 7.68] is a three-node triangular analytically integrated flat shell

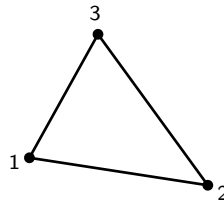


Figure 7.68: T18FSH

element with drilling rotation. In each node there are three translational degrees of freedom defined, u_x , u_y , and u_z , two rotational degrees of freedom around the axes in the element plane ϕ_x and ϕ_y , and one rotational degree of freedom around the axis normal to the element plane, ϕ_z . The plate bending is based on Kirchhoff theory and based on the formulation by Allman [3]. The membrane behaviour is also based on Allman's formulation [4]. The element is particularly suitable for postbuckling and nonlinear vibration analysis based on perturbation theory. This enhancement is based on Tiso's work [77]. Additionally, the element can consider geometric nonlinear behaviour. Because the element is analytically integrated, element output is only available at the element center and nodes.

7.3.4.11 Q24SF – quadrilateral, 4 nodes

The Q24SF element [Fig. 7.69] is a four-node quadrilateral isoparametric flat shell element

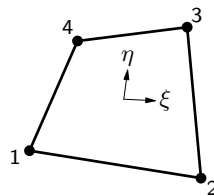


Figure 7.69: Q24SF

with drilling rotation. In each node there are three translational degrees of freedom defined, u_x , u_y , and u_z , two rotational degrees of freedom around the axes in the element plane ϕ_x and ϕ_y , and one rotational degree of freedom around the axis normal to the element plane, ϕ_z . The plate bending is according to the Mindlin–Reissner theory with and adapted transverse shear interpolation conform the Q12PL plate bending element [§ 7.3.4.2 p. 52]. The membrane behaviour is conform the Q12ME plane stress element

[§ 7.1.4.7 p. 27]. The geometry is interpolated by a linear function, the in-plane displacements $u_{x,y}$ by bi-linear and hierarchical bi-quadratic functions, the normal displacement u_z and the drilling rotation ϕ_z by bi-linear functions.

$$\begin{aligned}
 u_x(\xi, \eta) &= a_0 + a_1\xi + a_2\eta + a_3\xi\eta + a_4\xi^2 + a_5\xi^2\eta \\
 u_y(\xi, \eta) &= a_0 + a_1\xi + a_2\eta + a_3\xi\eta + a_4\eta^2 + a_5\xi\eta^2 \\
 u_z(\xi, \eta) &= b_0 + b_1\xi + b_2\eta + b_3\xi\eta \\
 \phi_i(\xi, \eta) &= b_0 + b_1\xi + b_2\eta + b_3\xi\eta
 \end{aligned}
 \tag{7.71}$$

Due to these polynomials, a rectangular element yields approximately the following strain and stress distribution along the element area. The strain ε_{xx} , the curvature κ_{xx} , the moment m_{xx} , the membrane force n_{xx} and the shear force q_{xz} are constant in x direction and vary linearly in y direction. The strain ε_{yy} , the curvature κ_{yy} , the moment m_{yy} , the membrane force n_{yy} and the shear force q_{yz} are constant in y direction and vary linearly in x direction. The only possible integration scheme is 2×2 which DIANA applies by default.

$$[n_\xi = 2, n_\eta = 2]$$

7.3.4.12 CT36F – triangle, 6 nodes

The CT36F element [Fig. 7.70] is a six-node triangular isoparametric flat shell element with

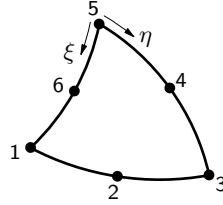


Figure 7.70: CT36F

drilling rotation. In each node there are three translational degrees of freedom defined, u_x , u_y , and u_z , two rotational degrees of freedom around the axes in the element plane ϕ_x and ϕ_y , and one rotational degree of freedom around the axis normal to the element plane, ϕ_z . The plate bending is according to the Mindlin–Reissner theory and is conform the CT18P plate bending element [§ 7.3.4.3 p. 53]. The membrane behaviour is conform the CT12M plane stress element [§ 7.1.4.3 p. 26]. The geometry and the displacements are interpolated by a quadratic functions. The integration perpendicular to the element face is direct. The polynomials for the translations u and the rotations ϕ can be expressed as

$$\begin{aligned}
 u_i(\xi, \eta) &= a_0 + a_1\xi + a_2\eta + a_3\xi\eta + a_4\xi^2 + a_5\eta^2 \\
 \phi_i(\xi, \eta) &= b_0 + b_1\xi + b_2\eta + b_3\xi\eta + b_4\xi^2 + b_5\eta^2
 \end{aligned}
 \tag{7.72}$$

Due to these polynomials, the following strain and stress distribution along the element area can be approximated. The strain ε_{xx} , the curvature κ_{xx} , the moment m_{xx} , the membrane force n_{xx} and the shear force q_{xz} vary linearly in x direction and vary quadratically in y direction. The strain ε_{yy} , the curvature κ_{yy} , the moment m_{yy} , the membrane force n_{yy} and the shear force q_{yz} vary linearly in y direction and vary quadratically in x direction. The stiffness for the drilling rotation ϕ_z is fictitious. The only possible integration scheme is 3-point which DIANA applies by default.

$$[n_{lc} = 3]$$

7.3.4.13 CQ48F – quadrilateral, 8 nodes

The CQ48F element [Fig. 7.71] is an eight-node quadrilateral isoparametric flat shell element with drilling rotation. In each node there are three translational degrees of freedom defined, u_x , u_y , and u_z , two rotational degrees of freedom around the axes in the element plane ϕ_x and ϕ_y , and one rotational degree of freedom around the axis normal to the element plane, ϕ_z . The plate bending is according to the Mindlin–Reissner theory and is conform the CQ24P plate bending element [§ 7.3.4.4 p. 53]. The membrane behaviour is conform the CQ16M plane stress element [§ 7.1.4.4 p. 26]. The element is based on a

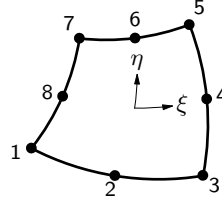


Figure 7.71: CQ48F

quadratic interpolation and only a numerical integration over the element area. The polynomials for the translations u and the rotations ϕ can be expressed as

$$\begin{aligned} u_i(\xi, \eta) &= a_0 + a_1\xi + a_2\eta + a_3\xi\eta + a_4\xi^2 + a_5\eta^2 + a_6\xi^2\eta + a_7\xi\eta^2 \\ \phi_i(\xi, \eta) &= b_0 + b_1\xi + b_2\eta + b_3\xi\eta + b_4\xi^2 + b_5\eta^2 + b_6\xi^2\eta + b_7\xi\eta^2 \end{aligned} \quad (7.73)$$

Due to these polynomials, a rectangular element yields approximately the following strain and stress distribution along the element area. The strain ε_{xx} , the curvature κ_{xx} , the moment m_{xx} , the membrane force n_{xx} and the shear force q_{xz} vary linearly in x direction and quadratically in y direction. The strain ε_{yy} , the curvature κ_{yy} , the moment m_{yy} , the membrane force n_{yy} and the shear force q_{yz} vary linearly in y direction and quadratically in x direction. The stiffness for the drilling rotation ϕ_z is fictitious. The only possible integration scheme is 2×2 which DIANA applies by default.

$[n_\xi = 2, n_\eta = 2]$

7.3.4.14 T15SH – triangular, 3 nodes

The T15SH element [Fig. 7.72] is a three-node triangular isoparametric curved shell el-

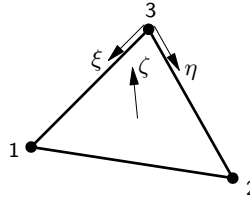


Figure 7.72: T15SH

ement. It is based on linear interpolation and area integration. The integration in ζ direction (thickness) may be Gauss or Simpson. The polynomials for the translations u and the rotations ϕ can be expressed as

$$\begin{aligned} u_i(\xi, \eta) &= a_0 + a_1\xi + a_2\eta \\ \phi_i(\xi, \eta) &= b_0 + b_1\xi + b_2\eta \end{aligned} \quad (7.74)$$

Typically, these polynomials yield approximately the following strain and stress distribution along the element area in a ζ lamina. The strain ε_{xx} , the curvature κ_{xx} , the moment m_{xx} , the membrane force n_{xx} and the shear force q_{xz} are constant in x direction and vary linearly in y direction. The strain ε_{yy} , the curvature κ_{yy} , the moment m_{yy} , the membrane force n_{yy} and the shear force q_{yz} are constant in y direction and vary linearly in x direction. The default integration scheme over the element area is a 3-point scheme.

$[n_{1c} = 3]$ Alternatively, it is possible to use the quadrilateral integration scheme. In this case T15SH behaves as a Q20SH element of which node 1 and node 4 coincide. The default in ζ direction (thickness) is 3-point Simpson, 2-point Gauss is a suitable option. Schemes higher than 3-point in ζ direction are only useful in case of nonlinear analysis.

$[n_\zeta = 3]$

7.3.4.15 Q20SH – quadrilateral, 4 nodes

The Q20SH element [Fig. 7.73] is a four-node quadrilateral isoparametric curved shell element. It is based on linear interpolation and Gauss integration over the $\xi\eta$ element area. The integration in ζ direction (thickness) may be Gauss or Simpson. To avoid shear locking, which results in a excessively stiff behaviour, DIANA automatically modifies the

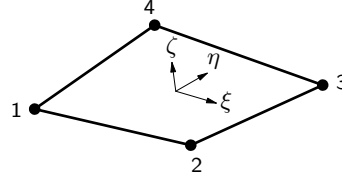


Figure 7.73: Q20SH

transverse shear strain fields. The polynomials for the translations u and the rotations ϕ can be expressed as

$$\begin{aligned} u_i(\xi, \eta) &= a_0 + a_1\xi + a_2\eta + a_3\xi\eta \\ \phi_i(\xi, \eta) &= b_0 + b_1\xi + b_2\eta + b_3\xi\eta \end{aligned} \quad (7.75)$$

Typically, for a rectangular element, these polynomials yield approximately the following strain and stress distribution along the element area in a ζ lamina. The strain ε_{xx} , the curvature κ_{xx} , the moment m_{xx} , the membrane force n_{xx} and the shear force q_{xz} are constant in x direction and vary linearly in y direction. The strain ε_{yy} , the curvature κ_{yy} , the moment m_{yy} , the membrane force n_{yy} and the shear force q_{yz} are constant in y direction and vary linearly in x direction. The only possible (and default) integration scheme over the element area is 2×2 . The default in ζ direction (thickness) is 3-point Simpson, 2-point Gauss is a suitable option. Schemes higher than 3-point in ζ direction are only useful in case of nonlinear analysis.

$[n_\xi = 2, n_\eta = 2]$
 $[n_\zeta = 3]$

7.3.4.16 CT30S – triangle, 6 nodes

The CT30S element [Fig. 7.74] is a six-node triangular isoparametric curved shell element.

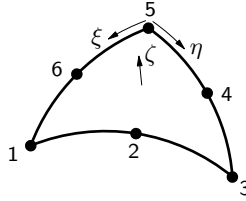


Figure 7.74: CT30S

It is based on quadratic interpolation and area integration. The integration in ζ direction (thickness) may be Gauss or Simpson. The polynomials for the translations u and the rotations ϕ can be expressed as

$$\begin{aligned} u_i(\xi, \eta) &= a_0 + a_1\xi + a_2\eta + a_3\xi\eta + a_4\xi^2 + a_5\eta^2 \\ \phi_i(\xi, \eta) &= b_0 + b_1\xi + b_2\eta + b_3\xi\eta + b_4\xi^2 + b_5\eta^2 \end{aligned} \quad (7.76)$$

Typically, these polynomials yield approximately the following strain and stress distribution along the element area in a ζ lamina. The strain ε_{xx} , the curvature κ_{xx} , the moment m_{xx} , the membrane force n_{xx} and the shear force q_{xz} vary linearly in x direction and quadratically in y direction. The strain ε_{yy} , the curvature κ_{yy} , the moment m_{yy} , the membrane force n_{yy} and the shear force q_{yz} vary linearly in y direction and quadratically in x direction. To avoid membrane and shear locking, a reduced 3-point integration scheme over the area should be used, which DIANA applies by default. The default in ζ direction (thickness) is 3-point Simpson, 2-point Gauss is a suitable option. Schemes higher than 3-point in ζ direction are only useful in case of nonlinear analysis.

$[n_c = 3]$

$[n_\zeta = 3]$

7.3.4.17 CQ40S – quadrilateral, 8 nodes

The CQ40S element [Fig. 7.75] is an eight-node quadrilateral isoparametric curved shell element. It is based on quadratic interpolation and Gauss integration over the $\xi\eta$ element area. The integration in ζ direction (thickness) may be Gauss or Simpson. The

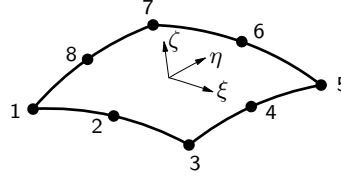


Figure 7.75: CQ40S

polynomials for the translations u and the rotations ϕ can be expressed as

$$\begin{aligned} u_i(\xi, \eta) &= a_0 + a_1\xi + a_2\eta + a_3\xi\eta + a_4\xi^2 + a_5\eta^2 + a_6\xi^2\eta + a_7\xi\eta^2 \\ \phi_i(\xi, \eta) &= b_0 + b_1\xi + b_2\eta + b_3\xi\eta + b_4\xi^2 + b_5\eta^2 + b_6\xi^2\eta + b_7\xi\eta^2 \end{aligned} \quad (7.77)$$

Typically, for a rectangular element, these polynomials yield approximately the following strain and stress distribution along the element area in a ζ lamina. The strain ε_{xx} , the curvature κ_{xx} , the moment m_{xx} , the membrane force n_{xx} and the shear force q_{xz} vary linearly in x direction and quadratically in y direction. The strain ε_{yy} , the curvature κ_{yy} , the moment m_{yy} , the membrane force n_{yy} and the shear force q_{yz} vary linearly in y direction and quadratically in x direction. To avoid membrane and shear locking, the only (and default) is a reduced 2×2 integration scheme over the area. The default in ζ direction (thickness) is 3-point Simpson, 2-point Gauss is a suitable option. Schemes higher than 3-point in ζ direction are only useful in case of nonlinear analysis.

$[n_\xi = 2, n_\eta = 2]$

$[n_\zeta = 3]$

7.3.4.18 CT45S – triangle, 9 nodes

The CT45S element [Fig. 7.76] is a six-node triangular isoparametric curved shell element.

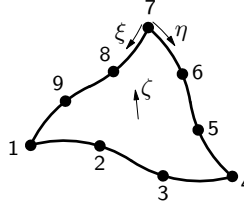


Figure 7.76: CT45S

It is based on third-order interpolation and area integration. The integration in ζ direction (thickness) may be Gauss or Simpson. The polynomials for the translations u and the rotations ϕ can be expressed as

$$\begin{aligned} u_i(\xi, \eta) &= a_0 + a_1\xi + a_2\eta + a_3\xi\eta + a_4\xi^2 + a_5\eta^2 \dots \\ \phi_i(\xi, \eta) &= b_0 + a_1\xi + b_2\eta + b_3\xi\eta + b_4\xi^2 + b_5\eta^2 \dots \end{aligned} \quad (7.78)$$

Typically, these polynomials yield approximately the following strain and stress distribution along the element area in a ζ lamina. The strain ε_{xx} , the curvature κ_{xx} , the moment m_{xx} , the membrane force n_{xx} and the shear force q_{xz} vary quadratically in x direction and third-order in y direction. The strain ε_{yy} , the curvature κ_{yy} , the moment m_{yy} , the membrane force n_{yy} and the shear force q_{yz} vary quadratically in y direction and third-order in x direction. By default DIANA applies a 7-point integration scheme over the element area, 4- and 6-point are suitable options. Schemes higher than 7-point are unsuitable. The default in ζ direction (thickness) is 3-point Simpson, 2-point Gauss is a suitable option. Schemes higher than 3-point in ζ direction are only useful in case of nonlinear analysis.

$[n_{lc} = 7]$

$[n_\zeta = 3]$

7.3.4.19 CQ60S – quadrilateral, 12 nodes

The CQ60S element [Fig. 7.77] is an twelve-node quadrilateral isoparametric curved shell

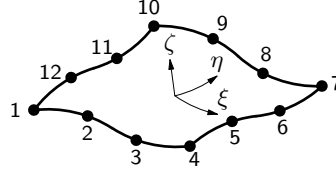


Figure 7.77: CQ60S

element. It is based on third-order interpolation and Gauss integration over the $\xi\eta$ element area. The integration in ζ direction (thickness) may be Gauss or Simpson. The polynomials for the translations u and the rotations ϕ can be expressed as

$$\begin{aligned} u_i(\xi, \eta) &= a_0 + a_1\xi + a_2\eta + a_3\xi\eta + a_4\xi^2 + a_5\eta^2 + a_6\xi^2\eta + a_7\xi\eta^2 \dots \\ \phi_i(\xi, \eta) &= b_0 + b_1\xi + b_2\eta + b_3\xi\eta + b_4\xi^2 + b_5\eta^2 + b_6\xi^2\eta + b_7\xi\eta^2 \dots \end{aligned} \quad (7.79)$$

Typically, for a rectangular element, these polynomials yield approximately the following strain and stress distribution along the element area in a ζ lamina. The strain ε_{xx} , the curvature κ_{xx} , the moment m_{xx} , the membrane force n_{xx} and the shear force q_{xz} vary quadratically in x direction and linearly in y direction. The strain ε_{yy} , the curvature κ_{yy} , the moment m_{yy} , the membrane force n_{yy} and the shear force q_{yz} vary quadratically in y direction and linearly in x direction. By default DIANA applies a 3×3 integration scheme over the element area, 2×2 is a suitable option. The default in ζ direction (thickness) is 3-point Simpson, 2-point Gauss is a suitable option. Schemes higher than 3-point in ζ direction are only useful in case of nonlinear analysis.

[$n_\xi = 3$, $n_\eta = 3$][$n_\zeta = 3$]

7.3.4.20 T18SH – triangular, 3 nodes

The T18SH element [Fig. 7.78] is a three-node triangular isoparametric curved shell ele-

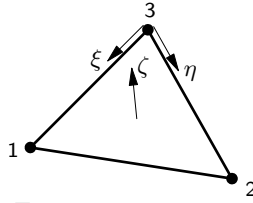


Figure 7.78: T18SH

ment with drilling rotation. In each node there are three translational degrees of freedom defined, u_x , u_y , and u_z , two rotational degrees of freedom around the axes in the element plane ϕ_x and ϕ_y , and one rotational degree of freedom around the axis normal to the element plane, ϕ_z . It is based on linear interpolation and area integration. The integration in ζ direction (thickness) may be Gauss or Simpson. The polynomials for the translations u and the rotations ϕ can be expressed as

$$\begin{aligned} u_i(\xi, \eta) &= a_0 + a_1\xi + a_2\eta \\ \phi_i(\xi, \eta) &= b_0 + b_1\xi + b_2\eta \end{aligned} \quad (7.80)$$

Typically, these polynomials yield approximately the following strain and stress distribution along the element area in a ζ lamina. The strain ε_{xx} , the curvature κ_{xx} , the moment m_{xx} , the membrane force n_{xx} and the shear force q_{xz} are constant in x direction and vary linearly in y direction. The strain ε_{yy} , the curvature κ_{yy} , the moment m_{yy} , the membrane force n_{yy} and the shear force q_{yz} are constant in y direction and vary linearly in x direction. The default integration scheme over the element area is a 3-point scheme.

Alternatively, it is possible to use the quadrilateral integration scheme. In this case T18SH behaves as a Q24SH element of which node 1 and node 4 coincide. The default in ζ direction (thickness) is 3-point Simpson, 2-point Gauss is a suitable option. Schemes higher than 3-point in ζ direction are only useful in case of nonlinear analysis.

[$n_{lc} = 3$][$n_\zeta = 3$]

7.3.4.21 Q24SH – quadrilateral, 4 nodes

The Q24SH element [Fig. 7.79] is a four-node quadrilateral isoparametric curved shell

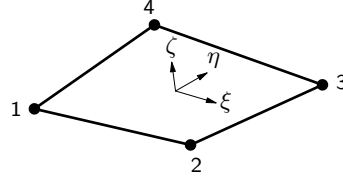


Figure 7.79: Q24SH

element with drilling rotation. In each node there are three translational degrees of freedom defined, u_x , u_y , and u_z , two rotational degrees of freedom around the axes in the element plane ϕ_x and ϕ_y , and one rotational degree of freedom around the axis normal to the element plane, ϕ_z . It is based on linear interpolation and Gauss integration over the $\xi\eta$ element area. The integration in ζ direction (thickness) may be Gauss or Simpson. To avoid shear locking, which results in an excessively stiff behaviour, DIANA automatically modifies the transverse shear strain fields. The polynomials for the translations u and the rotations ϕ can be expressed as

$$\begin{aligned} u_i(\xi, \eta) &= a_0 + a_1\xi + a_2\eta + a_3\xi\eta \\ \phi_i(\xi, \eta) &= b_0 + b_1\xi + b_2\eta + b_3\xi\eta \end{aligned} \quad (7.81)$$

Typically, for a rectangular element, these polynomials yield approximately the following strain and stress distribution along the element area in a ζ lamina. The strain ε_{xx} , the curvature κ_{xx} , the moment m_{xx} , the membrane force n_{xx} and the shear force q_{xz} are constant in x direction and vary linearly in y direction. The strain ε_{yy} , the curvature κ_{yy} , the moment m_{yy} , the membrane force n_{yy} and the shear force q_{yz} are constant in y direction and vary linearly in x direction. The only possible (and default) integration scheme over the element area is 2×2 . The default in ζ direction (thickness) is 3-point Simpson, 2-point Gauss is a suitable option. Schemes higher than 3-point in ζ direction are only useful in case of nonlinear analysis.

$[n_\xi = 2, n_\eta = 2]$
 $[n_\zeta = 3]$

7.3.4.22 CT36S – triangle, 6 nodes

The CT36S element [Fig. 7.80] is a six-node triangular isoparametric curved shell element

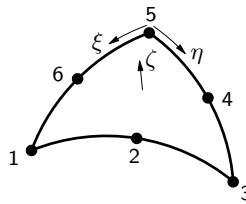


Figure 7.80: CT36S

with drilling rotation. In each node there are three translational degrees of freedom defined, u_x , u_y , and u_z , two rotational degrees of freedom around the axes in the element plane ϕ_x and ϕ_y , and one rotational degree of freedom around the axis normal to the element plane, ϕ_z . It is based on quadratic interpolation and area integration. The integration in ζ direction (thickness) may be Gauss or Simpson. The polynomials for the translations u and the rotations ϕ can be expressed as

$$\begin{aligned} u_i(\xi, \eta) &= a_0 + a_1\xi + a_2\eta + a_3\xi\eta + a_4\xi^2 + a_5\eta^2 \\ \phi_i(\xi, \eta) &= b_0 + b_1\xi + b_2\eta + b_3\xi\eta + b_4\xi^2 + b_5\eta^2 \end{aligned} \quad (7.82)$$

Typically, these polynomials yield approximately the following strain and stress distribution along the element area in a ζ lamina. The strain ε_{xx} , the curvature κ_{xx} , the moment m_{xx} , the membrane force n_{xx} and the shear force q_{xz} vary linearly in x direction and

quadratically in y direction. The strain ε_{yy} , the curvature κ_{yy} , the moment m_{yy} , the membrane force n_{yy} and the shear force q_{yz} vary linearly in y direction and quadratically in x direction. To avoid membrane and shear locking, a reduced 3-point integration scheme over the area should be used, which DIANA applies by default. The default in ζ direction (thickness) is 3-point Simpson, 2-point Gauss is a suitable option. Schemes higher than 3-point in ζ direction are only useful in case of nonlinear analysis.

7.3.4.23 CQ48S – quadrilateral, 8 nodes

The CQ48S element [Fig. 7.81] is an eight-node quadrilateral isoparametric curved shell

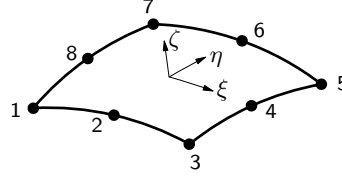


Figure 7.81: CQ48S

element with drilling rotation. In each node there are three translational degrees of freedom defined, u_x , u_y , and u_z , two rotational degrees of freedom around the axes in the element plane ϕ_x and ϕ_y , and one rotational degree of freedom around the axis normal to the element plane, ϕ_z . It is based on quadratic interpolation and Gauss integration over the $\xi\eta$ element area. The integration in ζ direction (thickness) may be Gauss or Simpson. The polynomials for the translations u and the rotations ϕ can be expressed as

$$\begin{aligned} u_i(\xi, \eta) &= a_0 + a_1\xi + a_2\eta + a_3\xi\eta + a_4\xi^2 + a_5\eta^2 + a_6\xi^2\eta + a_7\xi\eta^2 \\ \phi_i(\xi, \eta) &= b_0 + b_1\xi + b_2\eta + b_3\xi\eta + b_4\xi^2 + b_5\eta^2 + b_6\xi^2\eta + b_7\xi\eta^2 \end{aligned} \quad (7.83)$$

Typically, for a rectangular element, these polynomials yield approximately the following strain and stress distribution along the element area in a ζ lamina. The strain ε_{xx} , the curvature κ_{xx} , the moment m_{xx} , the membrane force n_{xx} and the shear force q_{xz} vary linearly in x direction and quadratically in y direction. The strain ε_{yy} , the curvature κ_{yy} , the moment m_{yy} , the membrane force n_{yy} and the shear force q_{yz} vary linearly in y direction and quadratically in x direction. To avoid membrane and shear locking, the only (and default) is a reduced 2×2 integration scheme over the area. The default in ζ direction (thickness) is 3-point Simpson, 2-point Gauss is a suitable option. Schemes higher than 3-point in ζ direction are only useful in case of nonlinear analysis.

7.3.4.24 CT30L – triangle, 6 nodes

The CT30L element [Fig. 7.82] is a six-node triangular isoparametric curved layered shell

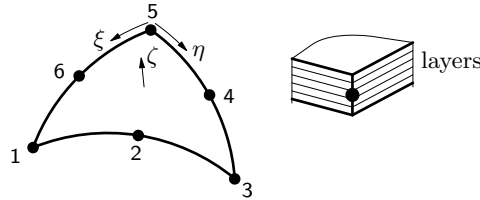


Figure 7.82: CT30L

element. It is based on quadratic interpolation and area integration per layer. The integration in ζ direction (layer thickness) may be Gauss or Simpson. The polynomials for the translations u and the rotations ϕ can be expressed as

$$\begin{aligned} u_i(\xi, \eta) &= a_0 + a_1\xi + a_2\eta + a_3\xi\eta + a_4\xi^2 + a_5\eta^2 \\ \phi_i(\xi, \eta) &= b_0 + b_1\xi + b_2\eta + b_3\xi\eta + b_4\xi^2 + b_5\eta^2 \end{aligned} \quad (7.84)$$

Typically, these polynomials yield approximately the following strain and stress distribution along the element area in a ζ lamina of layer. The strain ε_{xx} , the curvature κ_{xx} , the

moment m_{xx} , the membrane force n_{xx} and the shear force q_{xz} vary linearly in x direction and quadratically in y direction. The strain ε_{yy} , the curvature κ_{yy} , the moment m_{yy} , the membrane force n_{yy} and the shear force q_{yz} vary linearly in y direction and quadratically in x direction. To avoid membrane and shear locking, the only (and default) is a reduced 3-point integration scheme over the area of a layer. The default in ζ direction (layer thickness) is 3-point, 2-point is a suitable option. Schemes higher than 3-point in ζ direction are only useful in case of nonlinear analysis.

$[n_{lc} = 3]$
 $[n_{\zeta} = 3]$

7.3.4.25 CQ40L – quadrilateral, 8 nodes

The CQ40L element [Fig. 7.83] is an eight-node quadrilateral isoparametric curved layered

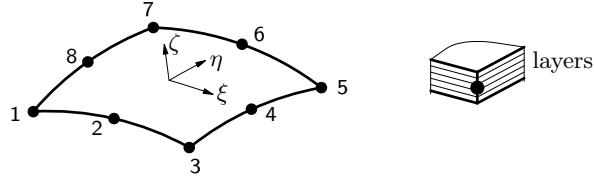


Figure 7.83: CQ40L

shell element. It is based on quadratic interpolation and Gauss integration over the $\xi\eta$ element area. The integration in ζ direction (layer thickness) may be Gauss or Simpson. The polynomials for the translations u and the rotations ϕ can be expressed as

$$\begin{aligned} u_i(\xi, \eta) &= a_0 + a_1\xi + a_2\eta + a_3\xi\eta + a_4\xi^2 + a_5\eta^2 + a_6\xi^2\eta + a_7\xi\eta^2 \\ \phi_i(\xi, \eta) &= b_0 + b_1\xi + b_2\eta + b_3\xi\eta + b_4\xi^2 + b_5\eta^2 + b_6\xi^2\eta + b_7\xi\eta^2 \end{aligned} \quad (7.85)$$

Typically, for a rectangular element, these polynomials yield approximately the following strain and stress distribution along the element area in a ζ lamina of a layer. The strain ε_{xx} , the curvature κ_{xx} , the moment m_{xx} , the membrane force n_{xx} and the shear force q_{xz} vary linearly in x direction and quadratically in y direction. The strain ε_{yy} , the curvature κ_{yy} , the moment m_{yy} , the membrane force n_{yy} and the shear force q_{yz} vary linearly in y direction and quadratically in x direction. To avoid membrane and shear locking, the only (and default) is a reduced 2×2 integration scheme over the area. The default in ζ direction (layer thickness) is 3-point, 2-point is a suitable option. Schemes higher than 3-point in ζ direction are only useful in case of nonlinear analysis.

$[n_{\xi} = 2, n_{\eta} = 2]$
 $[n_{lc} = 3]$

7.3.4.26 CT36L – triangle, 6 nodes

The CT36L element [Fig. 7.84] is a six-node triangular isoparametric curved layered shell

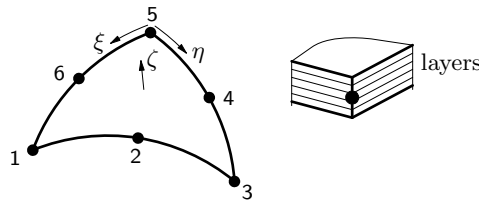


Figure 7.84: CT36L

element with drilling rotation. In each node there are three translational degrees of freedom defined, u_x , u_y , and u_z , two rotational degrees of freedom around the axes in the element plane ϕ_x and ϕ_y , and one rotational degree of freedom around the axis normal to the element plane, ϕ_z . It is based on quadratic interpolation and area integration per layer. The integration in ζ direction (layer thickness) may be Gauss or Simpson. The polynomials for the translations u and the rotations ϕ can be expressed as

$$\begin{aligned} u_i(\xi, \eta) &= a_0 + a_1\xi + a_2\eta + a_3\xi\eta + a_4\xi^2 + a_5\eta^2 \\ \phi_i(\xi, \eta) &= b_0 + b_1\xi + b_2\eta + b_3\xi\eta + b_4\xi^2 + b_5\eta^2 \end{aligned} \quad (7.86)$$

Typically, these polynomials yield approximately the following strain and stress distribution along the element area in a ζ lamina of layer. The strain ε_{xx} , the curvature κ_{xx} , the moment m_{xx} , the membrane force n_{xx} and the shear force q_{xz} vary linearly in x direction and quadratically in y direction. The strain ε_{yy} , the curvature κ_{yy} , the moment m_{yy} , the membrane force n_{yy} and the shear force q_{yz} vary linearly in y direction and quadratically in x direction. To avoid membrane and shear locking, the only (and default) is a reduced 3-point integration scheme over the area of a layer. The default in ζ direction (layer thickness) is 3-point, 2-point is a suitable option. Schemes higher than 3-point in ζ direction are only useful in case of nonlinear analysis. [$n_{lc} = 3$]
[$n_{\zeta} = 3$]

7.3.4.27 CQ48L – quadrilateral, 8 nodes

The CQ48L element [Fig. 7.85] is an eight-node quadrilateral isoparametric curved layered

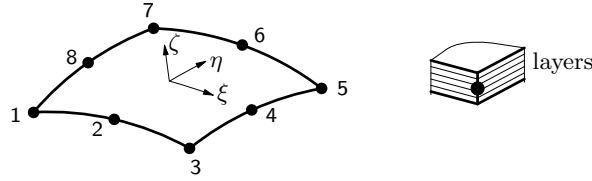


Figure 7.85: CQ48L

shell element with drilling rotation. In each node there are three translational degrees of freedom defined, u_x , u_y , and u_z , two rotational degrees of freedom around the axes in the element plane ϕ_x and ϕ_y , and one rotational degree of freedom around the axis normal to the element plane, ϕ_z . It is based on quadratic interpolation and Gauss integration over the $\xi\eta$ element area. The integration in ζ direction (layer thickness) may be Gauss or Simpson. The polynomials for the translations u and the rotations ϕ can be expressed as

$$\begin{aligned} u_i(\xi, \eta) &= a_0 + a_1\xi + a_2\eta + a_3\xi\eta + a_4\xi^2 + a_5\eta^2 + a_6\xi^2\eta + a_7\xi\eta^2 \\ \phi_i(\xi, \eta) &= b_0 + b_1\xi + b_2\eta + b_3\xi\eta + b_4\xi^2 + b_5\eta^2 + b_6\xi^2\eta + b_7\xi\eta^2 \end{aligned} \quad (7.87)$$

Typically, for a rectangular element, these polynomials yield approximately the following strain and stress distribution along the element area in a ζ lamina of a layer. The strain ε_{xx} , the curvature κ_{xx} , the moment m_{xx} , the membrane force n_{xx} and the shear force q_{xz} vary linearly in x direction and quadratically in y direction. The strain ε_{yy} , the curvature κ_{yy} , the moment m_{yy} , the membrane force n_{yy} and the shear force q_{yz} vary linearly in y direction and quadratically in x direction. To avoid membrane and shear locking, the only (and default) is a reduced 2×2 integration scheme over the area. The default in ζ direction (layer thickness) is 3-point, 2-point is a suitable option. Schemes higher than 3-point in ζ direction are only useful in case of nonlinear analysis. [$n_{\xi} = 2, n_{\eta} = 2$]
[$n_{lc} = 3$]

7.3.5 Integration Schemes

For all plate bending and shell elements, DIANA performs a numerical integration with an appropriate default scheme. For curved shell elements by default three integration points over the thickness of the element are defined and for layered shell elements three integration points per layer over the thickness. For efficiency you may want to reduce the number of integration points, or for achieving more accurate results with nonlinear material behaviour you may want to increase the number of integration points. You may choose an alternative scheme via a special data input, depending on the shape of the element. For more details see Volume *Element Library*.

7.3.6 Thickness and Shape Definition

Thickness of plate bending, flat, and curved shell elements may be uniform or nonuniform. If you only specify **t1** then the thickness is uniform [Fig. 7.86b]. Nonuniform (tapered) thickness varies linearly, quadratically, or cubically over the element area depending on

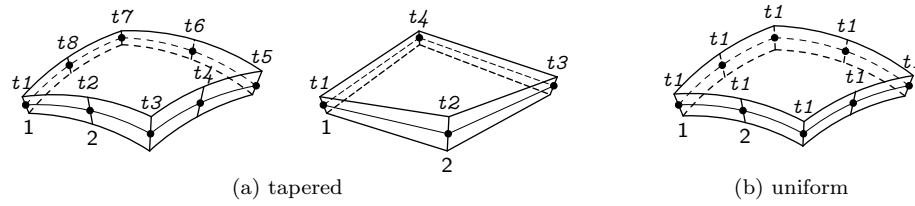


Figure 7.86: Thickness

the order of the element [Fig. 7.86a]. For nonuniform thickness the thickness is defined in every node of the element.

For curved shell elements, DIANA determines the direction in which the thickness is measured from the element shape. There are some predefined shapes for curved elements which you may specify with the element properties. If you do so, then DIANA checks whether the coordinates of the element nodes fit to the specified shape and produces an error message if this check fails. If none of the predefined shapes are appropriate, then you can specify an arbitrary shape by means of explicit thickness vectors for each element node. If you specify neither a predefined shape nor explicit thickness vectors then, by default, DIANA will apply an averaging procedure on the nodal normals which are initially determined by assuming a parabolic shape.

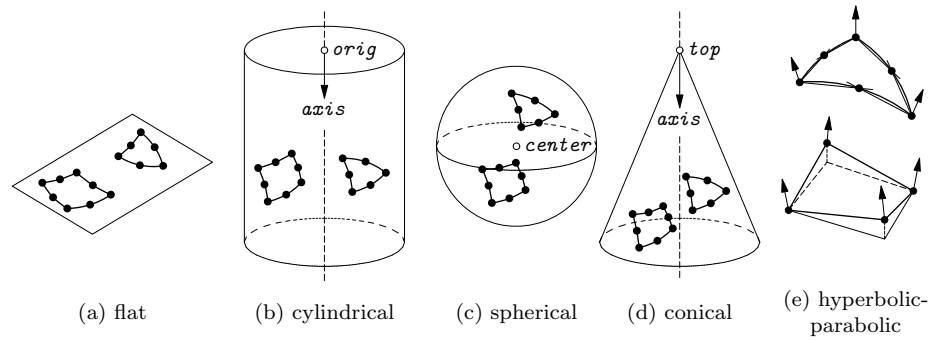


Figure 7.87: Predefined shapes

7.3.7 Shear Reduction and Geometry Factors

Since the actual transverse shear stress and strain vary parabolically over the thickness of flat and curved shell elements, the shear strain is an equivalent constant strain on a corresponding area. By default DIANA applies a shear reduction factor $S = 1.2$, which means that the constant shear stress yields approximately the same shear strain energy as the actual shear stress. You may customize the shear reduction in the model definition.

To calculate Cauchy stresses for flat shell elements, DIANA needs some geometry factors which it cannot derive from the element shape and thickness. Input depends on iso- or orthotropic geometry. If you do not specify any geometry factors, then DIANA assumes isotropic geometry with a shape factor $k_z = 1.5$. The shape factor k_z is used to calculate the maximum transfer shear stress and shear strain in the mid plane:

$$\sigma_{xz} = k_z \frac{q_{xz}}{t} \quad \sigma_{yz} = k_z \frac{q_{yz}}{t} \quad \gamma_{xz} = k_z \Psi_{yz} \quad (7.88)$$

7.3.8 Eccentric Connections for Curved Shell Elements

Curved shell elements may be connected eccentrically to their nodes. The eccentricity must be defined in the nodal element xyz directions [Fig. 7.88].

7.3.9 Linear Elastic Material Properties

The material properties to be input for linear elasticity are:

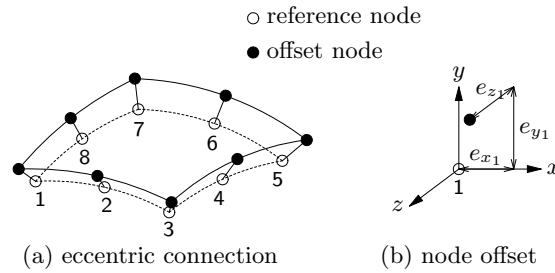


Figure 7.88: Eccentric connection

- Young's modulus E , always necessary.
- Poisson's ratio ν , necessary for two-dimensional and three-dimensional elements.
- Shear modulus G , necessary for orthotropic elasticity.
- Thermal expansion coefficient α , necessary to determine a temperature load.
- Concentration expansion coefficient γ , necessary to determine a concentration load.

Table 7.3 shows the required input for orthotropic elasticity for the various element families.

Table 7.3: LINEAR ORTHOTROPIC ELASTICITY

	plate bend.	fl. shell	cu. shell
Young's modulus	E_x	E_x	E_x
	E_y	E_y	E_y
	E_z	E_z	E_z
Poisson's ratio	ν_{xy}	ν_{xy}	ν_{xy}
		ν_{yz}	ν_{yz}
		ν_{xz}	ν_{xz}
Shear modulus	G_{xy}	G_{xy}	G_{xy}
		G_{yz}	G_{yz}
		G_{xz}	G_{xz}
Thermal exp.	α_x	α_x	α_x
	α_y	α_y	α_y
			α_z
Concentr. exp.	γ_x	γ_x	γ_x
	γ_y	γ_y	γ_y
			γ_z

The default xyz directions for orthotropic properties are the same as for the element. However, if the material xyz directions for a layer in a layered shell element do not coincide with the element axes, then you may specify these directions explicitly in addition to the orthotropic material properties by specifying a user-defined x axis in the respective material properties or by a user-specified fiber orientation angle θ in degrees. When the fiber orientation angle is specified, DIANA rotates the local xyz coordinate system around the z axis by θ . This option is particularly useful when the shell structure is curved, e.g. cylindrical, conical, spherical, etc.

7.4 Trusses and Cables

Truss elements are bars which must fulfill the condition that the dimensions d perpendicular to the bar axis are small in relation to the bar's length l [Fig. 7.89a]. The deformation of truss elements can only be the axial elongation Δl , there is neither bending nor shear deformation. Truss elements may be used to analyse bar structures with hinged connections like space decks, stiffeners in walls or structural parts connected by hinges. Also in modelling of discrete reinforcement bars, truss elements may be used.

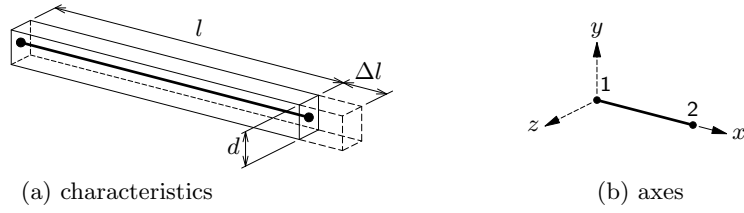


Figure 7.89: Truss elements

7.4.1 Element Axes

For all truss elements DIANA sets up an element x axis from the first to the second node [Fig. 7.89b]. For the regular elements this is the only axis.

7.4.2 Displacements

Regular elements have a minimal set of variables. This section only describes the basic variables, which are valid for all truss element types. The basic variables of truss elements

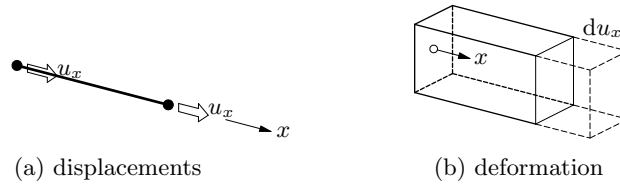


Figure 7.90: Truss elements, basic variables

are the translations u_x of the nodes in element x direction as shown in Figure 7.90a for a two-node element.

$$\mathbf{u}_e = \{u_x\} \quad (7.89)$$

7.4.3 Strains and Stresses

From the displacements u_x in the nodes, DIANA derives the deformation du_x of an infinitesimal part dx as shown in Figure 7.90b in its positive direction (elongation). This deformation yields the axial strain

$$\varepsilon = \{\varepsilon_{xx}\} = \frac{\partial u_x}{\partial x} \quad (7.90)$$

DIANA can calculate and output two types of stresses for truss elements: the normal Cauchy stress σ_{xx} and the generalized axial force N_x . From the basic strain of Eq. (7.90)

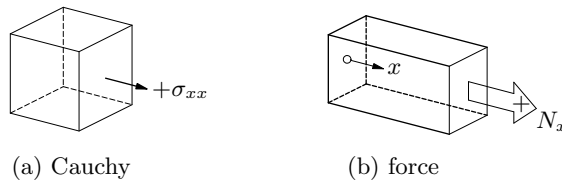


Figure 7.91: Truss elements, stresses

DIANA derives the following Cauchy stress.

$$\boldsymbol{\sigma} = \{\sigma_{xx}\} \quad (7.91)$$

Figure 7.91a shows this stress on a unit cube in its positive direction (tension). DIANA calculates the normal force in truss elements as shown in Figure 7.91b in its positive direction (tension).

7.4.4 Element Types

Truss element types are lines in DIANA. The elements may have linear or quadratic interpolation functions for the displacement field. For elements with linear interpolation functions there are only two nodes, whereas elements with quadratic interpolation functions have three nodes along the line. There are three types of truss elements: regular, enhanced, and cable elements. Compared to regular truss elements, the enhanced elements have additional degrees of freedom perpendicular to the bar axis. Therefore these elements can be used in geometrically nonlinear and in dynamic analyses. Another useful application of cable elements is the discrete modelling of prestress cables (tendons) in nonlinear analysis of reinforced concrete. Depending on the amount of additional degrees of freedom, these elements are suited for two-dimensional or three-dimensional analysis. The next paragraphs [§ 7.4.4.1 to § 7.4.4.5] give an overview of the different element types where the numbers display the relative node numbers for the element connectivity. The isoparametric coordinate ξ is also displayed for the different element types.

7.4.4.1 L2TRU, straight, 2 nodes

The L2TRU element [Fig. 7.92] is a two-node directly integrated (1-point) regular truss el-

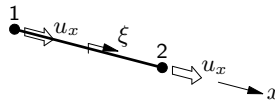


Figure 7.92: L2TRU

ement which may be used in one-, two-, and three-dimensional models. The interpolation polynomial for the displacement u_x can be expressed as

$$u_x(\xi) = a_0 + a_1\xi \quad (7.92)$$

This polynomial yields a strain ε_{xx} which is constant along the bar axis.

Caution: the mass distribution of an L2TRU element is complete in the element x axis, but incomplete in the model XYZ axes. Therefore this element must not be used in dynamic analysis, use the enhanced truss elements instead.

7.4.4.2 L4TRU, straight, 2 nodes, 2-D

The L4TRU element [Fig. 7.93] is a two-node directly integrated (1-point) enhanced truss

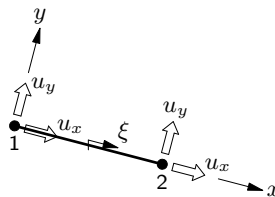


Figure 7.93: L4TRU

element. Due to the displacements u_y perpendicular to the bar axis, this element can be used in two-dimensional dynamic and geometrically nonlinear analysis. The interpolation polynomial for the displacements u_x and u_y can be expressed as

$$u_i(\xi) = a_0 + a_1\xi \quad (7.93)$$

This polynomial yields a strain ε_{xx} which is constant along the bar axis. Initially there is no stiffness for the displacements u_y , stiffness will arise in case of geometric nonlinearities.

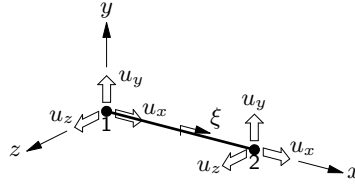


Figure 7.94: L6TRU

7.4.4.3 L6TRU, straight, 2 nodes, 3-D

The L6TRU element [Fig. 7.94] is a two-node directly integrated (1-point) enhanced truss element. Due to the displacements u_y and u_z perpendicular to the bar axis [Fig. 7.94], this element can be used in three-dimensional dynamic and geometrically nonlinear analysis. The interpolation polynomial for the displacements u_x , u_y and u_z can be expressed as

$$u_i(\xi) = a_0 + a_1\xi \quad (7.94)$$

This polynomial yields a strain ε_{xx} which is constant along the bar axis. Initially there is no stiffness for the displacements u_y and u_z , stiffness will arise in case of geometric nonlinearities.

7.4.4.4 CL6TR, curved, 3 nodes, 2-D

The CL6TR element [Fig. 7.95] is a three-node numerically integrated truss element with

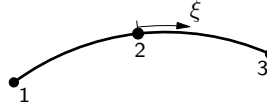


Figure 7.95: CL6TR

two displacements u_x and u_y in each node. This element can be used in two-dimensional dynamic and nonlinear analysis of cables. The interpolation polynomial for the displacements u_x and u_y can be expressed as

$$u_i(\xi) = a_0 + a_1\xi + a_2\xi^2 \quad (7.95)$$

Typically, this polynomial yields a strain which varies linearly along the bar axis. Strain and stress are purely axial. By default DIANA applies a 2-point Gauss integration scheme. Suitable options are 1 to 7-point Gauss, 2 to 10-point Newton-Cotes, and 2 to 10-point Lobatto.

[$n_\xi = 2$]

7.4.4.5 CL9TR, curved, 3 nodes, 3-D

The CL9TR element [Fig. 7.96] is a three-node numerically integrated truss element with

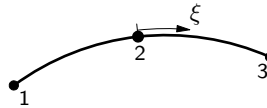


Figure 7.96: CL9TR

three displacements u_x , u_y and u_z in each node. This element can be used in three-dimensional dynamic and nonlinear analysis of cables. The interpolation polynomial for the displacements u_x , u_y and u_z can be expressed as

$$u_i(\xi) = a_0 + a_1\xi + a_2\xi^2 \quad (7.96)$$

Typically, this polynomial yields a strain which varies linearly along the bar axis. Strain and stress are purely axial. By default DIANA applies a 2-point Gauss integration scheme. Suitable options are 1 to 7-point Gauss, 2 to 10-point Newton-Cotes, and 2 to 10-point Lobatto.

[$n_\xi = 2$]

7.4.5 Cross-section

For all types of truss elements the cross-section is uniform along the element axis.

7.4.6 Linear Elastic Material Properties

The material properties to be input for linear elasticity are:

- Young's modulus E , always necessary.
- Thermal expansion coefficient α , necessary to determine a temperature load.
- Concentration expansion coefficient γ , necessary to determine a concentration load.

For trusses and cables no orthotropic elastic behaviour can be specified.

7.5 Beams

Beam elements are bars which must fulfill the condition that the dimensions d perpendicular to the bar axis are small in relation to the bar's length l [Fig. 7.97]. Beam elements

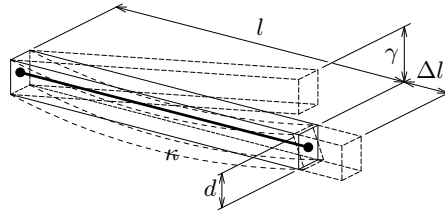


Figure 7.97: Beam elements, characteristics

may have axial deformation Δl , shear deformation γ , curvature κ and torsion, therefore they can describe axial force, shear force and moment. Beam elements are typically used to analyse two- and three-dimensional frames. In combination with continuum elements they can also be used to model stiffeners in plates or shells etc. DIANA offers three classes of beam elements:

- Class-I* – classical beam elements with directly integrated cross-sections. These elements may be used in linear and in geometric nonlinear analysis. Physic nonlinear analysis is limited to generalized stress-strain diagrams.
- Class-II* – fully numerically integrated classical beam elements. These elements may be used in linear and in geometric and physic nonlinear analysis.
- Class-III* – fully numerically integrated *Mindlin* beam elements. These elements may be used in linear and in geometric and physic nonlinear analysis.

All three classes of beam elements are numerically integrated along the beam axis, therefore the dimensions of the cross-section may vary along the beam axis.

- Classical beams

The classical beams (class-I and class-II) are two-node straight elements. The transverse displacement for these elements is a cubic Hermite shape function expressed in the nodal displacements and rotations. It is assumed that the cross-sections remain plane and perpendicular to the slope of the beam axis. Therefore these beam elements may be viewed as based on the *Bernoulli* theory. For the class-I beams, the incorporation of shear deformation is an option.

- Mindlin beams

For the fully numerically integrated Mindlin beam elements (class-III), the normal strain ε_{xx} varies linearly over the cross-section area and the transverse shear strains are forced to be constant. Since the actual transverse shear stress and strain vary

quadratically over the cross-section area, the shear strain is an equivalent constant strain on a corresponding area.

By default DIANA assumes a shear stress correction factor of 1.2 which, for homogeneous cross-sections, approximates the condition that the constant shear stress yields the same shear strain energy as the actual shear stress does. You may input alternative shear stress correction factors.

7.5.1 Element Axes

Two-dimensional beam elements have translational degrees of freedom in one plane, the element's xy plane. An element z axis is used to determine the orientation of this plane in XYZ space [Fig. 7.98]. If you specify the \bar{z} axis, then DIANA determines in each node

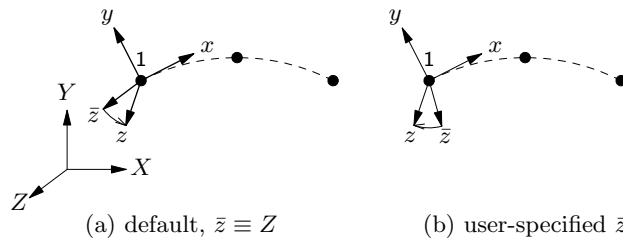


Figure 7.98: Axes for two-dimensional beams

an x axis tangential to the beam axis. Next the $y \perp \bar{z}x$ is set up and finally $z \perp xy$ is set up which is in fact a correction of the \bar{z} direction [Fig. 7.98a,b].

If the x axis coincides with the model Z direction then $y \perp \bar{z}x$ can not be done and the \bar{y} axis is chosen in model Y direction. In that case the element xyz axes are set up as follows: x in the direction of the beam axis, $z \perp x\bar{y}$ and $y \perp zx$.

The xyz axes in the first node of two-dimensional beams serve to describe the direction of the displacement degrees of freedom in the nodes. To describe the strains and stresses, the xy directions are locally rotated by z to $\hat{x}\hat{y}$ in such a way that the \hat{x} axis is tangential to the beam axis and that \hat{y} is perpendicular to it and $\hat{z} \equiv z$. The $\hat{x}\hat{y}\hat{z}$ axes serve also to input element loads in local element directions. If the beam is straight, then this transformation has no effect. DIANA then checks if all nodes of the element are in the plane perpendicular to the z axis, a fatal error message “non planar element” occurs if this is not true.

For three-dimensional beam elements in general there is no unique element plane and DIANA needs a \bar{z} direction to set up the element axes [Fig. 7.99]. If you specify the \bar{z}

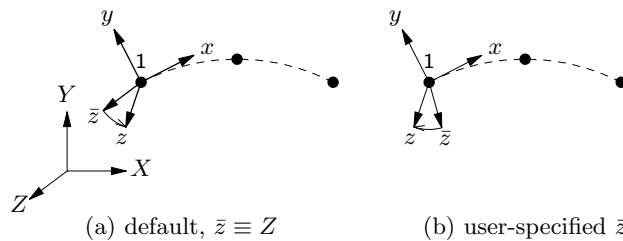


Figure 7.99: Axes for three-dimensional beams

axis, then DIANA determines in each node an x axis tangential to the beam axis. Next the $y \perp \bar{z}x$ is set up and finally $z \perp xy$ is set up which is in fact a correction of the \bar{z} direction.

If the x axis coincides with the model Z direction then $y \perp \bar{z}x$ can not be done and the \bar{y} axis is chosen in model Y direction. In that case the element xyz axes are set up as follows: x in the direction of the beam axis, $z \perp x\bar{y}$ and $y \perp zx$.

For class-III beams, you may specify a \bar{z} axis for each node by repeating the x , y and z components.

The xyz axes of three-dimensional beams serve to describe the direction of strains and stresses, for class-I beam elements they also describe the directions of the translations. The xyz directions are transformed locally to $\hat{x}\hat{y}\hat{z}$ in such a way that the \hat{x} axis is tangential to the beam axis and that \hat{y} and \hat{z} are perpendicular to it. These transformed axes serve also to input element loads in local element directions. If the beam is straight, then this transformation has no effect.

7.5.1.1 Displacements and Rotations

The basic variables of beam elements are the displacements in the nodes: translations u and rotations ϕ . The orientation of the displacements depends on the beam class and the dimensionality.

Displacements for class-I and class-II beams Eq. (7.97), are oriented in the local xyz directions [Fig. 7.100].

$$\text{two-dim.: } \mathbf{u}_e = \begin{Bmatrix} u_x \\ u_y \\ \phi_z \end{Bmatrix} \quad \text{three-dim.: } \mathbf{u}_e = \begin{Bmatrix} u_x \\ u_y \\ u_z \\ \phi_x \\ \phi_y \\ \phi_z \end{Bmatrix} \quad (7.97)$$

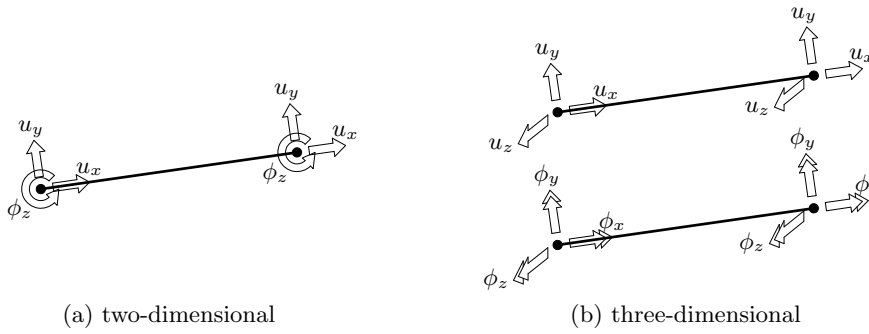


Figure 7.100: Displacements for class-I and class-II beams

Displacements for class-III beams Eq. (7.97), are oriented in the local xyz directions for two-dimensional beams [Fig. 7.101a] and in the global XYZ directions for three-dimensional beams [Fig. 7.101b].

$$\text{two-dim.: } \mathbf{u}_e = \begin{Bmatrix} u_x \\ u_y \\ \phi_z \end{Bmatrix} \quad \text{three-dim.: } \mathbf{u}_e = \begin{Bmatrix} u_X \\ u_Y \\ u_Z \\ \phi_X \\ \phi_Y \\ \phi_Z \end{Bmatrix} \quad (7.98)$$

7.5.2 Strains and Stresses

For the fully numerically integrated beam elements (class-II and class-III), DIANA derives the deformations for an infinitesimal part from the displacements in the nodes. The deformations that can be derived depend on the dimensionality of the beam element. From these deformations, DIANA derives the primary strains as described for the three classes of beam elements. The sign convention for strains is that an elongation yields a positive strain.

Figure 7.102 shows the deformations of the infinitesimal part in their positive direction. Note that flexural torsional deformation is not included in three-dimensional beam elements, therefore these elements do not describe warping of the cross-section.

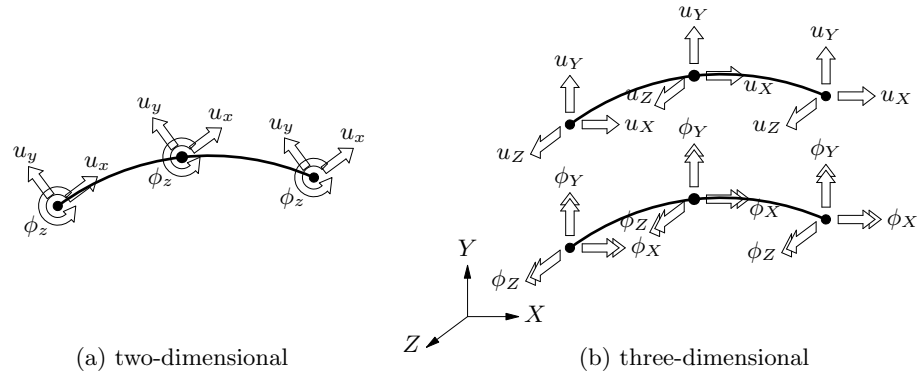


Figure 7.101: Displacements for class-III beams

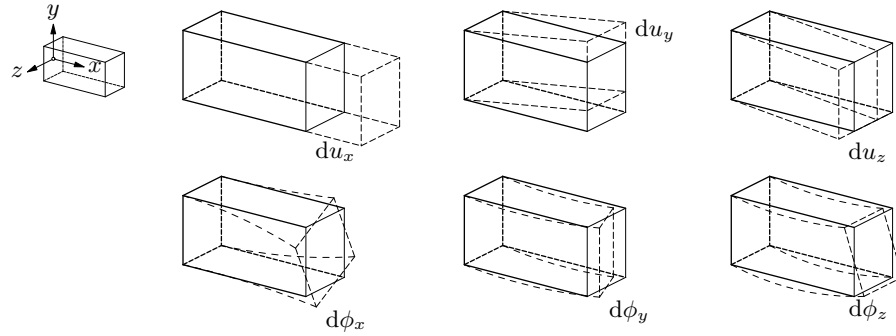


Figure 7.102: Deformation for three-dimensional beams

The primary strains of the two-dimensional class-III elements are the Green-Lagrange strains

$$\boldsymbol{\varepsilon} = \begin{Bmatrix} \varepsilon_{xx} \\ \gamma_{xy} \end{Bmatrix} \quad (7.99)$$

with

$$\varepsilon_{xx} = \frac{du_x}{dx} \quad \gamma_{xy} = \frac{du_x}{dy} + \frac{du_y}{dx} \quad (7.100)$$

The primary stresses are

$$\boldsymbol{\sigma} = \begin{Bmatrix} \sigma_{xx} \\ \sigma_{xy} \end{Bmatrix} \quad (7.101)$$

The primary strains of the three-dimensional class-III elements are the Green-Lagrange strains

$$\boldsymbol{\varepsilon} = \begin{Bmatrix} \varepsilon_{xx} \\ \gamma_{xy} \\ \gamma_{zx} \end{Bmatrix} \quad (7.102)$$

with

$$\varepsilon_{xx} = \frac{du_x}{dx} \quad \gamma_{xy} = \frac{du_x}{dy} + \frac{du_y}{dx} \quad \gamma_{zx} = \frac{du_x}{dz} + \frac{du_z}{dx} \quad (7.103)$$

The primary stresses are

$$\boldsymbol{\sigma} = \begin{Bmatrix} \sigma_{xx} \\ \sigma_{xy} \\ \sigma_{zx} \end{Bmatrix} \quad (7.104)$$

For beam elements, DIANA can calculate forces and moments in nodes and cross-sections and Cauchy stresses in stress points. The set of forces, moments and stresses depends on the dimensionality of the element.

For two-dimensional beams DIANA can calculate the forces, moments and Cauchy stresses of Eq. (7.105).

$$\mathbf{N} = \begin{Bmatrix} N_x \\ Q_y \end{Bmatrix} \quad \mathbf{M} = \{ M_z \} \quad \boldsymbol{\sigma} = \begin{Bmatrix} \sigma_{xx} \\ \sigma_{xy} = \sigma_{yx} \end{Bmatrix} \quad (7.105)$$

For three-dimensional beams DIANA can calculate the forces, moments and Cauchy stresses of Eq. (7.106).

$$\mathbf{N} = \begin{Bmatrix} N_x \\ Q_y \\ Q_z \end{Bmatrix} \quad \mathbf{M} = \begin{Bmatrix} M_x \\ M_y \\ M_z \end{Bmatrix} \quad \boldsymbol{\sigma} = \begin{Bmatrix} \sigma_{xx} \\ \sigma_{xy} = \sigma_{yx} \\ \sigma_{zx} = \sigma_{xz} \end{Bmatrix} \quad (7.106)$$

Figure 7.103 shows the primary moments and forces in a cross-section of an infinitesimal part of a three-dimensional class-I or -II beam element in their positive direction.

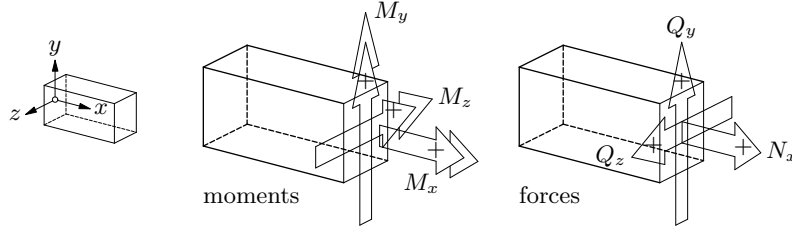


Figure 7.103: Moments and forces for three-dimensional class-I & -II beams

The sign convention for bending is that a positive bending moment yields a positive stress (tension) in the positive area ($+M_z$ works in the $-z$ direction!). The sign convention for torsion is that a positive torsion moment yields $+\sigma_{zx}$ in the $+y$ area. The sign convention for forces is that a positive force yields a positive stress.

Figure 7.104 shows the Cauchy stresses resulting from the moments and forces. The unit cube shows the complete stress situation of a three-dimensional beam element.

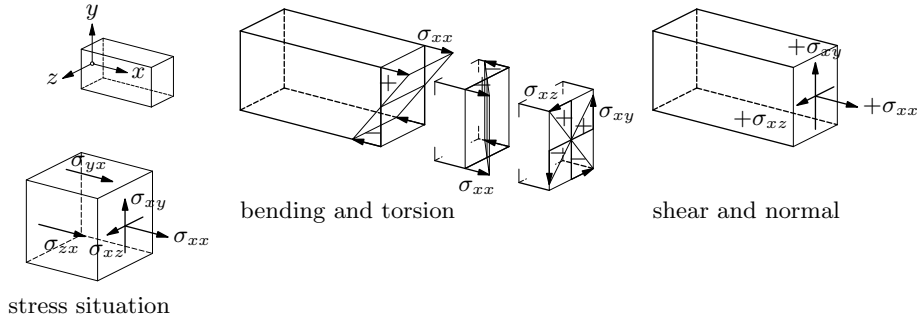


Figure 7.104: Cauchy stresses for three-dimensional beams

7.5.3 Element Types

Beam element types in DIANA are lines. The elements may have linear or quadratic interpolation functions for the displacement field. For elements with linear interpolation functions there are only two nodes at each edge, whereas for elements with quadratic interpolation function there are three nodes along the line. The next paragraphs [§ 7.5.3.1 to § 7.5.3.8] give an overview of the different element types where the numbers display the relative node numbers for the element connectivity. The isoparametric coordinates ξ , η , and ζ are also displayed for the different beam element types.

7.5.3.1 L6BEN – straight, 2 nodes, 2-D

The L6BEN element [Fig. 7.105] is a two-node, two-dimensional class-I beam element. Basic variables are the translations u_x and u_y and the rotation ϕ_z in the nodes [Eq. (7.97) p. 73]. The interpolation polynomials for the displacements can be expressed as

$$\begin{aligned} u_x(\xi) &= a_0 + a_1\xi \\ u_y(\xi) &= b_0 + b_1\xi + b_2\xi^2 + b_3\xi^3 \end{aligned} \quad (7.107)$$

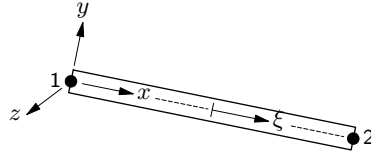


Figure 7.105: L6BEN

Due to these polynomials the strain ε_{xx} is constant and the curvature κ_z varies linearly along the center line of the beam. The primary strains of the L6BEN element are the elongation Δu and the curvature κ :

$$\varepsilon = \begin{Bmatrix} \Delta u_x \\ \kappa_z \end{Bmatrix} \quad (7.108)$$

with

$$\Delta u_x = u_x^{(2)} - u_x^{(1)} \quad \kappa_z = -\frac{d^2 u_y}{dx^2} \quad (7.109)$$

The primary stresses are the normal force N and the bending moment M :

$$\sigma = \begin{Bmatrix} N_x \\ M_z \end{Bmatrix} \quad (7.110)$$

$[n_\xi = 2]$ By default DIANA applies a 2-point Gauss integration scheme along the bar axis.

7.5.3.2 L12BE – straight, 2 nodes, 3-D

The L12BE element [Fig. 7.106] is a two-node, three-dimensional class-I beam element.

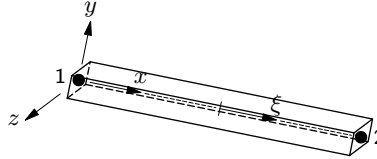


Figure 7.106: L12BE

Basic variables are the translations u_x , u_y and u_z and the rotations ϕ_x , ϕ_y and ϕ_z in the nodes [Eq. (7.97) p. 73]. The interpolation polynomials for the displacements can be expressed as

$$\begin{aligned} u_x(\xi) &= a_0 + a_1 \xi \\ u_y(\xi) &= b_0 + b_1 \xi + b_2 \xi^2 + b_3 \xi^3 \\ u_z(\xi) &= c_0 + c_1 \xi + c_2 \xi^2 + c_3 \xi^3 \end{aligned} \quad (7.111)$$

Due to these polynomials the strains are constant along the center line of the beam. The primary strains of the L12BE element are the elongation Δu , the torsion $\Delta \phi$ and the curvatures κ :

$$\varepsilon = \begin{Bmatrix} \Delta u_x \\ \Delta \phi \\ \kappa_y \\ \kappa_z \end{Bmatrix} \quad (7.112)$$

with

$$\begin{aligned} \Delta u_x &= u_x^{(2)} - u_x^{(1)} & \Delta \phi &= \phi_x^{(2)} - \phi_x^{(1)} \\ \kappa_y &= -\frac{d^2 u_z}{dx^2} & \kappa_z &= -\frac{d^2 u_y}{dx^2} \end{aligned} \quad (7.113)$$

The primary stresses are the normal force N and the moments M :

$$\sigma = \begin{Bmatrix} N_x \\ M_x \\ M_y \\ M_z \end{Bmatrix} \quad (7.114)$$

$[n_\xi = 2]$ By default DIANA applies a 2-point Gauss integration scheme along the bar axis.

7.5.3.3 L7BEN – straight, 2 nodes, 2-D

The L7BEN element [Fig. 7.107] is a two-node, two-dimensional class-II beam element.

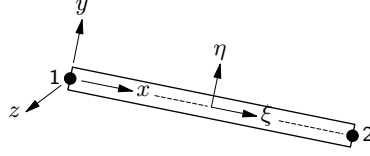


Figure 7.107: L7BEN

Basic variables are the translations u_x and u_y and the rotation ϕ_z in the nodes [Eq. (7.97) p. 73]. An additional variable is the elongation Δu_x . The interpolation polynomials for the displacements can be expressed as

$$\begin{aligned} u_x(\xi) &= a_0 + a_1\xi + a_2\xi^2 \\ u_y(\xi) &= b_0 + b_1\xi + b_2\xi^2 + b_3\xi^3 \end{aligned} \quad (7.115)$$

Due to these polynomials the strains vary linearly along the center line of the beam. The primary strain of the L7BEN element is the normal strain ε_{xx} . At an arbitrary off-axis location P this strain is

$$\varepsilon = \{ \varepsilon_{xx} \} = \frac{du_x}{dx} - \frac{d^2u_y}{dx^2} y_P \quad (7.116)$$

The primary stress is the normal Cauchy stress

$$\sigma = \{ \sigma_{xx} \} \quad (7.117)$$

By default DIANA applies a 2-point Gauss integration scheme along the bar axis. For $[n_\xi = 2]$ default schemes in the cross-section see Volume *Element Library*.

7.5.3.4 L13BE – straight, 2 nodes, 3-D

The L13BE element [Fig. 7.108] is a two-node, three-dimensional class-II beam element.

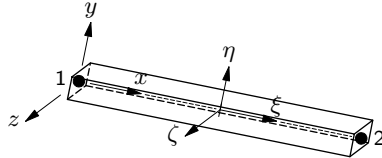


Figure 7.108: L13BE

Basic variables are the translations u_x , u_y and u_z and the rotations ϕ_x , ϕ_y and ϕ_z in the nodes [Eq. (7.97) p. 73]. An additional variable is the elongation Δu_x . The interpolation polynomials for the displacements can be expressed as

$$\begin{aligned} u_x(\xi) &= a_0 + a_1\xi + a_2\xi^2 \\ u_y(\xi) &= b_0 + b_1\xi + b_2\xi^2 + b_3\xi^3 \\ u_z(\xi) &= c_0 + c_1\xi + c_2\xi^2 + c_3\xi^3 \end{aligned} \quad (7.118)$$

Due to these polynomials the strains vary linearly along the center line of the beam. The primary strains for the L13BE element are

$$\varepsilon = \left\{ \begin{array}{c} \varepsilon_{xx} \\ \gamma_{xy} \\ \gamma_{zx} \end{array} \right\} \quad (7.119)$$

At an arbitrary off-axis location P these strains are

$$\begin{aligned}\varepsilon_{xx} &= \frac{d\bar{u}_x(y_P, z_P)}{dx} = \frac{du_x}{dx} - \frac{d^2u_y}{dx^2}y_P - \frac{d^2u_z}{dx^2}z_P \\ \gamma_{xy} &= \frac{d\bar{u}_x(y_P, z_P)}{dy} + \frac{du_y}{dx} \\ \gamma_{zx} &= \frac{d\bar{u}_x(y_P, z_P)}{dz} + \frac{du_z}{dx}\end{aligned}\quad (7.120)$$

The primary stresses for the L13BE element are

$$\boldsymbol{\sigma} = \begin{Bmatrix} \sigma_{xx} \\ \sigma_{xy} \\ \sigma_{zx} \end{Bmatrix} \quad (7.121)$$

[$n_\xi = 2$] By default DIANA applies a 2-point Gauss integration scheme along the bar axis. For default schemes in the cross-section see Volume *Element Library*.

7.5.3.5 L6BEA – straight, 2 nodes, 2-D

The L6BEA element [Fig. 7.109] is a two-node, two-dimensional class-III beam element.

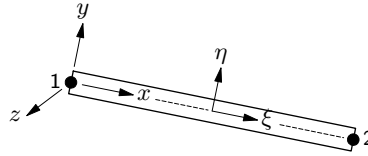


Figure 7.109: L6BEA

Basic variables are the translations u_x and u_y and the rotation ϕ_z in the nodes [Eq. (7.98) p. 73]. The interpolation polynomials for the displacements can be expressed as

$$\begin{aligned}u_x(\xi) &= a_0 + a_1\xi \\ u_y(\xi) &= b_0 + b_1\xi \\ \phi_z(\xi) &= c_0 + c_1\xi\end{aligned}\quad (7.122)$$

Due to these polynomials the strains are constant along the center line of the beam. For primary strain and stress see Eq. (7.99) on page 74. By default DIANA applies a 1-point Gauss integration scheme along the bar axis. For default schemes in the cross-section see Volume *Element Library*.

7.5.3.6 CL9BE – curved, 3 nodes, 2-D

The CL9BE element [Fig. 7.110] is a three-node, two-dimensional class-III beam element.

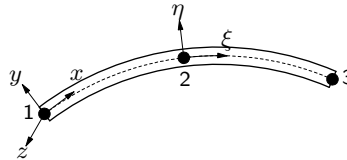


Figure 7.110: CL9BE

Basic variables are the translations u_x and u_y and the rotation ϕ_z in the nodes [Eq. (7.98) p. 73]. The interpolation polynomials for the displacements can be expressed as

$$\begin{aligned}u_x(\xi) &= a_0 + a_1\xi + a_2\xi^2 \\ u_y(\xi) &= b_0 + b_1\xi + b_2\xi^2 \\ \phi_z(\xi) &= c_0 + c_1\xi + c_2\xi^2\end{aligned}\quad (7.123)$$

Due to these polynomials the strains vary linearly along the center line of the beam. For primary strain and stress see Eq. (7.99) on page 74. By default DIANA applies a 2-point Gauss integration scheme along the bar axis. For default schemes in the cross-section see [n_ξ = 2] Volume *Element Library*.

Gauss integration with more than two integration points along the bar axis yields incorrect answers if the deformation is not pure bending!

7.5.3.7 L12BEA – straight, 2 nodes, 3-D

The L12BEA element [Fig. 7.111] is a two-node, three-dimensional class-III beam element.

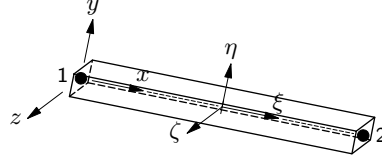


Figure 7.111: L12BEA

Basic variables are the translations u_x , u_y and u_z and the rotations ϕ_x , ϕ_y and ϕ_z in the nodes [Eq. (7.98) p. 73]. The interpolation polynomials for the displacements can be expressed as

$$\left. \begin{aligned} u_i(\xi) &= a_{i0} + a_{i1}\xi \\ \phi_i(\xi) &= b_{i0} + b_{i1}\xi \end{aligned} \right\} \quad i = x, y, z \quad (7.124)$$

Due to these polynomials the strains are constant along the center line of the beam. For primary strain and stress see Eq. (7.102) on page 74. By default DIANA applies a 1-point Gauss integration scheme along the bar axis. For default schemes in the cross-section see [n_ξ = 1] Volume *Element Library*.

7.5.3.8 CL18B – curved, 3 nodes, 3-D

The CL18B element [Fig. 7.112] is a three-node, three-dimensional class-III beam element.

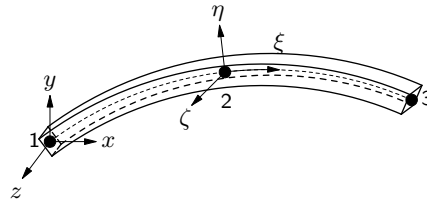


Figure 7.112: CL18B

Basic variables are the translations u_x , u_y and u_z and the rotations ϕ_x , ϕ_y and ϕ_z in the nodes [Eq. (7.98) p. 73]. The interpolation polynomials for the displacements can be expressed as

$$\left. \begin{aligned} u_i(\xi) &= a_{i0} + a_{i1}\xi + a_{i2}\xi^2 \\ \phi_i(\xi) &= b_{i0} + b_{i1}\xi + b_{i2}\xi^2 \end{aligned} \right\} \quad i = x, y, z \quad (7.125)$$

Due to these polynomials the strains vary linearly along the center line of the beam. For primary strain and stress see Eq. (7.102) on page 74. By default DIANA applies a 2-point Gauss integration scheme along the bar axis. For default schemes in the cross-section see [n_ξ = 2] Volume *Element Library*.

Gauss integration with more than two integration points in along the bar axis yields incorrect answers if the deformation is not pure bending!

7.5.4 Cross-section Definition

To describe the cross-sections of beam elements, DIANA offers you some *predefined shapes*. For class-I beam elements you may use the *profile library*, which is part of the DIANA installation, as an alternative for the predefined shapes. Arbitrary cross-section shapes may be input for all types of beam elements.

DIANA offers a set of predefined shapes as shown in Figure 7.113. For all these predefined shapes, the beam axis coincides with the elastic neutral axis. All dimensions

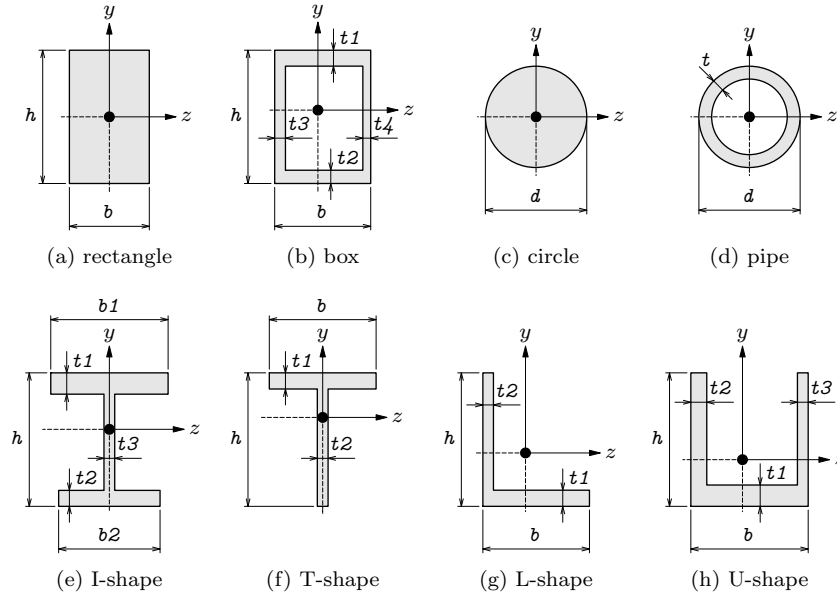


Figure 7.113: Predefined cross-sections

may be specified with one or more values. One value if the dimension is the same in all element nodes, more values for variable dimension: one for each respective node, i.e., a tapered cross-section. All dimensions must be greater than zero.

A library of profiles is available on system files in the DIANA installation. You may use this library to specify the cross-section of class-I beam elements. Profile data, like cross-section area A and moments of inertia I , have been stored in the library.

DIANA provides the following classes: HEA, HEAA, HEB, HEC, HEM, HER, HLA, HLAA, HLB, HLM, HLR, HXA, HXAA, HXB, HXM, HXR, IPE, IPEA, IPEO, IPER, IPEV.

If neither a predefined shape nor a profile is appropriate for a beam element cross-section then you may specify an arbitrary cross-section shape.

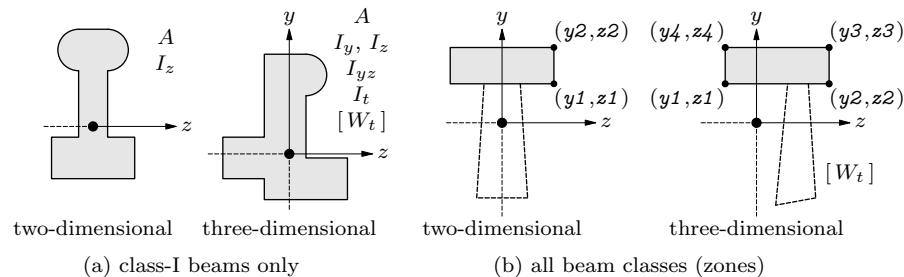


Figure 7.114: Arbitrary cross-sections

For class-I beams the cross-section properties can be specified explicitly by defining [Fig. 7.114a]:

- The area A of the cross-section. One value denotes a constant area of cross-section along the beam axis, more values (one for each element node) denote a tapered cross-section with variable A along the beam axis.

- The moment(s) of inertia around the local element axes: iz is I_z , iy is I_y , iyz is I_{yz} , and it is I_t where I_y , I_{yz} , and I_t are for three-dimensional beam elements only. One value for each I denotes constant moments of inertia along the beam axis, more values for each I (one for each element node) denote a tapered cross-section with variable I along the beam axis. The specified moments of inertia must satisfy the condition that

$$I_y + I_z > \sqrt{(I_y - I_z)^2 + 4I_{yz}^2}$$

- The torsional rigidity W_t for calculation of the average torsional shear stress: $\sigma_t = M_x/W_t$. Shear stresses will be calculated as $\sigma_{xy} = Q_y/A + \text{contribution } \sigma_t$ and $\sigma_{xz} = Q_z/A + \text{contribution } \sigma_t$. DIANA needs the torsional rigidity to calculate Cauchy stresses in user-specified stress points during post-processing. One value denotes a constant torsional rigidity along the beam axis, more values (one for each element node) denote a tapered cross-section with variable W_t along the beam axis.

For all beam classes an arbitrary cross-section may be specified with a number of quadrilateral (not necessarily rectangular!) zones [Fig. 7.114b]. For class-I beams the zone information is used to compute the moments of inertia I_z , I_y , I_{yz} , and I_t . Note that average torsional shear stresses for class-I beams can only be computed during post-processing if the torsional rigidity W_t is specified. For class-II and class-III beams the zones are numerically integrated. For class-II and class-III beams with zones, you may specify the material properties for each zone.

7.5.5 Integration Schemes

For all beam elements DIANA performs a numerical integration with an appropriate default scheme. The default of integration points in the ξ direction is defined in the table above. For the numerical integrated beams (class-II and class-III) the number of integration points in the cross-section is dependent on the defined shape of the cross-section. For efficiency you may want to reduce the number of integration points, or for achieving more accurate results with nonlinear material behaviour you may want to increase the number of integration points. You may choose an alternative scheme via a special data input, depending on the shape of the element. For more details see Volume *Element Library*.

7.5.6 Hinging and Sliding Connections

To model a hinged connection, appropriate for connection to beam, plate bending or shell elements, DIANA doubles the rotation degree of freedom. To model a sliding connection, appropriate for beam elements, DIANA doubles the translation degree of freedom.

7.5.7 Eccentric Connections

Class-I and class-III beam elements may be connected eccentrically to their nodes.

Class-II beam elements cannot be connected eccentrically to their nodes.

The eccentricity must be defined in the local element xyz directions [Fig. 7.115]. If only

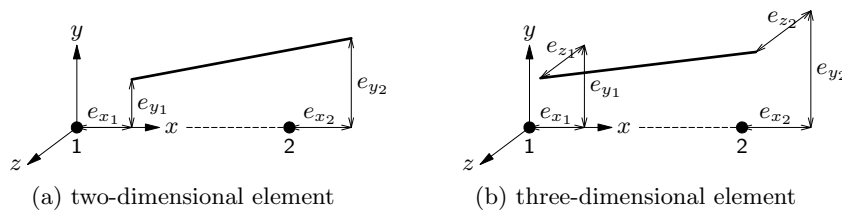


Figure 7.115: Eccentric connection class-I beams

one value is specified for each eccentricity, then the eccentricities are the same for all element nodes. DIANA assumes that element loads on eccentrically connected beams act on the reference axis [Fig. 7.116]. For instance a specified normal force line load q_y (in

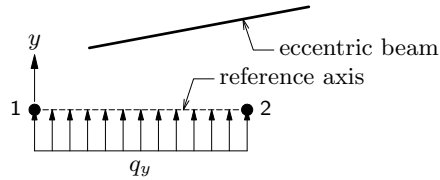


Figure 7.116: Normal force load on eccentric beam

element y direction) acts perpendicularly to the connection between the nodes. Such load introduces neither shear forces nor bending moments.

7.5.8 Linear Elastic Material Properties

The material properties to be input for linear elasticity are:

- Young's modulus E , always necessary.
- Poisson's ratio ν , necessary for two-dimensional and three-dimensional elements.
- Shear modulus G , necessary for orthotropic elasticity.
- Thermal expansion coefficient α , necessary to determine a temperature load.
- Concentration expansion coefficient γ , necessary to determine a concentration load.

For beams no orthotropic elastic behaviour can be specified.

7.6 Structural Interface Elements

The structural interface elements describe the interface behaviour in terms of a relation between the normal and shear tractions and the normal and shear relative displacements across the interface. Typical applications for structural interface elements are elastic bedding, nonlinear elastic bedding (for instance no-tension bedding), discrete cracking, bond-slip along reinforcement, friction between surfaces, joints in rock, masonry etc. The connectivity of interfaces does not change during the analysis, and therefore these elements should not be used in models with large slip (relative slip displacements in order of element edge or larger). With respect to shape and connectivity there are three types of structural interface elements:

- Nodal interface elements
Nodal interface elements are placed between two nodes. With these elements, the interface surface and directions are user-specified.
- Two-dimensional line interface elements
Two-dimensional line interface elements are placed between truss elements, beam elements or edges of two-dimensional elements. With these elements the interface surface and directions are evaluated automatically from the geometry of the element itself.
- Plane interface elements
Plane interface elements are placed between faces of three-dimensional elements. With these elements the interface surface and directions are also evaluated automatically from the geometry of the element itself.

7.6.1 Element Axes

For nodal interface elements, the specification of the surface allocation and axes depends on the dimensionality of the element: two-dimensional or three-dimensional.

For two-dimensional nodal interface elements the interface surface s and the direction normal to the interface surface (x axis) must be specified.

For three-dimensional nodal interface elements the interface surface s , the direction normal the interface surface (x axis) and the in-plane y axis must be specified.

The nodal surface allocation not only depends on the dimensions, but also on the interpolation of the surrounding continuum elements. Connection of four nodal interface elements to the edges of a face of a linear brick element gives an allocation of $\frac{1}{4}$ at the four corner nodes. Connection of eight nodal interface elements to the edges of a quadratic brick element gives $-\frac{1}{12}$ at the four corner nodes and $\frac{1}{3}$ at the mid-side nodes.

For two-dimensional line interface elements the out-of-plane direction (z axis) and the configuration of the interface element must be defined by the user. If you do not specify the z axis then DIANA takes the model Z direction by default ($z \equiv Z$). In any case the z axis must be perpendicular to the y axis ($z \perp y$). The configuration definition of the interface defines whether the interface element is applied in a plane stress, a plane strain or an axisymmetric model. It is also possible to define the configuration for a bond-slip interface of a truss element in a two-dimensional body element. Thickness input is required for plane stress and bond-slip: for plane stress it is the out-of-plane thickness, for bond-slip it is the (sum of the) perimeter(s) of the reinforcing bar(s). For plane strain and axisymmetry, the thickness input is not necessary. Thicknesses t are defined in the respective node pairs of the interface element. If you only specify one thickness value then the thickness is uniform.

For plane interface elements, i.e., triangles or quadrilaterals, the default x axis direction is in the tangential plane at the first node and points to the second node. If you prefer an x axis other than default, then you must specify its direction.

7.6.2 Displacements, Relative Displacements, and Traction

The basic variables for structural interfaces are the nodal displacements $\Delta \mathbf{u}_e$. The derived values are the relative displacements $\Delta \mathbf{u}$ and the tractions \mathbf{t} . The structural interface elements describe a relation between \mathbf{t} and $\Delta \mathbf{u}$ across the interface. DIANA can output the derived values in the integration points. The actual set of variables depends on the type of the interface element.

Two-dimensional nodal. Variables of two-dimensional structural nodal interfaces are oriented in the local xy axes.

$$\mathbf{u}_e = \begin{Bmatrix} u_x \\ u_y \end{Bmatrix} \quad \Delta \mathbf{u} = \begin{Bmatrix} \Delta u_{nx} \\ \Delta u_{sy} \end{Bmatrix} \quad \mathbf{t} = \begin{Bmatrix} t_{nx} \\ t_{sy} \end{Bmatrix} \quad (7.126)$$

The normal traction t_{nx} is perpendicular to the interface; the shear traction t_{sy} is tangential to the interface [Fig. 7.117].

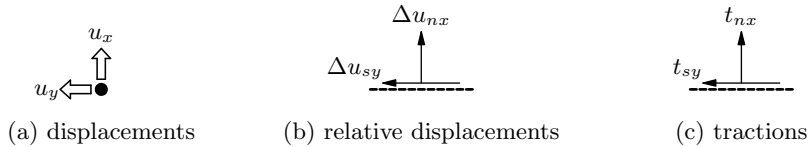


Figure 7.117: Variables of two-dimensional nodal interfaces

Two-dimensional line. Variables of two-dimensional structural interfaces are oriented in the local xy axes.

$$\mathbf{u}_e = \begin{Bmatrix} u_x \\ u_y \end{Bmatrix} \quad \Delta \mathbf{u} = \begin{Bmatrix} \Delta u_{sx} \\ \Delta u_{ny} \end{Bmatrix} \quad \mathbf{t} = \begin{Bmatrix} t_{sx} \\ t_{ny} \end{Bmatrix} \quad (7.127)$$

The normal traction t_{ny} is perpendicular to the interface; the shear traction t_{sx} is tangential to the interface [Fig. 7.118].

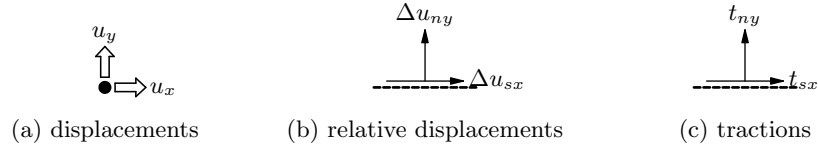


Figure 7.118: Variables of two-dimensional interfaces

Interfaces to shells. Variables of structural interfaces to shell elements are oriented in the local xyz axes. Compared to the two-dimensional interface elements [Eq. (7.127)], these elements additionally have a translational degree of freedom u_z and a rotational degree of freedom ϕ_x which give compatibility to curved shell elements.

$$\mathbf{u}_e = \begin{Bmatrix} u_x \\ u_y \\ u_z \\ \phi_x \end{Bmatrix} \quad \Delta \mathbf{u} = \begin{Bmatrix} \Delta u_{sx} \\ \Delta u_{ny} \\ \Delta u_{sz} \end{Bmatrix} \quad \mathbf{t} = \begin{Bmatrix} t_{sx} \\ t_{ny} \\ t_{sz} \end{Bmatrix} \quad (7.128)$$

The normal traction t_{ny} is perpendicular to the interface; the shear tractions t_{sx} and t_{sz} are tangential to the interface [Fig. 7.119].

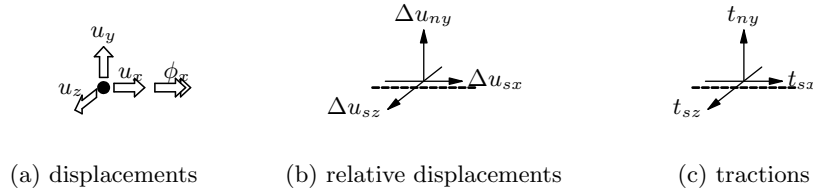


Figure 7.119: Variables for structural interfaces to shells

Line–solid connection. Variables are oriented in the local xyz axes [Fig. 7.120].

$$\mathbf{u}_e = \begin{Bmatrix} u_x \\ u_y \\ u_z \end{Bmatrix} \quad \Delta \mathbf{u} = \begin{Bmatrix} \Delta u_{sx} \\ \Delta u_{ny} \\ \Delta u_{nz} \end{Bmatrix} \quad \mathbf{t} = \begin{Bmatrix} t_{sx} \\ t_{ny} \\ t_{nz} \end{Bmatrix} \quad (7.129)$$

The shear traction t_{sx} is tangential to the interface; the normal tractions t_{ny} and t_{nz} are perpendicular to the interface [Fig. 7.120].

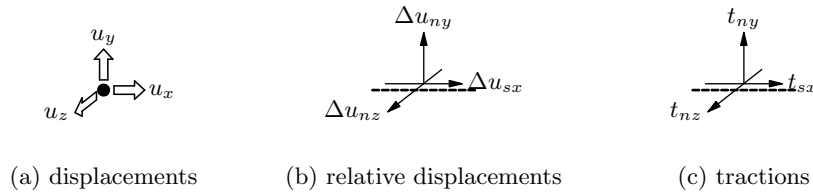


Figure 7.120: Variables of the line–solid interface elements

Three-dimensional nodal. Variables of three-dimensional structural nodal interfaces are oriented in the local xyz axes [Fig. 7.121].

$$\mathbf{u}_e = \begin{Bmatrix} u_x \\ u_y \\ u_z \end{Bmatrix} \quad \Delta \mathbf{u} = \begin{Bmatrix} \Delta u_{nx} \\ \Delta u_{sy} \\ \Delta u_{sz} \end{Bmatrix} \quad \mathbf{t} = \begin{Bmatrix} t_{nx} \\ t_{sy} \\ t_{sz} \end{Bmatrix} \quad (7.130)$$

The normal traction t_{nx} is perpendicular to the interface; the shear tractions t_{sy} and t_{sz} are tangential to the interface [Fig. 7.122].

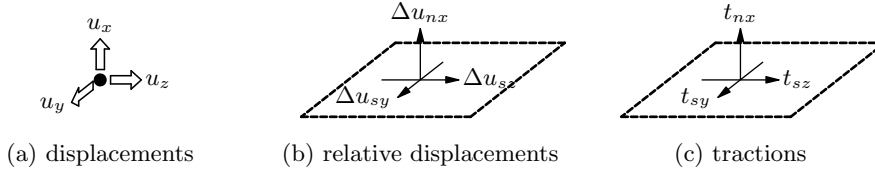


Figure 7.121: Variables of three-dimensional nodal interfaces

Three-dimensional surface. Variables of three-dimensional structural surface interfaces are oriented in the local xyz axes [Fig. 7.122].

$$\mathbf{u}_e = \begin{Bmatrix} u_x \\ u_y \\ u_z \end{Bmatrix} \quad \Delta \mathbf{u} = \begin{Bmatrix} \Delta u_{sx} \\ \Delta u_{sy} \\ \Delta u_{nz} \end{Bmatrix} \quad \mathbf{t} = \begin{Bmatrix} t_{sx} \\ t_{sy} \\ t_{nz} \end{Bmatrix} \quad (7.131)$$

The normal traction t_{nz} is perpendicular to the interface; the shear tractions t_{sx} and t_{sy} are tangential to the interface [Fig. 7.122].

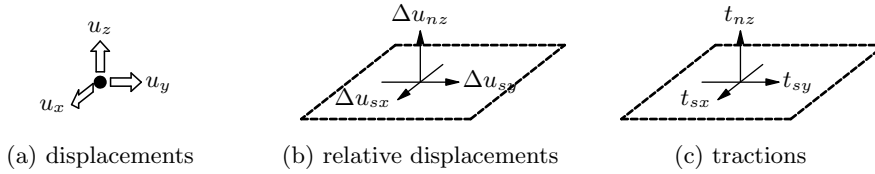


Figure 7.122: Variables of three-dimensional surface interfaces

7.6.3 Element Types

Structural interface element types in DIANA are sets of points, sets of lines, or sets of triangles or quadrilaterals. The lines, triangles and quadrilaterals may have linear, quadratic, or cubic interpolation functions for the displacement field. For elements with linear interpolation functions there are only two nodes at each edge, whereas for elements with quadratic interpolation functions there are three nodes at each edge, and elements with cubic interpolation functions have four nodes along each edge. The next paragraphs [§ 7.6.3.1 to § 7.6.3.14] give an overview of the different interface element types where the numbers display the relative node numbers for the element connectivity. The isoparametric coordinates ξ and η are also displayed for the different interface element types.

7.6.3.1 N4IF – 1+1 nodes, 2-D

The N4IF element is an interface element between two nodes in a two-dimensional con-

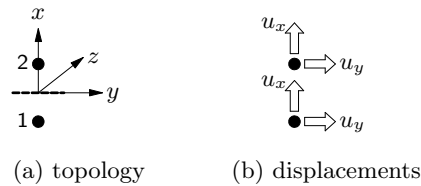


Figure 7.123: N4IF

figuration [Fig. 7.123]. The x and z axes are perpendicular to the interface, the y axis (tangential to the interface) is $\perp xz$. Variables are oriented in the xy axes [Eq. (7.127) p. 83].

7.6.3.2 N6IF – 1+1 nodes, 3-D

The N6IF element is an interface element between two nodes in a three-dimensional

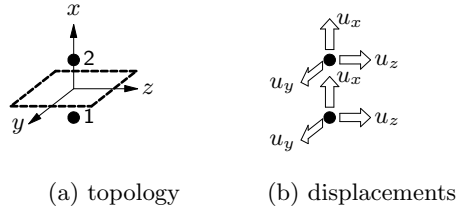


Figure 7.124: N6IF

configuration [Fig. 7.124]. The z axis (tangential to the interface) is $\perp xy$. Variables are oriented in the xyz axes [Eq. (7.131) p. 85].

7.6.3.3 L8IF – line, 2+2 nodes, 2-D

The L8IF element is an interface element between two lines in a two-dimensional configuration

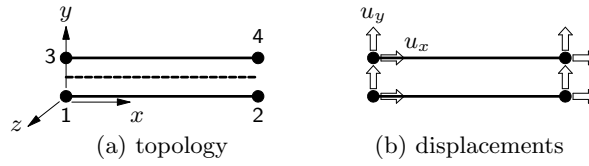


Figure 7.125: L8IF

ration [Fig. 7.125]. The local xy axes for the displacements are evaluated in the first node with x from node 1 to node 2. Variables are oriented in the xy axes [Eq. (7.127) p. 83]. The element is based on linear interpolation. By default DIANA applies a 2-point Newton–Cotes integration scheme. Other suitable options are 2- and 4-point Newton–Cotes, 2- and 3-point Gauss, 2-, 3- and 4-point Lobatto, and a nodal lumping scheme.

[$n_{\xi}=2$]

7.6.3.4 CL12I – line, 3+3 nodes, 2-D

The CL12I element is an interface element between two lines in a two-dimensional configuration

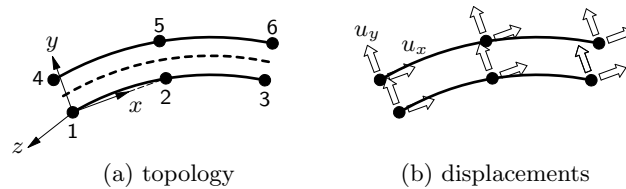


Figure 7.126: CL12I

ration [Fig. 7.126]. The local xy axes for the displacements are evaluated in the first node with x from node 1 to node 2. Variables are oriented in the xy axes [Eq. (7.127) p. 83]. The element is based on quadratic interpolation. By default DIANA applies a 3-point Newton–Cotes integration scheme. Suitable options are 3- and 5-point Newton–Cotes, 2- and 3-point Gauss, 3-, 4- and 5-point Lobatto, and a nodal lumping scheme.

[$n_{\xi}=3$]

7.6.3.5 T18IF – plane triangle, 3+3 nodes, 3-D

The T18IF element is an interface element between two planes in a three-dimensional configuration [Fig. 7.127]. The local xyz axes for the displacements are evaluated in the first node with x from node 1 to node 2 and z perpendicular to the plane. Variables are oriented in the local xyz axes [Eq. (7.131) p. 85]. The element is based on linear interpolation. By default DIANA applies a 3-point integration scheme. Suitable options are 1- and 4-point, and a nodal lumping scheme.

[$n_{lc}=3$]

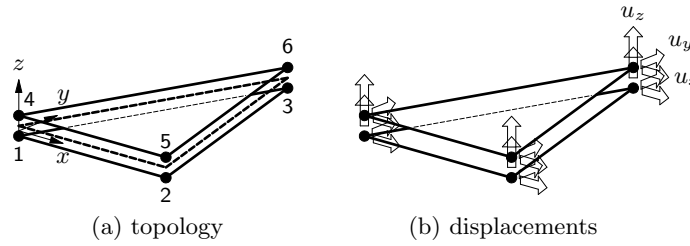


Figure 7.127: T18IF

7.6.3.6 Q24IF – plane quadrilateral, 4+4 nodes, 3-D

The Q24IF element is an interface element between two planes in a three-dimensional

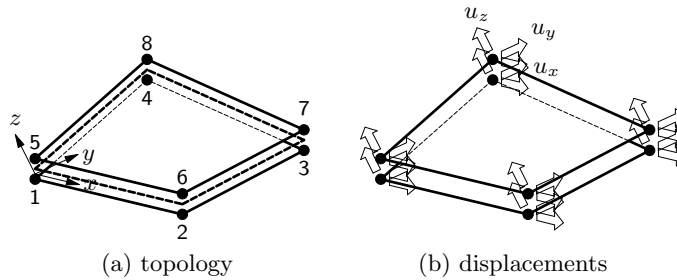


Figure 7.128: Q24IF

configuration [Fig. 7.128]. The local xyz axes for the displacements are evaluated in the first node with x from node 1 to node 2 and z perpendicular to the plane. Variables are oriented in the local xyz axes [Eq. (7.131) p. 85]. The element is based on linear interpolation. By default DIANA applies a 2×2 Newton–Cotes integration scheme. Suitable options are 2×2 and 4×4 Newton–Cotes, 2×2 and 3×3 Gauss, 2×2 , 3×3 and 4×4 Lobatto, and a nodal lumping scheme. [$n_{\xi}=2$, $n_{\eta}=2$]

7.6.3.7 CT36I – plane triangle, 6+6 nodes, 3-D

The CT36I element is an interface element between two planes in a three-dimensional

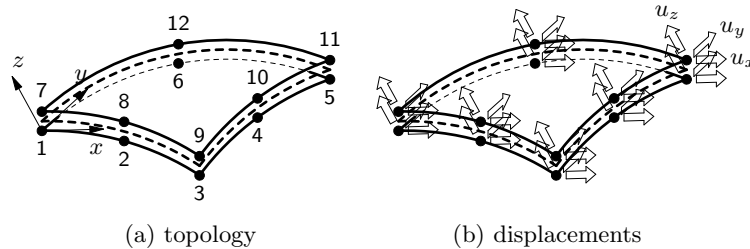


Figure 7.129: CT36I

configuration [Fig. 7.129]. The local xyz axes for the displacements are evaluated in the first node with x from node 1 to node 2 and z perpendicular to the plane. Variables are oriented in the local xyz axes [Eq. (7.131) p. 85]. The element is based on quadratic interpolation. By default DIANA applies a 6-point integration scheme. Alternative schemes are 3- and 4-point, and a nodal lumping scheme. [$n_{lc} = 6$]

7.6.3.8 CQ48I – plane quadrilateral, 8+8 nodes, 3-D

The CQ48I element is an interface element between two planes in a three-dimensional configuration [Fig. 7.130]. The local xyz axes for the displacements are evaluated in the first node with x from node 1 to node 2 and z perpendicular to the plane. Variables are oriented in the local xyz axes [Eq. (7.131) p. 85]. The element is based on quadratic interpolation. By default DIANA applies a 3×3 Newton–Cotes integration scheme. Suitable [$n_{\xi}=3$, $n_{\eta}=3$]

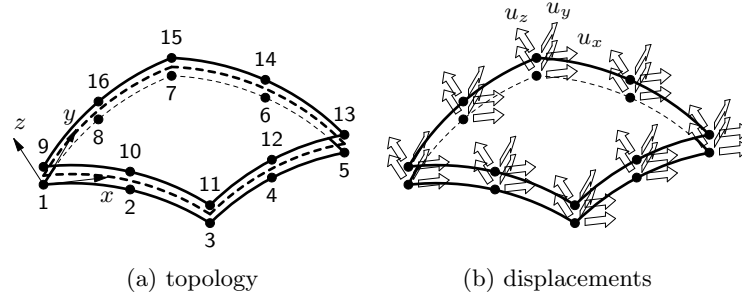


Figure 7.130: CQ48I

options are 3×3 and 5×5 Newton–Cotes, 3×3 and 4×4 Gauss, 3×3 , 4×4 and 5×5 Lobatto, and a nodal lumping scheme.

7.6.3.9 L16IF – line, 2+2 nodes, shell

The L16IF element is an interface element between two lines in a curved shell configuration

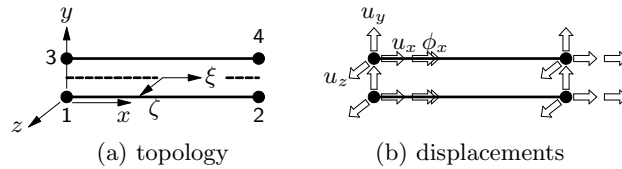


Figure 7.131: L16IF

[Fig. 7.131]. The local xyz axes for the displacements are evaluated with x from node 1 to node 2 and z in the ‘thickness’ direction. Variables are oriented in the local xyz axes [Eq. (7.128) p. 84]. The element is based on linear interpolation. By default DIANA applies a 2-point Newton–Cotes integration scheme in the longitudinal ξ direction and a 3-point Simpson scheme in the ‘thickness’ ζ direction.

[$n_{\xi}=2$]

[$n_{\zeta}=3$]

7.6.3.10 L20IF – line, 3+2 nodes, shell

The L20IF element is an interface element between two lines in a curved shell configuration

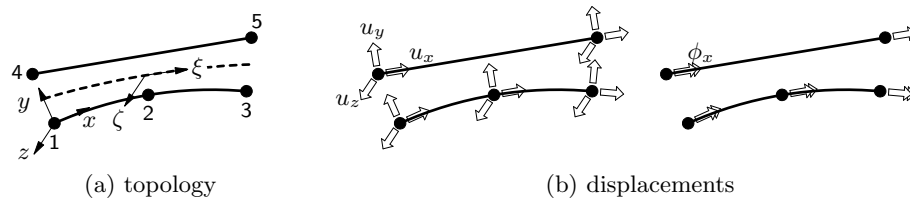


Figure 7.132: L20IF

[Fig. 7.132]. The local xyz axes for the displacements are evaluated in each node with x in the tangential direction and z in the ‘thickness’ direction. Variables are oriented in the local xyz axes [Eq. (7.128) p. 84]. The element is based on quadratic interpolation at the side with three nodes and on linear interpolation at the side with two nodes. Therefore it can connect a shell element with a linear interpolation function (T15SH, Q20SH) to a shell element with a quadratic interpolation function (CT30S, CQ40S).

As the interface behaviour is evaluated in the integration points only, and those integration points are connected with different interpolation to both sides of the element, incompatibilities of the displacements of both sides may occur.

[$n_{\xi}=3$] By default DIANA applies a 3-point Newton–Cotes integration scheme in the longitudinal ξ direction and a 3-point Simpson scheme in the ‘thickness’ ζ direction.

[$n_{\zeta}=3$]

7.6.3.11 CL24I – line, 3+3 nodes, shell

The CL24I element is an interface element between two lines in a curved shell configuration

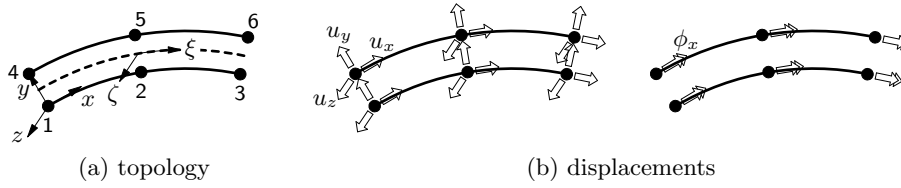


Figure 7.133: CL24I

[Fig. 7.133]. The local xyz axes for the displacements are evaluated in each node with x in the tangential direction and z in the ‘thickness’ direction. Variables are oriented in the local xyz axes [Eq. (7.128) p. 84]. The element is based on quadratic interpolation. By default DIANA applies a 3-point Newton–Cotes integration scheme in the longitudinal ξ direction and a 3-point Simpson scheme in the ‘thickness’ ζ direction. $[n_\xi=3]$
 $[n_\zeta=3]$

7.6.3.12 CL32I – line, 4+4 nodes, shell

The CL32I element is an interface element between two lines in a curved shell configuration

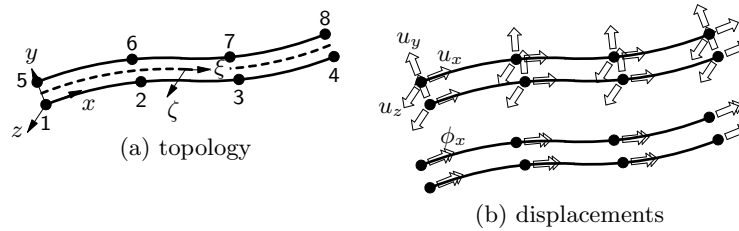


Figure 7.134: CL32I

[Fig. 7.134]. The local xyz axes for the displacements are evaluated in each node with x in the tangential direction and z in the ‘thickness’ direction. Variables are oriented in the local xyz axes [Eq. (7.128) p. 84]. The element is based on third-order interpolation. By default DIANA applies a 4-point Newton–Cotes integration scheme in the longitudinal ξ direction and a 3-point Simpson scheme in the ‘thickness’ ζ direction. $[n_\xi=4]$
 $[n_\zeta=3]$

7.6.3.13 L12IF – line, 2+2 nodes, line–solid connection

The L12IF element is an interface element between a line and an edge of a solid element

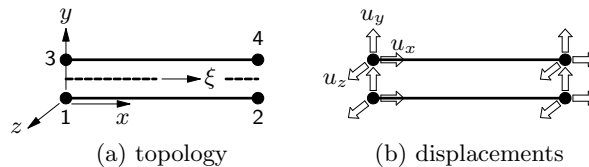


Figure 7.135: L12IF

in three-dimensional configuration [Fig. 7.135]. This element is only applicable in models for three-dimensional bond-slip analysis, typically it represents the bond area between a reinforcement bar and its surrounding material. The local xyz axes for the displacements are evaluated in each node with x in the tangential direction and z in a normal direction. Variables are oriented in the local xyz axes [Eq. (7.129) p. 84]. The element is based on linear interpolation. By default DIANA applies a 2-point Newton–Cotes integration scheme in the longitudinal ξ direction. $[n_\xi=2]$

7.6.3.14 CL18I – line, 3+3 nodes, line–solid connection

The CL18I element is an interface element between a line and an edge of a solid element

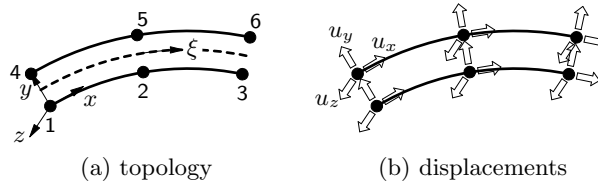


Figure 7.136: CL18I

in three-dimensional configuration [Fig. 7.136]. This element is only applicable in models for three-dimensional bond-slip analysis, typically it represents the bond area between a reinforcement bar and its surrounding material. The local xyz axes for the displacements are evaluated in each node with x in the tangential direction and z in a normal direction. Variables are oriented in the local xyz axes [Eq. (7.129) p.84]. The element is based on quadratic interpolation. By default DIANA applies a 3-point Newton–Cotes integration scheme in the longitudinal ξ direction.

[$n_{\xi}=3$]

7.6.4 Integration Schemes

For all line and plane structural interface elements DIANA performs a numerical integration with an appropriate default scheme. The default number of integration points in the ξ direction is defined in the table above. For different applications you may want to use different integration schemes. For sliding a Gauss scheme may be appropriate, but for discrete cracking it may be better to apply Newton–Cotes or lumped scheme with integration points at the outer ends of the interface where the element may open first. You may choose an alternative scheme via a special data input, depending on the shape of the element. For more details see Volume *Element Library*

7.6.5 Elastic Material Properties

Interface elements in linear structural analysis require input of the linear stiffness: D_{11} sets the relation between the normal traction t_n and the normal relative displacement Δu_n ; D_{22} sets the relation between the shear traction t_t and the shear relative displacement Δu_t ; For plane interfaces with different in plane sliding stiffnesses, D_{33} sets the relation between the shear traction t_s and the shear relative displacement Δu_s . If no stiffness D_{33} is given, it is assumed to be equal to stiffness D_{22} . The dimension of the stiffness moduli is force per area per length, i.e., stress per length, for instance N/mm³. Alternatively nonlinear elastic stiffness properties may be defined. Applications cover nonlinear elastic beddings, e.g. ‘no tension’ beddings with a constant stiffness for compression and zero stiffness for tension. The available nonlinear elastic model sets a multilinear relation between normal traction and perpendicular relative displacement or between the shear traction and the tangential relative displacement or both. Depending on the interface element type orthotropic normal stiffness, or orthotropic shear stiffness can be specified. The diagram domains for the input of the multilinear relations must cover the relative displacement domains used in the computation, i.e. all diagrams could start from negative relative displacements and end with positive relative displacements. The specified relative displacements must be in increasing order. For numerical reasons it is recommended to use a continuous slope at (0,0) in both diagrams which corresponds with the elastic stiffness.

7.7 Composed Elements

The main application of composed elements is in post-processing of analysis results. There are two classes of composed elements: composed line elements and composed surface elements.

In composed line elements the local forces and bending moments in a cross-section of the model can be calculated with reference to a line defined with composed line elements. For this purpose the primary Cauchy stresses or internal forces in the regular elements are integrated over the cross-section plane normal to the reference line. All types of regular elements and embedded reinforcements contribute to the cross-section forces and bending moments in a composed line element.

In composed surface elements the distributed local forces and bending moments along a line normal to a reference surface is calculated from the primary Cauchy stresses in solid elements and reinforcements located in solid elements that are intersected by the normal line.

Composed results are presented with reference to composed elements.

Composed elements do not have mechanical properties, such as stiffness or mass, by their own. Therefore, these elements do not influence the behaviour of the finite element model.

The mesh of regular elements may be unstructured and there are no topological constraints with the composed line or surface elements.

7.7.1 Element Axes

For composed line elements in general there is no unique element plane and DIANA needs a \bar{z} direction to set up the element axes. The specified direction of the \bar{z} axis may not be in the direction of the composed line element axis (tangent in the first node). From the \bar{z} axis, DIANA determines in each node an x axis tangential to the beam axis. Next the $y \perp \bar{z}x$ is set up and finally $z \perp xy$ is set up which is in fact a correction of the \bar{z} direction. If the x axis coincides with the model Z direction then $y \perp \bar{z}x$ can not be done and the \bar{y} axis is chosen in model Y direction. In that case the element xyz axes are set up as follows: x in the direction of the beam axis, $z \perp x\bar{y}$ and $y \perp zx$.

For the quadratic composed line element, you may specify a \bar{z} axis for each node by repeating the x , y and z components. The xyz axes of composed line elements serve to describe the direction of forces and bending moments. The xyz directions are transformed locally to $\hat{x}\hat{y}\hat{z}$ in such a way that the \hat{x} axis is tangential to the beam axis and that \hat{y} and \hat{z} are perpendicular to it.

For composed surface elements DIANA needs no special user input data to set up the element axes. By default, a local element \bar{x} axis points from the first to the second node of the element [Fig. 7.137a]. DIANA uses the \bar{x} axis to set up xyz axes in the nodes. The

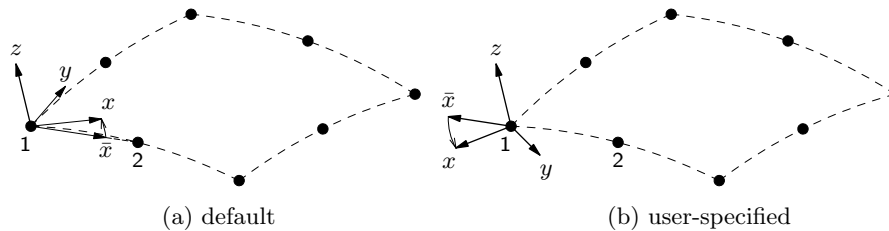


Figure 7.137: Axes

z axes are always set up perpendicular to the element plane, y is created $\perp z\bar{x}$ and finally $x \perp yz$ according to the right-hand-rule. Note that the default element axes are fully independent of the global model XYZ axes.

If you prefer x axes other than default, for instance to get the same axes directions for various elements, then you must specify an \bar{x} axis which DIANA uses to set up the real x axes [Fig. 7.137b]. First the z axis is put perpendicular to the element plane. Then $y \perp z\bar{x}$ is created and finally $x \perp yz$.

7.7.2 Forces and Bending Moments for Composed Line Elements

For composed line elements the calculated forces and bending moments are defined similar as for three-dimensional beams :

$$\mathbf{N} = \begin{Bmatrix} N_x \\ Q_y \\ Q_z \end{Bmatrix} \quad \mathbf{M} = \begin{Bmatrix} M_x \\ M_y \\ M_z \end{Bmatrix} \quad \boldsymbol{\sigma} = \begin{Bmatrix} \sigma_{xx} \\ \sigma_{xy} = \sigma_{yx} \\ \sigma_{zx} = \sigma_{xz} \end{Bmatrix} \quad (7.132)$$

Figure 7.138 shows the primary moments and forces in a cross-section of an infinitesimal part of a composed line element in their positive direction. The sign convention for

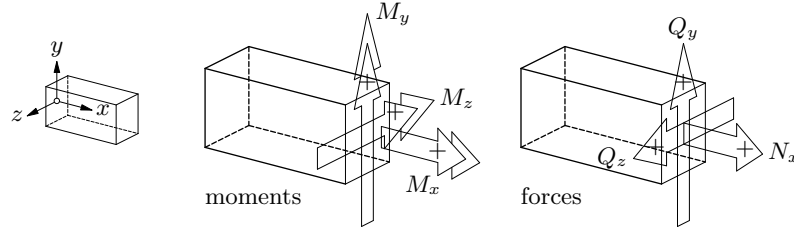


Figure 7.138: Moments and forces for composed line elements

bending is that a positive bending moment yields a positive stress (tension) in the positive area (+ M_z works in the $-z$ direction!). The sign convention for torsion is that a positive torsion moment yields $+\sigma_{zx}$ in the $+y$ area. The sign convention for forces is that a positive force yields a positive stress.

7.7.3 Distributed Forces and Distributed Bending Moments

For compose surface elements the calculated distributed forces and bending moments are defined similar as for shell elements:

$$\mathbf{m} = \begin{Bmatrix} m_{xx} \\ m_{yy} \\ m_{xy} = m_{yx} \end{Bmatrix} \quad \mathbf{f} = \begin{Bmatrix} n_{xx} \\ n_{yy} \\ n_{xy} = n_{yx} \\ q_{xz} \\ q_{yz} \end{Bmatrix} \quad (7.133)$$

The sign convention is that a positive moment yields positive stresses in the upper plane

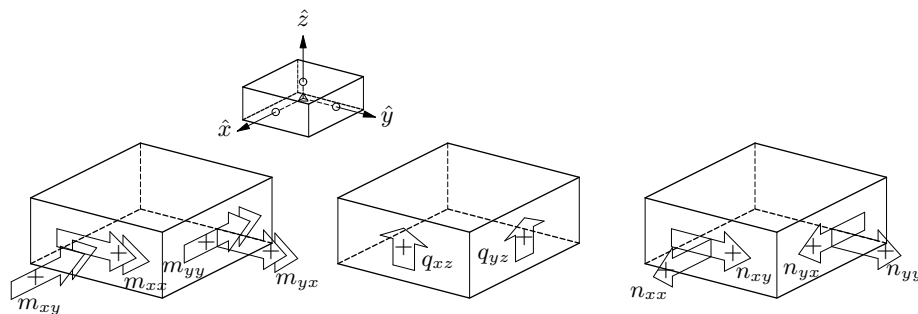


Figure 7.139: Generalized moments and forces

and that a positive shear force yields positive shear stresses.

In the model selection commands the user may define selection of composed elements, of elements, and of reinforcements. The following rules apply to the selection of model parts for composed element results: Only for selected composed elements results will be calculated and output. When elements are selected in the same selection, only the selected elements will contribute to the composed element results. Reinforcements located in non-selected elements will not contribute to composed element results. Selection of reinforcements, will be neglected, except when the command **REINFO NONE** is given. In that case reinforcements will not contribute to the composed element results.

7.7.4 Element Types

Composed element types in DIANA can be lines, triangles or quadrilaterals. The elements may have linear or quadratic interpolation functions. For elements with linear interpolation functions there are only two nodes at each edge, whereas for elements with quadratic interpolation functions there are three nodes at every edge. For triangles and quadrilaterals also elements with cubic interpolations functions, with four nodes at each edge, are available. The next paragraphs [§ 7.7.4.1 to § 7.7.4.8] give an overview of the different composed element types where the numbers display the relative node numbers for the element connectivity. The isoparametric coordinates ξ and η are also displayed for the different interface element types.

7.7.4.1 L2CMP – straight composed line, 2 nodes

The L2CMP element is a two-node straight line base element. No material data needs to

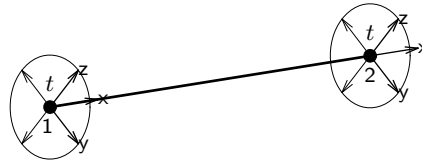


Figure 7.140: L2CMP

be defined for composed line elements. It is compulsory to define a local \bar{z} axis to define a unique element plane. The local element axes are defined similar to those of beam elements [§ 7.5.1 p. 72]. Optionally, a thickness can be defined to set a bandwidth for the location of structural nodes to be included in the composition.

7.7.4.2 CL3CM – curved composed line, 3 nodes

The CL3CM element is a three-node curved line base element. No material data needs to

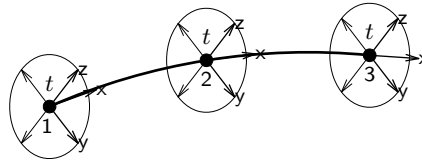


Figure 7.141: CL3CM

be defined for composed line elements. It is compulsory to define a local \bar{z} axis to define a unique element plane. The local element axes are defined similar to those of beam elements [§ 7.5.1 p. 72]. Optionally, a thickness can be defined to set a bandwidth for the location of structural nodes to be included in the composition.

7.7.4.3 T3CMP – triangular composed surface, 3 nodes

The T3CMP element is a three-node triangular composed surface element [Fig. 7.142].

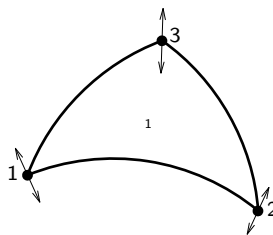


Figure 7.142: T3CMP

7.7.4.4 Q4CMP – quadrilateral composed surface, 4 nodes

The Q4CMP element is a four-node quadrilateral composed surface element [Fig. 7.143].

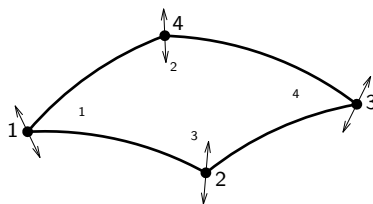


Figure 7.143: Q4CMP

7.7.4.5 CT6CM – triangular composed surface, 6 nodes

The CT6CM element is a six-node triangular curved composed surface element [Fig. 7.144].

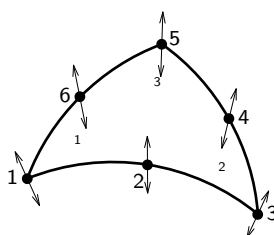


Figure 7.144: CT6CM

7.7.4.6 CQ8CM – quadrilateral composed surface, 8 nodes

The CQ8CM element is an eight-node quadrilateral curved composed surface element

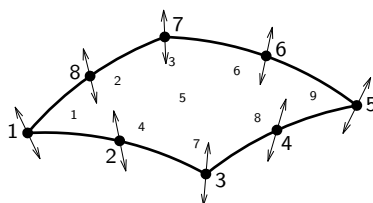


Figure 7.145: CQ8CM

[Fig. 7.145].

7.7.4.7 CT9CM – triangular composed surface, 9 nodes

The CT9CM element is a nine-node triangular curved composed surface element [Fig. 7.146].

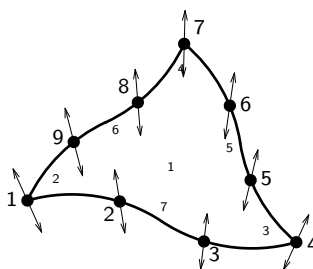


Figure 7.146: CT9CM

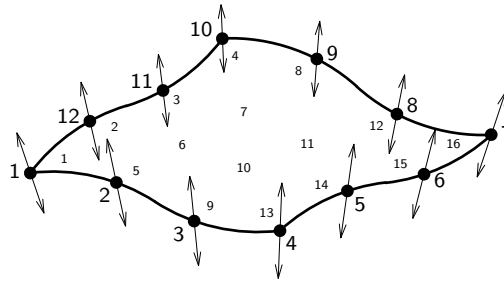


Figure 7.147: CQ12C

7.7.4.8 CQ12C – quadrilateral composed surface, 12 nodes

The CQ12C element is a twelve-node quadrilateral curved composed surface element [Fig. 7.147].

7.7.5 Thickness of Composed Line Elements

The thickness of a composed line element determines which nodes of structural elements become part of the composition: an element node will only be included in the composition if the node is positioned within the thickness in radial direction. If you do not specify it then DIANA assumes that the composed line element may extend to infinity in radial direction.

7.7.6 Thickness and Shape of Composed Surface Elements

DIANA applies the shape of the composed surface element to determine the direction in which the ‘thickness’ is measured, by setting up a normal vector in each node of the base element. There are some predefined shapes for composed surface elements which you may specify with the element properties. If you do so, then DIANA checks whether the coordinates of the element nodes fit to the specified shape and produces an error message if this check fails. The shape and thickness of a composed surface element determine

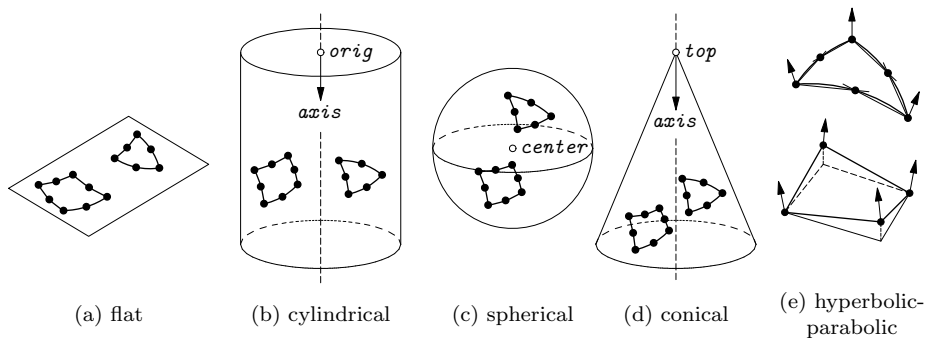


Figure 7.148: Predefined shapes

which solid elements become part of the composed elements: a solid element will only contribute to a composed surface element if all of its nodes are positioned within the thickness of the composed surface element. For this algorithm DIANA will assume that the thickness spans symmetrically along the two sides of the base element, i.e., half the thickness at each side. Unlike for the curved shells, the specification of the thickness is optional. If you do not specify it then DIANA assumes that the composed element may extend to infinity at both sides of the composed surface element.

7.8 Spring and Mass Elements

DIANA offers several types of spring and mass elements:

- Discrete spring/dashpots

These elements may not only serve as springs, but also as dashpots to simulate continuous damping on specific locations in the finite element model. Two-node spring/dashpot elements may be used to model the interaction between two points of the finite element model [Fig. 7.149a]. If one node of such a spring/dashpot

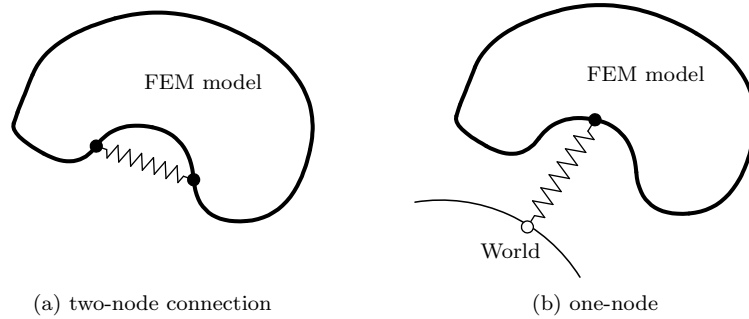


Figure 7.149: Discrete springs

element is fully supported, then the element models the interaction between the finite element model and its environment (the world). For the latter application, it is more convenient to use the one-node spring/dashpot elements, because it is not necessary to specify the support conditions [Fig. 7.149b]. There are separate one-node and two-node spring/dashpots to model *translation* or *rotation*.

For linear static analysis, the spring constant must be specified. For nonlinear analysis, various models for the spring stiffness, can be specified. For dynamic analysis, you may input either the spring stiffness or the damping coefficient, or both. For translation springs you may specify nonlinear elasticity via a force–elongation diagram [Fig. 7.150]. Please note the following:

To prevent ambiguity, the specified diagram must be monotonic, increasing or decreasing.

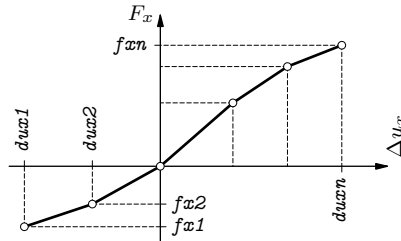


Figure 7.150: Force–elongation diagram for spring elements

For one-node spring/dashpots the input of the axis direction is obligatory. For two-node spring/dashpots DIANA will setup an axis in the direction from the first to the second node unless you explicitly specify an axis direction.

- Matrix spring elements

A matrix spring element can be either a one-node element [§ 7.8.1.5] or a two-node element [§ 7.8.1.6] with three translational and three rotational degrees of freedom. A coupling between the six degrees of freedom can be defined by the force versus displacement stiffness matrix. These elements are particular suited to model discrete beam behaviour or connections between beam elements and between beam and shell elements. This stress–strain relation is supposed to be symmetric and positive-definite. Therefore, only the upper half of the matrix has to be specified.

Afterwards the condition of the matrix is checked to be positive-definite.

$$\begin{Bmatrix} f_x \\ f_y \\ f_z \\ m_x \\ m_y \\ m_z \end{Bmatrix} = \begin{bmatrix} k_{11} & k_{12} & k_{13} & k_{14} & k_{15} & k_{16} \\ k_{12} & k_{22} & k_{23} & k_{24} & k_{25} & k_{26} \\ k_{13} & k_{23} & k_{33} & k_{34} & k_{35} & k_{36} \\ k_{14} & k_{24} & k_{34} & k_{44} & k_{45} & k_{46} \\ k_{15} & k_{25} & k_{35} & k_{45} & k_{55} & k_{56} \\ k_{16} & k_{26} & k_{36} & k_{46} & k_{56} & k_{66} \end{bmatrix} \cdot \begin{Bmatrix} u_{xx} \\ u_{yy} \\ u_{zz} \\ \phi_x \\ \phi_y \\ \phi_z \end{Bmatrix} \quad (7.134)$$

For matrix springs the user must define a x axis and a y axis. From the local x

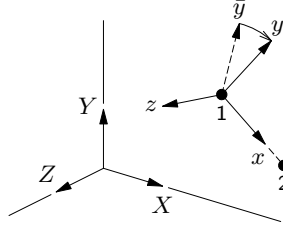


Figure 7.151: Local element axes for matrix spring elements

axis DIANA will setup the local y and z axes as follows. DIANA will setup the local z axis perpendicular to the local x axis and the specified provisional \bar{y} axis. Then, the definitive local y axis is setup perpendicular to z and x [Fig. 7.151].

- Base spring elements

Base spring elements are typically applied to model springy translational and rotational behaviour of supports in frame structures. These elements require the specification of the nonlinear rotational stiffness K_{ϕ_z} depending on the axial force F_x . The following stiffness properties must be defined for these elements:

- Linear spring stiffness for each translational and rotational degree of freedom.
- Ultimate value of the axial force: if the calculated axial force F_x becomes greater than $f d$ then DIANA will assume that all spring stiffnesses drop to zero [Fig. 7.152].

[$f d = 0$]

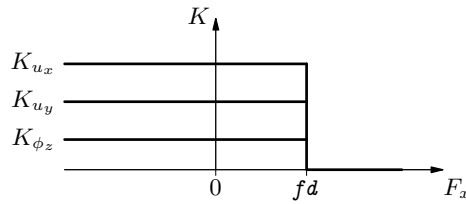


Figure 7.152: Spring stiffnesses depending on axial force (two-dimensional)

- Axial forces $F_x^{(i)}$, in increasing order, for which an M_z - ϕ_z relation diagram is specified (the nonlinear rotational stiffness K_{ϕ_z}), where m is the total number of specified diagrams [Fig. 7.153]. These forces must be less than the ultimate value and usually are negative, i.e., compression forces. ($f_1 < f_2 < \dots < f_m$)
- Rotations $\phi_z^{(j)}$ for which values M_z in the diagrams are specified, where n is the number of specified points in each diagram. ($f < f d$)
- Moment values of $M_z^{(i,j)}$ for each point in all the specified M_z - ϕ_z diagrams: first n values for the first diagram ($i = 1$), then n values for the second diagram ($i = 2$) and so on until the last n values for the last diagram ($i = m$).

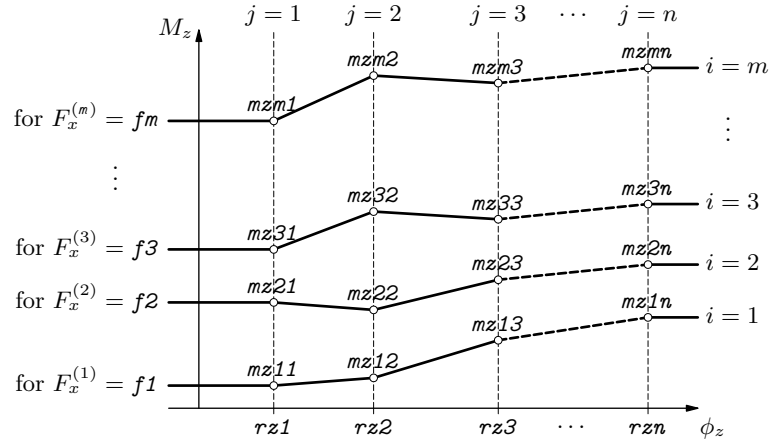


Figure 7.153: Input and application of nonlinear stiffness K_{ϕ_z} (two dimensional)

Figure 7.153 shows how DIANA will apply and interpret the specified diagrams. Diagrams for forces F_x in between the specified forces $F_x^{(i)}$ will be determined by linear interpolation. The first diagram ($i = 1$) will be applied for forces $F_x \leq f1$, and the last diagram ($i = m$) for forces $F_x \geq fm$. For each diagram M_z values for a particular ϕ_z will be determined by linear interpolation. Note that M_z will be assumed to be constant outside the ϕ_z range of the specified diagrams.

For base spring elements the axes are defined in the similar as for matrix springs.

- Point mass/dashpot elements These point elements may be applied to add mass or damping to the finite element model without influencing the stiffness. The point elements do not have any postanalysis results like strains or stresses. As point mass, these elements are typically used to correct the dead weight or to affect the inertia mass in a dynamic analysis. The element axes for point masses are defined in global directions.

7.8.1 Element Types

7.8.1.1 SP1TR – translation, 1 node

The SP1TR element is a one-node translation spring/dashpot [Fig. 7.154]. Basic variables

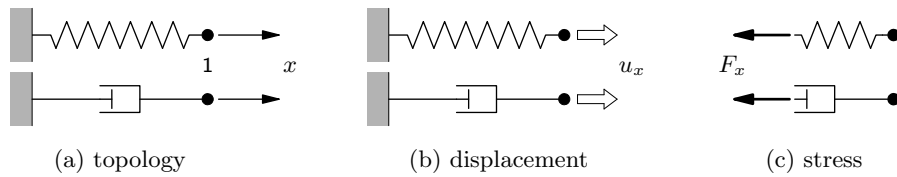


Figure 7.154: SP1TR

of the SP1TR element are the translation, the elongation and the axial force.

$$\mathbf{u}_e = \{ u_x \} \quad \boldsymbol{\varepsilon} = \{ \Delta u_x \} \quad \boldsymbol{\sigma} = \{ F_x \} \quad (7.135)$$

7.8.1.2 SP2TR – translation, 2 nodes

The SP2TR element is a two-node translation spring/dashpot [Fig. 7.155]. Basic variables of the SP2TR element are the translations, the elongation and the axial force.

$$\mathbf{u}_e = \{ u_x \} \quad \boldsymbol{\varepsilon} = \{ \Delta u_x \} = \{ u_x^{(2)} - u_x^{(1)} \} \quad \boldsymbol{\sigma} = \{ F_x \} \quad (7.136)$$

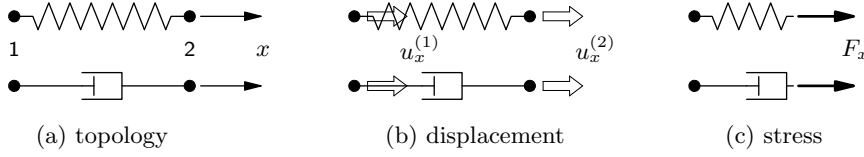


Figure 7.155: SP2TR

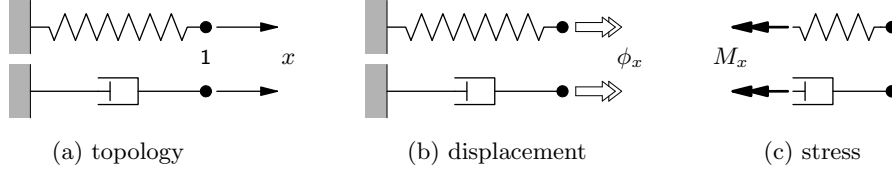


Figure 7.156: SP1RO

7.8.1.3 SP1RO – rotation, 1 node

The SP1RO element is a one-node rotation spring/dashpot [Fig. 7.156]. Basic variables of the SP1RO element are the rotation, the torsion and the axial moment.

$$\mathbf{u}_e = \{ \phi_x \} \quad \boldsymbol{\varepsilon} = \{ \Delta \phi_x \} \quad \boldsymbol{\sigma} = \{ M_x \} \quad (7.137)$$

7.8.1.4 SP2RO – rotation, 2 nodes

The SP2RO element is a two-node torsion spring/dashpot [Fig. 7.157]. Basic variables of

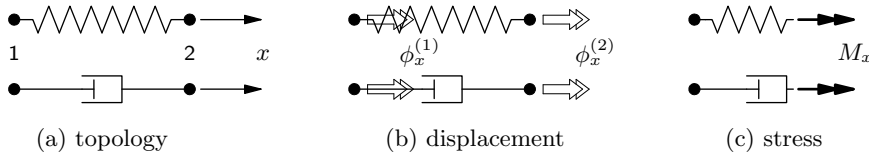


Figure 7.157: SP2RO

the SP2RO element are the rotations, the torsion and the axial moment.

$$\mathbf{u}_e = \{ \phi_x \} \quad \boldsymbol{\varepsilon} = \{ \Delta \phi_x \} = \{ \Delta \phi_x = \phi_x^{(2)} - \phi_x^{(1)} \} \quad \boldsymbol{\sigma} = \{ M_x \} \quad (7.138)$$

7.8.1.5 N6SPR – nodal spring, 1 node, 3-D

The N6SPR element [Fig. 7.158] is a one-node directly integrated generic spring element

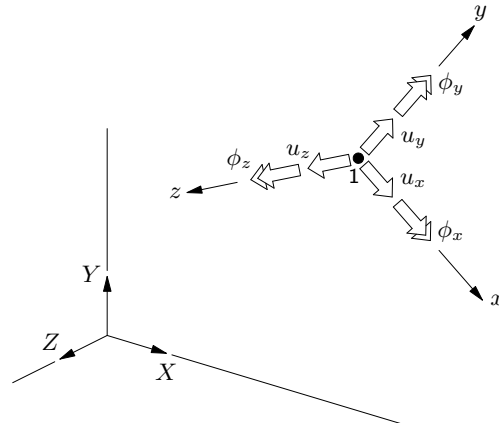


Figure 7.158: N6SPR

which can be used in three-dimensional models. This element is denoted as a matrix

spring element. The basic variables in the node of the N6SPR element are the x , y and z translations and rotations.

$$\mathbf{u}_e = \begin{Bmatrix} u_x \\ u_y \\ u_z \\ \phi_x \\ \phi_y \\ \phi_z \end{Bmatrix} \quad (7.139)$$

7.8.1.6 N12SPR – matrix spring, 2 node, 3-D

The N12SPR element [Fig. 7.159] is a two-node directly integrated generic spring element

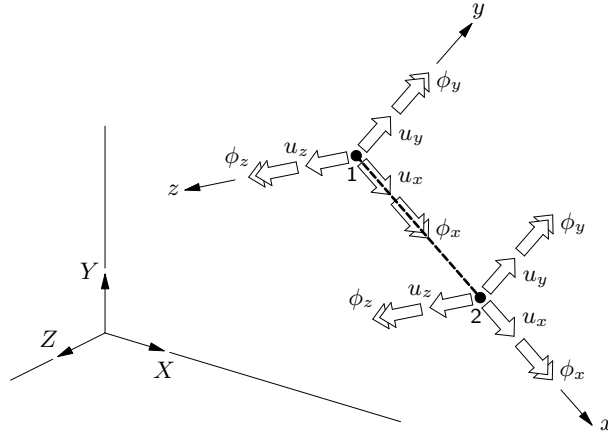


Figure 7.159: N12SPR

which can be used in three-dimensional models. This element is denoted as a matrix spring element. The basic variables in the node of the N12SPR element are the x , y and z translations and rotations.

$$\mathbf{u}_e = \begin{Bmatrix} u_x \\ u_y \\ u_z \\ \phi_x \\ \phi_y \\ \phi_z \end{Bmatrix} \quad (7.140)$$

7.8.1.7 SP6BA – base spring, 2 nodes, 2-D

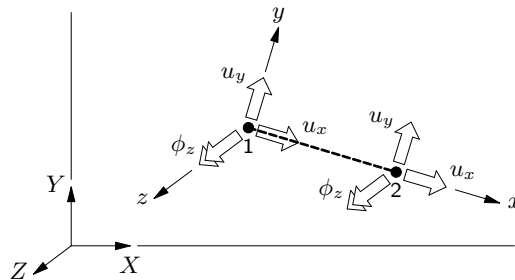


Figure 7.160: SP6BA

The SP6BA element [Fig. 7.160] is a two-node directly integrated base spring element which can only be used in two-dimensional models. The basic variables in the nodes of the SP6BA element are the x and y translations and the z rotation.

$$\mathbf{u}_e = \begin{Bmatrix} u_x \\ u_y \\ \phi_z \end{Bmatrix} \quad (7.141)$$

7.8.1.8 SP12BA – base spring, 2 nodes, 3-D

The SP12BA element [Fig. 7.161] is a two-node directly integrated base spring element

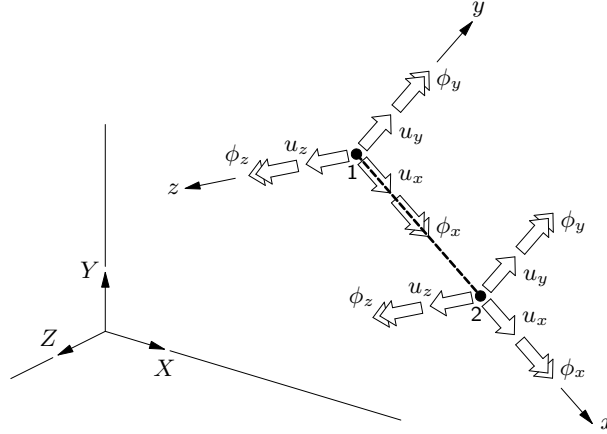


Figure 7.161: SP12BA

which can be used in three-dimensional models. The basic variables in the nodes of the SP12BA element are the x , y and z translations and rotations.

$$\mathbf{u}_e = \begin{Bmatrix} u_x \\ u_y \\ u_z \\ \phi_x \\ \phi_y \\ \phi_z \end{Bmatrix} \quad (7.142)$$

7.8.1.9 PT3T – translation, point mass/damping, 1 node

Translation mass can be used for dead weight in a linear analysis or as inertia mass in a dynamic analysis.

The stiffness at the node of a translation mass should be supplied by other elements. Thus, the point mass should be used in combination with three-dimensional elements. Otherwise, the direction without stiffness should be supported.

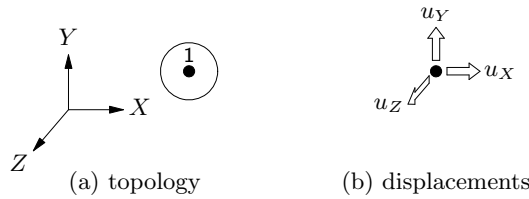


Figure 7.162: PT3T

The PT3T element is a one-node translation mass/damping element [Fig. 7.162], it acts as a concentrated mass in the finite element model. The basic (and only) variables of the PT3T element are the translations in the global XYZ directions.

$$\mathbf{u}_e = \begin{Bmatrix} u_X \\ u_Y \\ u_Z \end{Bmatrix} \quad (7.143)$$

7.8.1.10 PT3RO – rotation, point mass/damping, 1 node

Rotational mass can be used as inertia mass in a dynamic analysis.

The rotational stiffness at the node of a rotational mass element should be supplied by other elements. Otherwise, the direction without rotational stiffness should be supported.

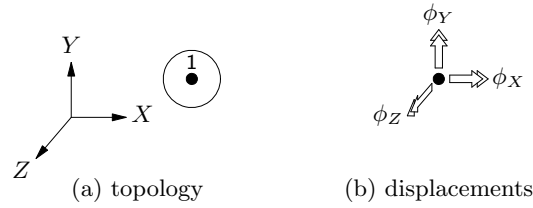


Figure 7.163: PT3RO

The PT3RO element is a one-node rotational mass element[Fig. 7.163], it acts as a concentrated rotation inertia in the finite element model and is only effective in a dynamic analysis. The basic (and only) variables of the PT3RO element are the rotations in the global XYZ directions.

$$\mathbf{u}_e = \begin{Bmatrix} \phi_X \\ \phi_Y \\ \phi_Z \end{Bmatrix} \quad (7.144)$$

7.9 Reinforcements

Reinforcements add stiffness to the finite element model. Reinforcements may be used to model steel reinforcements in concrete or ground anchors or pile foundations in soil.

In DIANA a reinforced concrete structure can be modeled by plain concrete elements and steel reinforcement bars. When the steel reinforcement is composed of a series of bars which are located at a fixed intermediate distance from each other, this can be modeled as a reinforcement grid. Bars in a grid may be located all in the same direction or being oriented in two orthogonal directions.

The elements and the reinforcements can be defined independent from each other, each with reference to their own geometry definition.

To the concrete elements different nonlinear material properties will be assigned than to the steel reinforcements. The concrete material model shall account for cracking failure under tensile stresses and crushing failure at compressive and shear stresses. The steel reinforcements can be modeled with Von Mises type elasto-plastic material models with user-defined hardening.

By default, the reinforcements are fully embedded in the elements in which they are located and are fully coupled, i.e. they do not allow relative slip. This type of reinforcements is called *embedded reinforcement* [§ 7.9.1].

Next to the embedded reinforcements, DIANA offers reinforcements for which the deformation of the reinforcements may be different than for the elements in which they are located, i.e. relative slip is allowed. These reinforcements are commonly used to model *bond-slip reinforcements* and *pile foundations* [§ 7.9.2 p. 110]. For this type, the reinforcements are connected with interface elements to the continuum elements in which they are located.

7.9.1 Embedded Reinforcements

The main characteristics of embedded reinforcements are:

- Reinforcements are embedded in structural elements, the so-called *mother elements*.

DIANA ignores the space occupied by an embedded reinforcement. The mother element neither diminishes in stiffness, nor in weight. The embedded reinforcement does not contribute to the weight (mass) of the element.

- Embedded reinforcements do not have degrees of freedom of their own.
- In embedded reinforcements the strains in the reinforcements are computed from the displacement field of the mother elements. This implies *perfect bond* between the reinforcement and the surrounding material. However, with an additional input option you can specify that the reinforcement is not bonded to the embedding elements [Vol. *Material Library*].

7.9.1.1 Basic Assumptions

Embedded reinforcements add stiffness to the finite element model. The main characteristics of embedded reinforcements are:

- Reinforcements are embedded in structural elements, the so-called *mother elements*.
- Standard reinforcements do not have degrees of freedom of their own.
- By default, reinforcement strains are computed from the displacement field of the mother elements. This implies *perfect bond* between the reinforcement and the surrounding material. Optionally, however, you can specify that the reinforcement is not bonded to the embedding elements.

The technique of embedding allows the lines of the reinforcement to deviate from the lines of the mesh. This permits the user to generate the finite element mesh without having to anticipate on the location of reinforcements.

A reinforcement is defined by the following information:

- Location in the model.
- Material properties.
- Physical properties, i.e. cross-section area.
- Integration schemes.

Special loadings can be applied to reinforcements.

DIANA offers two types of embedded reinforcements:

- Reinforcement bars

Reinforcement bars may be embedded in various families of elements: beams, plane stress (except three-dimensional plane stress elements), curved shell and solid. In finite element models with these elements, bar reinforcements have the shape of a line.

Bars may also be embedded in plane strain and axisymmetric elements where they have the shape of a point.

The total length of the bar is considered to be divided in several *particles* [Fig. 7.164-a]. By definition, a particle must be completely inside a structural element. The

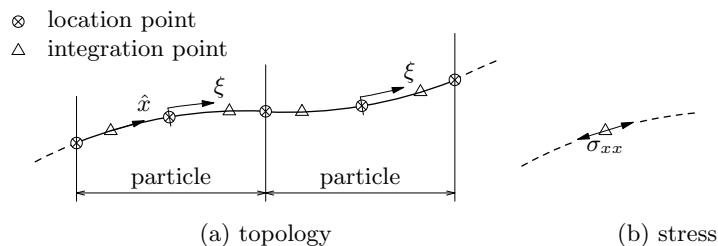


Figure 7.164: Reinforcement bar

so-called *location points* define the position of the particles in the finite element

model. Some location points are the intersections of the bar with the element boundaries. Other location points are in-between these intersections, these points define the curvature of the bar.

Usually, the location points are determined automatically by DIANA from input of larger *sections*; this process is called *preprocessing* of reinforcement location. In some cases it may be useful to specify the location points explicitly, which we call *element-by-element* input.

DIANA performs numerical integration of each particle of a reinforcement bar. In each integration point DIANA determines a vector tangential to the bar axis. The variables for a bar reinforcement are the strains ε_{xx} and the stresses σ_{xx} oriented in the direction of the vector.

- Reinforcement grids

Plane shaped reinforcement grids may be embedded in various families of elements: plane stress (except three-dimensional plane stress elements), curved shell and solid. Depending on the element family and the specified location points, the plane of the grid may be curved or flat.

Grids may also be embedded in plane strain or axisymmetric elements where they have the shape of a line.

The total area of the grid is considered to be divided in several *particles* [Fig. 7.165]. Each particle contributes to the stiffness of the element that embeds it. The definition of a particle depends on the dimensionality of the embedding structural element. Two-dimensional elements may be fully or partly covered by one or more

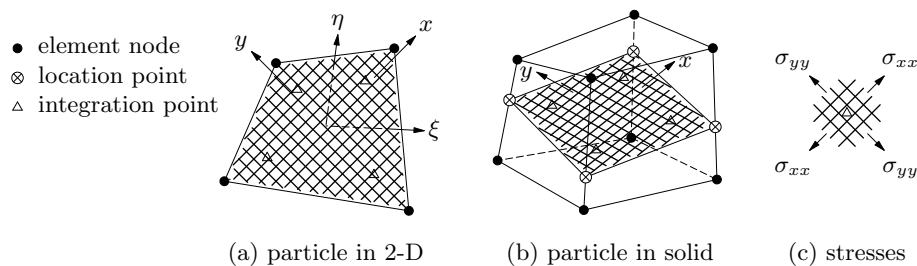


Figure 7.165: Reinforcement grid

particles of grid reinforcement. Solid elements embed a particle of grid reinforcement completely. The so-called location points define the position of the particles in the finite element model.

Usually, the embedding elements (and for solids the location points) are determined automatically by DIANA from input of larger *sections*; this process is called *preprocessing* of reinforcement location. In some cases it may be useful to specify the elements (and location points) explicitly, which we call *element-by-element* input.

7.9.1.2 Bar Reinforcement in Beam

Bar reinforcement can be embedded in beam elements of class-II and class-III.

Class-I beam elements cannot embed bar reinforcement.

DIANA assumes that a beam element embeds a particle of a reinforcement bar, if the bar intersects two different faces of the beam's envelope.

The envelope of a beam element is defined by six faces: two end faces and four lateral faces [Fig. 7.166a]. The end faces are the smallest possible rectangles (in local yz axes) that enclose the beam cross-section at the end nodes. For example, the envelope end face of a T-shape beam, is a rectangle with dimensions $b \times h$. Figure 7.166b shows two examples of embedded particles: the first one [b-top] with intersection points in the two

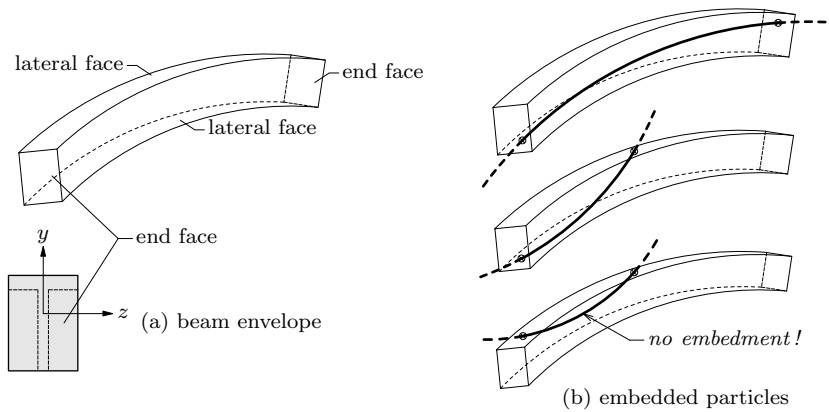


Figure 7.166: Bar reinforcement in beam element

end faces, the second one [b-mid] with an intersection in an end face and in a lateral face. The bottom of Figure 7.166b shows a particle with two intersection points in the same (lateral) face of the envelope, *this particle is not embedded in the beam element!*

If envelopes of different beam elements have some space in common, then some particles of a reinforcement bar may be multiply embedded!

To avoid multiply embedding of bars in beam elements, you may use element-by-element input.

7.9.1.3 Bar Reinforcement in Plane Stress Elements

Bar reinforcement can be embedded in all plane stress elements. Plane stress elements are automatically checked for embedding of bar reinforcements specified with sections.

A plane stress element embeds a particle of a bar section if it intersects one or two element edges, but none of them more than once.

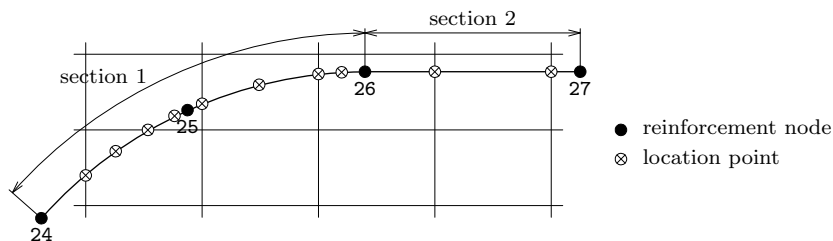


Figure 7.167: Bar reinforcement in plane stress element

7.9.1.4 Bar Reinforcement in Plane Strain Elements

Reinforcement bars may be embedded in all regular plane strain elements. Due to the nature of plane strain element models, the bars show up in the XY plane as points.

7.9.1.5 Bar Reinforcement in Axisymmetric Elements

Reinforcement bars may be embedded in all axisymmetric regular solid ring elements. Due to the nature of axisymmetry the bars are ring-shaped and show up in the XY plane as points.

7.9.1.6 Bar Reinforcement in Curved Shell Elements

Bar reinforcement can be embedded in all curved shell elements. Curved shell elements are automatically checked for embedding of bar reinforcements specified with sections.

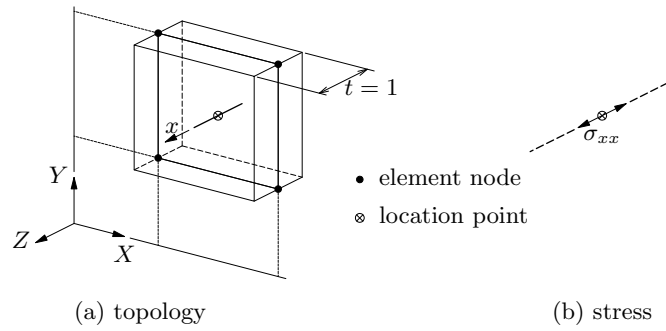


Figure 7.168: Bar in plane strain element

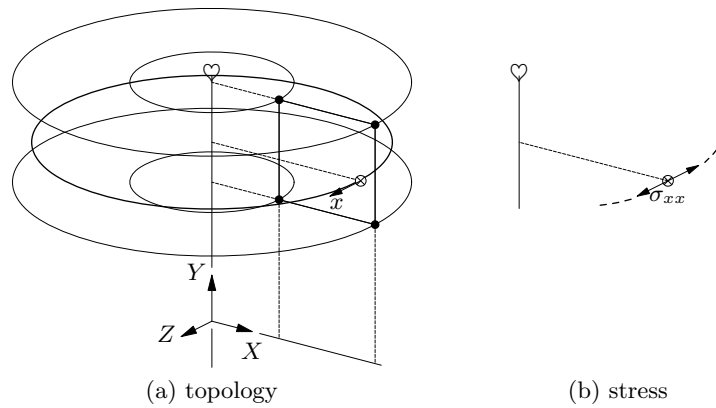


Figure 7.169: Bar in axisymmetric element

There are two conditions for a bar section to be embedded in a curved shell element: (1) it must intersect one or two element edges, but none of them more than once; (2) the computed location points must be inside the thickness domain of the element.

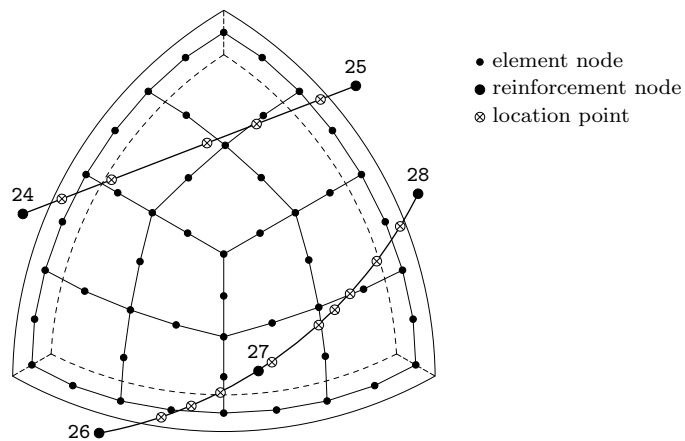


Figure 7.170: Bar in curved shell element

7.9.1.7 Bar Reinforcement in Solid Elements

Bar reinforcement can be embedded in all solid elements. To embed bar reinforcement in solid elements, DIANA needs for each solid element the location points of the particle that is embedded in that element.

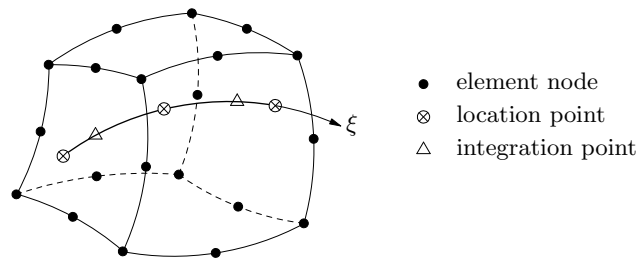


Figure 7.171: Bar in solid element

7.9.1.8 Grid Reinforcement in Plane Stress Elements

Grid reinforcement can be embedded in all plane stress elements.

A particle of a grid reinforcement may cover the complete area of the embedding plane stress element or only part of the area of the element.

This example specifies a grid reinforcement with one section [Fig. 7.172]. The nodes of the section are input with node numbers. The grid is embedded in a mesh of linear

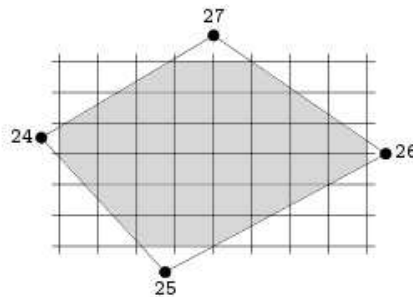


Figure 7.172: Grid reinforcement in plane stress element

quadrilateral plane stress elements. DIANA automatically determines which elements are fully or partly covered by the grid (the ones marked in gray).

7.9.1.9 Grid Reinforcement in Plane Strain Elements

Reinforcement grids may be embedded in all plane strain elements. Due to the nature of plane strain the grid shows up in the XY plane as a line between the location points. Plane strain elements are automatically checked for embedding of reinforcement grids

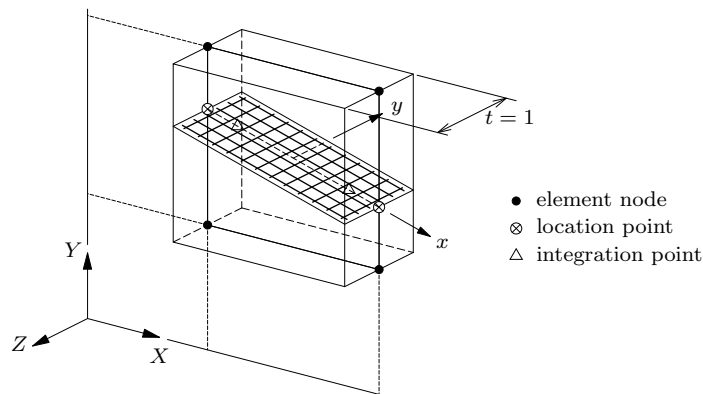


Figure 7.173: Grid reinforcement in plane strain element

specified with sections. In this case DIANA computes the location points that describe the particles of the grid within the plane strain elements.

A grid section is embedded in a plane strain element if it intersects one or two element edges, but none of them more than once.

7.9.1.10 Grid Reinforcement in Axisymmetric Elements

Reinforcement grids may be embedded in axisymmetric regular solid ring elements. Due to the nature of axisymmetry the grid shows up in the XY plane as a line between the location points. Axisymmetric solid ring elements are automatically checked for embedding

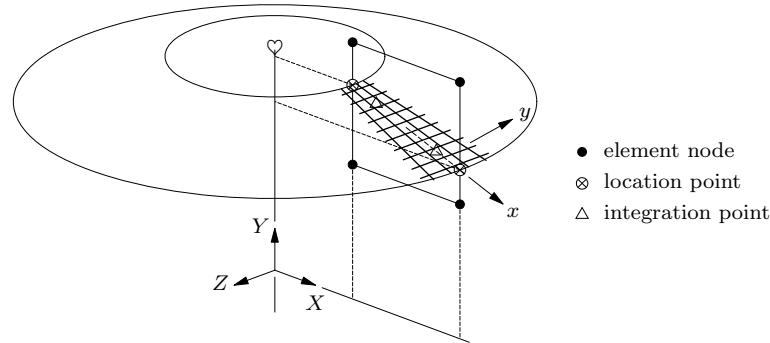


Figure 7.174: Grid reinforcement in axisymmetric element

of reinforcement grids specified with sections. In this case DIANA computes the location points that describe the particles of the grid within the axisymmetric elements.

A grid section is embedded in an axisymmetric element if it intersects one or two element edges, but none of them more than once.

7.9.1.11 Grid Reinforcement in Curved Shell Elements

Reinforcement grids can be embedded in all curved shell elements. A particle of a grid reinforcement may cover the complete area of the embedding curved shell element or part of the area. The location of the grid particle in the element is determined by location

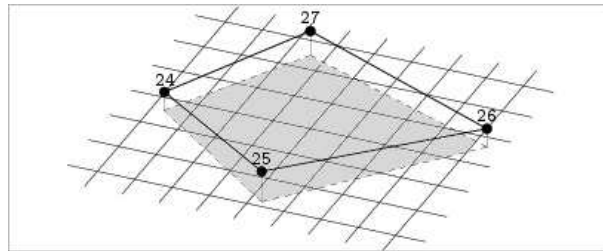


Figure 7.175: GRID section in curved shell elements (example)

points which must be within the thickness domain of the element. In other words: the eccentricity z cannot be greater than half the thickness t of the element at the node [Fig. 7.176b]. Curved shell elements are automatically checked for embedding of reinforcement grids specified with sections. In this case DIANA determines the structural elements that are covered by the grid section.

A particle of a grid section is embedded in a curved shell element if it is completely inside the thickness domain of the element and if it covers the element area completely.

7.9.1.12 Grid Reinforcement in Solid Elements

Reinforcement grids can be embedded in all solid elements. Typically, the location points are on the edges of the solid element but they can also be inside the element domain.

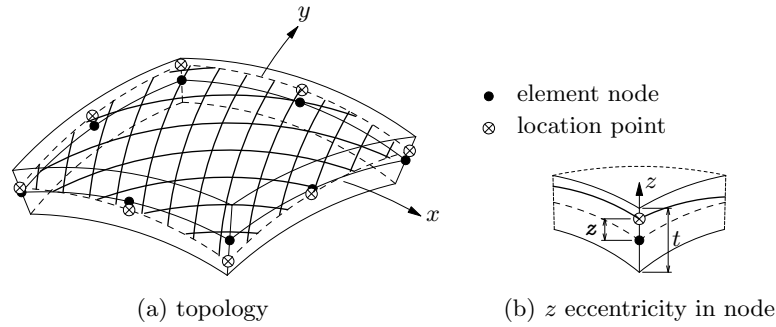


Figure 7.176: Grid reinforcement with eccentricities in curved shell element

Solid elements are automatically checked for embedding of reinforcement grids specified with sections. The contour of grid sections in solid elements may be quadrilateral or triangular.

Preprocessing of sections always yields triangular particles, therefore the integration scheme can only be specified with a single value.

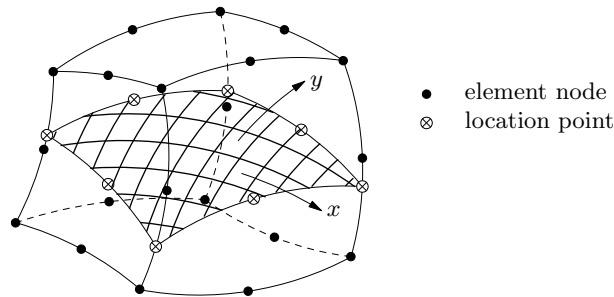


Figure 7.177: Grid reinforcement in solid element

This example specifies a grid reinforcement with one section [Fig. 7.178]. The section

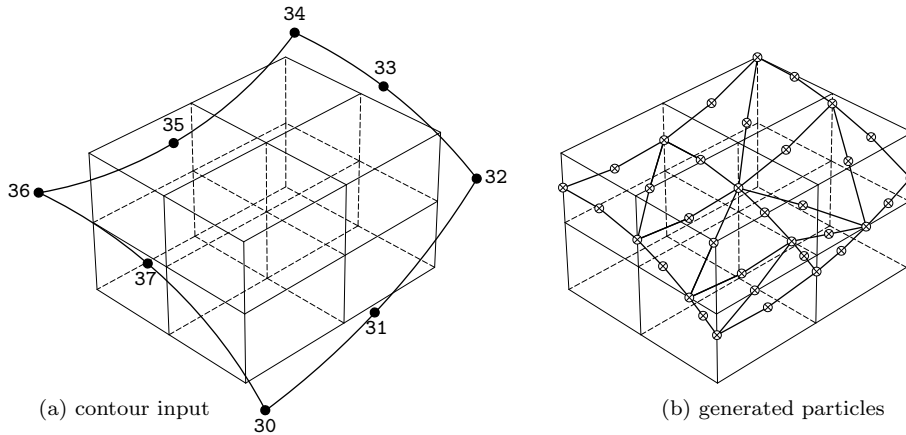


Figure 7.178: Triangularization of grid reinforcement in solid element

contour is a quadrilateral with curved edges, specified with eight nodes [Fig. 7.178a]. The generated grid particles are triangles with curved edges [Fig. 7.178b].

7.9.1.13 Loadings of Embedded Reinforcements

The most commonly used loading of reinforcements is the prestress. In DIANA the prestress loading can be applied as a uniform value for the reinforcement or as a variable prestress over the reinforcement. In the latter case the prestress is defined by the user in the model points of the reinforcement and interpolated by the program to the integration points of the reinforcements.

7.9.1.14 Stiffness Contribution

The contribution of the reinforcement stiffness to the stiffness of the respective mother element is automatically calculated. If for the respective element the not bonded option is defined, this reinforcement will not contribute to the stiffness matrix. The not bonded option can be used to simulate the pretensioning of prestress cables with reinforcement bars.

7.9.1.15 Material Models

The following material models can be used in combination with embedded reinforcements:

- Linear elastic
- Von Mises plasticity with optional hardening diagrams
- Menegotto-Pinto model for cyclic loading conditions
- Monti-Nuti model for cyclic loading conditions
- Dodd-Restrepo model for cyclic loading conditions
- Eurocode 2 EN 1992-1-2 for behaviour of reinforcement steel under fire load
- User-supplied subroutines for modelling user defined material models

7.9.1.16 Post-tensioning

In post-tensioning the reinforcement is located in the mother element, but is not yet bonded to the concrete. This is the case when e.g. a tendon is located in a reinforcement channel, but not yet grouted. When post-tensioning, the reinforcement may be stressed without affecting the deformation of the concrete in which it is located. This situation can be achieved by labeling the reinforcement as not bonded. In contrary to the default embedded reinforcement the not bonded option has the following properties:

- The stiffness of the reinforcement does not contribute to the stiffness of the mother element.
- The deformation of the mother element does not cause stresses or strains in the reinforcement.
- If a prestress is applied to a not bonded reinforcement than the equivalent element forces are applied as external forces to the element.

After the initial stresses have been applied to the not bonded reinforcement, the reinforcement will be fully grouted (bonded) to the mother element. The bonded or not bonded option is defined as a material property in DIANA. To change from not bonded reinforcements in earlier construction stages to bonded reinforcements in later construction stages, the not bonded option can be included in the material definition initially and removed for later stages, or the user can define the not bonded option in the material definition and define for which construction stages the not bonded behaviour must be replaced by bonded in the analysis command settings. This bonding is mostly associated with grouting the reinforcement channels. After bonding, the reinforcement contributes to the stiffness of the mother element and the reinforcement strains and stresses change upon deformation of the mother element.

7.9.2 Bond-Slip Reinforcements and Pile Foundations

The main characteristics of bond-slip reinforcements and pile foundations are:

- Bond-slip reinforcements are only available as embedded lines in regular plane stress, curved shell, or solid elements. In this case the reinforcement bar is internally modeled as a truss or beam elements, which are connected to the mother elements by line-plane, line-shell, or line-solid interface elements.

- In bond-slip reinforcements elastic or nonlinear bond-slip material behaviour may be defined for the line-plane, line-shell, or line-solid interfaces in the bond-slip reinforcements.
- Bond-slip reinforcements may be applied for modelling slip of steel reinforcement in concrete or for modelling interaction of pile foundations in soil and rock.

Bond-slip reinforcements and pile foundations are defined similar to embedded reinforcements, except that is explicitly specified that the reinforcement is discretized in truss or beam elements.

These truss or beam elements are connected by line-plane, line-shell, or line-solid interface elements to the mother elements in which they are located. For example, the bond-slip reinforcements can be used to describe the pull-out of an anchor of a bar reinforcement or to model the interaction of a pile foundation and the rock or soil. No integration schemes can be defined for bond-slip reinforcements, because in general beam, truss, and interface elements have different integration schemes. The applied elements for bond-slip reinforcements use the default integration schemes of these elements.

Bond-slip reinforcements are only available for lines in regular plane stress, curved shell, or solid elements with linear or quadratic interpolation functions.

7.9.2.1 Bar Reinforcement in Plane Stress

A bond-slip reinforcement bar in a plane stress element is defined similar to the description in [§ 7.9.1.3 p. 105]. The calculation of the intersections of the bar reinforcement and the plane stress elements can be done by reinforcement sections with location points being defined by nodes or by global XYZ coordinates. The type of line element that is internally used is automatically determined according to Table 7.4.

Table 7.4: ELEMENTS FOR BOND-SLIP REINFORCEMENTS IN PLANE STRESS

Mother element	T6MEM	Q8MEM	CT12M	CQ16M
Truss reinforcement	L4TRU	L4TRU	CL6TR	CL6TR
Beam reinforcement	L7BEN	L7BEN	CL9BE	CL9BE

7.9.2.2 Bar Reinforcement in Curved Shell

A bond-slip reinforcement bar in a curved shell is defined similar to the description in [§ 7.9.1.6 p. 105]. The calculation of the intersections of the reinforcement bar and the curved shell elements can be done by reinforcement sections with location points being defined by nodes or by global XYZ coordinates. The type of line element that is internally used is automatically determined according to Table 7.5.

Table 7.5: ELEMENTS FOR BOND-SLIP REINFORCEMENTS IN CURVED SHELL

Mother element	T15SH	Q20SH	CT30S	CQ40S
Truss reinforcement	L6TRU	L6TRU	CL9TR	CL9TR
Beam reinforcement	L13BE	L13BE	CL18B	CL18B

7.9.2.3 Bar Reinforcement in Solid

A bond-slip reinforcement bar in a solid is defined similar to the description in [§ 7.9.1.7 p. 106]. The calculation of the intersections of the reinforcement bar and the solid elements can be done by reinforcement sections with location points being defined by nodes or by global XYZ coordinates. The type of line element that is internally used is automatically determined according to Table 7.6 on the next page.

Table 7.6: ELEMENTS FOR BOND-SLIP REINFORCEMENTS IN SOLID

Mother element	TE12L	TP18L	HX24L	CTE30	CTP45	CHX60
Truss reinforcement	L6TRU	L6TRU	L6TRU	CL9TR	CL9TR	CL9TR
Beam reinforcement	L13BE	L13BE	L13BE	CL18B	CL18B	CL18B

7.9.2.4 Material and Geometry Parameters

The material parameters of the bond-slip reinforcements define both the material for the beam or truss elements as for the interface elements. All available material models for the regular beam and truss elements can be applied. For the interface elements linear elasticity, nonlinear elasticity, Coulomb friction, and the bond-slip models may be applied, see Volume *Material Library*. The geometry properties of the truss [§ 7.4 p. 67] or beam elements [§ 7.5 p. 71] must be defined as geometric properties of the reinforcement.

Note that for beam elements with an arbitrary cross-section shape specified with a number of quadrilateral zones, the maximum number of zones is four.

For the interface elements the outer diameter d or the perimeter of the reinforcement should be specified. For rectangular, box, circular, or pipe beam cross-sections [§ 7.5.4 p. 80], DIANA calculates the diameter and perimeter internally. Therefore, specification of the diameter or perimeter is not required for these predefined cross-sections. DIANA uses the diameter or perimeter to convert the shaft stresses into cross-section forces per unit length.

Furthermore, the outer diameter d of the reinforcement bar or pile should be specified.

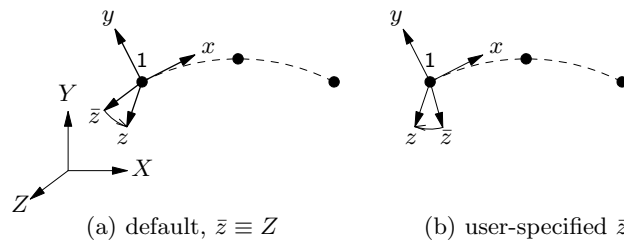


Figure 7.179: Axes for reinforcements in shell or solid elements

For reinforcements in shell or solid elements, there is no unique interface plane and DIANA needs a \bar{z} direction to set up the element axes [Fig. 7.179]. The specified direction of the \bar{z} axis may not be in the direction of the bar axis (tangent in the first node of the generated beam or truss elements).

If you specify the \bar{z} axis, then DIANA determines in each node an x axis tangential to the bar axis. Next the $y \perp \bar{z}x$ is set up and finally $z \perp xy$ is set up which is in fact a correction of the \bar{z} direction.

[$\bar{z} \equiv Z$] If you do not specify the \bar{z} axis then DIANA takes the global Z direction and you must check if this yields the correct element axes.

If the x axis coincides with the model Z direction then $y \perp \bar{z}x$ cannot be done and the \bar{y} axis is chosen in model Y direction. In that case the element xyz axes are set up as follows: x in the direction of the reinforcement axis, $z \perp x\bar{y}$ and $y \perp zx$.

For the interface along reinforcements in plane stress elements, DIANA always uses the global Z direction as element z direction.

7.9.2.5 Loading Input

When the begin and end nodes of bond-slip reinforcements are defined as nodes, instead of using global XYZ coordinates, nodal loads may be applied to these nodes. E.g. for defining a pre-tensioning anchor loading a pulling force can be applied in the begin or end node of the reinforcement bar and a pressure loading can be applied as reaction to the element faces of the concrete elements. Explicit specification of initial stresses is not allowed for bond-slip reinforcements.

End forces. In contrary to standard reinforcements the prestress in the reinforcement, including the anchor losses, can be calculated explicitly when using bond-slip reinforcements. In this case the user should define the anchor force P_0 explicitly and the prestress will be calculated in a full analysis step. Anchors can be applied at one or two ends of the reinforcement. The specification sequence determines the begin and end point of the reinforcement.

7.9.2.6 Connecting Bond-Slip Reinforcements to Elements

FX⁺ and jDIANA interfaces to DIANA generate reinforcements of which the location points are specified in global XYZ coordinates. Especially when pile foundations are described with bond-slip reinforcements the top ends of the pile foundations must be connected to the construction, which is defined by elements. Therefore, three special options to connect bond-slip reinforcements to elements are available:

- The node corresponding to the begin node of the bond-slip reinforcement is explicitly specified by the user. This node must be on the same location as the starting point of the reinforcement. Nodal loads, supports to this node will act on the begin node of the bond-slip reinforcement. When this node is connected to other elements, the corresponding degrees of freedom of the reinforcement and the elements in this node will be shared.
- The node corresponding to the end node of the bond-slip reinforcement is explicitly specified by the user. This node must be on the same location as the end point of the reinforcement. Nodal loads, supports to this node will act on the end node of the bond-slip reinforcement. When this node is connected to other elements, the corresponding degrees of freedom of the reinforcement and the elements in this node will be shared.
- The position of begin and end nodes of bond-slip reinforcements are checked against the position of nodes in a nodal group. When the begin or end node of a bond-slip reinforcement coincides with a node in the nodal group this node will be merged and the respective node number of the node in the group will be used for the begin or end node of the bond-slip reinforcement. E.g. when piles, being described by bond-slip reinforcements, need to be connected to a raft foundation modeled by shell elements, the nodes of the raft foundation are placed in a nodal group and this group name is referred to in the following option. The nodes will be merged when these positions match. Nodal loads, supports to this node will act on the node of the bond-slip reinforcement. When this node is connected to other elements, the corresponding degrees of freedom of the reinforcement and the elements in this node will be shared.

7.9.2.7 Tip Stiffness

At the extremities of a bond-slip reinforcement or a pile foundation a spring type of connection between the end of the reinforcement and the mother element in which the reinforcement end is located, may be defined. Situations where this functionality may be applied are an embedded anchor of a reinforcement or the tip of a pile foundation. The tip stiffness can be applied at one or both extremities of a bond-slip reinforcement. For the tip stiffness, either linear elasticity, or nonlinear elasticity may be applied, see Volume *Material Library*.

For the tip stiffness the cross-section area at the extremities of the reinforcement should be specified. For rectangular, box, circular, or pipe beam cross-sections [§ 7.5.4 p. 80] and truss cross-sections [§ 7.4.5 p. 71]), DIANA calculates the cross-section area internally. Therefore, specification of the cross-sectional is not required for trusses or these predefined beam cross-sections. DIANA uses the cross-section area to convert the tip stress into force.

7.9.2.8 Result Output

Results related to the beam or truss elements and the interface elements applied to model the bond-slip reinforcements are output as reinforcement results. Because beam or truss elements and interface elements may have different integration schemes and results from both these elements are projected in postprocessing to the same reinforcement particle, the output of results in other locations than the nodes is not allowed for bond-slip reinforcements. More information on applicable result items for bond-slip reinforcements can be found in Volume *Analysis Procedures*.

Chapter 8

Loading and Constraints

8.1 Nodal Variables

Nodal variables are also called degrees of freedom of the finite element model. Nodal variables are defined with reference to a basis type. A basis is connected to a node. A basis type can be of a vector type or a scalar type. Examples of vector type basis for nodal variables are translations and rotations and examples of scalar type basis for nodal variables are temperature and pressure head. For a vector type basis the directions of the nodal variables in the basis must be defined. Directions in one basis must be orthogonal and their number varies from 1 to 3. The number of directions in a basis is dependent on which types of elements are connected to the node of the basis. When a solid element is connected to a node, 3 nodal variables are required in the translation basis of this node. When only plane stress elements are connected to a node, only 2 nodal variables are required in the translation basis of this node and the vectors of these nodal variables must be oriented in the plane of the element, because a plane stress element has only translational nodal variables in the plane of the element. When e.g. a curved shell element is connected to a node, in this node both a translational basis and a rotational basis will be defined. In this case the translational basis will have 3 variables and the rotational basis will have 2 variables for a standard (without drilling) curved shell element and 3 variables for a curved shell element with drilling. For heat transfer analysis elements the degrees of freedom will be temperatures and for groundwater flow analysis the degrees of freedom will be potential heads. Thus, in DIANA the nodal variables are defined on the basis of which element types are connected to the nodes. That means that there are no unnecessary nodal variables in the model and it is not necessary to constraint unused nodal variables, e.g. out-of-plane displacements in a two-dimensional model. On the other hand when you use a three-dimensional beam for a two-dimensional analysis, the out-of-plane translation and in-plane rotations must be constraint. A nodal variable is identified by the node number, the basis type (e.g translation or rotation or temperature) and in case of a vector type of basis also the direction must be defined.

In the following paragraphs it will be explained how nodal variables can be constraint. Constraint to a nodal variable can be that the nodal variable is fixed to a user-defined value, e.g. zero, or a linear dependency relation between various nodal variables can be defined. An example of such dependency relation is that the translation in nodes 1-3 in the global X direction must be equal to the translation of node 4 in the global X direction.

The procedure of defining directions of vector degrees of freedom in DIANA is as follows:

First the nodal variables that are constraint or subject to a dependency relation are defined and numbered. When constraints and/or dependency relation are conflicting, DIANA will produce an error message. When in the basis several constraints are defined they must be orthogonal. Then DIANA checks all active elements in numerical order. Depending on the element types connected to a node the necessary basis and directions in a basis will be added.

8.2 Constraints and Supports

In DIANA constraints can be defined for nodal variables. In a structural analysis the constrained nodal variables are called supported nodes. When a nodal variable is supported, the value of the nodal variable is fixed to zero, unless the user defines a non-zero value for this variable. The set of system equations is defined such that the constrained nodal variable and unconstrained nodal variables are grouped. The linear system of equations can be expressed as:

$$\begin{bmatrix} \mathbf{K}_{uu} & \mathbf{K}_{uc}^T \\ \mathbf{K}_{uc} & \mathbf{K}_{cc} \end{bmatrix} \begin{Bmatrix} \mathbf{u}_u \\ \mathbf{u}_c \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}_u \\ \mathbf{f}_c \end{Bmatrix} \quad (8.1)$$

Here subscript $_u$ is associated with the *unconstrained* degrees of freedom and subscript $_c$ with the *constrained* degrees of freedom. Eq. (8.1) can be reduced to

$$\mathbf{K}_{uu}\mathbf{u}_u = \mathbf{f}_u - \mathbf{K}_{uc}\mathbf{u}_c \quad \text{or} \quad \hat{\mathbf{K}}\hat{\mathbf{u}} = \hat{\mathbf{f}} \quad (8.2)$$

Now all boundary conditions, tyings and loadings have been treated, the remaining system of equations can be solved.

Constraints or supports are defined in sets. These sets can be activated or deactivated in a phased construction analysis.

8.3 Linear Dependencies or Tyings

Linear dependencies between nodal variables are also 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 [53] 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.

Generally speaking, a tying consists of a degree of freedom in a *master* node, in the figures of this section drawn as \bullet , and one or more degrees of freedom in *slave* nodes, drawn as \circ . Figure 8.1 shows the notation convention for degrees of freedom for tyings. As for rigid supports, degrees of freedom for tyings are specified by means of a *type*:

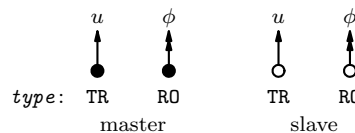


Figure 8.1: Tying degrees of freedom

translation or rotation, and a direction number.

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.

That means that the slave nodal variables are handled as constrained nodal variables \mathbf{u}_c when solving the systems set of equations, as explain in the previous paragraph.

In DIANA the user can define both single and multi-point tying. In single-point format you must explicitly specify all nodes for each tying separately. 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. 8.2]. The principle is that you specify the corner nodes of these edges, the so-called *vertices*, indicated with double circles. You also

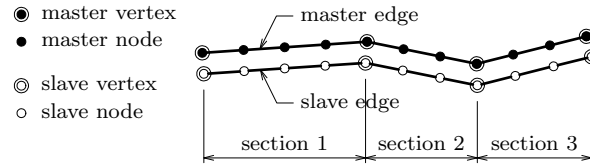


Figure 8.2: Multi-point tying

specify a set of master and slave nodes in arbitrary order, the 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. 8.2]. DIANA ignores specified nodes which are not a straight edge section.

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.

Linear dependencies or tyings are defined in sets, which can be activated or deactivated in a phased construction analysis.

The following types of tyings can be defined in DIANA:

8.3.1 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 as shown in Figure 8.3.

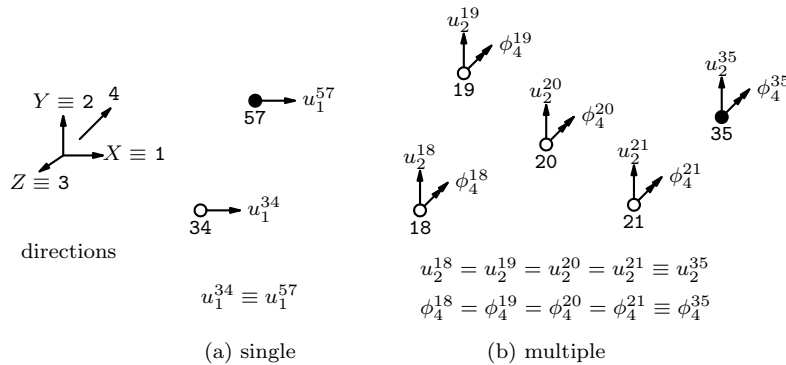


Figure 8.3: Equality tyings

8.3.2 Interconnections

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. A typical application of interconnection tyings is *Mesh refinement* as shown in the input for the mesh of Figure 8.4 on the following page.

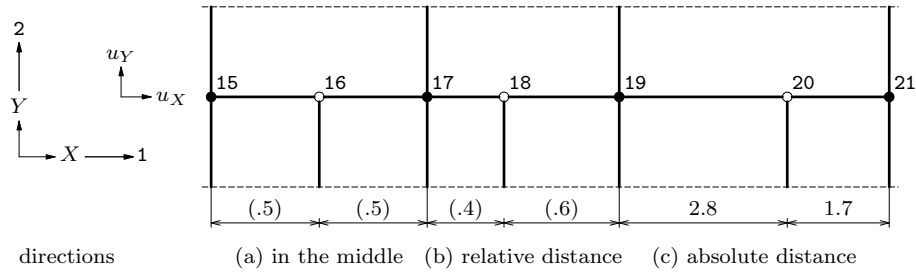


Figure 8.4: Mesh refinement

Interconnections in multi-point format with series of slave vertices and nodes, combined with series of master vertices and nodes are shown in Figure 8.5. If there are n

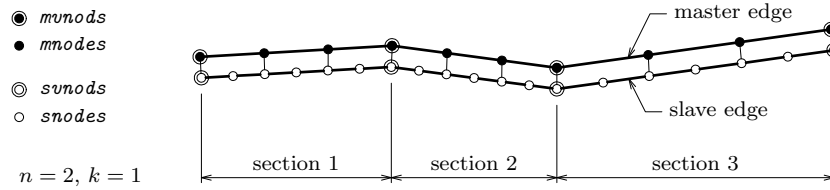


Figure 8.5: Multi-point interconnection

master nodes between two vertices and k slave nodes 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.

8.3.3 Eccentricities

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. Tying for eccentric connection may be input in single- or multi-point format.

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 8.6. In these examples, five

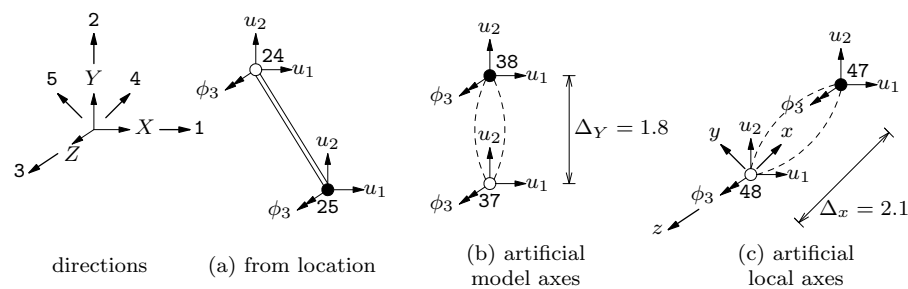


Figure 8.6: Eccentricity in two-dimensional model

directions are used. Figure 8.6a 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 8.6b 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 8.6c the artificial eccentricity Δ_x is specified in a local xyz coordinate system.

Eccentricities in multi-point format must be specified with series of slave vertices and nodes, combined with series of master vertices and nodes [Fig. 8.7]. For each edge section,

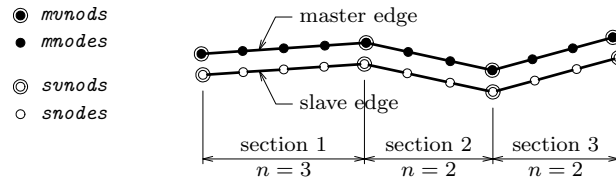


Figure 8.7: Multi-point eccentricity

the amount of master nodes n must be equal to the amount of slave nodes, i.e., each slave node is eccentrically connected to one master node.

8.3.4 General Connections

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.

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

The input data below presents two simple examples of general connections in two-dimensional space as shown in Figure 8.8. For the *Seesaw* connection [Fig. 8.8a] the Y

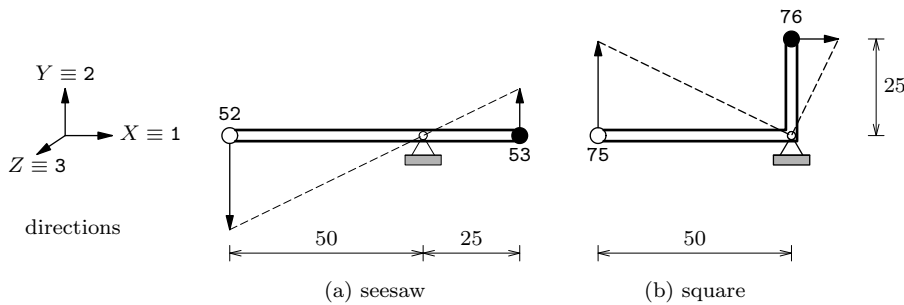


Figure 8.8: General connections in two-dimensional space

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. 8.8b] causes the slave's Y displacement to be twice as much as the master's X displacement.

General connections in multi-point format must be specified with series of slave vertices and nodes, combined with series of master vertices and nodes [Fig. 8.9].

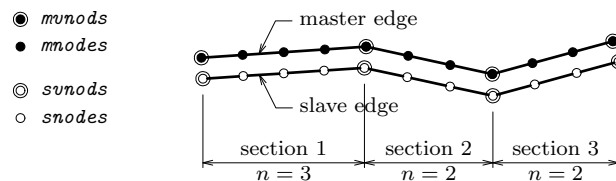


Figure 8.9: Multi-point general connection

8.3.5 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. 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.

There are some important restrictions to connection by automatic tyings:

- Currently the Automatic Tying option only works for class-III beams and curved shell elements [Vol. *Element Library*].

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.

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. 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.

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. For automatic tying of connected elements, it is not necessary to specify tyings explicitly in your model.

Automatic tying of connected elements is possible for the connections as shown in Figure 8.10 and outlined below.

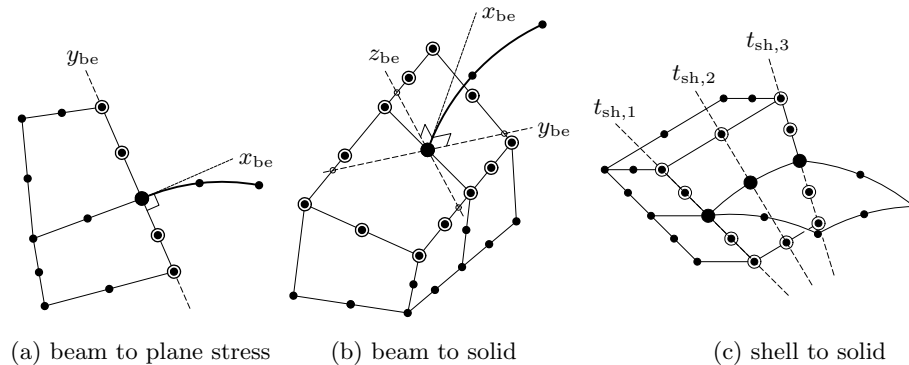


Figure 8.10: Automatic tying of connected elements

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 explicitly in your model. Automatic tying of loose elements is possible for the connections as shown in Figure 8.11 on the next page and outlined below.

8.4 Load Cases

There are various classes of loads input. 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*.

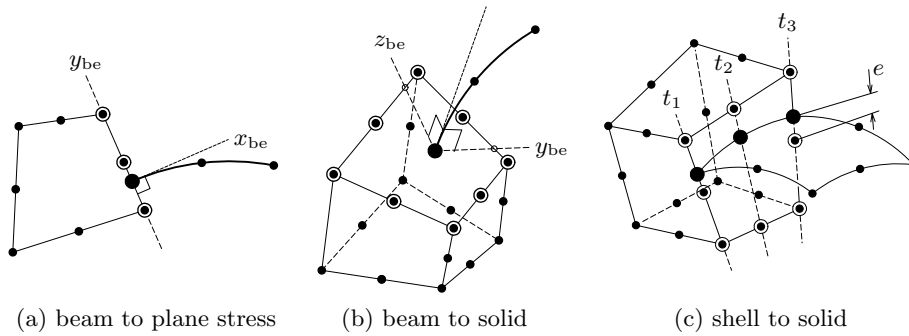


Figure 8.11: Automatic tying of loose elements

8.4.1 Nodal Loads

For structural analysis a nodal load can be a nodal force or a nodal moment. These loads can be defined in every node in the model in which a respective basis is defined (translations for force and rotations for moments).

8.4.2 Weight Loads

A weight load specification requires the definition of the direction of gravity, gravity acceleration and mass density for elements. The volume of the elements is calculated by DIANA. A weight load may be defined in several load cases, but the direction and acceleration are always equal. When different acceleration loads are applied in different directions and/or with different acceleration values, equivalent acceleration loads can be specified as alternative. Direction of gravity and gravity acceleration are used to convert pressure heads into pore-pressures for groundwater-flow analysis and therefore must be unique in the model.

8.4.3 Equivalent Acceleration Loads

An equivalent acceleration load specification requires the definition of the direction of gravity, gravity acceleration and mass density for elements. The volume of the elements is calculated by DIANA. An equivalent acceleration load may be defined in several load cases, with different accelerations and directions of acceleration. Even different equivalent acceleration loads may be defined in the same load case. This in contrary to weight loads for which the gravity acceleration and gravity direction must be the same for all weight loads in the model.

8.4.4 Centrifugal Loads

A centrifugal load specification requires the definition of axis of centrifugation, the rotation speed and mass density for elements. The volume of the elements is calculated by DIANA. A centrifugal load may be defined in several load cases, with different accelerations and directions of acceleration. Only one centrifugal load may be specified in a load case.

8.4.5 Fixed Displacement or Deformation Loads

Fixed or prescribed displacements, so-called deformation loads are defined for one or more constraint nodal variables or degrees of freedom. These degrees of freedom must be identified as constraint degrees of freedom, in order to make sure that the fixed displacement degree of freedom is considered as a constraint nodal variable [Eq. (8.1)]. That means that for a linear elastic analysis with several load cases, for which in only one load case a deformation load is defined, the degrees of freedom for which a deformation load is defined will be constraint for all load cases. This approach has the advantage analysis time is saved because the set of constraint and unconstraint variable is same for all load

cases or right hand side vectors. As a consequence the matrix need to be decomposed only one time, whereas for every load case only the substitution will to be done.

8.4.6 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. 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.*
- *Only one mobile load per load case may be defined.*
- *If a load case contains a mobile load, no other loadings are allowed in this load case.*

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. 8.12]. DIANA will automatically generate one load case for each position of the truck placed on

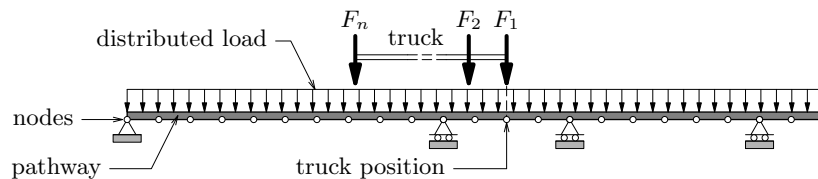


Figure 8.12: Mobile load definition

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 post-processing DIANA can calculate envelopes of results for all these load cases [Vol. *Pre- and Postprocessing*]. Other load cases, like weight or prestress, can be taken into account as well, when calculating envelopes of results.

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 8.12. 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. 8.13a], 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. 8.13b], and so on.

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. 8.13c]. 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. 8.13d]. 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. 8.13e]. 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.

Mobile loads may be defined explicitly, or with references to design codes.

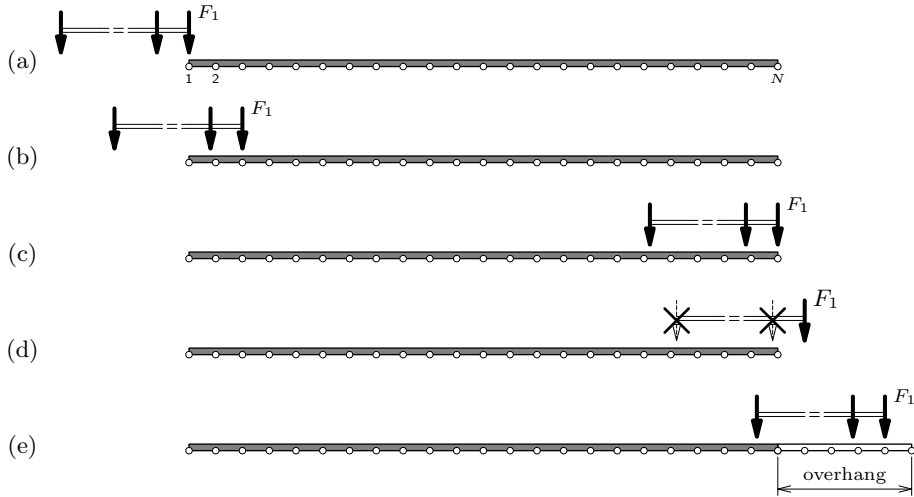


Figure 8.13: Ultimate truck positions

- Explicitly defined mobile load

For explicit input of mobile loads you must specify at least the truck load (F_1 to F_n and d_1 to d_n) and the distributed load q and the effective width of the carriage way. To simulate the influence of the transverse position of these loads you may specify load factors. By default DIANA neglects the influence of the transverse position of

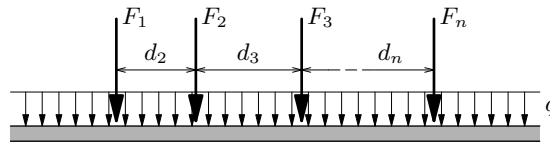


Figure 8.14: Mobile truck and distributed load

an explicitly specified mobile truck load. A multiplication factor f_F for the truck forces and a multiplication factor f_q for the distributed load can be defined.

- Dutch VOSB and VBB code

There are two Dutch codes which define mobile loads on bridges: the VOSB [56] for steel bridges and the VBB [55] for concrete bridges. According to these codes, the magnitude of the loads depend on the class as shown in Table 8.1.

Table 8.1: 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, and the width of the carriage way. Additionally, you may specify the transverse position of the truck, and the bump and load reduction factors.

The width w of the carriage way [Fig. 8.15] is subdivided in a number of lanes with width l . According to both Dutch codes, DIANA applies at most two lorries in transverse direction in a position which yields a maximum load contribution. 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. Values $v1$ to vn are the values

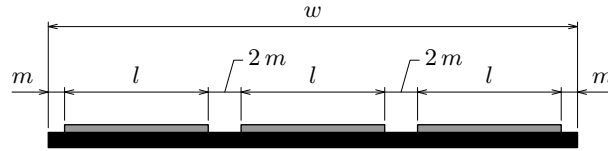


Figure 8.15: Lanes on a carriage way applied for VOSB code

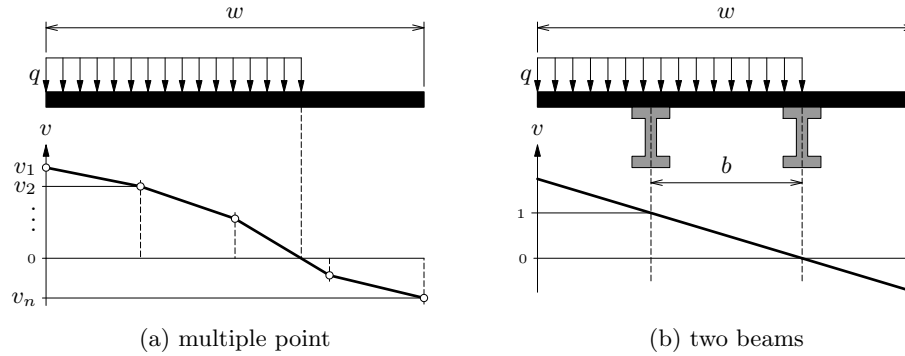


Figure 8.16: Influence line in transverse direction

of the influence line $v_{i=1,n}$. You may specify at most ten points. DIANA assumes ($n \leq 10$) an equidistant distribution of the points. If you only specify a single point, then the influence line is a constant. The two beams option is a short-cut input for the influence line if the finite element model simulates a bridge with two beams in transverse direction [Fig. 8.16b]. 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.

Alternatively to the influence lines for the effect in transverse direction a multiplication factor f_F for the truck forces and a multiplication factor f_q for the distributed load q can be defined.

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 [56, § A.2.10-11], DIANA uses a bump factor S and a load reduction factor B defined as

$$S = 1 + \frac{40}{100 + l} \quad \text{and} \quad B = 0.6 + \frac{40}{100 + l} \quad (8.3)$$

For the VBB [55, § 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.

- 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. 8.17]. The total width w of the carriage way depends on

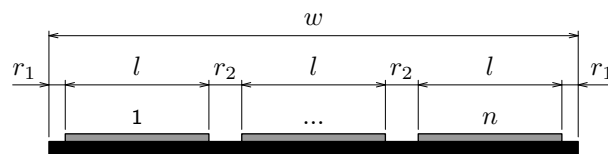


Figure 8.17: Lanes on a carriage way applied for ENV 1991-3 code

the number of lanes n , their width l , and the remaining width r :

$$w = n \times l + r \quad \text{with} \quad r = 2 \times r_1 + (n - 1) \times r_2 \quad (8.4)$$

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 8.2].

Table 8.2: 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	$w - n \times 3$

The ENV 1991-3 code indicates various classes of loading by a number [Table 8.3]. Each class gives the distributed traffic load q_{ik} , the distributed load on the remaining

Table 8.3: 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 8.3]. 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. 8.18]. 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. For the effect in transverse direction a

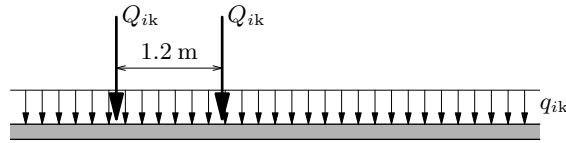


Figure 8.18: Mobile truck and distributed load according to the ENV 1991-3 code

multiplication factor for the truck axle loads f_F and a multiplication factor for the total distributed load f_q can be defined.

8.4.7 Wind and Water Loads

Wind and water loads are available for class-I and class-II beam elements. Wind and water wave loads will be evaluated as an average distributed loads on class-I and class-II beam elements.

Some geometry parameters must be specified if a wind or water load acts on the finite element model. These parameters are:

- Cross-sectional area A subjected to wind or water load, the dimension is $[length]$. For a PIPE or CIRCLE shape cross-section, the default value is the outer diameter d . For other cross-section shapes the input of this parameter is obligatory.

- Drag coefficient C_D . Typically $C_D = 1.3$.

For wind load the following parameters need to be specified:

- Roughness height z_0
- Displacement height d
- Friction wind speed v^*
- Wind direction
- Density of air ρ
- Global Z coordinate z_g of the ground. No loads will be generated below z_g .

For a water wave load the following parameters need to be specified

- Average water profile: $z1$ to zn are the heights z , $v1$ to vn the corresponding average velocities \bar{v} . Specify one \bar{v} for each z .
- Water wave direction
- Density of water ρ
- Global Z coordinate z_g of the ground. No loads will be generated below z_g .
- Global Z coordinate z_w of the still water level. No loads will be generated above z_w .

8.4.8 Influence Loads

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 an influence load, 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. The output of the influence field for a specific point may be selected.

Influence field determination cannot be combined with other loadings.

8.4.9 Element Point Loads

A concentrated force and/or moment load may be specified to act in a point of a beam, plate, flat shell, curved shell or solid element. The element at which the point load acts must be specified. The location at which the point load acts may be defined by a node number, which does not need to be a node in the connectivity of the element, or the global coordinates of the point or the iso-parametric coordinates of the point. The direction of the point load can be defined relative to the global coordinates system or relative to the element coordinate system.

8.4.10 Element Line Loads

Element line loads may be specified to act on line element types such as truss and beam, but also on edges of surface elements or solid elements. For trusses, two-dimensional bodies and three-dimensional bodies only distributed force loads may be specified (force per length). For beams, plates and shell elements also distributed moment loads may be specified (force*length/length). Element line loads can be defined with a constant value or may vary along the line. In case of variation for every node on the line or edge a value must be specified. The direction of the element line load can be defined relative to the global coordinates system or relative to the element coordinate system. In this

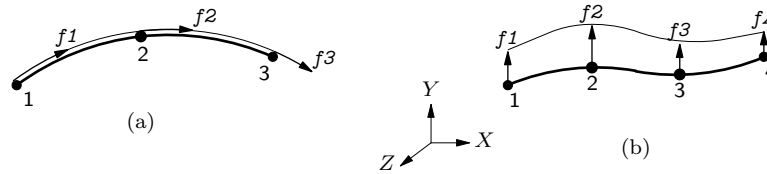


Figure 8.19: Line load examples

example element 1 must be a three-node truss element, a load along the beam axis varies quadratically from 30 in the first node to 25 in the second to 20 in the third [Fig. 8.19a]. Element 2 must be a four-node beam element, a load in the Y direction varies third-order from 20 in the first node to 25 in the second to 18 in the third to 15 in the fourth [Fig. 8.19b].

8.4.11 Element Face Loads

Element face loads may be specified to act on surface elements or solid elements. For a solid element the face of the solid needs to be identified. For two-dimensional bodies and three-dimensional bodies only distributed force loads may be specified (force per length per length). For plates and shell elements also distributed moment loads may be specified (force*length per length per length). Element face loads can be defined with a constant value or may vary over the face. In case of variation for every node on the face a value must be specified. The direction of the element face load can be defined relative to the global coordinates system or relative to the element coordinate system. In this example

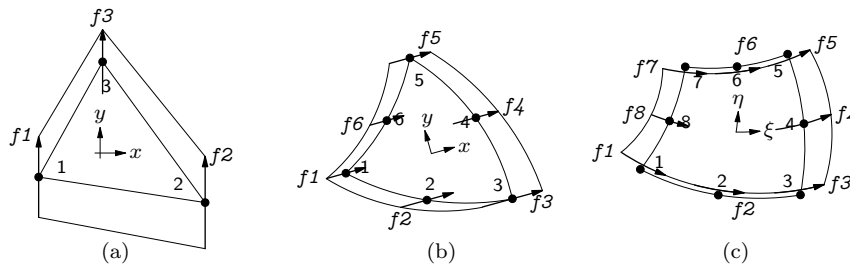


Figure 8.20: Face load examples

element 1 must be a three-node (linear) triangle, the distributed face load in y direction varies linearly from 100 in the first node to 114 in the second to 79 in the last node [Fig. 8.20a]. Element 2 must be a six-node (quadratic) triangle, the distributed face load in x direction varies quadratically from 100 in the first node to 134 in the second etc. to 86 in the last node [Fig. 8.20b]. Element 3 must be an eight-node (quadratic) quadrilateral, the distributed face load in ξ direction varies quadratically from 100 in the first node to 95 in the second etc. to 89 in the last node [Fig. 8.20c].

8.4.12 Element Volume Loads

Element volume loads may only be specified to act on solid elements. For solid element volume loads can only be distributed forces (force per length per length per length). Element volume loads can be defined with a constant value or may vary over the volume of the element. In case of variation for every node of the element a value must be specified. The direction of the element volume load can be defined relative to the global coordinates system or relative to the element coordinate system.

8.4.13 Initial Stress Loads

Initial stress loads or prestresses may be specified to act in two-dimensional body elements, three-dimensional body elements, plates and shell, beams and trusses, interface elements and embedded reinforcements. The prestress is always defined as the local stress tensor

for the respective element type, thus 6 components ($\sigma_{xx}, \sigma_{yy}, \sigma_{zz}, \sigma_{xy}, \sigma_{yz}, \sigma_{xz}$) for solid elements and only one component for and embedded bar reinforcement (σ_{xx}). Initial stresses are always defined in the local coordinate system of the element or reinforcement. Initial stresses may be defined as constant in an element or may vary over the element. In case of variation for every node of the element a stress tensor must be specified. However, for Class-I beams the initial stresses are not defined in terms of stresses, but as normal force and bending moments. Also for other directly integrated elements such as plate bending and flat shell elements the initial stresses are defined in terms of cross-section forces and bending moments. For numerical integrated beams and curved shell elements, next to the stress tensors in the mid-planes of the elements, gradient vector for the initial stress must be defined. These gradients of stress vectors allow to define variations of initial stresses in the cross-section of a beam or over the thickness of a shell element.

8.4.14 Temperature and Concentration Loads

A temperature definition itself does not lead to deformation, only a change of temperature does when a positive thermal expansion parameter is defined in the material properties for the respective elements. The same applies for concentrations, concentration changes and concentration expansion parameters. Temperature and concentration loads are considered as temperature changes or concentration changes. These loads may be specified to act in two-dimensional body elements, three-dimensional body elements, plates and shell, beams and trusses, interface elements and embedded reinforcements. The temperature and concentration load are defined as scalars. Temperatures and concentration loads may be defined as constant in an element or may vary over the element. In case of variation for every node of the element a value must be specified. However, for beams, plate bending and flat shell elements the initial stresses next to the temperature or concentration values in the mid-planes of the elements, gradient vector for the temperatures or concentrations must be defined. These gradients allow to define variations of temperature/concentration in the cross-section of a beam or over the thickness of a shell/plate element. For all types of beam elements, values *te1* to *ten* are the average temperatures

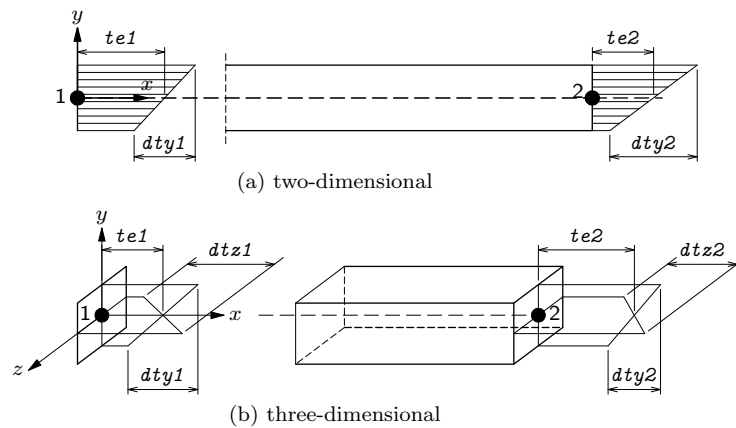


Figure 8.21: Temperature load examples for two-node beam

\bar{T} in the respective element nodes.

For *class-I* beam elements, values *dty1* to *dty_n* are the temperature *gradients* ΔT_y in the respective element nodes. For three-dimensional elements, values *dtz1* to *dtz_n* are the gradients ΔT_z in the respective element nodes. The gradient ΔT is defined as the variation of temperature per unit length going in the negative axis direction. Figure 8.21 shows positive gradients.

For *class-II* and *class-III* beam elements, values *dty1* to *dty_n* are the temperature *differences* ΔT_y between the extreme fibers in the respective element nodes. For three-dimensional elements, values *dtz1* to *dtz_n* are the differences ΔT_z in the respective element nodes. The difference ΔT is defined as the temperature in the extreme positive fiber (+y and +z) minus the temperature in the extreme negative fiber (−y and −z). Figure 8.21 shows positive differences.

For linear elements \bar{T} and ΔT vary linearly along the element axis, for quadratic elements quadratically etc. If you only specify one value `te1` and one value `dy1` (and `dtz1` for a three-dimensional beam) then the temperature distribution is uniform. For plate

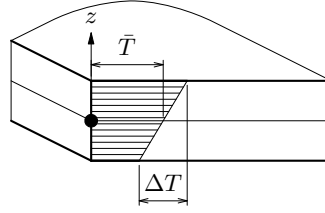


Figure 8.22: Temperature load

and shell elements the average temperature \bar{T} as well as the temperature gradient ΔT must be specified. T_0 and ΔT vary over the element area according to the interpolation order of the element: linearly, quadratically or higher-order. If you only specify `te1` and `dtz1` then the temperature distribution is uniform, else you must specify one T_0 and one ΔT for each of the element nodes.

The definition of the element concentration load C is analogous to temperature.

If the temperature or concentration distribution is of a higher order than the possible strain distribution, then the calculated strains and stresses may have a severe oscillating distribution.

8.4.15 Hydrostatic Pressure Loads

Hydrostatic pressure loads may be applied to plane strain, axisymmetric, solid elements, and to infinite, axisymmetric, or curved shell elements. Similar as for element edge loads, for the plane strain and axisymmetric elements the edge to which the hydrostatic acts must be defined. For solid element the face to which the hydrostatic load works is defined similar as for the element face loads. For these elements the hydrostatic pressure loads acts as a compressive load normal to the selected edges and/or faces. For shell elements the hydrostatic pressure load acts as a compressive load in the local element axes normal to the element. For shell elements the load direction can be inverted to load the other side of the shell. The fluid pressure is defined as

$$p = \max(0, \gamma_f (h - (\mathbf{x} - \mathbf{x}_{\text{ref}}))) \quad (8.5)$$

with γ_f the volumetric dead weight vector of the fluid as defined by the model fluid density, gravity direction and gravity acceleration, h the hydraulic head, and \mathbf{x}_{ref} the reference position for the hydraulic head.

8.4.16 Pore-pressure and Hydraulic Head Loads

DIANA offers several options to model a pore fluid in soil-like material. This section describes the two possibilities to specify a pore fluid load: as a *hydrostatic pressure* (neglected flow) or, more generally, as an *hydraulic load*. For both methods, you must specify some additional *material properties*.

You can apply a hydrostatic pore pressure distribution most conveniently by input of the position of a hydraulic head. For relevant *continuum elements* (plane strain, axisymmetric and solid), DIANA determines the isotropic load $\Delta\sigma_{ii}$ at position \mathbf{x} using $\Delta\sigma_{ii} = -Sp$. For *interface elements*, DIANA determines the normal traction load Δt_n at position \mathbf{x} using $\Delta t_n = -Sp$, where S is the degree of saturation and p the pore pressure: $p = \gamma_f (h - (\mathbf{x} - \mathbf{x}_{\text{ref}}))$, with γ_f the volumetric dead weight vector of the fluid, h the hydraulic head, and \mathbf{x}_{ref} the reference height for the hydraulic head.

8.4.17 Load Sets

Previously specified load cases may be combined to load sets, which form the load vectors of the system of equations (right hand side vectors). 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.

Chapter 9

Analysis Procedures

9.1 Linear Static Analysis

In a linear static analysis the following operations are performed subsequently:

- Evaluate model
- Solve the system of equations
- Output

9.1.1 Model Evaluation

During model evaluation the definitions of elements are checked, including material and geometrical properties which are assigned to elements; the element assembly, or ordering of element and nodal variables in the system set of degrees of freedom are defined; the element stiffness matrices are defined; and the load vectors are set-up. The following parameters can be set in the model evaluation:

- Extended test on element shape and aspect ratio

In the model evaluation command specification the user has the option to perform extended tests on the element shapes and aspect ratio. By default this test is not performed in a linear static analysis. The extended test checks whether the

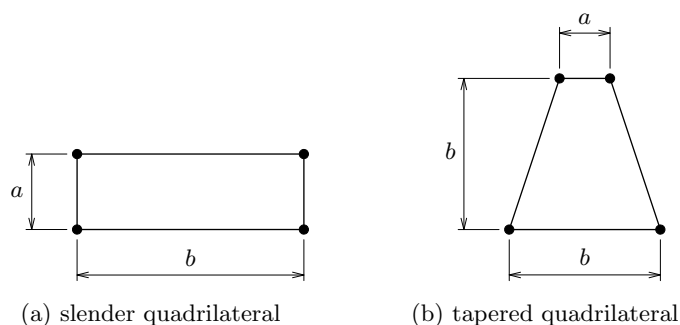


Figure 9.1: Element shapes

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. 9.1]. Table 9.1 shows some typical values of the shape parameters for the slender and tapered quadrilaterals of Figure 9.1. See Roddeman [63] for background theory. You may overrule the default values by specifying tolerance parameters for the shape and for the aspect ratio. If one of the shape parameters of an element is greater than the specified tolerance, DIANA will produce a warning message. The extended test option checks linear and quadratic elements of the following element

Table 9.1: 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

types: plane stress, plane strain, axisymmetric, plate bending, flat shell, and solid elements. Excluded are interfaces, beams, trusses etc. Especially for solid elements, the extended test consumes a considerable amount of processing time.

- Average nodal normal of shell elements

By default DIANA performs an averaging procedure on the nodal normals of all curved shell elements attached to a node. You may deactivate this operation. The tolerance angle in degrees for the averaging procedure can be specified by the user. For curved shell elements for which neither a predefined shape nor explicit thickness vectors are defined, 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 $\frac{1}{2}$ *angle* 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 *angle*.
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.

- Options for evaluation of reinforcements

By default DIANA does not evaluate embedded reinforcements in interface elements, but this operation can be activated optionally by selecting the option "Evaluate reinforcements in interface elements".

By default DIANA generates reinforcement particles in elements that are either fully or partly covered by a reinforcement grid section. In earlier versions grid reinforcement particles were only defined in those elements which are fully covered by a reinforcement section and not in elements that are partly covered by reinforcement grid sections. For getting compatibility with earlier versions you may select the option "Grid particles must cover element section fully".

- Tolerance for creating directions of variables in vector basis

DIANA will assemble the elements of the model and create the system degrees of freedom. The criterion ϵ for coincidence of a potential system degree of freedom and an existing one may be modified.

- Automatic tyings

By default automatic tyings are not defined in a linear static analysis. 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 [§ 8.3 p. 116] for further details and conditions.

9.1.2 Solve the System of Equations

After the model has been evaluated the linear system of equations is solved. DIANA offers four different solution methods:

- Intel PARDISO solution method, i.e. a Parallel Direct Sparse solver (default)
For models with many curved shell elements the Parallel Direct Sparse method (PARDISO) may be a good choice because this solver can handle the different order of contributions from rotations and translations to the stiffness matrix efficiently and this method provides multi-thread processing.
- Sparse Cholesky based solution method
With this solver, DIANA will automatically choose an optimal solution procedure which is initially based on a Sparse Cholesky method. If a factorization from a previous Sparse Cholesky solution exists, the automatic procedure uses this as a preconditioner in an iterative process.
- Iterative solution method
For models with more than 100.000 solid elements the iterative solver may be an efficient alternative because these models in general have a good numerical condition and iterative solver can provide a solution with minimum use of memory in relatively short time. DIANA provides multi-thread processing in the iterative solver.
- Out-of-core direct solution method

Substructuring is not useful for linear analysis. More information about solutions procedures is given in § 9.7.

9.1.3 Output

For a linear static analysis several results can be output, related to calculated displacements, forces, strains and stresses. In general results are output as fields. Displacements and nodal forces are usually presented as fields with reference to the nodes of the finite element mesh. Nodal result output gives per definition a smooth result field over boundaries of elements, because in each node only one result value is defined and result values in the elements are interpolated from the nodal values. On the other hand stresses and strains are usually presented as a field with reference to the elements of the mesh. For element based output fields results can be averaged per element (in the center of the element), or presented as values in the integration points of elements, or extrapolated to the nodes of the elements. When results are extrapolated to nodes, in each node there are as many result values as elements are sharing the respective node. Element field output in general does not show continuity in the result fields over the boundaries of elements. With the smooth option, element results can be averaged in the nodes and converted into nodal result fields.

A result field refers to a load case or a load-set. In linear static analysis DIANA can also calculate envelopes over different load sets for minimum or maximum values. For an envelope of minimum values over a selected number of load sets the minimum value over the selected load sets is presented in each location point.

The following typical results can be output for a linear static analysis:

- Displacements:
 - Total displacements
 - Prescribed displacements
 - Incremental displacements in a phased analysis
- Nodal forces:
 - Reaction forces in supports
 - External forces representing the nodal equivalents of user defined loads
 - Internal forces corresponding to the nodal equivalents of the element stresses

- Residual forces
- Strains:
 - Green–Lagrange strains in local (element xyz) or global (XYZ) coordinate system
 - Principal strains
 - Von Mises equivalent strains
 - Curvatures and elongations of beam, plate, and shell elements
 - Volumetric strains
 - Relative displacements in interface elements
- Stresses:
 - Cauchy stresses in local (element xyz) or global (XYZ) coordinate system
 - Principal stresses
 - Von Mises equivalent stresses
 - Cross-section forces and bending moments of truss, beam, and composed line elements in the local coordinate system
 - Distributed forces and bending moments of plate, shell, and composed surface elements in the local coordinate system
 - Tractions in interface elements (interface stresses)
 - Stress invariants
 - Hydrostatic pressure
 - Shear capacity with respect to the Mohr–Coulomb yield criterion for continuum elements and for interface elements
 - Safety factors with respect to the biaxial failure envelope
 - Stress gradients in reinforcement bars

In § 9.7 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.

9.1.3.1 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 § 7.1.3 on page 23, the strain–displacement relation is defined by the matrix \mathbf{B} as follows

$$\boldsymbol{\varepsilon} = \mathbf{B}\mathbf{u}_e \quad (9.1)$$

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

$$\boldsymbol{\varepsilon}(\xi, \eta, \zeta) = \mathbf{B}(\xi, \eta, \zeta)\mathbf{u}_e \quad (9.2)$$

will be evaluated at the standard values for ξ , η and ζ .

Equivalent Von Mises strain. DIANA calculates the equivalent Von Mises strain according to

$$\varepsilon_{eq} = \frac{2}{3} \sqrt{\frac{3(e_{xx}^2 + e_{yy}^2 + e_{zz}^2)}{2} + \frac{3(\gamma_{xy}^2 + \gamma_{yz}^2 + \gamma_{zx}^2)}{4}} \quad (9.3)$$

With the deviatoric strains:

$$\begin{aligned} e_{xx} &= +\frac{2}{3}\varepsilon_{xx} - \frac{1}{3}\varepsilon_{yy} - \frac{1}{3}\varepsilon_{zz} \\ e_{yy} &= -\frac{1}{3}\varepsilon_{xx} + \frac{2}{3}\varepsilon_{yy} - \frac{1}{3}\varepsilon_{zz} \\ e_{zz} &= -\frac{1}{3}\varepsilon_{xx} - \frac{1}{3}\varepsilon_{yy} + \frac{2}{3}\varepsilon_{zz} \end{aligned} \quad (9.4)$$

The engineering strains γ are defined as:

$$\gamma_{ij} = 2 \times \varepsilon_{ij} \quad (9.5)$$

For some calculations the strains are placed in a strain matrix \mathbf{E} which for the general three-dimensional strain situation is given by

$$\mathbf{E} = \begin{bmatrix} \varepsilon_{xx} & \varepsilon_{xy} & \varepsilon_{xz} \\ \varepsilon_{xy} & \varepsilon_{yy} & \varepsilon_{yz} \\ \varepsilon_{xz} & \varepsilon_{yz} & \varepsilon_{zz} \end{bmatrix} \quad (9.6)$$

Principal strains. DIANA calculates the principal strains $\varepsilon_{1,2,3}$ as the roots of Eq. (9.7) ordered such that $\varepsilon_1 \geq \varepsilon_2 \geq \varepsilon_3$.

$$\begin{bmatrix} \varepsilon_{xx} - \lambda & \varepsilon_{xy} & \varepsilon_{xz} \\ \varepsilon_{xy} & \varepsilon_{yy} - \lambda & \varepsilon_{yz} \\ \varepsilon_{xz} & \varepsilon_{yz} & \varepsilon_{zz} - \lambda \end{bmatrix} = 0 \quad (9.7)$$

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 $\varepsilon_1 \geq \varepsilon_2$.

Volumetric strain. DIANA calculates the volumetric strain ε_{vol} by summation of the principal strains:

$$e_{vol} = \varepsilon_1 + \varepsilon_2 + \varepsilon_3 \quad (9.8)$$

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.

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:

$$\mathcal{P}_S = D_{22} \times r \quad (9.9)$$

For this purpose it is strongly recommended to define D_{22} as representative for the shear modulus of the adjacent rock formations according to

$$D_{22} = \frac{E}{2(1+\nu)h} \quad (9.10)$$

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. It is the user's responsibility to make sure that the input of the fault ($h \geq 1$ m)

shear stiffness in the DIANA calculations is correct and based on Eq. (9.10) 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

$$\Delta \mathbf{u} = \begin{Bmatrix} \Delta u_x \\ \Delta u_y \\ \Delta u_z \end{Bmatrix} \quad (9.11)$$

where x denotes the normal direction and y and z the first and second tangential direction [Vol. *Element Library*]. The relative shear displacement r is derived as follows.

$$r = \begin{cases} -\frac{\Delta u_z}{\|\Delta u_z\|} \sqrt{\Delta u_z^2 + \Delta u_y^2} & \text{if } \Delta u_z \neq 0 \\ -\|\Delta u_y\| & \text{if } \Delta u_z = 0 \end{cases} \quad (9.12)$$

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.

9.1.3.2 Element Stresses

In order to determine the stresses, the stress-strain relation \mathbf{D} will be used.

$$\boldsymbol{\sigma} = \mathbf{D} (\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_0) + \boldsymbol{\sigma}_0 \quad (9.13)$$

Here $\boldsymbol{\varepsilon}_0$ and $\boldsymbol{\sigma}_0$ are input entities, whereas $\boldsymbol{\varepsilon}$ is determined according to Eq. (9.1). 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

$$\mathbf{S} = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{xy} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{yz} & \sigma_{zz} \end{bmatrix} \quad (9.14)$$

Equivalent Von Mises stress. DIANA calculates the equivalent Von Mises stress according to

$$\sigma_{eq} = \sqrt{\frac{3(s_{xx}^2 + s_{yy}^2 + s_{zz}^2)}{2} + 3(\sigma_{xy}^2 + \sigma_{yz}^2 + \sigma_{zx}^2)} \quad (9.15)$$

With the deviatoric stresses:

$$\begin{aligned} s_{xx} &= +\frac{2}{3}\sigma_{xx} - \frac{1}{3}\sigma_{yy} - \frac{1}{3}\sigma_{zz} \\ s_{yy} &= -\frac{1}{3}\sigma_{xx} + \frac{2}{3}\sigma_{yy} - \frac{1}{3}\sigma_{zz} \\ s_{zz} &= -\frac{1}{3}\sigma_{xx} - \frac{1}{3}\sigma_{yy} + \frac{2}{3}\sigma_{zz} \end{aligned} \quad (9.16)$$

Principal stresses. DIANA calculates the principal stresses $\sigma_{1,2,3}$ as the roots of Eq. (9.17) ordered such that $\sigma_1 \geq \sigma_2 \geq \sigma_3$.

$$\begin{bmatrix} \sigma_{xx} - \lambda & \sigma_{xy} & \sigma_{xz} \\ \sigma_{xy} & \sigma_{yy} - \lambda & \sigma_{yz} \\ \sigma_{zx} & \sigma_{yz} & \sigma_{zz} - \lambda \end{bmatrix} = 0 \quad (9.17)$$

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 $\sigma_1 \geq \sigma_2$.

Plastic yield. DIANA calculates the plastic yield Y_p from the deviatoric and shear stresses according to

$$Y_p = \frac{s_{xx}^2 + s_{yy}^2 + s_{zz}^2 + 2(\sigma_{xy}^2 + \sigma_{yz}^2 + \sigma_{zx}^2) - 2\sigma_y}{3} \quad (9.18)$$

Where σ_y is the yield strength.

Pressure. DIANA can calculate and output a ‘pressure’ p from three stress components according to

$$p = -\frac{1}{3}(\sigma_{xx} + \sigma_{yy} + \sigma_{zz}) \quad (9.19)$$

Stress invariants. DIANA calculates the stress invariants p' , q , and θ defined as

$$\begin{aligned} p' &= -\frac{1}{3}(\sigma_1 + \sigma_2 + \sigma_3) \\ q &= \sqrt{\frac{1}{2}((\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2)} \\ \theta &= \frac{1}{3} \arcsin \left(-\frac{27}{2} \frac{(\sigma_1 + p')(\sigma_2 + p')(\sigma_3 + p')}{q^3} \right) \end{aligned} \quad (9.20)$$

In this form, p' is the isotropic stress or mean stress, q is the equivalent shear stress, and θ is referred to as Lode’s 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: $|\sigma_1 - \sigma_3|$.

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 [50].

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.

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:

$$\sigma_{xx}, \sigma_{yy}, \sigma_{yx} = \sigma_{xy} \quad (9.21)$$

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:

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} \\ \sigma_{xy} & \sigma_{yy} \end{bmatrix} \quad (9.22)$$

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. 9.2]. Note that α is the angle between x and x' and β

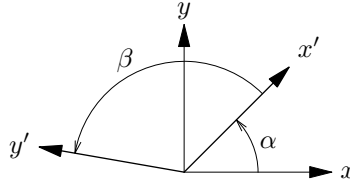


Figure 9.2: Reinforcement directions

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:

$$\sigma' = \frac{1}{\sin \beta^2} \mathbf{R}_\beta \mathbf{R}_\alpha \sigma \mathbf{R}_\alpha^T \mathbf{R}_\beta^T \quad (9.23)$$

where \mathbf{R}_α and \mathbf{R}_β are matrices defined by

$$\mathbf{R}_\alpha = \begin{bmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{bmatrix} \quad \mathbf{R}_\beta = \begin{bmatrix} \sin \beta & -\cos \beta \\ 0 & 1 \end{bmatrix} \quad (9.24)$$

It can be shown that the same transformation rule holds for moments (m_{xx} , m_{yy} , m_{xy}):

$$\mathbf{M}' = \frac{1}{\sin \beta^2} \mathbf{R}_\beta \mathbf{R}_\alpha \mathbf{M} \mathbf{R}_\alpha^T \mathbf{R}_\beta^T \quad (9.25)$$

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}):

$$\mathbf{N}' = \frac{1}{\sin \beta^2} \mathbf{R}_\beta \mathbf{R}_\alpha \mathbf{N} \mathbf{R}_\alpha^T \mathbf{R}_\beta^T \quad (9.26)$$

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' .

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

$$m'_1 = m'_{xx} \pm m'_{xy} \quad m'_2 = m'_{yy} \pm m'_{xy} \quad (9.27)$$

These formulas represent reinforcement moments per unit distance perpendicular to the reinforcement axes x' and y' [Fig. 9.3]. And thus, the following moments need to be

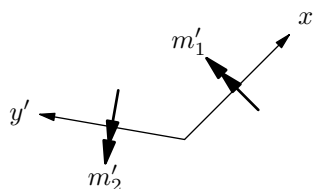


Figure 9.3: Reinforcement moments \perp reinforcement

supported by the reinforcement in the upper and lower plane of the plate.

$$m'_{1,\text{up}} = m'_{xx} + |m'_{xy}| \quad m'_{2,\text{up}} = m'_{yy} + |m'_{xy}| \quad (9.28)$$

$$m'_{1,\text{lo}} = m'_{xx} - |m'_{xy}| \quad m'_{2,\text{lo}} = m'_{yy} - |m'_{xy}| \quad (9.29)$$

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

$$q' = \sqrt{q_x^2 + q_y^2} \quad (9.30)$$

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.

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

$$n'_{1,up} = \frac{n'_{xx}}{2} + \left| \frac{n'_{xy}}{2} \right| \quad n'_{2,up} = \frac{n'_{yy}}{2} + \left| \frac{n'_{xy}}{2} \right| \quad (9.31)$$

$$n'_{1,lo} = \frac{n'_{xx}}{2} + \left| \frac{n'_{xy}}{2} \right| \quad n'_{2,lo} = \frac{n'_{yy}}{2} + \left| \frac{n'_{xy}}{2} \right| \quad (9.32)$$

These formulas represent forces in reinforcement directions per unit distance perpendicular to the reinforcement directions [Fig. 9.4]. Note that the forces in upper and lower

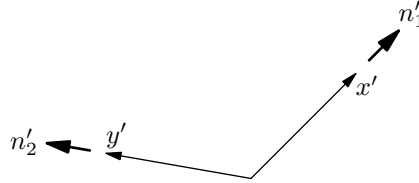


Figure 9.4: Forces in reinforcement directions

plane are the same, in the absence of bending moments. The latter four stress quantities are used to dimension reinforcement in plane stress elements. 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.

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. 9.5].

The combined reinforcement forces read, in x' -direction

$$n'^c_{xx,up} = \frac{n'_{xx}}{2} + \frac{m'_{xy}}{z_d} \quad n'^c_{xy,up} = \left| \frac{n'_{xy}}{2} + \frac{m'_{xy}}{z_d} \right| \quad (9.33)$$

$$n'^c_{xx,lo} = \frac{n'_{xx}}{2} - \frac{m'_{xy}}{z_d} \quad n'^c_{xy,lo} = \left| \frac{n'_{xy}}{2} - \frac{m'_{xy}}{z_d} \right| \quad (9.34)$$

and in y' -direction:

$$n'^c_{yy,up} = \frac{n'_{yy}}{2} + \frac{m'_{xy}}{z_d} \quad n'^c_{xy,up} = \left| \frac{n'_{xy}}{2} + \frac{m'_{xy}}{z_d} \right| \quad (9.35)$$

$$n'^c_{yy,lo} = \frac{n'_{yy}}{2} - \frac{m'_{xy}}{z_d} \quad n'^c_{xy,lo} = \left| \frac{n'_{xy}}{2} - \frac{m'_{xy}}{z_d} \right| \quad (9.36)$$

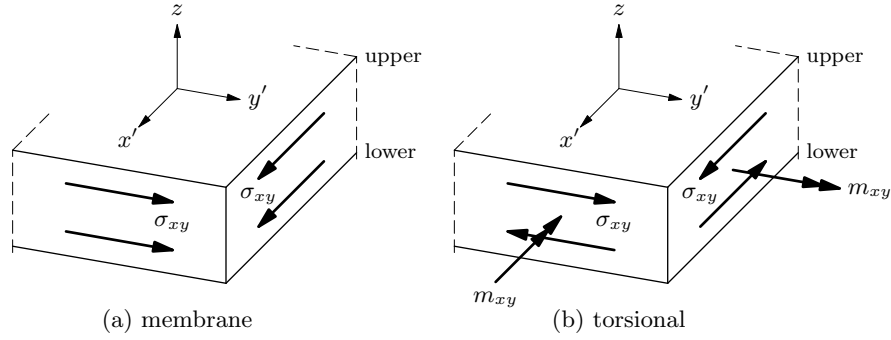


Figure 9.5: Shear stresses

where z_d denotes the ‘absolute internal beam arm’ for the moment. This beam arm follows from

$$z_d = (h_t - co) \times z_r \quad (9.37)$$

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

$$n_{1,up}^c = n_{xx,up}^c + n_{xy,up}^c \quad n_{2,up}^c = n_{yy,up}^c + n_{xy,up}^c \quad (9.38)$$

$$n_{1,lo}^c = n_{xx,lo}^c + n_{xy,lo}^c \quad n_{2,lo}^c = n_{yy,lo}^c + n_{xy,lo}^c \quad (9.39)$$

Additionally, the transverse shear forces q_x and q_y are used to dimension transverse shear reinforcement by means of the quantity:

$$q' = \sqrt{q_x^2 + q_y^2} \quad (9.40)$$

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 $n_{1,up}^c$ denotes a tension force and hence upper plane reinforcement in x' direction is necessary.

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. 9.6].

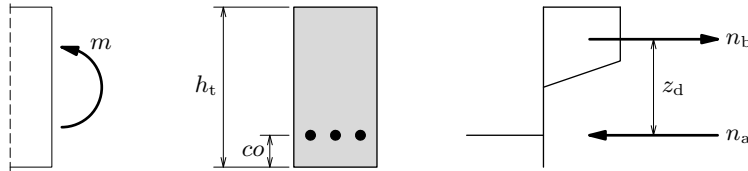


Figure 9.6: Internal beam arm for bending only

In the case of a compressive normal force, an additional eccentricity should be taken into account¹ [Fig. 9.7]: $n_b - n_a = n_e$ and $n_a \times z_d = m$.

$n_b + n_a = n_e$ and $(n_a - \frac{1}{2}n_e) \times z_d = m$ [Fig. 9.8].

¹ See for instance Dutch regulations VB74-84, art. E304.3.3, NEN 3880.

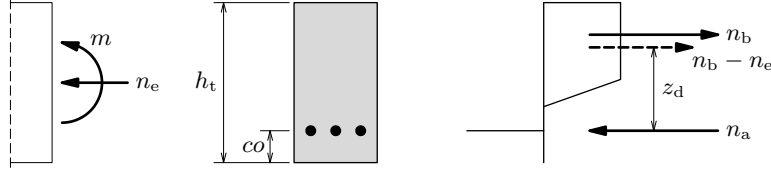


Figure 9.7: Internal beam arm for bending plus compressive force

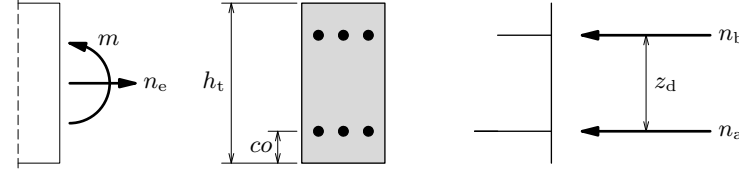


Figure 9.8: Internal beam arm for bending plus large tension force

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:

$$m'_{1,\text{up}} = 500 + 100 = 600 \text{ kN} \quad m'_{2,\text{up}} = 100 + 100 = 200 \text{ kN}$$

$$m'_{1,\text{lo}} = 500 - 100 = 400 \text{ kN} \quad m'_{2,\text{lo}} = 100 - 100 = 0 \text{ kN}$$

If there are only membrane forces, the required reinforcement should be determined from

$$n'_{1,\text{up}} = \frac{-250}{2} + \left| \frac{20}{2} \right| = -115 \text{ N/mm} \quad n'_{2,\text{up}} = \frac{-50}{2} + \left| \frac{20}{2} \right| = -15 \text{ N/mm}$$

$$n'_{1,\text{lo}} = \frac{-250}{2} + \left| \frac{20}{2} \right| = -115 \text{ N/mm} \quad n'_{2,\text{lo}} = \frac{-50}{2} + \left| \frac{20}{2} \right| = -15 \text{ N/mm}$$

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 $z_d = (550 - 50) \times 0.8 = 400 \text{ mm}$):

$$n'^c_{xx,\text{up}} = \frac{-250}{2} + \frac{500000}{400} = 1125 \text{ N/mm}$$

$$n'^c_{xy,\text{up}} = \left| \frac{20}{2} + \frac{100000}{400} \right| = 260 \text{ N/mm}$$

$$n'^c_{yy,\text{up}} = \frac{-50}{2} + \frac{100000}{400} = 225 \text{ N/mm}$$

and in the lower plane:

$$n'^c_{xx,\text{lo}} = \frac{-250}{2} - \frac{500000}{400} = -1375 \text{ N/mm}$$

$$n'^c_{xy,\text{lo}} = \left| \frac{20}{2} - \frac{100000}{400} \right| = 240 \text{ N/mm}$$

$$n'^c_{yy,\text{lo}} = \frac{-50}{2} - \frac{100000}{400} = -275 \text{ N/mm}$$

Hence, the following (extreme) values for the reinforcement forces are obtained, in the upper plane:

$$n_{1,\text{up}}^c = 1125 + 260 = 1385 \text{ N/mm} \quad n_{2,\text{up}}^c = 225 + 260 = 485 \text{ N/mm}$$

and in the lower plane:

$$n_{1,\text{lo}}^c = -1375 + 240 = -1135 \text{ N/mm} \quad n_{2,\text{lo}}^c = -275 + 240 = -35 \text{ N/mm}$$

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.

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:

$$\sigma'_{xx} = \frac{d\sigma_{xx}}{dx} \quad (9.41)$$

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 [§ 9.1.3.2] 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.

Shear capacity and hydrostatic pressure capacity DIANA calculates the shear-stress capacity ψ and the hydrostatic pressure capacity χ of stress against Mohr–Coulomb failure criterion, which are defined as:

$$\psi = \frac{q}{q_{\text{mc}}}, \quad (9.42)$$

and

$$\chi = \frac{3 - \sin \phi}{6 \sin \phi} (q_{\text{mc}} - q), \quad (9.43)$$

with

$$q = \sqrt{\frac{1}{2}((\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2)}, \quad (9.44)$$

and

$$q_{\text{mc}} = \frac{6 \sin \phi}{3 - \sin \phi} p + \frac{6c \cos \phi}{3 - \sin \phi}. \quad (9.45)$$

Where p is the hydrostatic pressure, and c and ϕ are the cohesion and the friction angle for the Mohr–Coulomb material, respectively. For interface elements q_{mc} is defined as

$$q_{\text{mc}} = c + p \cdot \tan \phi \quad (9.46)$$

and the pressure capacity χ is defined as

$$\chi = (q_{\text{mc}} - q) \cdot \tan \phi \quad (9.47)$$

where p is the normal traction and q is defined as the absolute sum of the combined shear tractions.

Safety factors with respect to the biaxial failure envelope. In DIANA a specific operation is available for output of Safety Factors for concrete under static and dynamic loading conditions with reference to a biaxial failure envelope:

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

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

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

$$r = \sqrt{(\sigma_1^2 + \sigma_2^2)} \quad (9.48)$$

To output safety factors FS for concrete under static and dynamic loading conditions with reference to a biaxial failure envelope, the user must define the static uniaxial compressive strength of the concrete f_c , the transformation factor from static to dynamic strength X_{rate} , and the transformation factor from apparent to actual stiffness X_{spec} . These two transformation factors are used to calculate the static tensile strength, and the dynamic compressive and tensile strength. Furthermore, the safety factor for the compressive and tensile strength under dynamic and static loading can be set by the user. The application of transformation factors and safety factors is specified later in this paragraph. The static compressive strength can be defined either by a constant value or as a function of time or maturity, using a multilinear diagram.

The static tensile strength f_t is calculated as follows:

$$f_t = X_{spec} \cdot f_c^{\frac{2}{3}} \quad (9.49)$$

The dynamic compressive strength f_{cd} and dynamic tensile strength f_{td} are defined by:

$$f_{cd} = X_{rate} \cdot f_c \quad (9.50)$$

$$f_{td} = X_{rate} \cdot f_t \quad (9.51)$$

The biaxial failure envelopes for concrete under static loading and under dynamic loading are defined in the stress surface which is expanded by the maximum principle stress σ_1 and the minimum principle stress σ_2 as displayed in [Fig. 9.9]

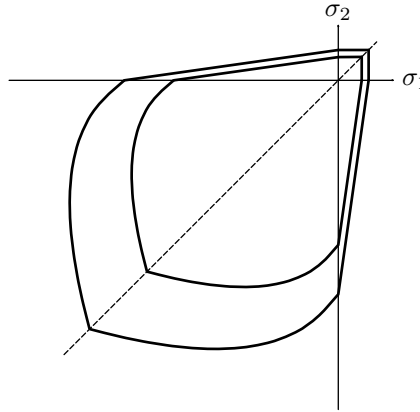


Figure 9.9: Static (inner) and dynamic (outer) concrete biaxial failure envelope

The failure envelope consists of three zones which are each defined as function of the ratio α between the maximum principle stress σ_1 and the minimum principle stress σ_2 :

$$\alpha = \frac{\sigma_1}{\sigma_2} \quad (9.52)$$

For zone 1 ($0 < \alpha < 1$):

$$R = \frac{(1 + 3.65\alpha)\sqrt{(1 + \alpha^2)}f_c}{(1 + \alpha)^2 f_{sc}} \quad (9.53)$$

For zone 2 ($\alpha < 0$):

$$R = \frac{\sqrt{(1 + \alpha^2)}f_c}{(1 - \alpha \frac{f_c}{f_t})f_{si}} \quad (9.54)$$

For zone 3 ($\alpha > 1$):

$$R = \frac{\sqrt{(1 + \alpha^2)} f_t}{\alpha f_{st}} \quad (9.55)$$

Where R is the distance from the origin to the failure envelope, and f_{sc} and f_{st} are the safety factors with reference to the load-cases:

- Static Usual
- Static Unusual
- Dynamic Unusual
- Dynamic Extreme

And where f_{si} is the intermediate safety factor:

$$f_{si} = f_{st} + \frac{f_{sc} - f_{st}}{f_c} |\sigma_1| \quad (9.56)$$

Note that all Safety Factors are limited to 100.

9.1.3.3 Nodal Forces

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

$$\mathbf{f}_{e,\text{in}} = \mathbf{K}_e \mathbf{u}_e^{\text{no}} - \int_{V_e} \mathbf{B}^T \mathbf{C} \boldsymbol{\varepsilon}_0 dV + \int_{V_e} \mathbf{B}^T \boldsymbol{\sigma}_0 dV \quad (9.57)$$

which can be rewritten as

$$\mathbf{f}_{e,\text{in}} = \int_{V_e} \mathbf{B} \boldsymbol{\sigma} dV \quad (9.58)$$

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.

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 $\mathbf{f}_{e,\text{ex}}$ as

$$\mathbf{f}_{e,\text{ex}}^{\text{el}} = \int_{V_e} \mathbf{N} \mathbf{p}_{V_e} dV + \int_{A_e} \mathbf{N} \mathbf{p}_{A_e} dA \quad (9.59)$$

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 $\mathbf{f}_{\text{ex}}^{\text{el}}$. The vector $\mathbf{f}_{\text{ex}}^{\text{pt}}$ contains the nodal point loads, so the total external nodal loads can be defined as

$$\mathbf{f}_{\text{ex}} = \mathbf{f}_{\text{ex}}^{\text{el}} + \mathbf{f}_{\text{ex}}^{\text{pt}} \quad (9.60)$$

² This definition has been made for pragmatic reasons only and has not much meaning in linear elastic analysis.

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} :

$$\mathbf{f}_r = \mathbf{f}_{ex} - \mathbf{f}_{in} \quad (9.61)$$

In case of static equilibrium the internal forces are equal to the external forces for unconstrained degrees of freedom:

$$\mathbf{f}_{in} = \mathbf{f}_{ex} \quad (9.62)$$

Therefore, the residual forces for unconstrained degrees of freedom are zero:

$$\mathbf{f}_r = \mathbf{f}_{ex} - \mathbf{f}_{in} = 0 \quad (9.63)$$

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:

$$\mathbf{f}_R = -\mathbf{f}_r \quad (9.64)$$

9.2 Eigenvalue Analysis

This section 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, effective mass, modal mass, modal damping factors, participation factors and vectors. By doing an eigenvalue analysis the finite element model with node based degrees of freedom can be converted into characteristic modes. Typical characteristic modes that can be calculated by an eigenvalue analysis are free vibration modes for a vibration or dynamic analysis and buckling modes for a stability analysis.

In an eigenvalue analysis the following operations are performed subsequently:

- Evaluate model
- Specify the type of eigenvalue problem
- Calculate the eigenvalues and eigenmodes
- Calculate the Rayleigh damping coefficients (optionally)
- Output

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. (10.93) p. 203] 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.

9.2.1 Model Evaluation

During model evaluation the definitions of elements are checked, including material and geometrical properties which are assigned to elements; the element assembly, or ordering of element and nodal variables in the system set of degrees of freedom are defined; the element stiffness matrices are defined; and the load vectors are set-up similar as for linear static analysis [§ 9.1.1].

9.2.2 Type of Eigenvalue Problem

DIANA offers three types of eigenvalue analysis: The standard eigenvalue problem, free vibration and linearized buckling.

9.2.2.1 Standard Eigenvalue problem

The simplest eigenvalue problem is the standard eigenproblem, which can be used for example to check whether the model is sufficiently constraint to perform any other analysis:

$$\mathbf{K}\phi = \lambda\phi \quad (9.65)$$

where $\mathbf{K}_{n \times n}$ is the symmetric stiffness matrix of the finite element model. The dimension of the stiffness matrix \mathbf{K} is defined by the number of independent equations n and is denoted as order n . There are n eigenvalues and corresponding eigenvectors ϕ_i satisfying Eq. (9.65). The i -th eigenpair is marked as (λ_i, ϕ_i) if the eigenvalues are ordered increasingly:

$$\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_{n-1} \leq \lambda_n \quad (9.66)$$

The solution for p eigenpairs can be written as

$$\mathbf{K}\Phi = \Phi\Lambda \quad (9.67)$$

where $\Phi_{n \times p}$ is the matrix with eigenvectors and $\Lambda_{p \times p}$ is a diagonal matrix with the corresponding eigenvalues.

For the stiffness matrix \mathbf{K} , DIANA can apply the linear elastic stiffness with a geometric-stress stiffness matrix added optionally. This solves the eigenproblem

$$(\mathbf{K} + \mathbf{K}_G)\phi = \lambda\phi$$

To setup the geometric stress-stiffness matrix \mathbf{K}_G from a stress field you may specify a load set number. Default is the lowest available load set number. An optional parameter specifies a multiplication factor.

Alternatively the tangential stiffness from a previously executed nonlinear analysis can be applied for \mathbf{K} . This solves the eigenproblem

$$\mathbf{K}_T\phi = \lambda\phi$$

Note that tangent stiffness matrices are only available when the nonlinear analysis involves a Newton-Raphson iteration method. 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.

9.2.2.2 Free Vibration

A very frequently occurring generalized eigenproblem is the *free vibration* equation to be solved in the mode superposition method written as

$$\mathbf{K}\phi = \omega^2\mathbf{M}\phi \quad (9.68)$$

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. (9.67) the solution for p natural frequencies squared and the corresponding mode shape vectors of Eq. (9.68) can be written as

$$\mathbf{K}\Phi = \mathbf{M}\Phi\Omega^2 \quad (9.69)$$

For the stiffness matrix \mathbf{K} , DIANA can apply the linear elastic stiffness with a geometric-stress stiffness matrix added optionally. This solves the eigenproblem

$$(\mathbf{K} + \mathbf{K}_G)\phi = \omega^2\mathbf{M}\phi$$

To setup the geometric stress-stiffness matrix \mathbf{K}_G from a stress field you may specify a load set number. Default is the lowest available load set number. An optional parameter specifies a multiplication factor. A typical example is the calculation of the natural frequencies of a prestressed structure, for instance a guitar string.

Alternatively the tangential stiffness from a previously executed nonlinear analysis can be applied for \mathbf{K} . 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 involves a Newton-Raphson iteration method. 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.

9.2.2.3 Linearized Buckling

Another possible generalized eigenproblem can be encountered in *stability* analysis. In stability analysis the problem is solved by linearized buckling analysis with

$$\mathbf{K} \phi = \lambda \mathbf{K}_G \phi \quad (9.70)$$

where \mathbf{K} is the stiffness matrix and \mathbf{K}_G the geometric stress-stiffness matrix. However, the analysis description is also applicable to the solution of the other problems. For example, to solve the standard eigenproblem in Eq. (9.65), the mass matrix must be replaced by an identity matrix \mathbf{I} and in case of a linearized buckling analysis Eq. (9.70) the mass matrix \mathbf{M} must be replaced by the geometric stress-stiffness matrix \mathbf{K}_G .

9.2.3 Execute Eigenvalue Analysis

The user specifies whether the Arnoldi based procedure or the FEAST method [60] based eigenvalue analysis procedure must be applied. Both methods make use of parallel processing. The FEAST method based eigenvalue analysis calculates all eigenpairs within a user-defined range using the Intel MKL Extended Eigensolver [81], whereas the Arnoldi method based eigenvalue analysis calculates a user-defined number of eigenpairs.

9.2.4 Calculate Rayleigh Damping Coefficients

In a free vibration eigenvalue analysis you may calculate the Rayleigh damping coefficients a and b from two frequencies to set up the damping matrices \mathbf{C} according to

$$\mathbf{C} = a \mathbf{M} + b \mathbf{K} \quad (9.71)$$

The frequencies are specified explicitly or as eigenfrequencies either by their mode numbers or by the cumulative effective mass contribution in global directions.

9.2.4.1 Shifting

In an eigenvalue problem, a user-specified shift factor μ 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. Shifting the stiffness matrix \mathbf{K} may help you to find this eigenvalue and -mode.

- *Perturbation analysis.* In case of perturbation analysis, 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

$$\hat{\mathbf{K}}\phi = \hat{\omega}^2 \mathbf{M}\phi \quad (9.72)$$

$$\hat{\mathbf{K}} = \mathbf{K} + \mu \mathbf{M} \quad (9.73)$$

$$\hat{\omega}^2 = \omega^2 + \mu \quad (9.74)$$

in which $\hat{\mathbf{K}}$ is the modified stiffness matrix, the eigenvalue $\lambda \equiv \omega^2$ is related by Eq. (9.74) and the eigenvector ϕ is unchanged.

9.2.5 Output

DIANA always writes the following general information resulting from an eigenvalue analysis to the standard output file *file.out*: eigenvalues [§ 9.2.5.1] and relative errors [§ 9.2.5.2]. For a free vibration eigenvalue analysis additional information may be written: generalized masses [§ 9.2.6], participation factors and vectors [§ 9.2.6.1], modal damping factors [§ 9.2.6.4], effective masses [§ 9.2.6.2]. and modal masses [§ 9.2.6.3]. The eigenmodes may also be output [§ 9.2.5.3].

9.2.5.1 Eigenvalues and Eigenfrequencies

The calculated natural circular frequencies ω of the generalized eigenproblem of Eq. (9.68) are expressed in radians per second. The corresponding period T may be computed with

$$\omega T = 2\pi \quad \therefore \quad T = \frac{2\pi}{\omega} \quad (9.75)$$

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

$$f = \frac{1}{T} = \frac{\omega}{2\pi} \quad (9.76)$$

and expressed in hertz, whereby one hertz is one cycle per second (CPS). For the standard eigenproblem as given in Eq. (9.65), the eigenvalue λ can be obtained by taking the frequency ω squared

$$\lambda \equiv \omega^2 = 4\pi^2 f^2 \quad (9.77)$$

The quantities ω and λ can be determined from relation Eq. (9.76) and Eq. (9.77) respectively.

If the eigenproblem has been shifted by a factor μ according to Eq. (9.73), DIANA corrects the actual frequencies according to Eq. (9.74) with

$$\omega = (\hat{\omega}^2 - \mu)^{\frac{1}{2}} = \left(4\pi^2 \hat{f}^2 - \mu\right)^{\frac{1}{2}} \quad (9.78)$$

and the eigenvalues with

$$\lambda = (\hat{\omega}^2 - \mu) = \left(4\pi^2 \hat{f}^2 - \mu\right) \quad (9.79)$$

where $\hat{\omega}$ and \hat{f} are the shifted frequencies and ω and f are the output frequencies.

The resulting eigenvalues are output in ascending order of the absolute value, their type depends on the type of eigenvalue analysis performed.

- The standard eigenproblem yields the ‘real’ eigenvalues λ from Eq. (9.65) on page 146.

- The free vibration eigenproblem yields the natural frequencies $f = \omega/2\pi$ from Eq. (9.68) on page 146. 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. 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. (9.70) on page 147.

9.2.5.2 Relative Errors

The error measures ϵ_i for each eigenvalue and eigenmodes approximation is output. The error measure for the calculated eigenvalue and eigenvector approximation $\tilde{\lambda}$ and $\tilde{\phi}$ is determined as

$$\epsilon_i = \frac{\|\mathbf{K}\tilde{\phi}_i - \tilde{\lambda}_i \mathbf{M}\phi_i\|_2}{\|\mathbf{K}\tilde{\phi}_i\|_2} \quad (9.80)$$

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.

9.2.5.3 Eigenmodes and Eigenvectors

DIANA can output the eigenmodes or eigenvectors, i.e. the displacements or deformed geometry of a structure, after an eigenvalue analysis. Prior to outputting them, DIANA normalizes the eigenmodes in such a way that the largest translation displacement component has a value of 1. Alternatively, DIANA can normalize 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. (9.82) on page 149. For the other types of eigenvalue analysis the mass matrix will be replaced by the appropriate matrix.

9.2.6 Generalized Mass

In a free vibration analysis DIANA will output for each eigenfrequency f_i the corresponding generalized mass m_{ii} by

$$m_{ii} = \phi_i^T \mathbf{M} \phi_i \quad (9.81)$$

With the eigenvectors ϕ_i normalized such that

$$m_{ii} = 1 \quad (9.82)$$

In DIANA all eigenvectors are normalized such that the corresponding generalized mass is equal to one.

9.2.6.1 Participation Factors and Vectors

In a free vibration analysis DIANA will output for each eigenfrequency f_i the corresponding participation factor γ_i [Eq. (9.83) p. 149] and the direction dependent participation factors Γ_i for translations and, if available, rotations in global X , Y , and Z direction.

Participation factors. For each calculated frequency f_i , DIANA determines the corresponding participation factor by

$$\gamma_i = \frac{\phi_i^T \mathbf{M} \mathbf{i}}{m_{ii}} \quad (9.83)$$

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. (9.82).

Participation vectors. For each calculated frequency f_i , DIANA determines the corresponding participation vector by

$$\phi_{p.i} = \gamma_i \phi_i \quad (9.84)$$

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:

$$\sum_{i=1}^n \gamma_i \phi_i = \mathbf{i} \quad (9.85)$$

Direction dependent participation factors. In a free vibration analysis DIANA will output for each eigenfrequency f_i the corresponding direction dependent participation factors Γ_i for translations and, if available, rotations in global X , Y , and Z direction.

$$\begin{aligned} \Gamma_{t_{Xi}} &= \frac{\mathbf{l}_{t_{Xi}}}{m_{ii}} \\ \Gamma_{t_{Yi}} &= \frac{\mathbf{l}_{t_{Yi}}}{m_{ii}} \\ \Gamma_{t_{Zi}} &= \frac{\mathbf{l}_{t_{Zi}}}{m_{ii}} \\ \Gamma_{r_{Xi}} &= \frac{\mathbf{l}_{r_{Xi}}}{m_{ii}} \\ \Gamma_{r_{Yi}} &= \frac{\mathbf{l}_{r_{Yi}}}{m_{ii}} \\ \Gamma_{r_{Zi}} &= \frac{\mathbf{l}_{r_{Zi}}}{m_{ii}} \end{aligned} \quad (9.86)$$

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

$$\mathbf{l}_i = \phi_i^T \mathbf{M} \mathbf{r} \quad (9.87)$$

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. (9.82).

9.2.6.2 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.

$$\begin{aligned} m_{eff.t_{X.i}} &= \frac{\mathbf{l}_{t_{Xi}}^2}{m_{ii}} \\ m_{eff.t_{Y.i}} &= \frac{\mathbf{l}_{t_{Yi}}^2}{m_{ii}} \\ m_{eff.t_{Z.i}} &= \frac{\mathbf{l}_{t_{Zi}}^2}{m_{ii}} \end{aligned} \quad (9.88)$$

Where \mathbf{l}_t are the coefficient vectors according to Eq. (9.87). The effective mass is also printed as percentage of the total structural mass of the model for each eigenfrequency f_i . Furthermore, the cumulative percentage of effective mass with respect to the total structural mass is printed.

For fluid-structure analysis the mass contribution of the fluid may result in (cumulative) percentages larger than 100 %, because the percentage is given with respect to the total mass of only the structural part of the model.

9.2.6.3 Modal Mass

In a free vibration analysis DIANA will output for each eigenfrequency f_i the corresponding modal masses $m_{mod.ii}$ by

$$m_{mod.ii} = \phi_i^T \mathbf{M} \phi_i \quad (9.89)$$

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. The modal mass is also printed as percentage of the total structural mass of the model for each eigenfrequency f_i .

For fluid-structure analysis the mass contribution of the fluid may result in percentages larger than 100 %, because the percentage is given with respect to the total mass of only the structural part of the model.

9.2.6.4 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 according to

$$h_i = \frac{\sum_{j=1}^n h_{e,j} \phi_i^T \mathbf{K}_j \phi_i}{\sum_{j=1}^n \phi_i^T \mathbf{K}_j \phi_i} \quad (9.90)$$

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.

9.3 Buckling Analysis

In this section we will discuss how to perform an 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.

The set of equations representing nodal equilibrium are written as

$$\mathbf{r}(\mathbf{u}) = \mathbf{f}(\mathbf{u}) \quad (9.91)$$

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

$$\mathbf{r}(\mathbf{u}_{crit}) = \mathbf{f}(\mathbf{u}_{crit}) \quad (9.92)$$

and

$$\mathbf{r}(\mathbf{u}_{crit} + \delta \mathbf{u}) = \mathbf{f}(\mathbf{u}_{crit} + \delta \mathbf{u}) \quad (9.93)$$

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 $\mathbf{u}_{crit} + \delta \mathbf{u}$ is ‘close to’ \mathbf{u}_{crit} , Eq. (9.93) can be linearized with respect to the nodal degrees of freedom:

$$\mathbf{r}(\mathbf{u}_{crit}) + \left(\frac{\partial \mathbf{r}}{\partial \mathbf{u}} \right)_{crit} \delta \mathbf{u} \approx \mathbf{f}(\mathbf{u}_{crit}) + \left(\frac{\partial \mathbf{f}}{\partial \mathbf{u}} \right)_{crit} \delta \mathbf{u} \quad (9.94)$$

Introducing the tangent stiffness matrix

$$\mathbf{K} = \left(\frac{\partial \mathbf{r}}{\partial \mathbf{u}} \right)_{crit} \quad (9.95)$$

leads to

$$\mathbf{r}(\mathbf{u}_{\text{crit}}) + \mathbf{K}\delta\mathbf{u} \approx \mathbf{f}(\mathbf{u}_{\text{crit}}) + \left(\frac{\partial\mathbf{f}}{\partial\mathbf{u}}\right)_{\text{crit}} \delta\mathbf{u} \quad (9.96)$$

And subtracting Eq. (9.92) yields

$$\mathbf{K}\delta\mathbf{u} \approx \left(\frac{\partial\mathbf{f}}{\partial\mathbf{u}}\right)_{\text{crit}} \delta\mathbf{u} \quad (9.97)$$

For conservative loading, the external force vector \mathbf{f} does not depend on \mathbf{u} and thus

$$\mathbf{K}\delta\mathbf{u} \approx \mathbf{0} \quad (9.98)$$

9.3.1 Applicability of Elements

This section outlines which of the stability analysis options may be applied for the various structural elements. Table 9.2 outlines the applicability of the various options for stabil-

Table 9.2: STABILITY ANALYSIS FOR STRUCTURAL ELEMENTS

	Truss	Beam	Pl. stress	Plate bend.	Pl. strain	Axisymm.	Fl. shell	Cu. shell	Solid
Euler	*	*	fk	-	g	g	i	*	g
Initial displacements	*	c	fk	-	gj	gh	i	*	g
Imperfections	a	cd	fk	-	gj	gh	i	*	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. (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. The spline elements are the only flat shell elements that can be used in stability analysis. These elements accept initial displacements and imperfections and 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.

9.3.2 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. However, often it is a relatively simple and effective method to get a fair impression of a structure’s buckling modes. Only geometrical nonlinear effects are taken into account, and thus the tangent stiffness matrix reads (see Bathe [6])

$$\mathbf{K} = \mathbf{K}_L + \mathbf{K}_{NL} \quad (9.99)$$

where

$$\mathbf{K}_L = \int_{V_0} \mathbf{B}_L^T \mathbf{D} \mathbf{B}_L dV_0 \quad \text{with:} \quad \mathbf{B}_L = \mathbf{B}_{L0} + \mathbf{B}_{L1} \quad (9.100)$$

with \mathbf{B}_{L1} due to an initial displacement effect. See Bathe [6] for a definition of the matrices \mathbf{B}_{L0} and \mathbf{B}_{L1} . Furthermore

$$\mathbf{K}_{NL} = \int_{V_0} \mathbf{B}_{NL}^T \boldsymbol{\tau} \mathbf{B}_{NL} dV_0 \quad (9.101)$$

where \mathbf{B}_{NL} is defined in Bathe [6]. 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. (9.91) linearized is known, i.e., \mathbf{u}_{lin} results from

$$\mathbf{K}_{L0}\mathbf{u}_{\text{lin}} = \mathbf{f} \quad (9.102)$$

where

$$\mathbf{K}_{L0} = \int_{V_0} \mathbf{B}_{L0}^T \mathbf{D} \mathbf{B}_{L0} dV_0 \quad (9.103)$$

with constant external loading \mathbf{f} . It can be questioned whether solutions \mathbf{u}_{crit} satisfying Eq. (9.92) and Eq. (9.98) exist such that

$$\mathbf{u}_{\text{crit}} = \lambda_{\text{crit}} \mathbf{u}_{\text{lin}} \quad (9.104)$$

Note that $\lambda_{\text{crit}} \mathbf{u}_{\text{lin}}$ would result from a loading $\mathbf{f}_{\text{crit}} = \lambda_{\text{crit}} \mathbf{f}$ on the linear system, as can be shown by multiplication of Eq. (9.102) by λ_{crit} :

$$\lambda_{\text{crit}} \mathbf{K}_{L0} \mathbf{u}_{\text{lin}} = \lambda_{\text{crit}} \mathbf{f} \quad (9.105)$$

Parameter λ_{crit} will be determined from the instability condition Eq. (9.98). 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} :

$$\mathbf{K}_{LL}(\mathbf{u}_{\text{lin}}) = \int_{V_0} (\mathbf{B}_{L1}^T \mathbf{D} \mathbf{B}_{L0} + \mathbf{B}_{L0}^T \mathbf{D} \mathbf{B}_{L1}) dV_0 \quad (9.106)$$

- The second Piola–Kirchhoff stresses:

$$\boldsymbol{\tau} \approx \lambda_{\text{crit}} \boldsymbol{\sigma}_{\text{lin}} \quad (9.107)$$

Application of these approximations leads to

$$\left(\mathbf{K}_{L0} + \lambda_{\text{crit}} (\mathbf{K}_{LL}(\mathbf{u}_{\text{lin}}) + \mathbf{K}_G(\mathbf{u}_{\text{lin}})) \right) \delta \mathbf{u} = \mathbf{0} \quad (9.108)$$

where

$$\mathbf{K}_G(\mathbf{u}_{\text{lin}}) = \int_{V_0} \mathbf{B}_{NL}^T \boldsymbol{\sigma}_{\text{lin}} \mathbf{B}_{NL} dV_0 \quad (9.109)$$

and

$$\mathbf{K}_{L0} = \int_{V_0} \mathbf{B}_{L0}^T \mathbf{D} \mathbf{B}_{L0} dV_0 \quad (9.110)$$

Eq. (9.108) is satisfied by nontrivial solutions for $\delta \mathbf{u}$ if

$$\det \left(\mathbf{K}_{L0} + \lambda_{\text{crit}} (\mathbf{K}_{LL}(\mathbf{u}_{\text{lin}}) + \mathbf{K}_G(\mathbf{u}_{\text{lin}})) \right) = 0 \quad (9.111)$$

And so, the general stability conditions Eq. (9.92) and Eq. (9.93) are replaced by the relatively simple conditions Eq. (9.105) and Eq. (9.111) in linear stability analysis. Eq. (9.111) is solved as a generalized eigenproblem written as

$$\mathbf{K}_{L0} \phi_i = -\lambda_{i \text{ crit}} (\mathbf{K}_{LL}(\mathbf{u}_{\text{lin}}) + \mathbf{K}_G(\mathbf{u}_{\text{lin}})) \phi_i \quad (9.112)$$

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*.

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 9.10 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. (9.92) and Eq. (9.98) are satisfied. The force F_{crit} is called

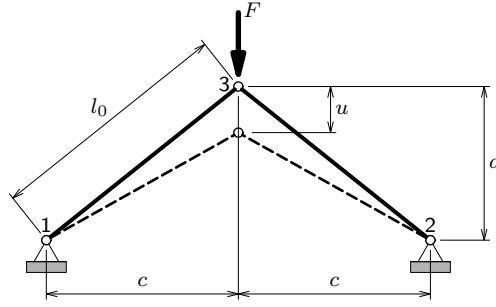


Figure 9.10: Arch, loading and deformation

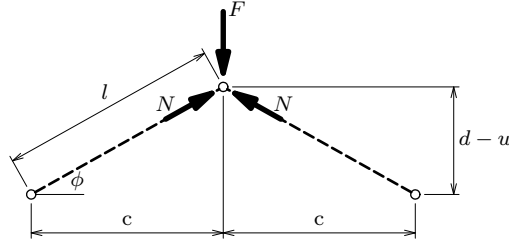


Figure 9.11: Arch, equilibrium node 3 deformed state

the critical load. Firstly, the equilibrium condition Eq. (9.91) will be examined in more detail. Load F is in equilibrium with the normal forces N in the bars as shown in Figure 9.11. Mathematically this reads:

$$2N \sin \phi = 2N \frac{d-u}{l} = F \quad (9.113)$$

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:

$$N = -A_0 \frac{l}{l_0} \tau \quad (9.114)$$

where A_0 is the initial cross-section of the bars. The second Piola–Kirchhoff stress τ is given by

$$\tau = E \gamma \quad (9.115)$$

where E denotes Young's modulus (the bars are assumed to be isotropic) and γ represents the Green–Lagrange strain given by

$$\gamma = \frac{1}{2} \left(\frac{c^2 + (d-u)^2}{l_0^2} - 1 \right) \quad (9.116)$$

Combination of Eq. (9.105) and Eq. (9.108) yields

$$-2 \frac{A_0 \tau}{l_0} (d-u) = F \quad (9.117)$$

Taking the first variation of Eq. (9.117) leads to:

$$2 \frac{A_0}{l_0} \left(\left(\frac{Ed^2}{l_0^2} \right) - \left(2 \frac{Edu}{l_0^2} - \frac{Eu^2}{l_0^2} \right) + (\tau) \right) = 0 \quad (9.118)$$

The first term in the left-hand-side of Eq. (9.118) 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:

$$\tau = \sigma = E \varepsilon \quad (9.119)$$

where ε is the geometrical linear (small) strain:

$$\varepsilon = -\frac{d}{l_0^2} u \quad (9.120)$$

These assumptions lead to a more simple expression for Eq. (9.118):

$$2 \frac{A_0 E d}{l_0^3} (d - 3u) = 0 \quad (9.121)$$

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. (9.121) becomes satisfied for such F . The required value for F_{crit} is determined by substituting $u = \frac{1}{3}d$ into Eq. (9.117).

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. (9.118) without applying any approximations leads to:

$$\frac{EA_0}{l_0^3} (c^2 - l_0^2 + 3(d - u)^2) = 0 \quad (9.122)$$

Eq. (9.122) is satisfied if:

$$u_{\text{crit}} = d \left(1 + \frac{1}{\sqrt{3}} \right) \quad \text{or:} \quad u_{\text{crit}} = d \left(1 - \frac{1}{\sqrt{3}} \right) \quad (9.123)$$

with a load F_{crit} following from Eq. (9.117). 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 .

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.
2. If appropriate, add initial imperfections [§ 9.3.2.1].
3. Perform a stability analysis.
4. Output the buckling modes (optional).

9.3.2.1 Initial Imperfections

Many of DIANA's elements may be applied in Euler stability analysis with initial imperfections [Table 9.2]. 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.

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:

$$\mathbf{K}_{L0}^I = \mathbf{K}_{L0}^I(\mathbf{u}^I) \quad (9.124)$$

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:

$$\det\left(\mathbf{K}_{L0}^I(\mathbf{u}^I) + \lambda(\mathbf{K}_{LL}(\mathbf{u}_{lin}) + \mathbf{K}_G(\mathbf{u}_{lin}))\right) = 0 \quad (9.125)$$

For a fixed imperfection pattern (i.e., user-specified or random), an eigenvalue calculation on Eq. (9.125) 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. (9.125) 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. (9.125), now including \mathbf{u}^I , is performed to add the effect of the imperfections.

9.3.3 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.

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. (9.93) 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. (9.91) $\mathbf{r}(\mathbf{u}_{pb}) = \mathbf{f}(\mathbf{u}_{pb})$ 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

$$\mathbf{u}_{pb} = \lambda\mathbf{u}_{lin} + a_i\phi_i + a_ja_i\mathbf{u}_{ij} \quad (9.126)$$

where \mathbf{u}_{ij} is called the second order displacement vector and a_i should be interpreted as amplitude of the respective mode. In literature³ it is shown that \mathbf{u}_{ij} must be calculated by solving the system

$$(\mathbf{K}_{L0} + \lambda_p\lambda_1\mathbf{K}_G(\mathbf{u}_{lin}))\mathbf{u}_{ij} = \mathbf{f}_{ij} \quad (9.127)$$

where $\lambda_p \neq 1$ must be specified by the user. Applying the orthogonality conditions

$$\phi_k^T \mathbf{K}_{L0} \mathbf{u}_{ij} = 0 \quad \text{with } k = 1, \dots, M \quad (9.128)$$

where \mathbf{f}_{ij} is defined as the mode interaction load vector

$$\mathbf{f}_{ij} = \int_{V_0} \left(\mathbf{B}_{L0}^T \boldsymbol{\sigma}_{ij} + \mathbf{B}_{NL}^T(\phi_i) \boldsymbol{\sigma}_j + \mathbf{B}_{NL}^T(\phi_j) \boldsymbol{\sigma}_i \right) dV_0 \quad (9.129)$$

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

$$P(a_i, \lambda) = \frac{1}{2} \sum_{I=1}^M \left(1 - \frac{\lambda}{\lambda_I} \right) a_I a_I + A_{ijk} a_i a_j a_k + A_{ijkl} a_i a_j a_k a_l \quad (9.130)$$

³See for instance Koiter [43], Van Erp [24], Byskov & Hutchinson [16].

where

$$A_{ijk} = \frac{1}{2} \int_{V_0} \sigma_{ij} \epsilon_k dV_0 \quad (9.131)$$

and

$$A_{ijkl} = \frac{1}{8} \int_{V_0} \sigma_{ij} \epsilon_{kl} dV_0 - \frac{1}{2} \mathbf{u}_{ij} \mathbf{f}_{kl} \quad (9.132)$$

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. (9.126).

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.

9.3.3.1 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 [77] and Rahman [61]. In these works functional notation introduced by Budiansky [14] 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.

Functional notation. If \mathbf{u} and $\boldsymbol{\epsilon}$ denote displacement and strain fields respectively, then in functional notation strain–displacement relation is written as

$$\boldsymbol{\epsilon} = L_1(\mathbf{u}) + \frac{1}{2} L_2(\mathbf{u}) \quad (9.133)$$

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

$$L_2(\mathbf{u} + \mathbf{v}) = L_2(\mathbf{u}) + 2L_{11}(\mathbf{u}, \mathbf{v}) + L_2(\mathbf{v}) \quad (9.134)$$

From Eq. (9.134) it follows that

$$L_{11}(\mathbf{u}, \mathbf{v}) = L_{11}(\mathbf{v}, \mathbf{u}) \quad (9.135)$$

$$L_{11}(\mathbf{u}, \mathbf{u}) = L_2(\mathbf{u}) \quad (9.136)$$

The perturbation method. The perturbation method based postbuckling analysis [§ 9.3.3 p. 157] 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} , ϵ , σ) 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 , ϵ_0 , σ_0)

$$\begin{aligned}\mathbf{u} &= \mathbf{u}_0(\lambda) + \mathbf{u}_i \xi_i + \mathbf{u}_{ij} \xi_i \xi_j + \dots \\ \epsilon &= \epsilon_0(\lambda) + \epsilon_i \xi_i + \epsilon_{ij} \xi_i \xi_j + \dots \\ \sigma &= \sigma_0(\lambda) + \sigma_i \xi_i + \sigma_{ij} \xi_i \xi_j + \dots\end{aligned}\quad (9.137)$$

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 ϵ and σ are the corresponding strains and stresses. In Eq. (9.137) 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:

$$\epsilon_i = L_1(\mathbf{u}_i) + L_{11}(\mathbf{u}_c, \mathbf{u}_i) \quad (9.138)$$

$$\sigma_i = H(\epsilon_i) \quad (9.139)$$

$$\sigma_c \cdot \delta \epsilon_c + \sigma_c \cdot L_{11}(\mathbf{u}_i, \delta \mathbf{u}) = 0 \quad (9.140)$$

where the prebuckling quantities with the subscript $(\cdot)_c$ 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:

$$\xi_I(\lambda - \lambda_I) = a_{Ijk} \lambda_I \xi_j \xi_k + b_{Ijkl} \lambda_I \xi_j \xi_k \xi_l + \dots \quad (9.141)$$

In view of Eq. (9.141), 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}

$$\epsilon_{ij} = L_1(\mathbf{u}_{ij}) + L_{11}(\mathbf{u}_c, \mathbf{u}_{ij}) + \frac{1}{2} L_{11}(\mathbf{u}_i, \mathbf{u}_j) \quad (9.142)$$

$$\sigma_{ij} = H(\epsilon_{ij}) \quad (9.143)$$

$$\sigma_{ij} \cdot \delta \epsilon_c + \sigma_c \cdot L_{11}(\mathbf{u}_{ij}, \delta \mathbf{u}) + \frac{1}{2} [\sigma_i \cdot L_{11}(\mathbf{u}_j, \delta \mathbf{u}) + \sigma_j \cdot L_{11}(\mathbf{u}_i, \delta \mathbf{u})] = 0 \quad (9.144)$$

The second order modes \mathbf{u}_{ij} are further subject to the following orthogonality condition

$$\dot{\sigma}_c \cdot L_{11}(\mathbf{u}_i, \mathbf{u}_{ij}) + \sigma_1 \cdot L_{11}(\dot{\mathbf{u}}_c, \mathbf{u}_{ij}) + \sigma_2 \cdot L_{11}(\dot{\mathbf{u}}_c, \mathbf{u}_i) = 0 \quad (9.145)$$

where $(\dot{\cdot}) = \frac{\partial}{\partial \lambda}(\cdot)$.

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 [24], and Byskov & Hutchinson [16] 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.

$$P(\xi, \lambda) = \frac{1}{2} \sum_{I=1}^m \left(1 - \frac{\lambda}{\lambda_I}\right) (\lambda_I \hat{\Delta}_I) \xi_I^2 + A_{ijk} \xi_i \xi_j \xi_k + A_{ijkl} \xi_i \xi_j \xi_k \xi_l \quad (9.146)$$

where $\hat{\Delta}_I$, A_{ijk} and A_{ijkl} are defined as

$$\hat{\Delta}_I = 2\sigma_I \cdot L_{11}(\dot{\mathbf{u}}_c, \mathbf{u}_I) + \dot{\sigma}_c \cdot L_2(\mathbf{u}_I) \quad (9.147)$$

$$A_{ijk} = \frac{1}{2} L_1(\mathbf{u}_i) \cdot \mathbf{H} L_{11}(\mathbf{u}_j, \mathbf{u}_k) \quad (9.148)$$

$$A_{ijkl} = \frac{1}{4} \{ L_1(\mathbf{u}_k) \cdot \mathbf{H}(L_{11}(\mathbf{u}_l, \mathbf{u}_{ij})) + L_1(\mathbf{u}_l) \cdot \mathbf{H}(L_{11}(\mathbf{u}_k, \mathbf{u}_{ij})) \\ + L_1(\mathbf{u}_{ij}) \cdot \mathbf{H}(L_{11}(\mathbf{u}_k, \mathbf{u}_l)) + \frac{1}{2} L_{11}(\mathbf{u}_i, \mathbf{u}_j) \cdot \mathbf{H}(L_{11}(\mathbf{u}_k, \mathbf{u}_l)) \} \quad (9.149)$$

The reduced set of Eq. (9.141) can be obtained from Eq. (9.146) 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} :

$$b_{Ijkl} = \frac{2}{\lambda_I \hat{\Delta}_I} (A_{Ijkl} + A_{jklI}) \quad (9.150)$$

9.3.4 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. 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.

9.3.5 Effect of imperfection

In § 9.3.2.1 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. (9.141) is modified to

$$\xi_I(\lambda - \lambda_I) = a_{Ijk} \lambda_I \xi_j \xi_k + b_{Ijkl} \lambda_I \xi_j \xi_k \xi_l - \alpha_I \lambda_I \bar{\xi} - \beta_I (\lambda - \lambda_I) \bar{\xi} + \dots \quad (9.151)$$

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

$$\alpha_I = (1/\lambda_I \hat{\Delta}_I) [\boldsymbol{\sigma}_I \cdot L_{11}(\hat{\mathbf{u}}, \mathbf{u}_c) + \boldsymbol{\sigma}_c \cdot L_{11}(\hat{\mathbf{u}}, \mathbf{u}_I)] \quad (9.152)$$

$$\beta_I = (1/\hat{\Delta}_I) \{ \boldsymbol{\sigma}_I \cdot L_{11}(\hat{\mathbf{u}}, \dot{\mathbf{u}}_c) + \dot{\boldsymbol{\sigma}}_c \cdot L_{11}(\hat{\mathbf{u}}, \mathbf{u}_I) + H[L_{11}(\dot{\mathbf{u}}_c, \mathbf{u}_I)] \cdot L_{11}(\hat{\mathbf{u}}, \mathbf{u}_c) \\ - \alpha_I \lambda_I [\boldsymbol{\sigma}_I \cdot L_{11}(\dot{\mathbf{u}}_c, \mathbf{u}_I) + (1/2) \dot{\boldsymbol{\sigma}}_c \cdot L_{11}(\mathbf{u}_I, \mathbf{u}_I) \\ + H[L_{11}(\dot{\mathbf{u}}_c, \mathbf{u}_I)] \cdot L_{11}(\dot{\mathbf{u}}_c, \mathbf{u}_I)] \} \quad (9.153)$$

In case of a linear prebuckling state $\alpha_I = \beta_I = 1$. Eq. (9.151) 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. (9.151) with varying imperfection amplitude $\bar{\xi}$ at very little additional computational expense.

9.3.5.1 Finite Element Implementation

After some algebraic manipulation of Eqs. (9.138), (9.139), and (9.140) 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

$$[\mathbf{K}_{tb} + (\lambda_i - \lambda_b) [\mathbf{K}_D(\mathbf{q}_b, \dot{\mathbf{q}}_b) + \mathbf{K}_G(\dot{\boldsymbol{\sigma}}_b)]] \mathbf{q}_i = 0 \quad (9.154)$$

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. (9.154) 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}_{t_b} . Then, a linearized buckling analysis is done at the base state, which is nothing but the solution of the eigenvalue problem [Eq. (9.154)].

Eqs. (9.142), (9.143), (9.144) are the equations for determination of the second order modes in functional notation. In terms of finite element matrices they can be written as

$$[\mathbf{K}_{t_b} + \phi(\lambda_c - \lambda_b)[\dot{\mathbf{K}}_D(\mathbf{q}_b, \dot{\mathbf{q}}_b) + \dot{\mathbf{K}}_G(\dot{\boldsymbol{\sigma}}_b)]\mathbf{q}_{ij} = \mathbf{g}_{ij} \quad (9.155)$$

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

$$\begin{aligned} \mathbf{g}_{ij} = & \frac{1}{2} [(\mathbf{B}_L + \mathbf{B}_{NL}(\mathbf{q}_c))^T \mathbf{H} \mathbf{B}_{NL}(\mathbf{q}_i) \mathbf{q}_j \\ & + \mathbf{B}_{NL}^T(\mathbf{q}_i) \mathbf{H} [\mathbf{B}_L + \mathbf{B}_{NL}(\mathbf{q}_c)] \mathbf{q}_j \\ & + \mathbf{B}_{NL}^T(\mathbf{q}_j) \mathbf{H} [\mathbf{B}_L + \mathbf{B}_{NL}(\mathbf{q}_c)] \mathbf{q}_i] \end{aligned} \quad (9.156)$$

where \mathbf{H} is the stress-strain matrix. The associated orthogonality constraint as defined by Eq. (9.145) can be translated to finite element context as

$$\mathbf{q}_i^T [\mathbf{K}_D(\mathbf{q}_b, \dot{\mathbf{q}}_b) + \mathbf{K}_G(\dot{\boldsymbol{\sigma}}_b)] \mathbf{q}_{ij} + \frac{1}{2} [\mathbf{H} \mathbf{B}_{NL}(\mathbf{q}_i) \mathbf{q}_i]^T [\mathbf{B}_{NL}(\mathbf{q}_i) \dot{\mathbf{q}}_b] = 0 \quad (9.157)$$

Solution of Eq. (9.155) together with Eq. (9.157) gives the second order modes \mathbf{q}_{ij} .

A concise expression for A_{Ijkl} defined in Eq. (9.149) can be obtained as

$$A_{Ijkl} = \frac{1}{8} \int_v [\mathbf{H} \mathbf{B}_{NL}(\mathbf{q}_i) \mathbf{q}_j]^T [\mathbf{B}_{NL}(\mathbf{q}_k) \mathbf{q}_l] dv - \frac{1}{2} \mathbf{q}_{ij}^T \mathbf{g}_{kl} \quad (9.158)$$

In Eq. (9.158) the integration sign implies integration over the entire structure. The post-buckling coefficients b_{Ijkl} are then computed according to Eq. (9.150). The imperfection form factors (α_I , β_I) are evaluated at each integration point using Eqs. (9.152), (9.153) 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:

$$\mathbf{q}_c \approx \mathbf{q}_b, \quad \dot{\mathbf{q}}_c \approx \dot{\mathbf{q}}_b \quad (9.159)$$

Now $\ddot{\mathbf{q}}_b$ can be approximated as

$$\ddot{\mathbf{q}}_b = \frac{\dot{\mathbf{q}}_b - \dot{\mathbf{q}}_{b-1}}{\Delta \lambda} \quad (9.160)$$

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.

9.3.5.2 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 [14] in a multi-mode context with the inclusion of prebuckling nonlinearity. Due to inertial forces the equilibrium equation takes the form

$$\boldsymbol{\sigma} \cdot \delta \boldsymbol{\epsilon} - \mathbf{f} \cdot \delta \mathbf{u} + M(\ddot{\mathbf{u}}) \cdot \delta \mathbf{u} = \mathbf{0} \quad (9.161)$$

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

$$M(\mathbf{u}) \cdot \mathbf{v} = M(\mathbf{v}) \cdot \mathbf{u} \quad (9.162)$$

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. (9.137) as

$$\begin{aligned} \mathbf{u} &= \lambda F(t) \mathbf{u}_0 + \mathbf{u}_i \xi_i(t) + \mathbf{u}_{ij} \xi_i(t) \xi_j(t) + \dots \\ \boldsymbol{\epsilon} &= \lambda F(t) \boldsymbol{\epsilon}_0 + \boldsymbol{\epsilon}_i \xi_i(t) + \boldsymbol{\epsilon}_{ij} \xi_i(t) \xi_j(t) + \dots \\ \boldsymbol{\sigma} &= \lambda F(t) \boldsymbol{\sigma}_0 + \boldsymbol{\sigma}_i \xi_i(t) + \boldsymbol{\sigma}_{ij} \xi_i(t) \xi_j(t) + \dots \end{aligned} \quad (9.163)$$

By repeating the same procedure as for the static case and neglecting the inertial forces associated with the prebuckling displacements one obtains

$$\begin{aligned} \left(\frac{1}{\omega_I^2} \right) \ddot{\xi}_I(t) + \left[1 - \frac{\lambda F(t)}{\lambda_I} \right] \xi_I(t) + a_{Ijk} \xi_j(t) \xi_k(t) \\ + b_{Ijkl} \xi_j(t) \xi_k(t) \xi_l(t) = \left[\frac{\lambda F(t)}{\lambda_I} \right] \bar{\xi}_I \end{aligned} \quad (9.164)$$

where ω_I^2 is defined as

$$\omega_I^2 = \frac{\boldsymbol{\sigma}_I \cdot \boldsymbol{\epsilon}_I}{M(\mathbf{u}_I) \cdot \mathbf{u}_I} \quad (9.165)$$

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. (9.164) is solved with a standard Runge–Kutta scheme of the 4th order.

In order to account for prebuckling nonlinearity Eq. (9.165) is written as

$$\omega_I^2 = \frac{-\lambda_I \hat{\Delta}_I}{M(\mathbf{u}_I) \cdot \mathbf{u}_I} \quad (9.166)$$

It can be noticed that, the quantity $\boldsymbol{\sigma}_I \cdot \boldsymbol{\epsilon}_I$ in Eq. (9.165) is replaced by $-\lambda_c \hat{\Delta}_I$ in Eq. (9.166) where $\hat{\Delta}_I$ is defined by Eq. (9.147). Finally, Eq. (9.164) is modified to

$$\begin{aligned} \left(\frac{1}{\omega_I^2} \right) \ddot{\xi}_I(t) + \left[1 - \frac{\lambda F(t)}{\lambda_I} \right] \xi_I(t) + a_{Ijk} \xi_j(t) \xi_k(t) + b_{Ijkl} \xi_j(t) \xi_k(t) \xi_l(t) \\ = \alpha_I \bar{\xi}_I - \beta_I \left[1 - \frac{\lambda F(t)}{\lambda_I} \right] \bar{\xi}_I \end{aligned} \quad (9.167)$$

where α_I and β_I are imperfection form factors which are evaluated using Eqs. (9.152) and (9.153). In case of linear prebuckling state both α_I and β_I are unity and Eq. (9.167) becomes identical to Eq. (9.164). 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. (9.164) 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. (9.163).

9.4 Reinforcement Grid Design Checking Analysis

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.

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 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.

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 [50].

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.

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.

Consider the Cauchy stresses following from a linear elastic analysis:

$$\sigma_{xx}, \sigma_{yy}, \sigma_{yx} = \sigma_{xy} \quad (9.168)$$

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:

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} \\ \sigma_{xy} & \sigma_{yy} \end{bmatrix} \quad (9.169)$$

This matrix contains the components of the Cauchy stress tensor with respect to the element x and y axes. However, stress components must be transferred to the axes of the reinforcement, x' and y' .

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. 9.12].

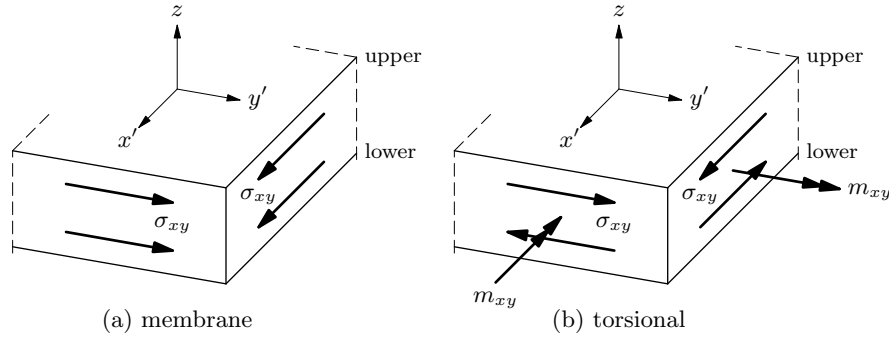


Figure 9.12: Shear stresses

The combined reinforcement forces read, in x' -direction

$$n'_{xx,\text{up}} = \frac{n'_{xx}}{2} + \frac{m'_{xx}}{z_d} \quad n'_{xy,\text{up}} = \left| \frac{n'_{xy}}{2} + \frac{m'_{xy}}{z_d} \right| \quad (9.170)$$

$$n'_{xx,\text{lo}} = \frac{n'_{xx}}{2} - \frac{m'_{xx}}{z_d} \quad n'_{xy,\text{lo}} = \left| \frac{n'_{xy}}{2} - \frac{m'_{xy}}{z_d} \right| \quad (9.171)$$

and in y' -direction:

$$n'_{yy,\text{up}} = \frac{n'_{yy}}{2} + \frac{m'_{yy}}{z_d} \quad n'_{xy,\text{up}} = \left| \frac{n'_{xy}}{2} + \frac{m'_{xy}}{z_d} \right| \quad (9.172)$$

$$n'_{yy,\text{lo}} = \frac{n'_{yy}}{2} - \frac{m'_{yy}}{z_d} \quad n'_{xy,\text{lo}} = \left| \frac{n'_{xy}}{2} - \frac{m'_{xy}}{z_d} \right| \quad (9.173)$$

where z_d denotes the ‘absolute internal beam arm’ for the moment. This beam arm follows from

$$z_d = (h_t - co) \times z_r \quad (9.174)$$

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_t depends on the ratio of bending moments and membrane forces. Several examples for determination of z_t are given further.

Finally, the reinforcement forces per unit length perpendicular to the reinforcement directions x' and y' are given by

$$n'_{1,\text{up}} = n'_{xx,\text{up}} + n'_{xy,\text{up}} \quad n'_{2,\text{up}} = n'_{yy,\text{up}} + n'_{xy,\text{up}} \quad (9.175)$$

$$n'_{1,\text{lo}} = n'_{xx,\text{lo}} + n'_{xy,\text{lo}} \quad n'_{2,\text{lo}} = n'_{yy,\text{lo}} + n'_{xy,\text{lo}} \quad (9.176)$$

Additionally, the transverse shear forces q_x and q_y are used to dimension transverse shear reinforcement by means of the quantity:

$$q' = \sqrt{q_x^2 + q_y^2} \quad (9.177)$$

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 $n'_{1,\text{up}}$ denotes a tension force and hence upper plane reinforcement in x' direction is necessary.

9.4.1 Specific Model Properties

The reinforcement grids to be considered in the design check must be identified by the user. 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.

Specific input parameters for the reinforcement grids in a design checking analysis are:

- the diameter ϕ_x of the bars in the reinforcement grid in local x direction. **phi_x** ($\phi_x \geq 0$) specifies the diameter ϕ_y of the bars in the reinforcement grid in local y direction. ($\phi_y \geq 0$)
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.
- the spacing of the bars in the grid reinforcement in local x direction s_x . **spacing_x** ($s_x \geq 0$) specifies the spacing of the bars in the grid reinforcement in local y direction s_y .
- the relative internal arm z_r . When this parameter is not defined, the default value of 0.9 will be used.
- the length of the line that is required for calculating average results. The default value is two times the effective height of the element or composed element in which the grid is located.
- the Young’s modulus E of the reinforcement grid.
- the design value of the yield stress σ_{yd} .
- the average tensile strength of the concrete f_{ctm} . Alternatively the concrete properties may be specified with reference to the Eurocode 2 EN 1992-1-1 model code regulation.
- the characteristic compressive strength of the concrete f_{ck} . Alternatively the concrete properties may be specified with reference to the Eurocode 2 EN 1992-1-1 model code regulation.

- the environmental class given in Eurocode 2 EN 1992-1-1: X0, XC1, XC2, XC3, XC4, XD1, XS1, XD2, XS2, XD3, or XS3. When the environmental class is defined in the material properties, it overrules the global environmental class defined for the analysis commands.

The directions of the bars are defined with the local x and y axes of the reinforcement grid, which can be set in the usual way. DIANA will automatically calculate and apply the equivalent thickness parameter for the reinforcement grids and apply this parameter in the model for follow up analyses with other applications.

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.

9.4.2 Reinforcement Grid Design Check

In a reinforcement grid design check analysis the following operations are performed subsequently:

- Design code specification
- Model evaluation
- Solve the system of equations
- Define load combinations
- Define load envelopes
- Output results

9.4.3 Eurocode 2 EN 1992-1-1 Definitions

The national annex code, the environmental and construction class can be defined. Table 9.3 presents the parameters according to the chosen annex.

Table 9.3: 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

9.4.4 Model Evaluation

During model evaluation the definitions of elements are checked, including material and geometrical properties which are assigned to elements; the elements assembly, or ordering of element and nodal variables in the system set of degrees of freedom are defined. The following parameters can be set in the model evaluation:

- Extended test on element shape and aspect ratio [§ 9.1.1 p. 131]
- Average nodal normal of shell elements [§ 9.1.1 p. 132]
- Option for evaluation of reinforcements [§ 9.1.1 p. 132]
- Tolerance for creating directions of variables in vector basis [§ 9.1.1 p. 132]
- Automatic tyings [§ 9.1.1 p. 132]

9.4.5 Define Load Combinations

Optionally, one or more load combinations can be defined. Each load case combination may contain only a single mobile load case. Each load combination may be specified with a name. These names can be referred to in the load envelope definitions of in the output results selection. You should specify whether the load combination is of type ‘Ultimate Limit State’ or ‘Servicability Limit State’, to allow selections in following steps.

9.4.6 Define Load Envelopes

Optionally, one or more load envelopes can be defined. 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.

9.4.7 Output Results

You can obtain output of reinforcement grid design check analysis results via one or more output command blocks which select the analysis results to be output. For every output block you can select the device. Available devices are: Tabulated output, DIANA Ppf, FX⁺ for DIANA, or iDIANA output. You must specify the required output. By default there is no output for reinforcement grid design analysis. When specifying the required output you may select for which load sets, load combinations, or load envelopes you want to get the selected output results.

The selected result items may be output either constant per reinforcement or constant per element or different for every integration point with results presented as valid for the integration points or extrapolated to the nodes of an element. There is also the possibility to average the results in the direction normal to the reinforcement bars over a user defined length, which is related to the thickness of the plate. For the options of constant results per reinforcement or per element 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 the option of different per integration point is specified the result values in each integration point of the grid will be output individually. If the option per node is specified the result values in each node of the grid reinforcement is calculated and output individually. If the averaged option 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 a user specified length is defined. The average results over the reinforcement particles that are intersected by this line are calculated and displayed in the integration points.

There are two types of output:

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. These checks are strongly related to the calculation of the reinforcement forces and moments. The combined reinforcement forces, n'_{xx} , n'_{yy} , n'_{xy} for reinforcement grids above the neutral plane are defined as:

$$n'_{xx} = \frac{n_{xx}}{2} + \frac{m_{xx}}{z_d} \quad (9.178)$$

$$n'_{yy} = \frac{n_{yy}}{2} + \frac{m_{yy}}{z_d} \quad (9.179)$$

$$n'_{xy} = \left| \frac{n_{xy}}{2} + \frac{m_{xy}}{z_d} \right| \quad (9.180)$$

For reinforcement grids below the neutral plane the combined reinforcement are defined as:

$$n'_{xx} = \frac{n_{xx}}{2} - \frac{m_{xx}}{z_d} \quad (9.181)$$

$$n'_{yy} = \frac{n_{yy}}{2} - \frac{m_{yy}}{z_d} \quad (9.182)$$

$$n'_{xy} = \left| \frac{n_{xy}}{2} - \frac{m_{xy}}{z_d} \right| \quad (9.183)$$

The required area of the reinforcement in local directions is defined as:

$$A_{\text{req},x} = (n'_{xx} + n'_{xy})/f_{yd} \quad (9.184)$$

$$A_{\text{req},y} = (n'_{yy} + n'_{xy})/f_{yd} \quad (9.185)$$

The applied area of the reinforcement in local directions is defined as:

$$A_x = \left(\frac{1}{4} \pi \phi_x^2 \right) / s_x \quad (9.186)$$

$$A_y = \left(\frac{1}{4} \pi \phi_y^2 \right) / s_y \quad (9.187)$$

The ratio of the required area over the applied area in local directions is defined as:

$$A_{\text{rat},x} = \frac{A_{\text{req},x}}{A_x} \quad (9.188)$$

$$A_{\text{rat},y} = \frac{A_{\text{req},y}}{A_y} \quad (9.189)$$

The ratio of the required area in local x direction over the required area in local y direction is defined as:

$$A_{\text{req},xy} = \min(A_{\text{req},x}/A_{\text{req},y}, A_{\text{req},y}/A_{\text{req},x}) \quad (9.190)$$

For the applied area holds:

$$A_{xy} = \min(A_x/A_y, A_y/A_x) \quad (9.191)$$

The unity check on the minimum reinforcement area as defined in §9.2.1.1 of the Eurocode[57] is defined as:

$$A_{s,\min} = \min(A_{\min1}, A_{\min2}) \quad (9.192)$$

$$A_{\min1} = \max(0.0013 * d, 0.26 * f_{ctm} * d / f_{yk}) \quad (9.193)$$

Where f_{yk} is the characteristic value for the yield stress:

$$f_{yk} = \gamma_s * f_{yd} \quad (9.194)$$

$$A_{\min 2} = 1.2 * A_{\text{req}} \quad (9.195)$$

$$\text{UCMIN}_x = A_{s,\min}/A_x \quad (9.196)$$

$$\text{UCMIN}_y = A_{s,\min}/A_y \quad (9.197)$$

The unity check on the maximum reinforcement area as defined in §9.2.1.1 of the Eurocode[57] is defined as:

$$A_{\max} = 0.04 * A_c \quad (9.198)$$

Where A_c represents the concrete area of the cross-section.

$$\text{UCMAX}_x = \frac{A_x}{A_{\max}} \quad (9.199)$$

$$\text{UCMAX}_y = \frac{A_y}{A_{\max}} \quad (9.200)$$

Next to the unity check related to the ‘Ultimate Limit State’ there are the output items related to ‘Serviceability Limit State’, the coverage check and the shear force check. These are unity check output related to the spacing (UCSPA) and diameter (UCPHI) of the reinforcement. The unity check related to the shear force (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.

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[57].

The effective stress in the reinforcement bars is calculated as:

$$\sigma = f_{yd} * \frac{A_{\text{req}}}{A} \quad (9.201)$$

From Table 7.2N of the Eurocode[57] and the calculated effective stress σ , the maximum required bar diameter ϕ_s^* is calculated. This maximum bar diameter will be adapted using Eq. 7.7N of the Eurocode:

$$\phi_s = \phi_s^* (F_{ctm}/2.9) * \frac{1}{2} * h_t / (2 * (h - d)) \quad (9.202)$$

The adapted maximum bar diameter is used to calculate the unity check related to the diameter of the reinforcement bar in local directions:

$$\text{UCPHI}_x = \frac{\phi_x}{\phi_s} \quad (9.203)$$

$$\text{UCPHI}_y = \frac{\phi_y}{\phi_s} \quad (9.204)$$

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:

$$\text{UCSPA}_x = \frac{s_x}{s_{\max}} \quad (9.205)$$

$$\text{UCSPA}_y = \frac{s_y}{s_{\max}} \quad (9.206)$$

The design check on the coverage is a model check independent of the loading. The nominal coverage c_{nom} equals Eq. 4.1 of the Eurocode:

$$c_{\text{nom}} = c_{\min} + \Delta c_{\text{dev}} \quad (9.207)$$

The minimum coverage c_{\min} can be calculated with Eq. 4.2 of the Eurocode:

$$c_{\min} = \max(c_{\min,b}, c_{\min,dur} + \Delta c_{dur,g} - \Delta c_{dur,st} - \Delta c_{dur,add}, 10 \text{ mm}) \quad (9.208)$$

$c_{\min,b}$ 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. $c_{\min,dur}$ is based on table 4.4N of the Eurocode and depends on the environmental and construction class. The unity checks on coverage can be defined as:

$$UCCOV_x = \frac{c_{nom,x}}{co_x} \quad (9.209)$$

$$UCCOV_y = \frac{c_{nom,y}}{co_y} \quad (9.210)$$

The design checks on shear force are related to the Ultimate Limit State. The design concrete shear $V_{Rd,c}$ can be calculated using Eq. 6.2a of the Eurocode:

$$V_{Rd,c} = \left(C_{Rd,c} \cdot k \cdot (100 \cdot \rho_1 \cdot f_{ck})^{1/3} + k_1 \cdot \sigma_{cp} \right) \cdot d \quad (9.211)$$

The minimum value for $V_{Rd,c}$ is given in Eq. 6.2b of the Eurocode:

$$V_{Rd,c} = (v_{\min} + k_1 \cdot \sigma_{cp}) \cdot d \quad (9.212)$$

With

$$k = 1 + \sqrt{(200/d)} \leq 2.0 \quad (9.213)$$

$$\rho_1 = A_{sl}/(b_w \cdot d) \leq 0.02 \quad (9.214)$$

$$A_{sl,y} = \frac{1}{4} \pi \phi_y^2 1000 / s_y \quad (9.215)$$

$$A_{sl,x} = \frac{1}{4} \pi \phi_x^2 1000 / s_x, \quad (9.216)$$

$$\sigma_{cp} = n_{xx}/d < 0.2 f_{cd} \quad (9.217)$$

Where f_{ck} and f_{cd} in MPa and d in mm.

The parameters $C_{Rd,c}$, v_{\min} and k_1 depend on the chosen annex. The unity checks on the shearforce can be defined as:

$$UCSHR_x = q_{xz}/V_{Rd,c,x} \quad (9.218)$$

$$UCSHR_y = q_{yz}/V_{Rd,c,y} \quad (9.219)$$

9.5 Stiffness Adaptation Analysis

As alternative for a full nonlinear analysis for calculating load distributions, deformations, crack patterns, and crack openings in constructions with nonlinear materials, such as reinforced concrete, a stiffness adaptation analysis 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.

In stiffness adaptation analysis both standard linear elastic material as well as non-linear 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 9.13 dis-

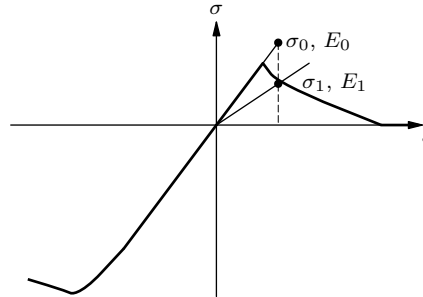


Figure 9.13: Uniaxial stress-strain curve with stress and stiffness reduction

plays 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.

In a stiffness adaptation analysis different loadings 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. 9.14]. 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

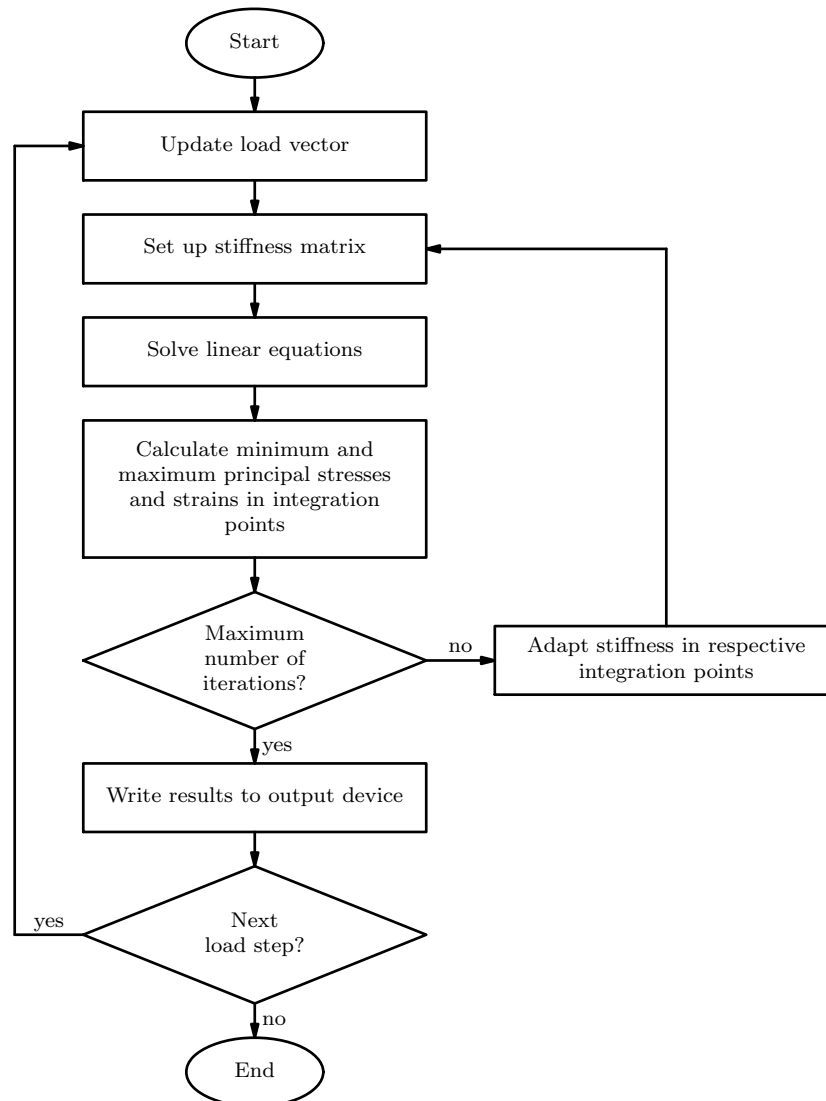


Figure 9.14: Flowchart of analysis sequence for stiffness adaptation analysis

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.

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 the analysis times of the stiffness adaptation method will be shorter, and no advanced analysis procedures and settings are required. Stiffness adaption 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 is recommended.

In the present implementation of stiffness adaptation analysis combination with other modules, such as phased analysis is not allowed.

In stiffness adaption analysis nonlinear stress-strain curves can be assigned to the following element types:

- Two-dimensional plane stress (membrane) elements
- Plane strain and axisymmetric elements, excluding axisymmetric shells and infinite shells

- Beam elements
- Truss elements
- Curved shell elements, excluding layered shell elements
- Solid elements
- Bar and grid reinforcements

Other element types can be applied in a stiffness adaptation analysis, but only with linear elastic material properties.

9.5.1 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. Alternatively, predefined materials from concrete and steel model codes may be used.

For direct input for material for which the stiffness should be adapted for material failure, an initial Young's modulus and Poisson's ration must be specified as well as uniaxial stress-strain curves for the tensile and compressive regime. For a stiffness adaptation analysis model you can choose a predefined tension softening function [Fig. 9.15]. Beyond the tensile strength f_t the shape of these curves is like the tension softening curves for the multi-directional fixed crack models.

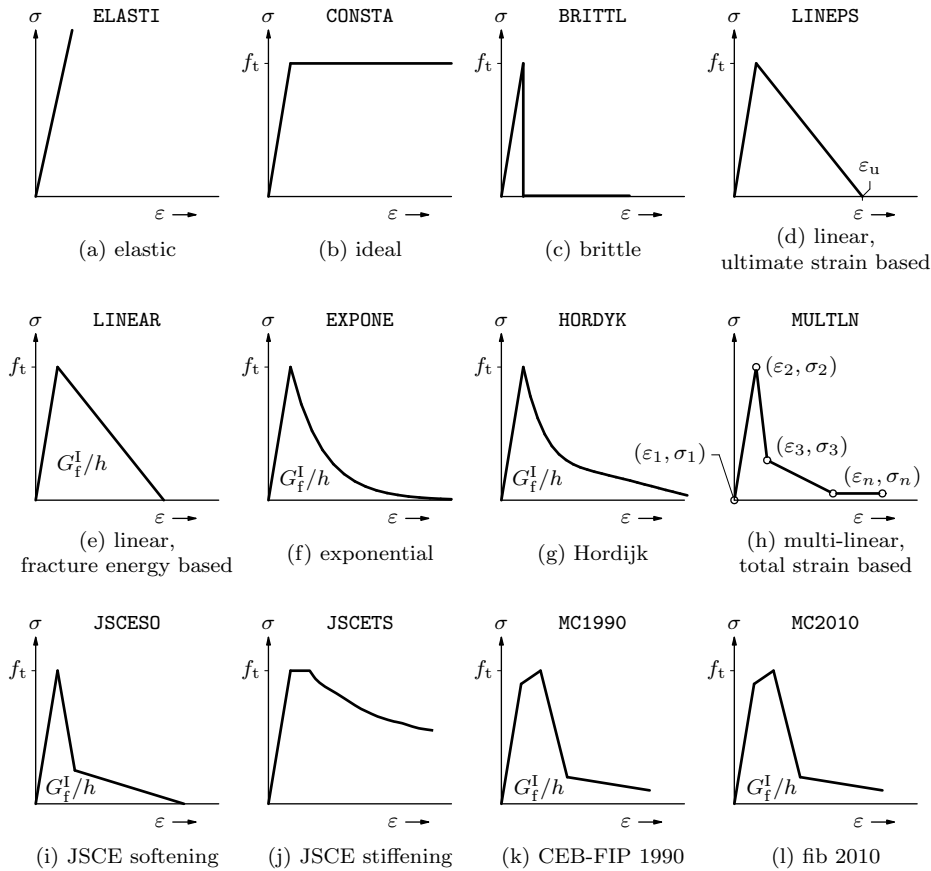
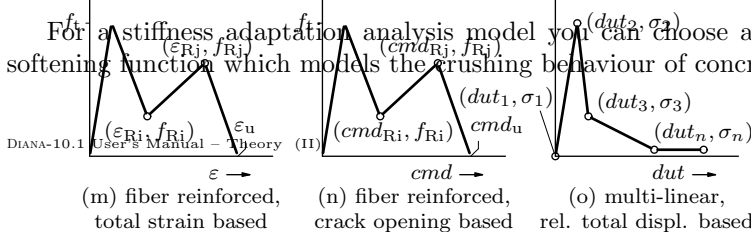


Figure 9.15: Predefined tension softening for stiffness adaptation analysis

For a stiffness adaptation analysis model you can choose a predefined compressive softening function which models the crushing behaviour of concrete [Fig. 9.16].



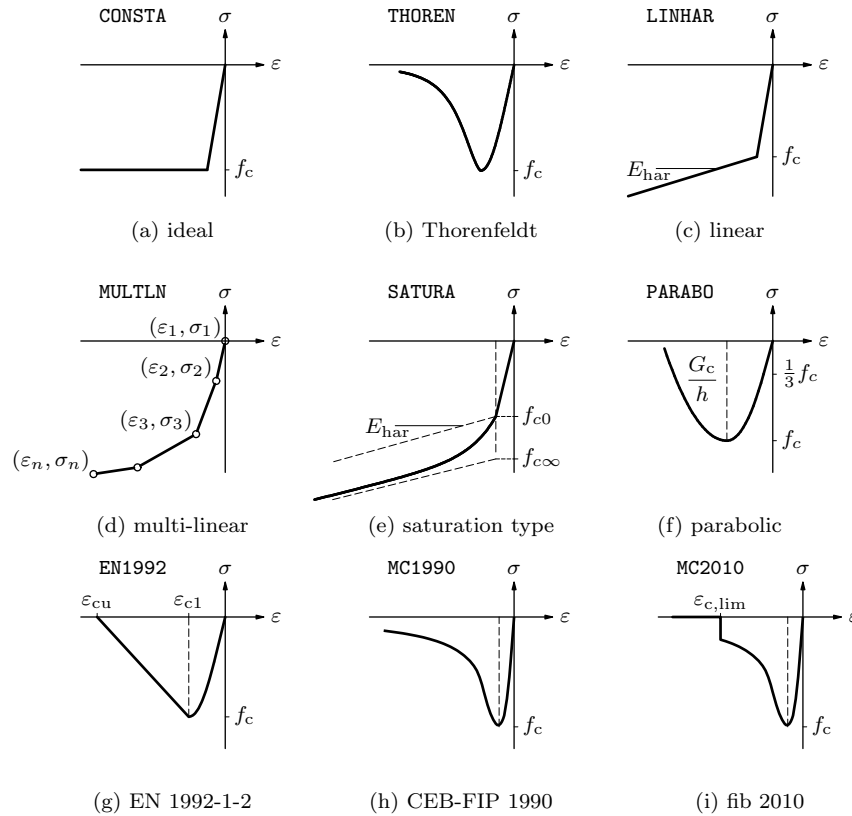


Figure 9.16: Predefined compression behaviour for stiffness adaptation analysis

9.5.2 Performing a Stiffness Adaptation Analysis

In a stiffness adaptation check analysis the following operations are performed subsequently:

- Model evaluation
- Solve the system of equations
- Execute steps
- Output results

9.5.3 Model Evaluation

During model evaluation the definitions of elements are checked, including material and geometrical properties which are assigned to elements; the elements assembly, or ordering of element and nodal variables in the system set of degrees of freedom are defined. The following parameters can be set in the model evaluation:

- Extended test on element shape and aspect ratio [§ 9.1.1 p. 131]
- Average nodal normal of shell elements [§ 9.1.1 p. 132]
- Option for evaluation of reinforcements [§ 9.1.1 p. 132]
- Tolerance for creating directions of variables in vector basis [§ 9.1.1 p. 132]
- Automatic tyings [§ 9.1.1 p. 132]

9.5.4 Solve the System of Equations

See linear static analysis [§ 9.1.2 p. 133]

9.5.5 Execute Steps

This command block may be repeated. For every execute steps block an load is applied and stiffness changes and results are calculated. You should specify the load step number to be incremented to the external applied load, the load increment sizes and the maximum number of iterations for each load increment step.

9.5.6 Output Results

9.6 Phased Construction 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.

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.

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. Then we will explain the analysis procedure for a changing 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. 9.17].

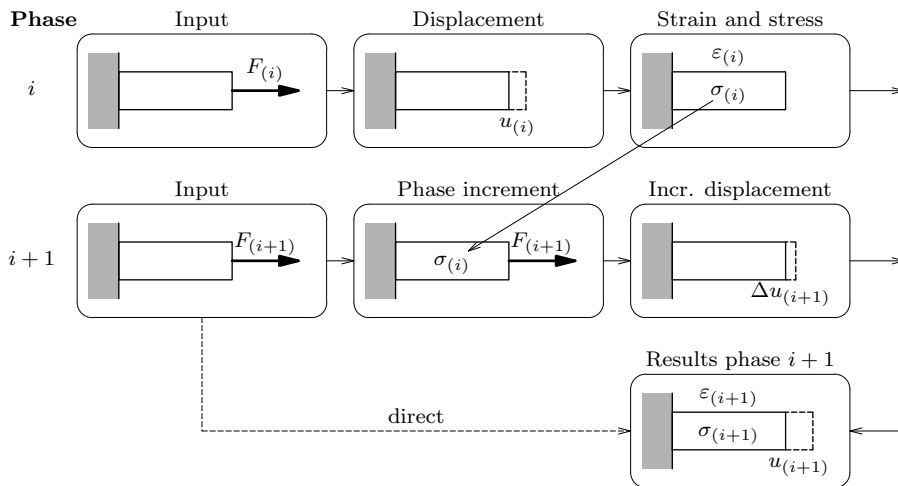


Figure 9.17: Phases with unchanged model

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

$$F_{(i)} \Rightarrow \begin{cases} u_{(i)} \\ \varepsilon_{(i)} \\ \sigma_{(i)} \end{cases} \quad (9.220)$$

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 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

$$\text{Increment: } F_{(i+1)} - F_{\sigma_{(i)}} \Rightarrow \begin{cases} \Delta u_{(i+1)} \\ \Delta \varepsilon_{(i+1)} \\ \Delta \sigma_{(i+1)} \end{cases} \quad (9.221)$$

To terminate a phase in the analysis, DIANA adds the incremental results Eq. (9.221) to the total results of the previous phase Eq. (9.220), which yields the total results of the current phase:

$$\text{Total: } F_{(i+1)} \Rightarrow \begin{cases} u_{(i+1)} &= u_{(i)} + \Delta u_{(i+1)} \\ \varepsilon_{(i+1)} &= \varepsilon_{(i)} + \Delta \varepsilon_{(i+1)} \\ \sigma_{(i+1)} &= \sigma_{(i)} + \Delta \sigma_{(i+1)} \end{cases} \quad (9.222)$$

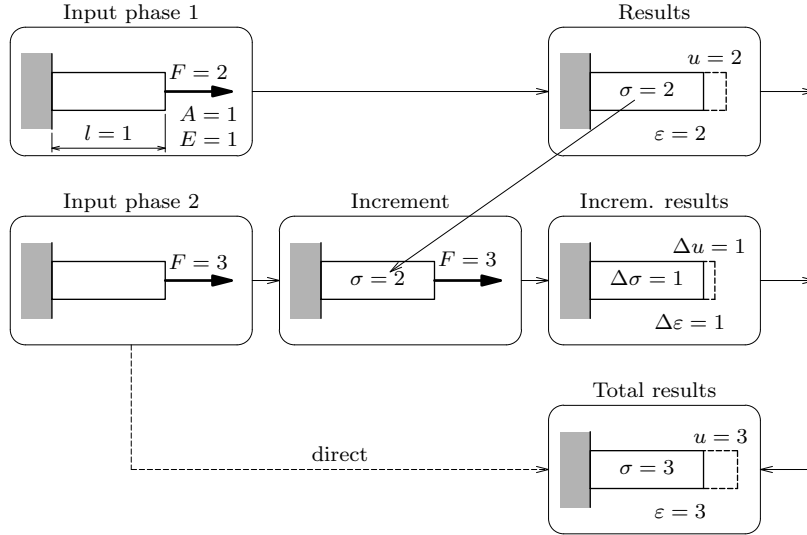
Direct analysis. Alternatively we could have obtained this total result directly in a regular analysis, without the phased analysis option, by applying the total load $F_{(i+1)}$ on the model

$$F_{(i+1)} \Rightarrow \begin{cases} u_{(i+1)} \\ \varepsilon_{(i+1)} \\ \sigma_{(i+1)} \end{cases} \quad (9.223)$$

This equation is very much like the one for the previous phase Eq. (9.220). 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.*

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. 9.18]. We assume 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

Figure 9.18: Increasing load F

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 $\Delta \epsilon = 1$ 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 $\epsilon = 3$ 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.

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. 9.19]. For this example we assume specific properties of the bar: the length is l , the cross-section

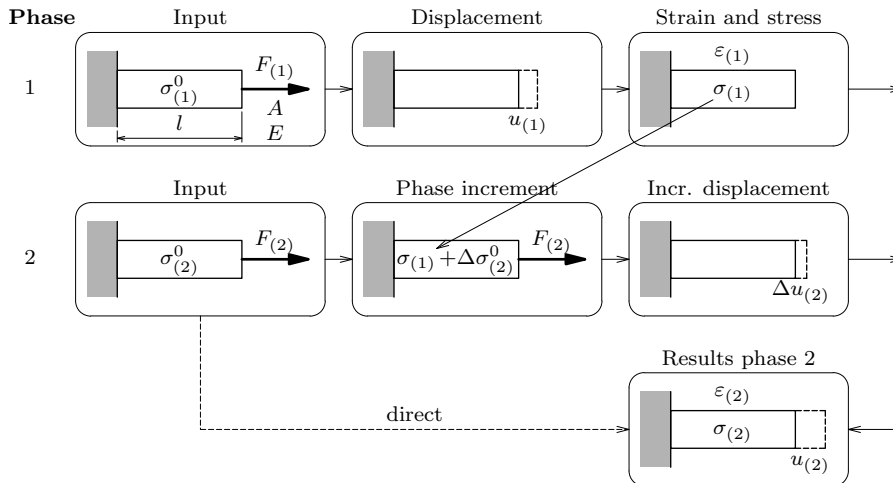


Figure 9.19: Phases with unchanged model and initial stress

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

$$\Delta\sigma_{(1)} = \sigma_{(1)} - \sigma_{(1)}^0 \quad (9.224)$$

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

$$F_{(1)} + F_{\sigma_{(1)}^0} \Rightarrow \begin{cases} u_{(1)} = \Delta u_{(1)} = u_{F_{(1)}} + u_{\sigma_{(1)}^0} \\ \varepsilon_{(1)} = \Delta\varepsilon_{(1)} = \varepsilon_{F_{(1)}} + \varepsilon_{\sigma_{(1)}^0} \\ \sigma_{(1)} = \sigma_{(1)}^0 + \Delta\sigma_{(1)} = \sigma_{(1)}^0 + \Delta\varepsilon_{(1)}E \end{cases} \quad (9.225)$$

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. (9.222). Substitution of the models’ properties in Eq. (9.225) gives

$$\begin{cases} u_{(1)} = \Delta u_{(1)} = \frac{F_{(1)}l}{EA} - \frac{\sigma_{(1)}^0 l}{E} \\ \varepsilon_{(1)} = \Delta\varepsilon_{(1)} = \frac{u_{(1)}}{l} \\ \sigma_{(1)} = \frac{F_{(1)}}{A} \end{cases} \quad (9.226)$$

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

$$\Delta\sigma_{(2)}^0 = \sigma_{(2)}^0 - \sigma_{(1)}^0 \quad (9.227)$$

The new stress in phase 2 now becomes

$$\bar{\sigma}_{(2)}^0 = \sigma_{(1)} + \Delta\sigma_{(2)}^0 = \sigma_{(2)}^0 + \Delta\varepsilon_{(1)}E \quad (9.228)$$

This results in an incremental displacement $\Delta u_{(2)}$ from which DIANA determines the incremental strains and stresses:

$$\text{Increment: } F_{(2)} + F_{\Delta\sigma_{(2)}^0} + F_{\sigma_{(1)}} \Rightarrow \begin{cases} \Delta u_{(2)} \\ \Delta\varepsilon_{(2)} \\ \Delta\sigma_{(2)} \end{cases} \quad (9.229)$$

And substitution of the models’ properties in Eq. (9.229) gives

$$\text{Increment: } \begin{cases} \Delta u_{(2)} = \frac{F_{(2)}l}{EA} - \frac{F_{(1)}l}{EA} - \frac{\sigma_{(2)}^0 l}{E} + \frac{\sigma_{(1)}^0 l}{E} \\ \Delta\varepsilon_{(2)} = \frac{F_{(2)}}{EA} - \frac{F_{(1)}}{EA} - \frac{\sigma_{(2)}^0}{E} + \frac{\sigma_{(1)}^0}{E} \\ \Delta\sigma_{(2)} = \frac{F_{(2)}}{A} - \frac{F_{(1)}}{A} - \sigma_{(2)}^0 + \sigma_{(1)}^0 \end{cases} \quad (9.230)$$

The total results for phase 2 are now equal to those of phase 1 Eq. (9.225), with the addition of the calculated increments Eq. (9.229):

$$\text{Total: } \begin{cases} u_{(2)} = u_{(1)} + \Delta u_{(2)} = u_{F_{(2)}} + u_{\sigma_{(2)}^0} \\ \varepsilon_{(2)} = \varepsilon_{(1)} + \Delta\varepsilon_{(2)} \\ \sigma_{(2)} = \bar{\sigma}_{(2)}^0 + \Delta\sigma_{(2)} = \sigma_{(1)} + \Delta\sigma_{(2)}^0 + \Delta\sigma_{(2)} \end{cases} \quad (9.231)$$

Substitution of Eq. (9.226) and Eq. (9.230) in Eq. (9.231) for the models' properties gives

$$\text{Total: } \begin{cases} u_{(2)} = \frac{F_{(2)}l}{EA} - \frac{\sigma_{(2)}^0 l}{E} \\ \varepsilon_{(2)} = \frac{F_{(2)}}{EA} - \frac{\sigma_{(2)}^0}{E} \\ \sigma_{(2)} = \frac{F_{(2)}}{A} \end{cases} \quad (9.232)$$

Direct. In the equations Eq. (9.232) 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. 9.20]. Again we assume that all relevant physical and material

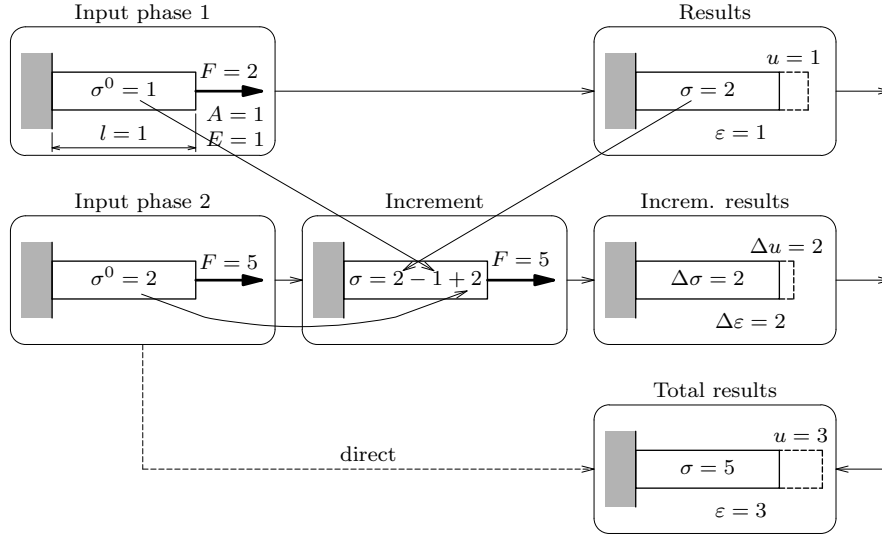


Figure 9.20: Increasing load F and initial stress σ^0

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. (9.231) 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.

9.6.1 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,⁴ 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 9.21 shows what happens in phased analysis when in phase 2 an element is added to the model. Note that the total stresses and strains

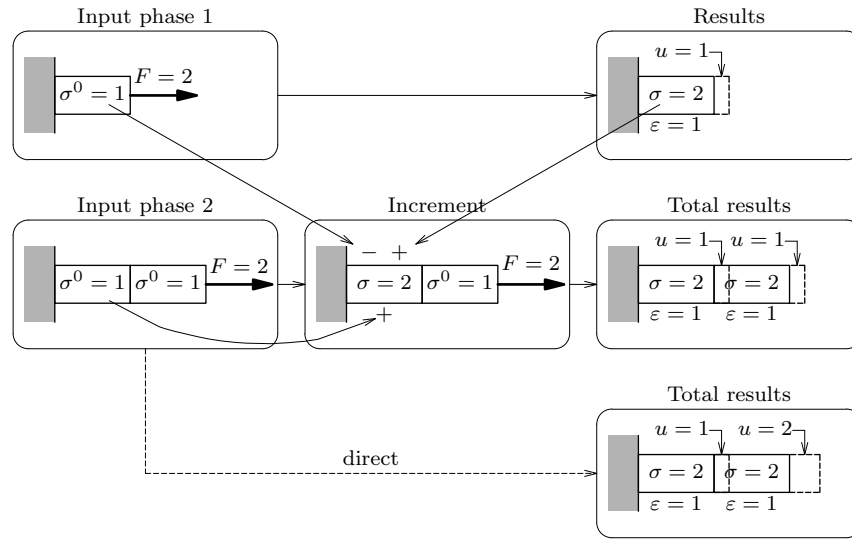


Figure 9.21: Addition of an element

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.

9.7 Solving Linear Sets of Equations

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

$$\begin{bmatrix} \mathbf{K}_{uu} & \mathbf{K}_{uc}^T \\ \mathbf{K}_{uc} & \mathbf{K}_{cc} \end{bmatrix} \begin{Bmatrix} \mathbf{u}_u \\ \mathbf{u}_c \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}_u \\ \mathbf{f}_c \end{Bmatrix} \quad (9.233)$$

Here subscript u is associated with the *unconstrained* degrees of freedom and subscript c with the *constrained* degrees of freedom. Eq. (9.233) can be reduced to

$$\mathbf{K}_{uu}\mathbf{u}_u = \mathbf{f}_u - \mathbf{K}_{uc}\mathbf{u}_c \quad \text{or} \quad \hat{\mathbf{K}}\hat{\mathbf{u}} = \hat{\mathbf{f}} \quad (9.234)$$

⁴It is not necessary that you define the complete model beforehand.

Now all boundary conditions, tyings and loadings have been treated, the remaining system of equations can be solved. For convenience we write this equation in a more simple form

$$\boxed{\mathbf{K}\mathbf{u} = \mathbf{f}} \quad (9.235)$$

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 $n \times n$ matrix⁵ with a symmetric structure, \mathbf{f} is a known right-hand-side vector and \mathbf{u} the unknown solution vector to be computed.

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.

9.7.1 Direct Solution Methods

A general approach to solve the linear system of Eq. (9.235) is first to *factorize* \mathbf{K} :

$$\mathbf{K} \Rightarrow \mathbf{L}\mathbf{U} \quad (9.236)$$

$$\mathbf{K} \Rightarrow \mathbf{L}\mathbf{D}\mathbf{U} \quad (9.237)$$

where \mathbf{L} , \mathbf{U} and \mathbf{D} are lower, upper and diagonal matrices respectively. The process of Eq. (9.237) is known as *LDU decomposition*. After the factorization, the solution vector \mathbf{u} can be computed by successive forward and backward substitution to solve the triangular system of equations Eq. (9.235)

$$\mathbf{L}\mathbf{w} = \mathbf{f} \quad \mathbf{D}\mathbf{U}\mathbf{u} = \mathbf{w} \quad (9.238)$$

Under the assumption that \mathbf{K} is a positive definite symmetric matrix the decomposition becomes $\mathbf{L}\mathbf{L}^T$ or $\mathbf{L}\mathbf{D}\mathbf{L}^T$.

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

$$k_{ij} \leftarrow k_{ij} - k_{ik} \frac{k_{kj}}{k_{kk}} \quad (9.239)$$

the entry in location (i, j) of the original matrix was zero. It has been proved, see George & Liu [28], that the number of operations to compute the triangular factor \mathbf{L} of a symmetric matrix \mathbf{K} is equal to

$$\frac{1}{2} \sum_{i=1}^{N-1} (\eta(L(*, i)) - 1) \times (\eta(L(*, i)) + 2) \quad (9.240)$$

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 [41]. 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. (9.235) 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.

⁵*sparse* = ‘with many zero-terms’.

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 $\mathbf{Ly} = \mathbf{Pf}$ and $\mathbf{DL}^T\mathbf{x} = \mathbf{y}$ and then set $\mathbf{u} = \mathbf{P}^T\mathbf{x}$.

The first two steps are symbolic operations and have to be performed only once. In subsequent load steps such as in stiffness adaptation analysis or nonlinear analysis the analysis flow is 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.

- 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 [28]. This algorithm fully uses the nonzero pattern of the system matrix at the expense of introducing indirections of the inner loops. Prior to the factorization, the Sparse Cholesky solver reorders the system matrix according to the Metis algorithm, see Karypis & Kumar [41]. 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.

- 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 [69, 70]. 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.

9.7.2 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

$$\mathbf{u}_{i+1} = \mathbf{u}_i + \gamma_i \mathbf{P}^{-1}(\mathbf{f} - \mathbf{Ku}_i) \quad (9.241)$$

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 $\mathbf{P}\mathbf{x} = \mathbf{y}$.

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

$$\kappa = \frac{\lambda_{\max}}{\lambda_{\min}} \quad (9.242)$$

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 [§ 9.7.2.4 p. 184].

Many algorithms exist that compute the iteration parameters [65]. 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 9.22 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.

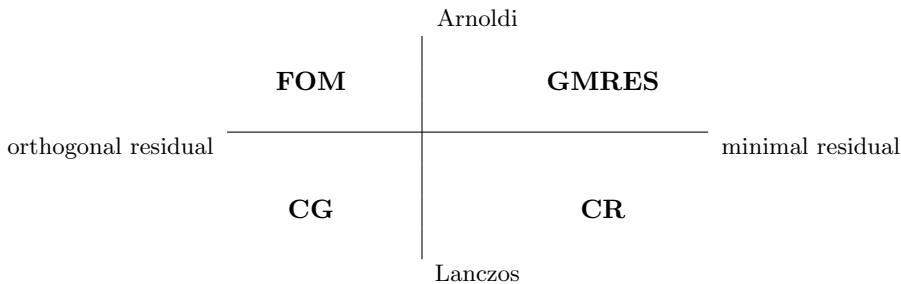


Figure 9.22: Krylov-based iterative solution methods.

9.7.2.1 Conjugate Gradient

The Conjugate Gradient method, see Hestenes & Stiefel [34], 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.(9.242). The following bound on the error $\mathbf{u} - \mathbf{u}_i$ at iteration i of Conjugate Gradient is well-known [65]:

$$\|\mathbf{u} - \mathbf{u}_i\|_{\mathbf{K}} \leq 2 \left[\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right]^i \|\mathbf{u} - \mathbf{u}_0\|_{\mathbf{K}} \quad (9.243)$$

It follows that the ratio between the largest and smallest eigenvalue should be small for CG to converge fast.

9.7.2.2 Generalized Minimal Residual

The GMRES method, see Saad & Schultz [66], 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} .

9.7.2.3 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 [48]: 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

$$\mathbf{K} = \sum_{i=1}^n \mathbf{L}_i^T \mathbf{K}_i \mathbf{R}_i \quad (9.244)$$

The right restriction operator can be used to extract subdomain matrices and vectors from the global matrices and vectors:

$$\begin{aligned} \mathbf{K}_i &= \mathbf{R}_i \mathbf{K} \mathbf{R}_i^T \\ \mathbf{x}_i &= \mathbf{R}_i \mathbf{x} \end{aligned} \quad (9.245)$$

The left operator can be used to assemble the local vectors into a global vector, which is useful for inner products:

$$\mathbf{x} = \sum_{i=1}^n \mathbf{L}_i^T \mathbf{x}_i \quad (9.246)$$

The parallel computations in Schwarz domain decomposition involve different types of data exchange, which involve multiplications with the left and right restrictors [48, 67].

9.7.2.4 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:

$$\mathbf{P}^{-1} \mathbf{K} \mathbf{u} = \mathbf{P}^{-1} \mathbf{f} \quad (9.247)$$

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. (9.247), and from the right.

Incomplete LU-decomposition. By default DIANA applies *Incomplete LU-decomposition preconditioning*, generally known as ILU preconditioning, see Meijerink & Van Der Vorst [49]. 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}

$$\mathbf{P} = \mathbf{L} \cdot \mathbf{U} \approx \mathbf{K} \quad (9.248)$$

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

$$\mathbf{P}_{ij} = \sum_k \mathbf{L}_{ik} \cdot \mathbf{U}_{kj} \quad \text{if } \mathbf{K}_{ij} \neq 0 \quad (9.249)$$

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 [64]). 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.

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.

The natural choice for the termination criterion of the iterative process is based on the reduction of the residual \mathbf{r}_i ,

$$\mathbf{r}_i = \mathbf{f} - \mathbf{K} \cdot \mathbf{u}_i \quad (9.250)$$

with \mathbf{u}_i the approximation of the solution \mathbf{u} after i iterations. The iteration is stopped if

$$\|\mathbf{r}_i\| < \epsilon \|\mathbf{f}\| \quad (9.251)$$

with ϵ a user-defined tolerance. with ϵ a user-defined tolerance.

9.7.3 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.

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.

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.

9.7.4 Substructuring

Substructuring is a standard technique in Finite Element Analysis (see Bathe [5, § 8.2.4]). Substructuring is one of the two common domain decomposition methods [65, 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

$$\begin{bmatrix} \mathbf{A}_1 & \mathbf{0} & \mathbf{B}_1 \\ \mathbf{0} & \mathbf{A}_2 & \mathbf{B}_2 \\ \mathbf{B}_1^T & \mathbf{B}_2^T & \mathbf{C} \end{bmatrix} \quad (9.252)$$

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. (9.252) can be factorized as follows

$$\begin{bmatrix} \mathbf{A}_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_2 & \mathbf{0} \\ \mathbf{B}_1^T & \mathbf{B}_2^T & \mathbf{C}^* \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{A}_1^{-1}\mathbf{B}_1 \\ \mathbf{0} & \mathbf{I} & \mathbf{A}_2^{-1}\mathbf{B}_2 \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \quad (9.253)$$

where \mathbf{I} is the identity matrix and \mathbf{C}^* the so-called Schur complement

$$\mathbf{C}^* = \mathbf{C} - \mathbf{B}_1^T \mathbf{A}_1^{-1} \mathbf{B}_1 - \mathbf{B}_2^T \mathbf{A}_2^{-1} \mathbf{B}_2 \quad (9.254)$$

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:

$$\mathbf{B}^T \mathbf{A}^{-1} \mathbf{B} = \mathbf{B}^T (\mathbf{LL}^T)^{-1} \mathbf{B} = (\mathbf{L}^{-1} \mathbf{B})^T \mathbf{L}^{-1} \mathbf{B} \quad (9.255)$$

After all substructures have been assembled, it remains to solve a system of the form

$$\mathbf{C}^* \mathbf{u} = \mathbf{f} \quad (9.256)$$

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.

Part III

Linear Dynamic and Nonlinear Finite Element Analysis

Chapter 10

Linear Dynamic Analysis

The governing equation of motion for a linear dynamic finite element system reads

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{C}\dot{\mathbf{u}} + \mathbf{K}\mathbf{u} = \mathbf{f}(t) \quad (10.1)$$

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

$$\mathbf{u} = \mathbf{u}_{tr} + \mathbf{u}_{st} \quad (10.2)$$

The response \mathbf{u}_{tr} depends on the conditions with which the response is initiated and is induced by the initial conditions

$${}^0\mathbf{u} = \mathbf{u}(t_0) \quad \text{and} \quad {}^0\dot{\mathbf{u}} = \dot{\mathbf{u}}(t_0) \quad (10.3)$$

By contrast \mathbf{u}_{st} is the particular solution which satisfies the forcing function of $\mathbf{f}(t)$ of Eq. (10.1), 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. If the initial transient part is substantially important then you must analyse the complete transient response.

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.

10.1 Mass and Damping

10.1.1 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. [36]. 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.

In dynamic analysis DIANA requires the input of mass parameters. There are two forms of mass input: mass density for structural elements or concentrated mass via point elements.

10.1.1.1 Mass Density

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.

10.1.1.2 Concentrated Mass

Concentrated mass in dynamic analysis can be modeled 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. In static analysis, the concentrated mass acts as concentrated loading for dead weight.

10.1.2 Damping

There are various forms of damping input: viscous damping for structural elements, continuous damping (dashpots) via discrete spring/dashpot elements or point mass/damping elements. 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 [76].

10.1.2.1 Modal Damping

A modal frequency response analysis 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

$$\phi_i^T \mathbf{C} \phi_j = 2\omega_i \xi_i \delta_{ij} \quad (10.4)$$

where ω is the natural angular frequency and ξ_i is the damping ratio. The critical damping factor is

$$c_{\text{crit}} = 2\sqrt{km} \quad (10.5)$$

Where k is the generalized stiffness $\phi_i^T \mathbf{K} \phi_j \delta_{ij}$ and m is the generalized mass $\phi_i^T \mathbf{M} \phi_j \delta_{ij}$.

10.1.2.2 Viscous Damping

Applying a nonmodal solution technique, it is necessary to evaluate the damping matrix \mathbf{C} for structural elements explicitly and usually viscous damping effects can be included by assumption of *Rayleigh* damping which is of the form

$$\mathbf{C} = a\mathbf{M} + b\mathbf{K} \quad (10.6)$$

where a and b are constants to be determined from given damping ratios. Viscous damping is a form of damping which is proportional to the velocity.

10.1.2.3 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

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{K}(1 + i\gamma)\mathbf{u} = \mathbf{f}(t) \quad (10.7)$$

where γ is the structural damping factor. Equations like Eq. (10.7) can only be solved in the frequency domain.

10.1.2.4 Continuous Damping

Continuous damping may be specified via discrete spring/dashpot elements or one-node point mass/damping elements.

10.2 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 [15] 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.

$$\mathbf{f}_{\text{push}} = \sum_j a * \mathbf{M} * \phi_i * \mathbf{x}_j \quad (10.8)$$

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]. It is possible to define multiple pushover loads in one load case.

10.3 Frequency Response Analysis

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 [§ 10.3.3 p. 194]. 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. (10.1) 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.

$$\mathbf{M}\ddot{\mathbf{u}}(t) + \mathbf{C}\dot{\mathbf{u}}(t) + \mathbf{K}\mathbf{u}(t) = \mathbf{f}e^{i\Omega t} \quad (10.9)$$

$$\mathbf{u}(t) = \mathbf{u}e^{i\Omega t} \quad (10.10)$$

The response for each excitation frequency Ω_f can be determined from Eq. (10.9) 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. (10.10) with \mathbf{u} as complex displacement vector with in- and out-of-phase components. Substitution of the solution Eq. (10.10) in the equation of motion Eq. (10.9) results in

$$(\mathbf{K} - \Omega^2\mathbf{M} + i\Omega\mathbf{C})\mathbf{u} = \mathbf{f} \quad (10.11)$$

$$\mathbf{u} = (\mathbf{K} - \Omega^2\mathbf{M} + i\Omega\mathbf{C})^{-1} \mathbf{f} = \mathbf{H}(\Omega) \mathbf{f} \quad (10.12)$$

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. (10.11): an indirect method based on *mode superposition* [§ 10.3.1], and a *direct* method [§ 10.3.2].

10.3.1 Mode Superposition

The principle of the mode superposition technique may be employed providing the system of equations of Eq. (10.1) 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:

$$\mathbf{u}(\Omega) = \sum_{i=1}^{N_p} \phi_i \alpha_i(\Omega) \quad (10.13)$$

where ϕ_i is the i -th eigenvector of the undamped eigenproblem [§ 9.2 p. 145] and $\alpha_i(\Omega)$ is the i -th generalized modal displacement. Substitution of this equation in Eq. (10.11) 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

$$\mathbf{u} = \mathbf{H}(\Omega)\mathbf{f} = \sum_{i=1}^{N_p} \phi_i (K_{ii} - \Omega^2 M_{ii}, i\Omega C_{ii})^{-1} \phi_i^T \mathbf{f} \quad (10.14)$$

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. (10.14) is no longer correct, a direct solution method is more suitable.

10.3.2 Direct Solution

The direct solution method requires the solution of a complex system of equations. Complex arithmetic can be avoided by transformation of Eq. (10.11) to a system with real coefficients:

$$\begin{bmatrix} \mathbf{K} - \Omega^2 \mathbf{M} & +\Omega \mathbf{C} \\ -\Omega \mathbf{C} & \mathbf{K} - \Omega^2 \mathbf{M} \end{bmatrix} \begin{Bmatrix} \Im \mathbf{u} \\ \Re \mathbf{u} \end{Bmatrix} = \begin{Bmatrix} \Im \mathbf{f} \\ \Re \mathbf{f} \end{Bmatrix} \quad (10.15)$$

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. (10.15) is regular but not always positive definite and can only be solved with an LDU-decomposition [§ 9.7.1 p. 181].

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.

10.3.3 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

$$u_i(t) = \hat{u}_i \cos(\Omega t - \phi_i) \quad (10.16)$$

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. (10.1) may be assumed to have the form

$$\mathbf{u}(t) = \mathbf{u}_e e^{i\Omega t} \quad (10.17)$$

where $\mathbf{u} = \Re \mathbf{u} + i \Im \mathbf{u}$ is the complex spatial displacement vector. The element stresses have the same form

$$\boldsymbol{\sigma}_e(t) = \boldsymbol{\sigma}_e e^{i\Omega t} \quad (10.18)$$

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

$$\boldsymbol{\sigma}_e = \mathbf{D} \mathbf{B} \mathbf{u}_e \quad (10.19)$$

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

$$\sigma_i(t) = \hat{\sigma}_i \cos(\Omega t - \phi_i) \quad (10.20)$$

10.4 Response Spectrum Analysis

Response Spectrum Analysis predicts the linear response of a system subjected to a base excitation spectrum. The implementation in DIANA is based on the Response Spectrum Method as described by Gupta [32]. 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 in DIANA 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

$$\mathbf{M}\ddot{\mathbf{u}}(t) + \mathbf{K}\mathbf{u}(t) = -\mathbf{M}\mathbf{i}\ddot{u}_{su}(t) \quad (10.21)$$

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

$$\mathbf{f}_k = \mathbf{K}\mathbf{u}(t) \quad (10.22)$$

and the pseudo-inertia forces

$$\mathbf{f}_m = \mathbf{M}(\ddot{\mathbf{u}}(t) + \mathbf{i}\ddot{u}_{su}(t)) = \mathbf{M}\mathbf{a}(t) \quad (10.23)$$

Equilibrium of the forces gives $\mathbf{f}_m = -\mathbf{f}_k$ which results in

$$\mathbf{a} = \omega^2 \mathbf{u} \quad (10.24)$$

The absolute maximum value of the pseudo-acceleration \mathbf{a} is called the spectral acceleration S_A

$$S_A(\omega) = \max |\mathbf{a}(t)| = \omega^2 S_D(\omega) \quad (10.25)$$

where the spectral displacement S_D is

$$S_D(\omega) = \max |\mathbf{u}(t)| \quad (10.26)$$

Response Spectrum Analysis is based upon the mode superposition principle. Substitution of the mode superposition equation Eq. (10.13) in Eq. (10.21), premultiplication by $\boldsymbol{\phi}_i^T$, and modal vectors scaled such that

$$\boldsymbol{\phi}_i^T \mathbf{M} \boldsymbol{\phi}_i = 1 \quad \text{and} \quad \boldsymbol{\phi}_i^T \mathbf{K} \boldsymbol{\phi}_i = \omega_i^2 \quad (10.27)$$

gives

$$\ddot{\alpha}_i + \omega_i^2 \alpha_i = \gamma_i \ddot{u}_{su} \quad (10.28)$$

where α is the generalized modal displacement and γ_i is called the modal participation factor, which is equal to

$$\gamma_i = \phi_i^T \mathbf{M} \mathbf{i} \quad (10.29)$$

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

$$\mathbf{u}_{i,\max} = \gamma_i \phi_i S_{D_i} \quad (10.30)$$

Therefore, the maximum force vector for the i th mode can be calculated by

$$\mathbf{f}_{i,\max} = \mathbf{K} \mathbf{u}_{i,\max} = \mathbf{K} \gamma_i \phi_i S_{D_i} = \mathbf{M} \gamma_i \phi_i S_{A_i} \quad (10.31)$$

It is obvious that the upper bound of the combined forces is given by the absolute sum of the modal forces, the ABS rule:

$$\mathbf{f}_{\text{abs}} = \sum_{i=1}^N |\mathbf{f}_{i,\max}| \quad (10.32)$$

In general, it is unlikely that the maximum values of the modal forces would occur at the same time. Goodman, Rosenblueth & Newmark [31] 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:

$$\mathbf{f}_s = \sqrt{\sum_{i=1}^N \mathbf{f}_{i,\max}^2} \quad (10.33)$$

A method to take the correlation among the modes into account is the Complete Quadratic Combination, the CQC rule:

$$\mathbf{f}_s = \sqrt{\sum_{i=1}^N \sum_{j=1}^N |\rho_{ij} \mathbf{f}_{i,\max} \mathbf{f}_{j,\max}|} \quad (10.34)$$

In this approach 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:

$$\rho_{ij} = \frac{8\sqrt{\xi_i \xi_j} (r_{ij} \xi_i + \xi_j) r_{ij}^{\frac{3}{2}}}{(1 - r_{ij}^2)^2 + 4\xi_i \xi_j r_{ij} (1 + r_{ij}^2) + 4(\xi_i^2 + \xi_j^2) r_{ij}^2} \quad (10.35)$$

where

$$r_{ij} = \frac{\omega_i}{\omega_j} \quad (10.36)$$

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.

10.5 Transient Dynamic Analysis

This section is an introduction to the background theory of dynamic analysis with time integration. Elaborate discussions on dynamics can be found in Bathe [6, Ch. 9], Craig [21], Hughes [38, §9], Zienkiewicz [83, §9], and Géradin & Rixen [29, Ch. 7]. Using Module NONLIN, such analysis may be combined with physical or geometrical nonlinearities, see part 11. 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

$$\mathbf{M}\ddot{\mathbf{u}}(t) + \mathbf{C}\dot{\mathbf{u}}(t) + \mathbf{f}_{\text{int}}(\mathbf{u}, \dot{\mathbf{u}}, \boldsymbol{\varepsilon}, \boldsymbol{\sigma}, t, \dots) = \mathbf{f}_{\text{ext}}(t) \quad (10.37)$$

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. (10.37) can be written as

$$\mathbf{M}\ddot{\mathbf{u}}(t) + \mathbf{C}\dot{\mathbf{u}}(t) + \mathbf{K}\mathbf{u}(t) = \mathbf{f}_{\text{ext}}(t) \quad (10.37a)$$

For geometrical or physical nonlinear analysis or both, \mathbf{f}_{int} must be calculated for the actual stress distribution satisfying all nonlinear conditions.

$$\mathbf{f}_{\text{int}} = \int \mathbf{B}^T \boldsymbol{\sigma} \quad (10.38)$$

For the transient response of a nonlinear analysis, the solution of the second order differential equation Eq. (10.37) 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. (10.37) holds

$$\mathbf{M}^t \ddot{\mathbf{u}} + \mathbf{C}^t \dot{\mathbf{u}} + {}^t\mathbf{f}_{\text{int}}(\mathbf{u}, \dot{\mathbf{u}}, \boldsymbol{\varepsilon}, \boldsymbol{\sigma}, t, \dots) = {}^t\mathbf{f}_{\text{ext}} \quad (10.39)$$

If the time integration procedure being used requires

$$\mathbf{M}^{t+\Delta t} \ddot{\mathbf{u}} + \mathbf{C}^{t+\Delta t} \dot{\mathbf{u}} + {}^{t+\Delta t}\mathbf{f}_{\text{int}}(\mathbf{u}, \dot{\mathbf{u}}, \boldsymbol{\varepsilon}, \boldsymbol{\sigma}, t, \dots) = {}^{t+\Delta t}\mathbf{f}_{\text{ext}} \quad (10.40)$$

to obtain ${}^{t+\Delta t}\ddot{\mathbf{u}}$, then the time integration scheme will be called *implicit*. If, on the other hand, Eq. (10.40) 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 $\Delta \mathbf{u} = {}^{t+\Delta t}\mathbf{u} - {}^t\mathbf{u}$.

10.5.1 Newmark

A widely used integration scheme is the Newmark method, which consists of the following equations.

$${}^{t+\Delta t}\dot{\mathbf{u}} = {}^t\dot{\mathbf{u}} + ((1 - \gamma) {}^t\ddot{\mathbf{u}} + \gamma {}^{t+\Delta t}\ddot{\mathbf{u}}) \Delta t \quad (10.41)$$

$${}^{t+\Delta t}\mathbf{u} = {}^t\mathbf{u} + {}^t\dot{\mathbf{u}} \Delta t + \left(\left(\frac{1}{2} - \beta\right) {}^t\ddot{\mathbf{u}} + \beta {}^{t+\Delta t}\ddot{\mathbf{u}}\right) \Delta t^2 \quad (10.42)$$

To find ${}^{t+\Delta t}\ddot{\mathbf{u}}$ and ${}^{t+\Delta t}\dot{\mathbf{u}}$ the equations are solved by using first Eq. (10.42)

$${}^{t+\Delta t}\ddot{\mathbf{u}} = \frac{1}{\beta \Delta t^2} \Delta \mathbf{u} - \frac{1}{\beta \Delta t} {}^t\dot{\mathbf{u}} - \frac{1 - 2\beta}{2\beta} {}^t\ddot{\mathbf{u}} \quad (10.43)$$

which is then substituted in Eq. (10.41) 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 $\gamma \geq \frac{1}{2}$ excessive low frequency mode damping is introduced, and the method is first order accurate.

10.5.2 Euler Backward

The Euler Backward method consists of the following equations.

$${}^{t+\Delta t}\dot{\mathbf{u}} = \frac{\Delta \mathbf{u}}{\Delta t} \quad (10.44)$$

$${}^{t+\Delta t}\ddot{\mathbf{u}} = \frac{{}^{t+\Delta t}\dot{\mathbf{u}} - {}^t\dot{\mathbf{u}}}{\Delta t} \quad (10.45)$$

10.5.3 Hilber–Hughes–Taylor

The Hilber–Hughes–Taylor method [35] (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. (10.41) and Eq. (10.42) 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:

$$\begin{aligned} \mathbf{M} {}^{t+\Delta t}\ddot{\mathbf{u}} + (1 + \alpha)\mathbf{C} {}^{t+\Delta t}\dot{\mathbf{u}} - \alpha\mathbf{C} {}^t\dot{\mathbf{u}} + \\ (1 + \alpha) {}^{t+\Delta t}\mathbf{f}_{\text{int}} - \alpha {}^t\mathbf{f}_{\text{int}} = {}^{t+(1+\alpha)\Delta t}\mathbf{f}_{\text{ext}} \end{aligned} \quad (10.46)$$

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.

10.5.4 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 [6]. In the Wilson- θ method the acceleration is assumed to vary linearly in time from t' to $t' + \theta\Delta t'$ with $\theta \geq 1$ [Fig. 10.1]. 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

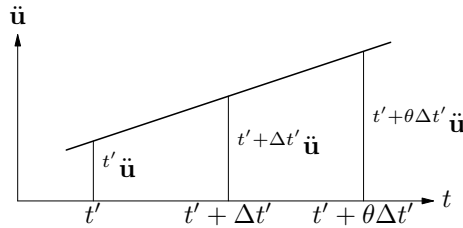


Figure 10.1: Linear acceleration

equations for ${}^{t'+\theta\Delta t'}\dot{\mathbf{u}}$ and ${}^{t'+\theta\Delta t'}\ddot{\mathbf{u}}$.

$${}^{t'+\theta\Delta t'}\ddot{\mathbf{u}} = \frac{6}{\theta^2\Delta t'^2} \left({}^{t'+\theta\Delta t'}\mathbf{u} - {}^t\mathbf{u} \right) - \frac{6}{\theta\Delta t'} {}^t\dot{\mathbf{u}} - 2 {}^t\ddot{\mathbf{u}} \quad (10.47)$$

$${}^{t'+\theta\Delta t'}\dot{\mathbf{u}} = \frac{3}{\theta\Delta t'} \left({}^{t'+\theta\Delta t'}\mathbf{u} - {}^t\mathbf{u} \right) - 2 {}^t\dot{\mathbf{u}} - \frac{\theta\Delta t'}{2} {}^t\ddot{\mathbf{u}} \quad (10.48)$$

The dynamic equilibrium equation is considered at $t' + \theta\Delta t'$.

$${}^{t'+\theta\Delta t'}\mathbf{M} {}^{t'+\theta\Delta t'}\ddot{\mathbf{u}} + {}^{t'+\theta\Delta t'}\mathbf{C} {}^{t'+\theta\Delta t'}\dot{\mathbf{u}} + {}^{t'+\theta\Delta t'}\mathbf{f}_{\text{int}} = {}^{t'+\theta\Delta t'}\mathbf{f}_{\text{ext}} \quad (10.49)$$

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 $t' \rightarrow t' + \theta\Delta t'$:

$${}^{t'+\theta\Delta t'}\mathbf{f}_{\text{ext}} = {}^t\mathbf{f}_{\text{ext}} + \theta \left({}^{t'+\Delta t'}\mathbf{f}_{\text{ext}} - {}^t\mathbf{f}_{\text{ext}} \right) \quad (10.50)$$

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.

10.5.5 Runge–Kutta

Runge–Kutta methods are a large class of methods for solving initial value problems of the form

$$\frac{d\mathbf{y}}{dt} = \mathbf{f}(\mathbf{y}, t) \quad (10.51)$$

By using the velocity $\mathbf{v} = \dot{\mathbf{u}}$ as an extra variable, the second order differential equation Eq. (10.37) is split into a system of two first order differential equations

$$\begin{aligned} \dot{\mathbf{u}} &= \mathbf{v} \\ \dot{\mathbf{v}} &= \mathbf{M}^{-1}(\mathbf{f}_{\text{ext}}(t) - \mathbf{C}\mathbf{v} - \mathbf{f}_{\text{int}}(\mathbf{u}, \mathbf{v}, \boldsymbol{\varepsilon}, \boldsymbol{\sigma}, t, \dots)) \end{aligned} \quad (10.52)$$

This system is of the same form as Eq. (10.51), so the Runge–Kutta method can be applied. A single step of an s -stage Runge–Kutta method is described by

$$\begin{aligned} {}^{t_i}\mathbf{y} &= {}^t\mathbf{y} + \Delta t \sum_{j=1}^s a_{ij} \mathbf{f}({}^{t_j}\mathbf{y}, t_j) \\ {}^{t+\Delta t}\mathbf{y} &= {}^t\mathbf{y} + \Delta t \sum_{j=1}^s b_j \mathbf{f}({}^{t_j}\mathbf{y}, t_j) \end{aligned} \quad (10.53)$$

with

$$t_i = t + c_i \Delta t \quad (10.54)$$

See for instance Hairer and Warner [33, § 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. (10.37) 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} :

$$\mathbf{e} = \frac{\alpha - \hat{\alpha}}{\alpha} \left(({}^{t+\Delta t}\mathbf{y} - {}^t\mathbf{y}) - \frac{1}{\alpha} ({}^{t_1}\mathbf{y} - {}^t\mathbf{y}) \right) \quad (10.55)$$

with $\hat{\alpha} = 1 - \frac{5}{4}\sqrt{2}$. The error estimate \mathbf{e} can be used to control the time step size, see Hairer and Warner [33, § IV.8], or Ellsiepen and Hartmann [23].

10.5.6 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 [82, 8] and Olson & Bathe [58]. Effects of large scale flow in the fluid are excluded. Attention is paid to the discretization method, the numerical solution techniques and simplifications.

Figure 10.2 shows a general fluid–structure interaction geometry. The solid extends

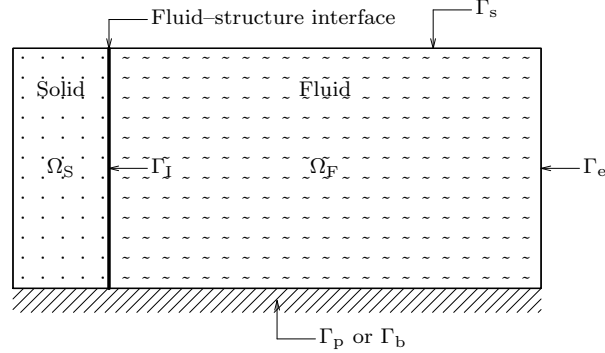


Figure 10.2: Fluid–structure interaction

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 .

10.5.6.1 Solid

In the solid, the discretization in the familiar form is given below and will be assumed throughout:

$$\mathbf{M}_S \ddot{\mathbf{u}} + \mathbf{C}_S \dot{\mathbf{u}} + \mathbf{K}_S \mathbf{u} + \mathbf{f}_I = \mathbf{f}_S^{\text{ex}}(t) \quad (10.56)$$

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.

10.5.6.2 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

$$\nabla^2 p = \frac{1}{c^2} \ddot{p} \quad (10.57)$$

where p is the pressure (compression positive) and c the wave speed, given by

$$c^2 = \frac{\beta}{\rho} \quad (10.58)$$

where β is the bulk modulus and ρ the density. Appropriate boundary conditions of the following form can be imposed.

Solid boundary. The conditions applying to the surface Γ_I being the interface between the fluid and structure, can be written as

$$\frac{\partial p}{\partial n} = -\rho_F \mathbf{n}_F^T \ddot{\mathbf{u}}_F \quad \text{and} \quad \boldsymbol{\sigma} \mathbf{n}_S = p_F \mathbf{n}_F \quad \text{on} \quad \Gamma_I \quad (10.59)$$

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. (10.59)

$$\frac{\partial p}{\partial n} = -\rho_F \mathbf{n}_F^T \ddot{\mathbf{u}}_S \quad (10.60)$$

Prescribed conditions.

$$p = \bar{p} \quad \text{on} \quad \Gamma_p \quad (10.61)$$

where \bar{p} is a prescribed pressure often to be zero along part of the boundary Γ_p .

Free surface.

$$p = \rho_F g u_z \quad \text{on} \quad \Gamma_s \quad (10.62)$$

where g is the gravity acceleration and z is directed normal to the free surface. Noting that

$$\ddot{u}_z = -\frac{1}{\rho_F} \frac{\partial p}{\partial z} \quad (10.63)$$

Eq. (10.62) can be written as

$$\frac{\partial p}{\partial z} = -\frac{1}{g} \ddot{p} \quad \text{on} \quad \Gamma_s \quad (10.64)$$

which is the linearized free surface condition for first order waves.

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 exists. In the existence of only outgoing waves, incoming waves are supposed to be absent, giving a solution of the form

$$p = f'(x - ct) \quad (10.65)$$

where a positive x is the outward direction. The radiation boundary condition is now obtained by eliminating f' and is given by

$$\frac{\partial p}{\partial x} = -\frac{1}{c} \dot{p} \quad \text{on} \quad \Gamma_e \quad (10.66)$$

where c is the wave speed given by Eq. (10.58). 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.

Bottom. The conditions applying to the surface Γ_b being the bottom of the fluid reservoir, can be written as

$$\frac{\partial p}{\partial n} = -\frac{1 - \alpha_B}{c(1 + \alpha_B)} \dot{p} \quad \text{on} \quad \Gamma_b \quad (10.67)$$

where c is the wave speed given by Eq. (10.58) 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 [25] and Küçükarslan et al. [46].

10.5.6.3 Discretized Coupled Equations

A standard finite element discretization has used approximating p in terms of nodal values \mathbf{p}

$$p \approx \mathbf{N}_F \mathbf{p}^e \quad (10.68)$$

and the discretization gives a system of equations in a form

$$\mathbf{M}_F \ddot{\mathbf{p}} + \mathbf{C}_F \dot{\mathbf{p}} + \mathbf{K}_F \mathbf{p} + \mathbf{r}_I = 0 \quad (10.69)$$

where \mathbf{M}_F , \mathbf{C}_F , \mathbf{K}_F and \mathbf{r}_I are defined in terms of the following element matrices:

$$[M_{ij}]_F^e = \frac{1}{g} \int_{\Gamma_s^e} N_i N_j d\Gamma + \frac{1}{c^2} \int_{\Omega_F} N_i N_j d\Omega \quad (10.70)$$

with Γ_s the free surface and Ω_F the fluid reservoir

$$[C_{ij}]_F^e = \frac{1}{c_s} \int_{\Gamma_e^e} N_i N_j d\Gamma + \frac{1 - \alpha_B}{c(1 + \alpha_B)} \int_{\Gamma_b^e} N_i N_j d\Gamma \quad (10.71)$$

with Γ_e the radiation boundary and Γ_b the reservoir bottom

$$[K_{ij}]_F^e = \int_{\Omega_F^e} \nabla N_i \nabla N_j d\Omega \quad \text{with } \Omega_F \text{ the fluid domain} \quad (10.72)$$

$$[r_i]_I^e = \int_{\Gamma_I^e} N_i \rho_F n_k \ddot{u}_k d\Gamma \quad \text{with } \Gamma_I \text{ the fluid-structure interface} \quad (10.73)$$

The contribution \mathbf{f}_I from Eq. (10.56) can be written as

$$[f_{ik}]_I^e = - \int_{\Gamma_I^e} N_i^u n_k p d\Gamma \quad (10.74)$$

or

$$\mathbf{f}_I^e = -\mathbf{R}^{eT} \mathbf{p}^e \quad (10.75)$$

and likewise the contribution \mathbf{r}_I from Eq. (10.69) as

$$[r_i]_I^e = \int_{\Gamma_I^e} N_i^p \rho_F n_k \ddot{u}_k d\Gamma \quad (10.76)$$

or

$$\mathbf{r}_I^e = \rho_F \mathbf{R}^e \ddot{\mathbf{u}}^e \quad (10.77)$$

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

$$\begin{bmatrix} \mathbf{M}_S & \mathbf{O}^T \\ \rho_F \mathbf{R} & \mathbf{M}_F \end{bmatrix} \begin{Bmatrix} \ddot{\mathbf{u}} \\ \ddot{\mathbf{p}} \end{Bmatrix} + \begin{bmatrix} \mathbf{C}_S & \mathbf{O}^T \\ \mathbf{O} & \mathbf{C}_F \end{bmatrix} \begin{Bmatrix} \dot{\mathbf{u}} \\ \dot{\mathbf{p}} \end{Bmatrix} + \begin{bmatrix} \mathbf{K}_S & -\mathbf{R}^T \\ \mathbf{O} & \mathbf{K}_F \end{bmatrix} \begin{Bmatrix} \mathbf{u} \\ \mathbf{p} \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}_S(t) \\ \mathbf{0} \end{Bmatrix} \quad (10.78)$$

The technique to be used for solving the system of equations Eq. (10.78) strongly depends on the form of the forcing function $\mathbf{f}_S(t)$.

10.5.6.4 Frequency Domain Analysis

If the forcing function of Eq. (10.78) has been expressed in, or can be transformed to a periodic form as

$$\mathbf{f}_S(t) = \hat{\mathbf{f}}_S e^{i\omega t} \quad (10.79)$$

then for linear problems the steady-state solution will exist in the same form, thus

$$\mathbf{u}(t) = \hat{\mathbf{u}} e^{i\omega t} \quad \text{and} \quad \mathbf{p}(t) = \hat{\mathbf{p}} e^{i\omega t} \quad (10.80)$$

Now a complex expression of the solution is obtained and can be written in the matrix form

$$\begin{bmatrix} -\omega^2 \mathbf{M}_S + \mathbf{K}_S + i\omega \mathbf{C}_S & -\mathbf{R}^T \\ -\omega^2 \rho_F \mathbf{R} & -\omega^2 \mathbf{M}_F + \mathbf{K}_F + i\omega \mathbf{C}_F \end{bmatrix} \begin{Bmatrix} \hat{\mathbf{u}} \\ \hat{\mathbf{p}} \end{Bmatrix} = \begin{Bmatrix} \hat{\mathbf{f}}_S \\ \mathbf{0} \end{Bmatrix} \quad (10.81)$$

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. (10.81) can be written

$$\hat{\mathbf{p}} = [-\omega^2 \mathbf{M}_F + \mathbf{K}_F + i\omega \mathbf{C}_F]^{-1} \omega^2 \rho_F \mathbf{R} \hat{\mathbf{u}} \quad (10.82)$$

or with $\mathbf{H}_F(\omega) = [-\omega^2 \mathbf{M}_F + \mathbf{K}_F + i\omega \mathbf{C}_F]^{-1}$ as frequency response function

$$\hat{\mathbf{p}} = \omega^2 \rho_F \mathbf{H}_F(\omega) \mathbf{R} \hat{\mathbf{u}} \quad (10.83)$$

On substitution for $\hat{\mathbf{p}}$ the result from above into the first subsystem of equations implied in Eq. (10.81) an additional fluid matrix $\hat{\mathbf{K}}_I$ is obtained and is given as

$$\hat{\mathbf{K}}_I = -\omega^2 \rho_F \mathbf{R}^T \mathbf{H}_F(\omega) \mathbf{R} \quad (10.84)$$

In the above the matrix $\hat{\mathbf{K}}_I$ is a complex quantity and can be written in a form

$$\mathbf{K}_I = -\omega^2 \rho_F \Re(\mathbf{R}^T \mathbf{H}_F(\omega) \mathbf{R}) - \omega^2 \rho_F \Im(\mathbf{R}^T \mathbf{H}_F(\omega) \mathbf{R}) \quad (10.85)$$

or

$$\mathbf{K}_I = -\omega^2 \tilde{\mathbf{M}}_F + i\omega \tilde{\mathbf{C}}_F \quad (10.86)$$

where $\tilde{\mathbf{M}}_F$ and $\tilde{\mathbf{C}}_F$ are denoted as the *added mass* matrix and the *added damping* matrix respectively:

$$\tilde{\mathbf{M}}_F = \Re(\mathbf{R}^T \mathbf{H}_F(\omega) \mathbf{R}) \quad (10.87)$$

$$\tilde{\mathbf{C}}_F = -\omega \Im(\mathbf{R}^T \mathbf{H}_F(\omega) \mathbf{R}) \quad (10.88)$$

The structural matrix now takes the form

$$\left[-\omega^2 (\mathbf{M}_S + \tilde{\mathbf{M}}_F) + \mathbf{K}_S + i\omega (\mathbf{C}_S + \tilde{\mathbf{C}}_F) \right] \hat{\mathbf{u}} = \hat{\mathbf{f}}_S \quad (10.89)$$

which can be solved with a direct solution procedure [§ 10.3.2 p. 194].

10.5.7 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.,

$$p = 0 \quad \text{respectively on } \Gamma_s \text{ and } \Gamma_e \quad (10.90)$$

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. (10.69) as well as the second of Eq. (10.78) become zero. The pressure vector \mathbf{p} can now be obtained directly in terms of $\ddot{\mathbf{u}}$ as

$$\mathbf{p} = -\mathbf{K}_F^{-1} \rho_F \mathbf{R} \ddot{\mathbf{u}} \quad (10.91)$$

On substitution into the first of Eq. (10.78) the structural matrix now becomes of the general form

$$(\mathbf{M}_S + \tilde{\mathbf{M}}_F) \ddot{\mathbf{u}} + \mathbf{C}_S \dot{\mathbf{u}} + \mathbf{K}_S \mathbf{u} = \mathbf{f}_S(t) \quad (10.92)$$

where the added mass is simply given as

$$\tilde{\mathbf{M}}_F = \rho_F \mathbf{R}^T \mathbf{K}_F^{-1} \mathbf{R} \quad (10.93)$$

10.5.7.1 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 [§ 10.3.1 p. 194]. Now the solution \mathbf{u} will be obtained by superposition of the response in each mode:

$$\mathbf{u}(\Omega) = \sum_{i=1}^{\infty} \phi_i \alpha_i(\Omega) \quad (10.94)$$

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. (10.92) with damping neglected

$$\left(\mathbf{M}_S + \tilde{\mathbf{M}}_F \right) \ddot{\mathbf{u}} + \mathbf{K}_S \mathbf{u} = 0 \quad (10.95)$$

Next, the set of equations Eq. (10.92) are transformed to global coordinates and a decoupled set of equations is obtained provided that the damping matrix \mathbf{C}_S is proportional [§ 10.1.2.3 p. 192].

10.5.7.2 Time Domain Analysis

In case of an arbitrary transient loading, the response of the simplified problem in Eq. (10.92) can now be obtained by a direct time integration method [§ 10.5 p. 196]. After the *added mass* matrix $\tilde{\mathbf{M}}_F$, given by Eq. (10.93), has been determined via the solution of Eq. (10.91), the transient analysis can be carried out in the usual way.

10.6 Specific Elements

10.6.1 Boundary Surface Elements

These line and surface elements may be applied to add mass and/or stiffness properties to an outer surface of a finite element model. These elements may be applied either to add distributed mass to a finite element model without influencing the stiffness of the model, or to model the free field medium in a dynamic analysis.

Line boundary surface elements are to be placed on the outer edge of a two-dimensional model or on the upper face of a line interface element. Plane boundary surface elements are to be placed on the outer surface of a three-dimensional model or on the upper face of a plane interface element.

The formulation of the line and plane boundary surface elements is fully iso-parametric. This means that the quadratic line elements may be straight as well as curved, and that the plane elements may be flat as well as curved.

Distributed translational mass Boundary surface elements can be used as distributed mass elements to affect the inertia mass in a dynamic analysis, such as e.g. the contribution of fluid to the dynamic behaviour of the structure in a fluid-structure interaction analysis. The stiffness at the nodes of a distributed translational mass should be supplied by other elements. The distributed translational mass elements do not have any post-analysis results like strains or stresses.

Note that the added masses as defined for distributed mass elements do not have a contribution in the weight load, in contrary to point-mass elements [§ 7.8.1.9 p. 101] which do have both an inertia contribution in a dynamic analysis and a contribution to the weight load vector.

Free field medium Boundary surface elements can be used to model a free field medium in a dynamic analysis. The mass and stiffness properties of these elements can be used to take the effects of the free field medium into account in a dynamic seismic analysis.

For line surface elements, the local x axis is tangential to the line and point from the first node to the second node. The normal perpendicular to the boundary surface y must point into the free field medium and an approximate direction of \tilde{y} can be defined directly in the geometry definition.

With the \tilde{y} direction a normal z perpendicular to the element xy -plane is calculated as $z \perp x\tilde{y}$ and the actual normal will be calculated as $\vec{y} \perp zx$, possibly resulting in a direction that is slightly different from the specified direction.

If the approximate \tilde{y} direction is not defined, the the global Z axis is used as the normal to the element xy -plane and the actual normal direction is calculated as $\vec{y} \perp Zx$.

For plane boundary surface elements, i.e. triangles or quadrilaterals, the x axis and y axis are in the tangential plane; by default the x axis points from the first node to the second node and the y axis points from the first node to the last node. The default normal z is setup perpendicular to the tangential element xy -plane as $z \perp xy$ and after that the actual y axis is setup as $y \perp zx$.

If the element x axis is preferred other than the default, a preferred direction \tilde{x} can be defined in the geometry definition. The actual x and y axes are now calculated as $y \perp z\tilde{x}$ and $x \perp yz$.

The normal z perpendicular to the boundary surface must point into the free field medium and an approximate direction of \tilde{z} can be defined directly in the geometry definition. When the approximated normal direction \tilde{z} does not point into the same direction as the calculated element normal z , the direction of the normal is reversed $\vec{z} = -z$, and finally the actual y axis is set up as $y \perp \vec{z}x$.

For line and plane free field medium boundary surface elements, DIANA performs a numerical integration scheme with an appropriate default scheme as indicated for the individual elements. You may choose an alternative scheme via a special data input, depending on the shape of the element.

For line and plane distributed mass boundary surface elements, DIANA does not perform a numerical integration scheme.

10.6.1.1 L4TM – straight line, 2 nodes, 2-D

The L4TM element [Fig. 10.3] is a two-node line element which acts as a surface boundary

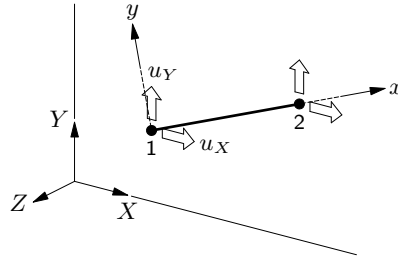


Figure 10.3: L4TM

in the finite element model. This element may be applied either to add distributed mass to a finite element model without influencing the stiffness of the model, or to model the free field medium in a dynamic analysis. The local element x axis is in the direction of the line and the local element y axis is normal to the line element. By default DIANA applies a 1-point Gauss integration scheme along the bar axis when the element is used to model the free field medium.

$[n_\xi = 1]$

When the boundary surface elements are specified for modelling the behaviour of the free field medium in a dynamic analysis, specific input of mass density and stiffness properties is required. When the boundary surface elements are only specified to affect the inertia mass in a dynamic analysis, only specific input of distributed mass is required.

The basic (and only) variables of the L4TM element are the translations in the global XY directions.

$$\mathbf{u}_e = \begin{Bmatrix} u_X \\ u_Y \end{Bmatrix} \quad (10.96)$$

10.6.1.2 CL6TM – curved line, 3 nodes, 2-D

The CL6TM element [Fig. 10.4] is a three-node line element which acts as a surface bound-

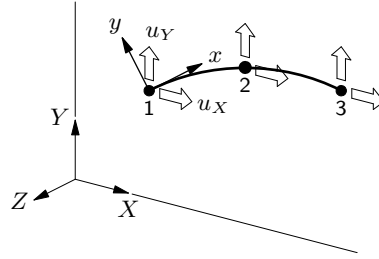


Figure 10.4: CL6TM

ary in the finite element model. This element may be applied either to add distributed mass to a finite element model without influencing the stiffness of the model, or to model the free field medium in a dynamic analysis. The local element x axis is in the direction of the line and the local element y axis is normal to the line element. By default DIANA applies a 2-point Gauss integration scheme along the bar axis when the element is used to model the free field medium.

[$n_\xi = 2$]

When the boundary surface elements are specified for modelling the behaviour of the free field medium in a dynamic analysis, specific input of mass density and stiffness properties is required. When the boundary surface elements are only specified to affect the inertia mass in a dynamic analysis, only specific input of distributed mass is required.

The basic (and only) variables of the CL6TM element are the translations in the global XY directions.

$$\mathbf{u}_e = \begin{Bmatrix} u_X \\ u_Y \end{Bmatrix} \quad (10.97)$$

10.6.1.3 L6TM – straight line, 2 nodes, 3-D

The L6TM element [Fig. 10.5] is a two-node line element which acts as a three-dimension-

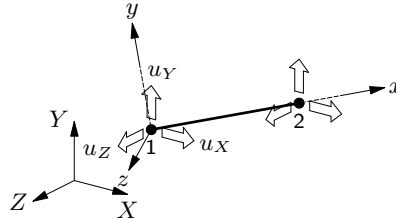


Figure 10.5: L6TM

al distributed line mass element. This element may be applied to add distributed mass to a finite element model without influencing the stiffness of the model. The local element x axis is in the direction of the line and the local element y and z axes are normal to the line element. Only specific input of distributed mass is required.

The basic (and only) variables of the L6TM element are the translations in the global XYZ directions.

$$\mathbf{u}_e = \begin{Bmatrix} u_X \\ u_Y \\ u_Z \end{Bmatrix} \quad (10.98)$$

10.6.1.4 CL9TM – curved line, 3 nodes, 3-D

The CL9TM element [Fig. 10.6] is a three-node line element which acts as a three-dimensional distributed line mass element. This element may be applied to add distributed mass to a finite element model without influencing the stiffness of the model. The local

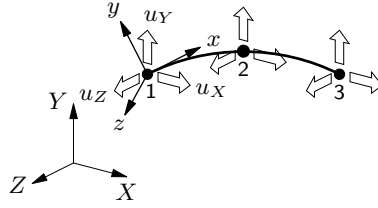


Figure 10.6: CL9TM

element x axis is in the direction of the line and the local element y and z axes are normal to the line element. Only specific input of distributed mass is required.

The basic (and only) variables of the CL9TM element are the translations in the global XYZ directions.

$$\mathbf{u}_e = \begin{Bmatrix} u_X \\ u_Y \\ u_Z \end{Bmatrix} \quad (10.99)$$

10.6.1.5 T9TM – triangle, 3 nodes, 3-D

The T9TM element [Fig. 10.7] is a three-node triangular element which acts as a surface

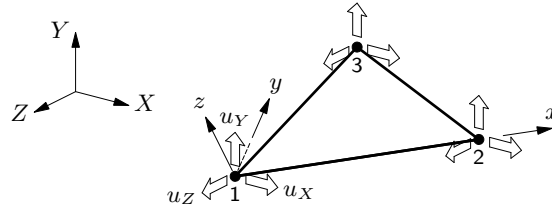


Figure 10.7: T9TM

boundary in the finite element model. This element may be applied either to add distributed mass to a finite element model without influencing the stiffness of the model, or to model the free field medium in a dynamic analysis. The local element x axis and y axis are in the plane of the line and the local element z axis is normal to the plane. By default DIANA applies a 1-point integration scheme when the element is used to model the free field medium. [$n_{lc} = 1$]

When the boundary surface elements are specified for modelling the behaviour of the free field medium in a dynamic analysis, specific input of mass density and stiffness properties is required. When the boundary surface elements are only specified to affect the inertia mass in a dynamic analysis, only specific input of distributed mass is required.

The basic (and only) variables of the T9TM element are the translations in the global XYZ directions.

$$\mathbf{u}_e = \begin{Bmatrix} u_X \\ u_Y \\ u_Z \end{Bmatrix} \quad (10.100)$$

10.6.1.6 Q12TM – quadrilateral, 4 nodes, 3-D

The Q12TM element [Fig. 10.8] is a four-node quadrilateral element which acts as a

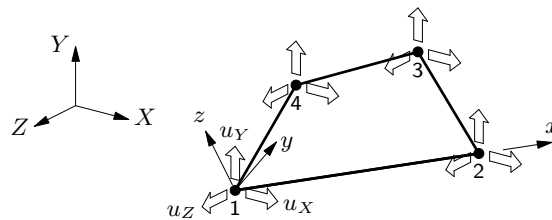


Figure 10.8: Q12TM

$[n_\xi = 2, n_\eta = 2]$

surface boundary in the finite element model. This element may be applied either to add distributed mass to a finite element model without influencing the stiffness of the model, or to model the free field medium in a dynamic analysis. The local element x axis and y axis are in the plane of the line and the local element z axis is normal to the plane. By default DIANA applies a 2×2 integration scheme when the element is used to model the free field medium.

When the boundary surface elements are specified for modelling the behaviour of the free field medium in a dynamic analysis, specific input of mass density and stiffness properties is required. When the boundary surface elements are only specified to affect the inertia mass in a dynamic analysis, only specific input of distributed mass is required.

The basic (and only) variables of the Q12TM element are the translations in the global XYZ directions.

$$\mathbf{u}_e = \begin{Bmatrix} u_X \\ u_Y \\ u_Z \end{Bmatrix} \quad (10.101)$$

10.6.1.7 CT18TM – triangle, 6 nodes, 3-D

The CT18TM element [Fig. 10.9] is a six-node triangular element which acts as a surface

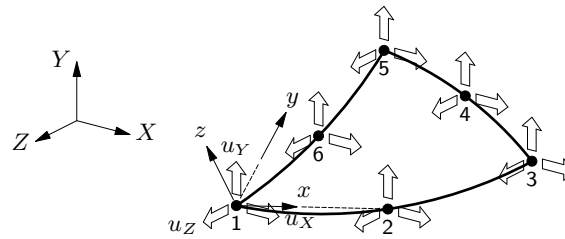


Figure 10.9: CT18TM

boundary in the finite element model. This element may be applied either to add distributed mass to a finite element model without influencing the stiffness of the model, or to model the free field medium in a dynamic analysis. The local element x axis and y axis are in the plane of the line and the local element z axis is normal to the plane. By default DIANA applies a 3-point integration scheme when the element is used to model the free field medium.

$[n_{lc} = 3]$

When the boundary surface elements are specified for modelling the behaviour of the free field medium in a dynamic analysis, specific input of mass density and stiffness properties is required. When the boundary surface elements are only specified to affect the inertia mass in a dynamic analysis, only specific input of distributed mass is required.

The basic (and only) variables of the CT18TM element are the translations in the global XYZ directions.

$$\mathbf{u}_e = \begin{Bmatrix} u_X \\ u_Y \\ u_Z \end{Bmatrix} \quad (10.102)$$

10.6.1.8 CQ24TM – quadrilateral, 8 nodes, 3-D

The CQ24TM element [Fig. 10.10] is a eight-node quadrilateral element which acts as a surface boundary in the finite element model. This element may be applied either to add distributed mass to a finite element model without influencing the stiffness of the model, or to model the free field medium in a dynamic analysis. The local element x axis and y axis are in the plane of the line and the local element z axis is normal to the plane. By default DIANA applies a 2×2 integration scheme when the element is used to model the free field medium.

$[n_\xi = 2, n_\eta = 2]$

When the boundary surface elements are specified for modelling the behaviour of the free field medium in a dynamic analysis, specific input of mass density and stiffness properties is required. When the boundary surface elements are only specified to affect the inertia mass in a dynamic analysis, only specific input of distributed mass is required.

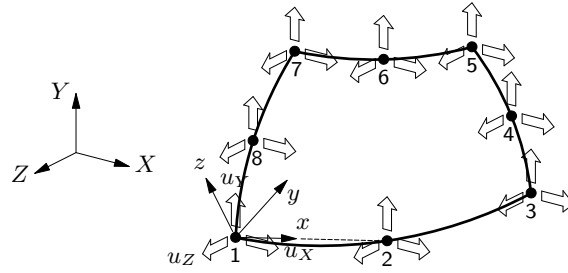


Figure 10.10: CQ24TM

The basic (and only) variables of the CQ24TM element are the translations in the global XYZ directions.

$$\mathbf{u}_e = \begin{Bmatrix} u_X \\ u_Y \\ u_Z \end{Bmatrix} \quad (10.103)$$

10.6.2 Fluid–Structure Interface Elements

Fluid–structure interface elements are used in dynamic fluid–structure interaction analysis [§ 10.5.6 p. 200] to couple the fluid to the structural domain. The coupling is realized on its interface by continuity of the pressure of the fluid and the normal displacement of the structure. For all fluid–structure interface elements you must specify the fluid density ρ_f as a material property. With respect to shape and connectivity there are two types of fluid–structure interface elements:

Line interface elements to be placed between edges of two-dimensional structural elements (like axisymmetric, plane stress or plane strain) and two-dimensional flow elements.

Plane interface elements to be placed between faces of solid (three-dimensional) structural elements and three-dimensional flow elements.

The formulation of the line and plane interface elements is fully isoparametric. This means that the quadratic line interface elements may be straight as well as curved, and the plane interface elements may be flat as well as curved.

For fluid–structure interface elements, the node sequence in the connectivity specification determines the fluid and the structural sides of the element.

The normal direction of a fluid–structure interface element must point outward into the fluid domain. DIANA determines the local xyz directions through the topology of the element.

For line elements the x axis points from node 1 to node 2 and the z axis parallel to the global Z axis, then the local y axis determines the normal direction. For plane elements the x axis points from node 1 to node 2, and an \bar{y} axis from node 1 to the last ‘structural’ node, then the z axis determines the normal direction. If you prefer a normal axis other than default, then you must specify its direction.

The basic variables for fluid–structural interfaces are the nodal displacements \mathbf{u}_e and the fluid nodal pressures \mathbf{p} . The derived values are the relative displacements $\Delta\mathbf{u}$ and the tractions \mathbf{t} .

Variables of two-dimensional fluid–structural interfaces are oriented in the local xy axes.

$$\mathbf{u}_e = \begin{Bmatrix} u_x \\ u_y \end{Bmatrix} \quad \mathbf{p} = \{ p_x \} \quad (10.104)$$

The normal traction t_x is perpendicular to the interface; the shear traction t_y is tangential to the interface. p_x is a scalar value. The fluid–structure interface elements need to be placed between the edges of two-dimensional quadratic structural elements and two-dimensional flow elements [Fig. 10.11].

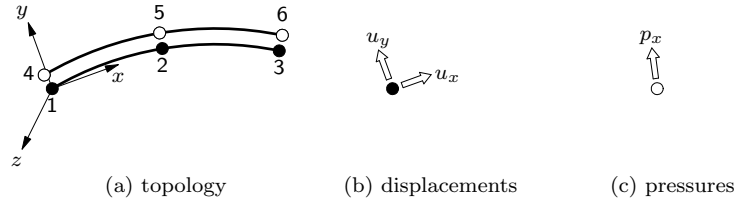


Figure 10.11: Variables of two-dimensional fluid-structural interfaces

Variables of three-dimensional fluid-structural interfaces are oriented in the local xyz axes.

$$\mathbf{u}_e = \begin{Bmatrix} u_x \\ u_y \\ u_z \end{Bmatrix} \quad \mathbf{p} = \{ p_x \} \quad (10.105)$$

The normal traction t_x is perpendicular to the interface; the shear traction t_y and t_z are tangential to the interface. p_x is a scalar value. The fluid-structure interface elements

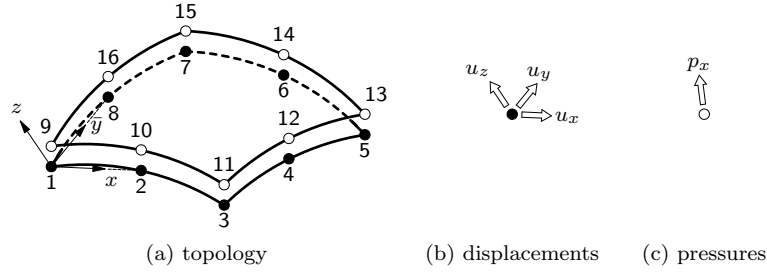


Figure 10.12: Variables of three-dimensional fluid-structural interfaces

need to be placed between the edges of three-dimensional quadratic structural elements and three-dimensional flow elements [Fig. 10.12].

Fluid-structure interface elements have two types of basic degrees of freedom: displacements and pressures. While solving the structural domain, the pressure degrees of freedom must be suppressed to get a solvable system of equations. DIANA automatically generates these supports. Note that these supports are additional to the regular structural supports.

10.6.2.1 BL4S2 – line, 2+2 nodes, 2-D

The BL4S2 element is a fluid-structure line interface element to be placed between the

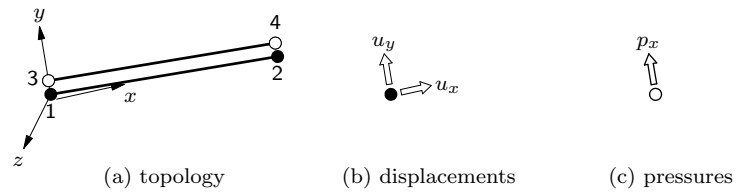


Figure 10.13: BL4S2

edges of two-dimensional linear structural elements and two-dimensional linear flow elements [Fig. 10.13]. Based on isoparametric linear interpolation for the displacements and linear interpolation for the pressures this element models the continuity between the normal displacement on the structural side and the pressure on the fluid side. The element xy axes for the displacements in the nodes are evaluated in node 1: x from node 1 to node 2, and z parallel to the global Z axis. The matching y axis must point in the direction of the fluid domain. The pressure p_x acts in a direction perpendicular to the interface element. The element is numerically integrated via a Gauss scheme. By default DIANA [n_ξ=1] applies a 1-point Gauss integration scheme, 2-point Gauss is a suitable option.

10.6.2.2 BCL6S3 – line, 3+3 nodes, 2-D

The BCL6S3 element is a fluid–structure line interface element to be placed between the

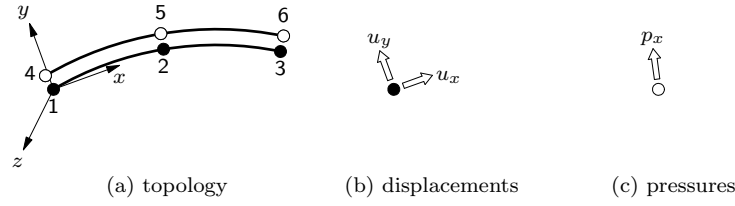


Figure 10.14: BCL6S3

edges of two-dimensional quadratic structural elements and two-dimensional quadratic flow elements [Fig. 10.14]. Based on isoparametric quadratic interpolation for displacements and pressures this element models the continuity between the normal displacement on the structural side and the pressure on the fluid side. The element xy axes for the displacements in the nodes are evaluated in node 1: x from node 1 to node 2, and z parallel to the global Z axis. The matching y axis must point in the direction of the fluid domain. The pressure p_x acts in a direction perpendicular to the interface element. The element is numerically integrated via a Gauss scheme. By default DIANA applies a 2-point Gauss integration scheme, 3-point Gauss is a suitable option. [$n_{\xi}=2$]

10.6.2.3 BCL6S2 – line, 3+2 nodes, 2-D

The BCL6S2 element is a fluid–structure line interface element to be placed between

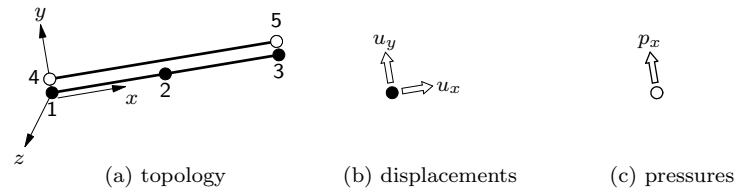


Figure 10.15: BCL6S2

the edges of two-dimensional quadratic structural elements and two-dimensional linear flow elements [Fig. 10.15]. Based on isoparametric quadratic interpolation for the displacements and linear interpolation for the pressures this element models the continuity between the normal displacement on the structural side and the pressure on the fluid side. The element xy axes for the displacements in the nodes are evaluated in node 1: x from node 1 to node 2, and z parallel to the global Z axis. The matching y axis must point in the direction of the fluid domain. The pressure p_x acts in a direction perpendicular to the interface element. The element is numerically integrated via a Gauss scheme. By default DIANA applies a 2-point Gauss integration scheme, 3-point Gauss is a suitable option. [$n_{\xi}=2$]

10.6.2.4 BQ12S4 – quadrilateral, 4+4 nodes, 3-D

The BQ12S4 element is a fluid–structure plane interface element to be placed between

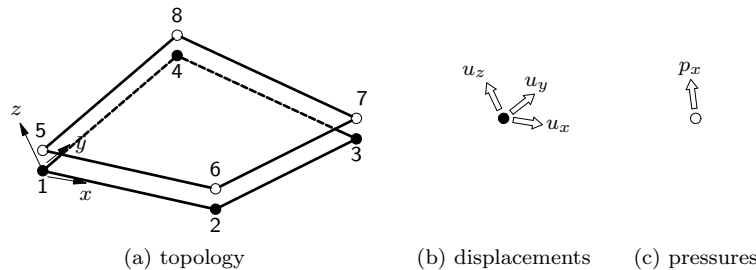


Figure 10.16: BQ12S4

the faces of linear solid (three-dimensional) structural elements and three-dimensional linear flow elements [Fig. 10.18]. Based on isoparametric linear interpolation for the displacements and linear interpolation for the pressures it models the continuity between the normal displacement on the structural side and the pressure on the fluid side. The element xyz axes for the displacements in the nodes are evaluated in node 1: x from node 1 to node 2 and \bar{y} from node 1 to node 4. The matching z axis must point in the direction of the fluid domain. The \bar{y} axis is corrected to $y \perp zx$. The pressure p_x acts in a direction perpendicular to the interface element. The element is numerically integrated via a Gauss scheme. By default DIANA applies a 2×2 Gauss integration scheme.

$[n_\xi=2, n_\eta=2]$

10.6.2.5 BQ24S8 – quadrilateral, 8+8 nodes, 3-D

The BQ24S8 element is a fluid–structure plane interface element to be placed between

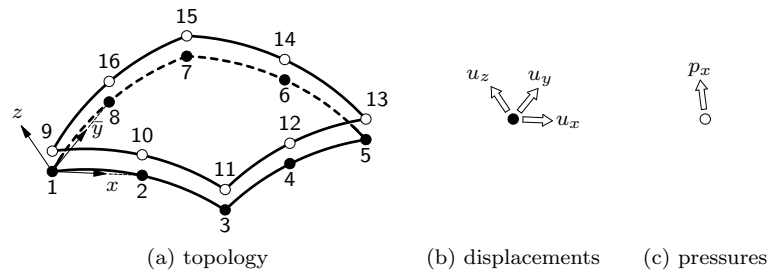


Figure 10.17: BQ24S8

the faces of quadratic solid (three-dimensional) structural elements and three-dimensional quadratic flow elements [Fig. 10.17]. Based on isoparametric quadratic interpolation for the displacements and the pressures this element models the continuity between the normal displacement on the structural side and the pressure on the fluid side. The element xyz axes for the displacements in the nodes are evaluated in node 1: x from node 1 to node 2 and \bar{y} from node 1 to node 8. The matching z axis must point in the direction of the fluid domain. The \bar{y} axis is corrected to $y \perp zx$. The pressure p_x acts in a direction perpendicular to the interface element. The element is numerically integrated via a Gauss scheme. By default DIANA applies a 3×3 Gauss integration scheme, 2×2 Gauss is a suitable option.

$[n_\xi=3, n_\eta=3]$

10.6.2.6 BQ24S4 – quadrilateral, 8+4 nodes, 3-D

The BQ24S4 element is a fluid–structure plane interface element to be placed between

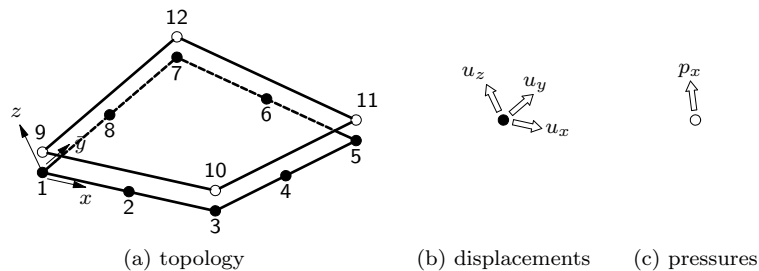


Figure 10.18: BQ24S4

the faces of quadratic solid (three-dimensional) structural elements and three-dimensional linear flow elements [Fig. 10.18]. Based on isoparametric quadratic interpolation for the displacements and linear interpolation for the pressures it models the continuity between the normal displacement on the structural side and the pressure on the fluid side. The element xyz axes for the displacements in the nodes are evaluated in node 1: x from node 1 to node 2 and \bar{y} from node 1 to node 8. The matching z axis must point in the direction of the fluid domain. The \bar{y} axis is corrected to $y \perp zx$. The pressure p_x acts in a direction perpendicular to the interface element. The element is numerically integrated

$[n_\xi=3, n_\eta=3]$ via a Gauss scheme. By default DIANA applies a 3×3 Gauss integration scheme, 2×2 Gauss is a suitable option.

10.6.2.7 BT9S3 – triangular, 3+3 nodes, 3-D

The BT9S3 element is a fluid–structure plane interface element to be placed between

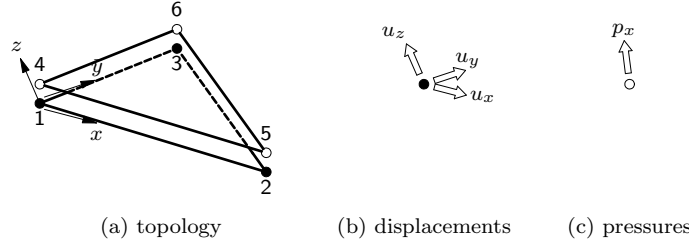


Figure 10.19: BT9S3

the faces of linear solid (three-dimensional) structural elements and three-dimensional linear flow elements [Fig. 10.19]. Based on isoparametric linear interpolation for the displacements and linear interpolation for the pressures it models the continuity between the normal displacement on the structural side and the pressure on the fluid side. The element xyz axes for the displacements in the nodes are evaluated in node 1: x from node 1 to node 2 and \bar{y} from node 1 to node 3. The matching z axis must point in the direction of the fluid domain. The \bar{y} axis is corrected to $y \perp zx$. The pressure p_x acts in a direction perpendicular to the interface element. By default DIANA applies a 3-point integration scheme.

$[n_\xi=3]$

10.6.2.8 BT18S6 – triangular, 6+6 nodes, 3-D

The BT18S6 element is a fluid–structure plane interface element to be placed between

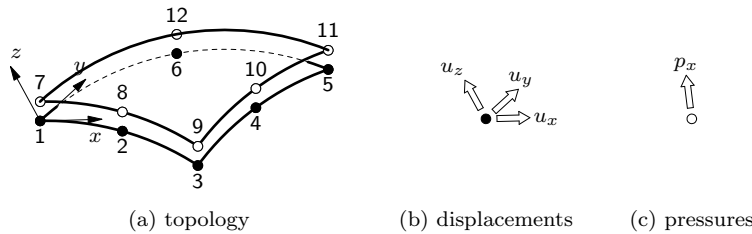


Figure 10.20: BT18S6

the faces of quadratic solid (three-dimensional) structural elements and three-dimensional quadratic flow elements [Fig. 10.20]. Based on isoparametric quadratic interpolation for the displacements and linear interpolation for the pressures it models the continuity between the normal displacement on the structural side and the pressure on the fluid side. The element xyz axes for the displacements in the nodes are evaluated in node 1: x from node 1 to node 2 and \bar{y} from node 1 to node 6. The matching z axis must point in the direction of the fluid domain. The \bar{y} axis is corrected to $y \perp zx$. The pressure p_x acts in a direction perpendicular to the interface element. By default DIANA applies a 4-point integration scheme.

$[n_\xi=4]$

10.6.2.9 BT18S3 – triangular, 6+3 nodes, 3-D

The BT18S3 element is a fluid–structure plane interface element to be placed between the faces of quadratic solid (three-dimensional) structural elements and three-dimensional linear flow elements [Fig. 10.21]. Based on isoparametric quadratic interpolation for the displacements and linear interpolation for the pressures it models the continuity between

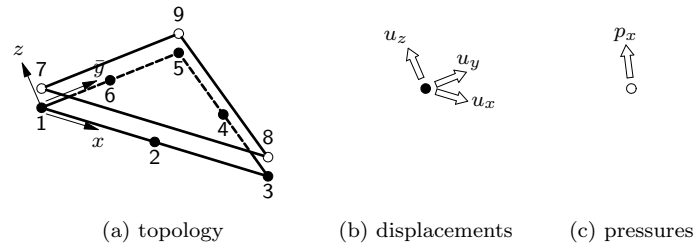


Figure 10.21: BT18S3

the normal displacement on the structural side and the pressure on the fluid side. The element xyz axes for the displacements in the nodes are evaluated in node 1: x from node 1 to node 2 and \bar{y} from node 1 to node 6. The matching z axis must point in the direction of the fluid domain. The \bar{y} axis is corrected to $y \perp zx$. The pressure p_x acts in a direction perpendicular to the interface element. By default DIANA applies a 4-point integration

[$n_{\xi}=4$]

Chapter 11

Nonlinear Analysis

11.1 Geometrical Nonlinear Analysis

11.1.1 Large Displacements, Rotations and Strains

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 11.1 shows two examples of structures for which the small displacement and rotation assumptions do no longer hold.

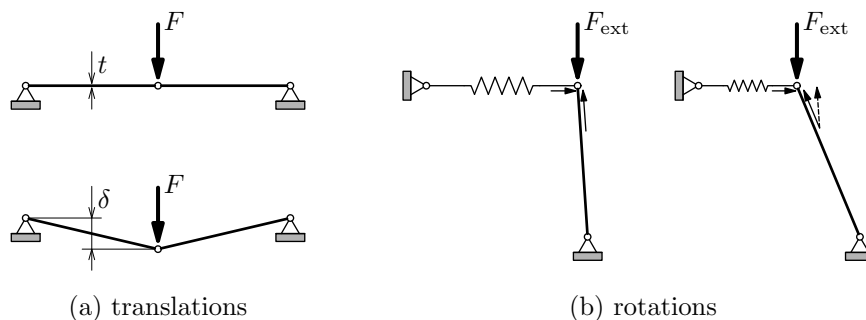


Figure 11.1: Large displacements

Figure 11.1a 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 11.1b 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 [6, § 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.

11.1.1.1 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. 11.2]. The

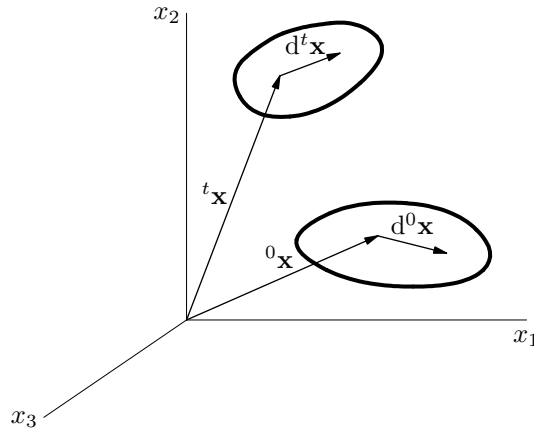


Figure 11.2: Coordinate Frames

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

$${}^t\mathbf{u} = {}^t\mathbf{x} - {}^0\mathbf{x} \quad (11.1)$$

The deformation gradient \mathbf{F} expresses the rotation and stretch of $d\mathbf{x}$. The deformation gradient is

$${}^t_0\mathbf{F} = {}^t\mathbf{x}_0 \overleftarrow{\nabla} = \frac{\partial {}^t\mathbf{x}}{\partial {}^0\mathbf{x}} \quad (11.2)$$

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

$$\mathbf{F} = \mathbf{R} \cdot \mathbf{U} = \mathbf{V} \cdot \mathbf{R} \quad (11.3)$$

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

$${}^tV = J {}^0V \quad ; \quad J = \det {}^t_0\mathbf{F} \quad (11.4)$$

The velocity of a point is denoted as

$$\dot{\mathbf{u}} = {}^t\dot{\mathbf{x}} \quad (11.5)$$

The velocity gradient is

$$\mathbf{L} = \dot{\mathbf{u}} \overleftarrow{\nabla} = \frac{\partial {}^t \dot{\mathbf{u}}}{\partial {}^t \mathbf{x}} \quad (11.6)$$

The spin $\mathbf{\Omega}$ and rate of deformation \mathbf{D} are defined in terms of the local velocity gradient

$$\mathbf{D} = \frac{1}{2} (\mathbf{L} + \mathbf{L}^T) \quad \text{and} \quad \mathbf{\Omega} = \frac{1}{2} (\mathbf{L} - \mathbf{L}^T) \quad (11.7)$$

The Green–Lagrange strain is defined as

$${}^t_0 \mathbf{E} = \frac{1}{2} ({}^t_0 \mathbf{F}^T \cdot {}^t_0 \mathbf{F} - \mathbf{I}) \quad (11.8)$$

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

$$\boldsymbol{\sigma} \mathbf{n} \, dA = d\mathbf{f} \quad (11.9)$$

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

$$\delta W = \int_V \boldsymbol{\sigma} \delta \boldsymbol{\varepsilon} \, dV \quad (11.10)$$

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

$${}^t_0 \mathbf{S} = \det {}^t_0 \mathbf{F} \cdot {}^t_0 \mathbf{F}^{-1} \cdot {}^t \boldsymbol{\sigma} \cdot {}^t_0 \mathbf{F}^{-T} \quad (11.11)$$

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

$$\overset{\nabla}{\boldsymbol{\sigma}} = \dot{\boldsymbol{\sigma}} + \boldsymbol{\sigma} \cdot \mathbf{\Omega} - \mathbf{\Omega} \cdot \boldsymbol{\sigma} \quad (11.12)$$

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.

11.1.1.2 Total Lagrange

In a Total Lagrange description, strain and stress measures are defined with reference to the undeformed geometry.

11.1.1.3 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.

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

$$\delta W = \delta W_1 + \delta W_2 + \delta W_3 = 0 \quad \forall \delta \mathbf{v} \quad (11.13)$$

$$\delta W_1 = \int_V \vec{\nabla} \delta \mathbf{v} : \boldsymbol{\sigma} \, dV \quad (11.14)$$

$$\delta W_2 = \int_V \rho \ddot{\mathbf{u}} \cdot \delta \mathbf{v} \, dV \quad (11.15)$$

$$\delta W_3 = \int_S \mathbf{t} \cdot \delta \mathbf{v} \, dS + \int_V \mathbf{f} \cdot \delta \mathbf{v} \, dV \quad (11.16)$$

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

$$\overset{\nabla}{\boldsymbol{\sigma}} = \dot{\boldsymbol{\sigma}} + \boldsymbol{\sigma} \cdot \boldsymbol{\Omega} - \boldsymbol{\Omega} \cdot \boldsymbol{\sigma} = \frac{\dot{\rho}}{\rho} \boldsymbol{\sigma} + \mathbb{L} : \mathbf{D} \quad (11.17)$$

with fourth order constitutive tensor \mathbb{L} and deformation rate \mathbf{D} . Using this rate type constitutive model we can derive δW_1 from

$$\delta W_1 = \int_0^t \delta \dot{W}_1 \, dt \quad (11.18)$$

After taking the time derivative of Eq. (11.14) and using Eq. (11.17) the required expression for $\delta \dot{W}_1$ reads, see Van den Boogaard [11] and Huétink [37]:

$$\delta \dot{W}_1 = \int_V \left(\delta \mathbf{D} : (\mathbb{L} - \mathbb{I} \cdot \boldsymbol{\sigma} - \boldsymbol{\sigma} \cdot \mathbb{I}) : \mathbf{D} + (\vec{\nabla} \delta \mathbf{v} \cdot \dot{\mathbf{u}} \vec{\nabla}) : \boldsymbol{\sigma} \right) \, dV \quad \forall \delta \mathbf{v} \quad (11.19)$$

with \mathbb{I} the fourth order unit tensor with components $\mathbb{I}_{ijkl} = \delta_{ik} \delta_{jl}$.

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.

The displacement field is discretized in space by means of interpolation functions Ψ

$$\mathbf{u} = \sum_n \Psi_n \mathbf{u}_n \quad (11.20)$$

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

$$\mathbf{D} = \sum_n \mathbf{B}_n \dot{\mathbf{u}}_n \quad \mathbf{B}_n = \frac{1}{2} \left((\mathbb{I} + \mathbb{T}) : (\vec{\nabla} \Psi_n \mathbf{I}) \right) \quad (11.21)$$

where \mathbb{T} is the transpose operator with components $\mathbb{T}_{ijkl} = \delta_{jk} \delta_{il}$, and where \mathbf{v} is the velocity.

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

$$\mathbf{M}^t \ddot{\mathbf{u}} + \mathbf{C}^t \dot{\mathbf{u}} + {}^t \mathbf{f}_{\text{int}} = {}^t \mathbf{f}_{\text{ext}} \quad (11.22)$$

where \mathbf{M} is the mass matrix, derived from Eq. (11.15); \mathbf{C} is the damping matrix; \mathbf{f}_{ext} is the external force vector, derived from Eq. (11.16). 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. (11.14).

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.

$$(a\mathbf{M} + b\mathbf{C} + {}^{t+\Delta t}\mathbf{K}) \cdot (\Delta \mathbf{u}_{(i)} - \Delta \mathbf{u}_{(i-1)}) = {}^{t+\Delta t}\mathbf{f}_{\text{ext}} - {}^{t+\Delta t}\mathbf{f}_{\text{int.}(i-1)} \quad (11.23)$$

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 $t + \Delta t$.

$${}^{t+\Delta t}\mathbf{f}_{\text{int.}(i-1)} = \sum_m {}^{t+\Delta t} \int_V \mathbf{B}_m^* : \boldsymbol{\sigma} \, dV \quad (11.24)$$

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 .

$$\mathbf{B}_m^* = \frac{1}{2} \left((\mathbf{I} \Psi_m \overleftarrow{\nabla}) : (\mathbb{I} + \mathbb{T}) \right) \quad (11.25)$$

The stress update is performed via integration of the Jaumann rate, assuming a constant velocity inside the step:

$$\dot{\mathbf{u}} = \frac{\Delta \mathbf{u}}{\Delta t} \quad (11.26)$$

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:

$${}^{t+\Delta t}_{t+\Delta t} \boldsymbol{\sigma} = {}^t\mathbf{R} \cdot \left({}^t\mathbf{R} \cdot {}^t\boldsymbol{\sigma} \cdot {}^t\mathbf{R}^T + \int_t^{t+\Delta t} \dot{\boldsymbol{\sigma}} \, dt \right) \cdot {}^t\mathbf{R}^T \quad (11.27)$$

$$\int_t^{t+\Delta t} \dot{\boldsymbol{\sigma}} \, dt = \mathbb{L} : {}_{t+\alpha\Delta t}\mathbf{D} \Delta t \quad (11.28)$$

The transformation tensor

$${}^t\mathbf{R} = {}^{t+\alpha\Delta t}(\mathbf{F} \cdot \mathbf{U}^{-1}) \quad (11.29)$$

represents the rotation of the reference coordinate frame at t to the reference coordinate frame at $t + \alpha \Delta t$.

The stiffness matrix consists of different contributions:

$$\mathbf{K} = \sum_n \sum_m (\mathbf{K}_{mn} + \mathbf{K}_{mnS}) \quad (11.30)$$

with \mathbf{K}_{mn} from Eq. (11.19):

$$\mathbf{K}_{mn} = \int_V \mathbf{B}_m^* : (\mathbb{L} - \mathbb{I} \cdot \boldsymbol{\sigma} - \boldsymbol{\sigma} \cdot \mathbb{I}) : \mathbf{B}_n + \mathbf{I} \left(\overrightarrow{\nabla} \Psi_m \cdot \boldsymbol{\sigma} \cdot \overrightarrow{\nabla} \Psi_n \right) dV \quad (11.31)$$

where \mathbf{B}_n comes from Eq. (11.21) and \mathbf{B}_m^* from Eq. (11.25). The contribution from a nonconservative traction load comes from Eq. (11.16):

$$\mathbf{K}_{mnS} = - \int_S \Psi_m \mathbf{t} \overrightarrow{\nabla}_S \Psi_n \, dS \quad (11.32)$$

11.1.2 Non-conservative Loads

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.

11.1.3 Specific Elements

DIANA offers specific elements for contact analysis, which is a special kind of geometric nonlinearity. Therefore, contact elements in the finite element model require a nonlinear analysis. The contact algorithm prevents penetration of one material body into another material body. The no-penetration conditions gives rise to an extra set of conditions on top of the equilibrium equations. These extra contact conditions are satisfied by using a Constrained Minimization technique, see Van Gijzen [30].

Note that the required Constrained Minimization technique to solve the contact conditions is only available when a direct linear equation solver is used.

The contact leads to a contact force F_n , in the direction normal to the plane of contact [Fig. 11.3]. This normal force in turn leads to a tangential friction force F_t , by assuming

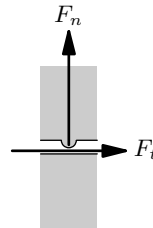


Figure 11.3: Coulomb friction

the Coulomb friction law to be valid

$$F_t = \mu F_n \quad (11.33)$$

where μ is the Coulomb friction coefficient. If the internal body stresses are less than this force μF_n , then the two bodies stick at the contacting point. If the internal body stresses reach the force μF_n , then slipping starts with a frictional force $F_t = \mu F_n$.

Contact elements are special interface elements to model zones of possible contact. A contact zone must be modeled by two types of contact elements: A surface containing contacter elements and a surface containing target elements. The user can define the properties of the contact elements. The only input for the contact elements themselves is the connectivity. Geometry or loading data is not applicable for these elements.

If the contact area can be split in several contact zones which do not have mutual interaction, i.e. the nodes of the contacter elements in a specific zone will not make contact with target elements in the other zones, the contact elements of separated zones can be put in different contact sets to increase the performance of the contact evaluation. In this case the contact evaluation is performed per individual contact set.

In a model for contact analysis you must classify a contact element as a ‘contacter’ or as a ‘target’. DIANA recognizes and effectuates contact between contact elements if a contacter touches the ‘outside’ of a target. Therefore, you must pay special attention to the axes direction which defines the outside of the target element. For line contact elements, i.e., two-dimensional analysis, the element y axis defines the ‘outside’ of the target element [Fig. 11.4]. For triangular and quadrilateral contact elements, i.e., three-dimensional analysis, the element z axis defines the ‘outside’ of the element [Fig. 11.5].

Nodes of targets and nodes of contacters may not be supported in the same direction.

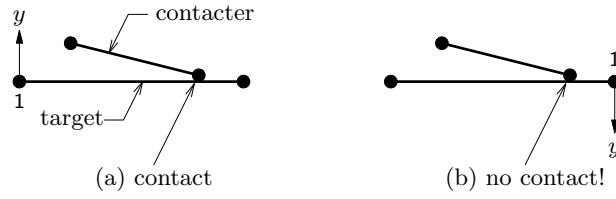


Figure 11.4: Two-dimensional contact algorithm

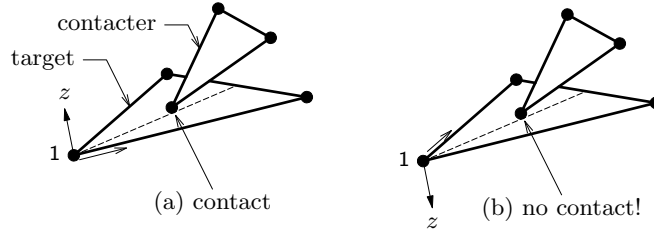


Figure 11.5: Three-dimensional contact algorithm

The basic variables in the nodes of the contact shell elements are the translations u_X , u_Y and u_Z in the global XYZ directions.

$$\mathbf{u}_e^{(2D)} = \begin{Bmatrix} u_X \\ u_Y \end{Bmatrix} \quad ; \quad \mathbf{u}_e^{(3D)} = \begin{Bmatrix} u_X \\ u_Y \\ u_Z \end{Bmatrix} \quad (11.34)$$

Primary stresses for the contact elements are the Cauchy stresses in the local element xyz axes.

$$\boldsymbol{\sigma}^{(2D)} = \begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \end{Bmatrix} \quad ; \quad \boldsymbol{\sigma}^{(3D)} = \begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \end{Bmatrix} \quad (11.35)$$

From the Cauchy stresses DIANA derives the distributed forces.

$$\mathbf{f}^{(2D)} = \begin{Bmatrix} q_{xx} \\ n_{yy} \end{Bmatrix} \quad ; \quad \mathbf{f}^{(3D)} = \begin{Bmatrix} q_{xx} \\ q_{yy} \\ n_{zz} \end{Bmatrix} \quad (11.36)$$

Figure 11.6 shows the orientation of stresses for two-dimensional and three-dimensional contact elements.

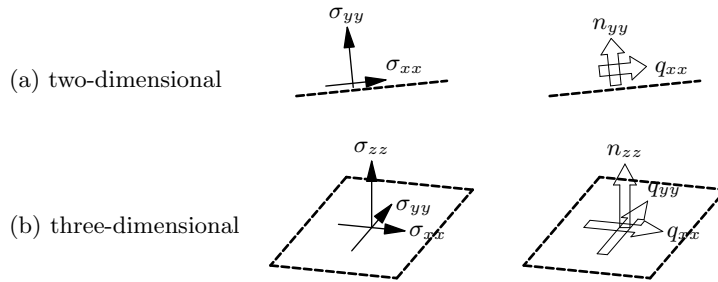


Figure 11.6: Stresses for contact elements

In contact analysis, DIANA checks if contact occurs between the contact elements. Via the material properties you must specify the two types of contact elements: the contactors and the targets. Generally speaking, the nodes of a contactor element cannot penetrate a target element [Fig. 11.7a]. Moreover you may specify some ‘material’ properties for target elements which describe the contact behaviour, for instance a check criterion for penetration [Fig. 11.7b].

The contact status is available as output result for a contact analysis.

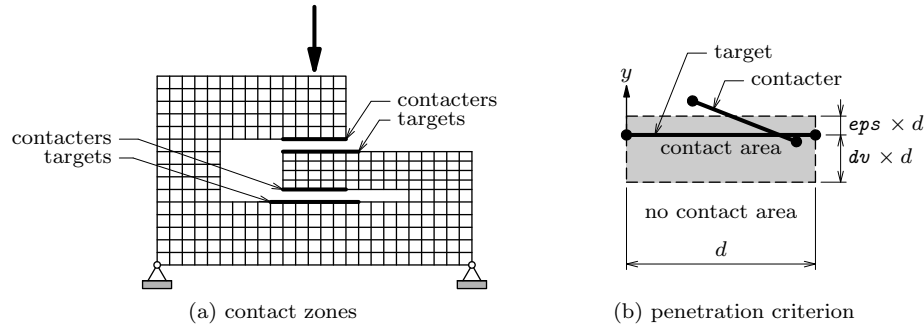


Figure 11.7: Contact behaviour

11.1.3.1 L4CT – line, 2 nodes, 2-D

The L4CT element [Fig. 11.8] is a two-node line element (linear interpolation) for two-

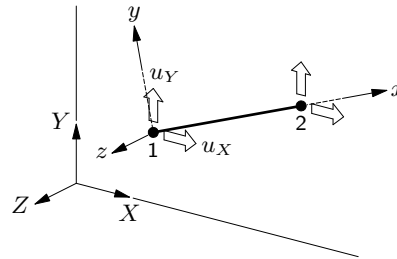


Figure 11.8: L4CT

dimensional contact analysis in the XY plane. DIANA derives the element axes from the node locations in the deformed model: the x axis from the first to the second node, the z axis is always parallel to the global Z axis and the y axis $\perp zx$ according to the right-hand-rule. The variables for the L4CT element are the displacements u in the global XY directions.

11.1.3.2 CL6CT – line, 3 nodes, 2-D

The CL6CT element [Fig. 11.9] is a three-node line element (quadratic interpolation) for

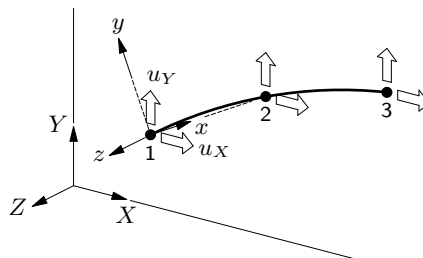


Figure 11.9: CL6CT

two-dimensional contact analysis in the XY plane. DIANA derives the element axes from the node locations in the deformed model: the x axis from the first to the second node, the z axis is always parallel to the global Z axis and the y axis $\perp zx$ according to the right-hand-rule. The variables for the CL6CT element are the displacements u in the global XY directions.

11.1.3.3 T9CT – triangle, 3 nodes, 3-D

The T9CT element [Fig. 11.10] is a three-node triangular element (linear interpolation) for three-dimensional contact analysis. DIANA derives the element axes from the node

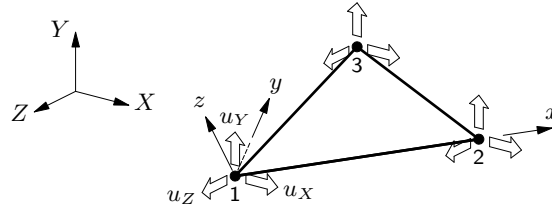


Figure 11.10: T9CT

locations in the deformed model: the x axis from the first to the second node, the z axis perpendicular to the element plane and the y axis $\perp zx$ according to the right-hand-rule. The variables for the T9CT element are the displacements u in the global XYZ directions.

11.1.3.4 Q12CT – quadrilateral, 4 nodes, 3-D

The Q12CT element [Fig. 11.11] is a four-node quadrilateral element (linear interpolation)

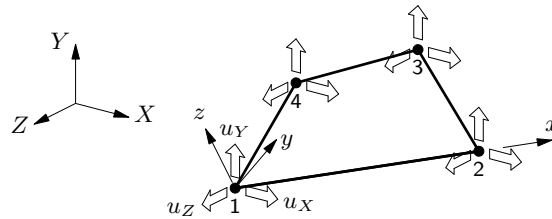


Figure 11.11: Q12CT

for three-dimensional contact analysis. DIANA derives the element axes from the node locations in the deformed model: the x axis from the first to the second node, the z axis approximately perpendicular to the element plane and the y axis $\perp zx$ according to the right-hand-rule. The variables for the Q12CT element are the displacements u in the global XYZ directions.

11.1.3.5 CT18C – triangle, 6 nodes, 3-D

The CT18C element [Fig. 11.12] is a six-node triangular element (quadratic interpolation)

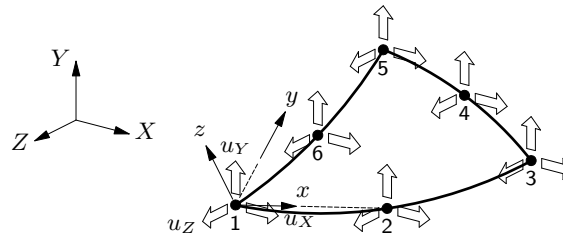


Figure 11.12: CT18C

for three-dimensional contact analysis. DIANA derives the element axes from the node locations in the deformed model: the x axis from the first to the second node, the z axis approximately perpendicular to the element plane and the y axis $\perp zx$ according to the right-hand-rule. The variables for the CT18C element are the displacements u in the global XYZ directions.

11.1.3.6 CQ24C – quadrilateral, 8 nodes, 3-D

The CQ24C element [Fig. 11.13] is an eight-node quadrilateral element (quadratic interpolation) for three-dimensional contact analysis. DIANA derives the element axes from the node locations in the deformed model: the x axis from the first to the second node, the z axis approximately perpendicular to the element plane and the y axis $\perp zx$ according to the right-hand-rule. The variables for the CQ24C element are the displacements u in the global XYZ directions.

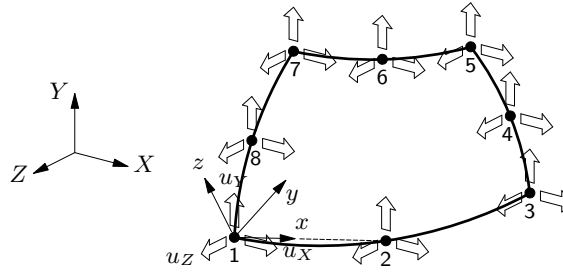


Figure 11.13: CQ24C

the right-hand-rule. The variables for the CQ24C element are the displacements u in the global XYZ directions.

11.2 Physical Nonlinear Analysis

11.2.1 Nonlinear Elasticity

DIANA offers some material models for nonlinear elasticity which behave hypoelastic, i.e., the behaviour in loading and unloading is similar. For granular materials, the elastic behaviour is quite different from the elastic behaviour of other structural materials because in general, the stiffness increases with the stress level [Fig. 11.14].

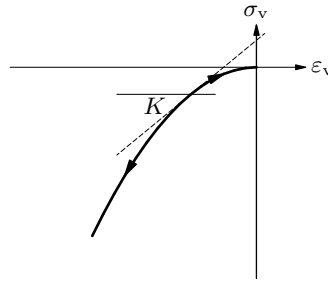


Figure 11.14: Hypoelastic behaviour of granular materials

DIANA supports three models, specifically suited for granular materials: the *Grains model* [§ 11.2.1.1 p. 225], the *Boyce model* [§ 11.2.1.2 p. 225], and the *Jardine model* [§ 11.2.1.3 p. 225]. The formulation is given in stress and strain invariants which are defined as follows. The hydrostatic pressure, or mean stress p , is defined by

$$p = \frac{1}{3} (\sigma_{xx} + \sigma_{yy} + \sigma_{zz}) = \frac{1}{3} I_1 \quad (11.37)$$

The effective deviatoric stress q is defined as

$$q = \sqrt{\frac{1}{2} (\sigma_{xx} - \sigma_{yy})^2 + \frac{1}{2} (\sigma_{yy} - \sigma_{zz})^2 + \frac{1}{2} (\sigma_{zz} - \sigma_{xx})^2 + 3 (\sigma_{xy}^2 + \sigma_{yz}^2 + \sigma_{zx}^2)} \quad (11.38)$$

The related strain invariants (volumetric and shear) are defined as

$$\varepsilon_v = (\varepsilon_{xx} + \varepsilon_{yy} + \varepsilon_{zz}) \quad (11.39)$$

and

$$\varepsilon_s = \sqrt{\frac{2}{9} (\varepsilon_{xx} - \varepsilon_{yy})^2 + \frac{2}{9} (\varepsilon_{yy} - \varepsilon_{zz})^2 + \frac{2}{9} (\varepsilon_{zz} - \varepsilon_{xx})^2 + \frac{1}{3} \gamma_{xy}^2} \quad (11.40)$$

The bulk modulus (compression modulus) K and the shear modulus G are defined as

$$K = \frac{p}{\varepsilon_v} \quad G = \frac{q}{3\varepsilon_s} \quad (11.41)$$

Finally, the stiffness matrix \mathbf{D} is given by

$$\mathbf{D} = \begin{bmatrix} K + \frac{4}{3}G & K - \frac{2}{3}G & K - \frac{2}{3}G & 0 \\ K - \frac{2}{3}G & K + \frac{4}{3}G & K - \frac{2}{3}G & 0 \\ K - \frac{2}{3}G & K - \frac{2}{3}G & K + \frac{4}{3}G & 0 \\ 0 & 0 & 0 & G \end{bmatrix} \quad (11.42)$$

11.2.1.1 Grains Model

The grains (or standard) model, proposed by Allaart [2], gives the hydrostatic pressure and the effective deviatoric stress as nonlinear functions of the isotropic and deviatoric strain invariants:

$$\begin{cases} p &= \frac{1}{2} \sqrt[n]{K_1 \varepsilon_v + \sqrt{(K_1 \varepsilon_v)^2 + \beta (6G_1 \varepsilon_s)^2}} \\ q &= 3G_1 p^{1-n} \varepsilon_s \end{cases} \quad (11.43)$$

with the factor β given by

$$\beta = \frac{K_1 (1-n)}{6G_1} \quad (11.44)$$

The reference values of the bulk and shear moduli, K_1 and G_1 respectively, as well as the value of the power n are determined by experiments. Given the current strain state, the stress invariants p and q are given by Eq. (11.43) and the moduli K and G are determined. The (secant) stiffness matrix is then easily calculated with Eq. (11.42).

11.2.1.2 Boyce Model

The model proposed by Boyce [13] describes the nonlinear elastic behaviour of granular materials under repeated loading. The relations between the isotropic and shear strain invariants and the hydrostatic and effective deviatoric stresses read

$$\begin{cases} \varepsilon_v &= \frac{1}{K_1} \left(\frac{p}{p_0} \right)^{n-1} \left(1 - \beta \frac{q^2}{p^2} \right) p \\ \varepsilon_s &= \frac{1}{3G_1} \left(\frac{p}{p_0} \right)^{n-1} q \end{cases} \quad (11.45)$$

with the reference pressure p_0 with a value equal to -1 kPa and the reference values of the bulk and shear moduli, K_1 and G_1 respectively. These reference values as well as the value of the power n are determined by experiments, with the factor β given by Eq. (11.44).

11.2.1.3 Jardine Model

The model proposed by Jardine et al. [39] is based on the relation between the secant Young's modulus and the axial strain, measured in an undrained triaxial compression test, i.e., a test in which a cylindrical specimen is loaded by an increasing axial load while the radial stress is kept constant, see for instance Jardine et al. [40]. The secant Young's modulus is defined as:

$$E_u = \frac{\sigma_a - \sigma_{a;0}}{\varepsilon_a} \quad (11.46)$$

In which E_u is the secant Young's modulus, ε_a is the axial strain, σ_a is the axial stress, and $\sigma_{a;0}$ is the initial axial stress (for which $\varepsilon_a = 0$). In the triaxial test axial stresses are measured. With Eq. (11.46) E_u follows directly from these measurements.

The basic assumption of the Jardine model is that the relation between the secant Young's modulus and the axial strain over the elastic range can be expressed in the form:

$$E_u = G + (F - G) \cos\left(\alpha \left(\log \frac{\varepsilon_a}{C}\right)^\gamma\right) \quad (11.47)$$

This relation can be visualized as a stretched periodic function in a diagram of E_u vs. ε_a [Fig. 11.15]. The relation contains five independent parameters (C , D , E , F , and G)

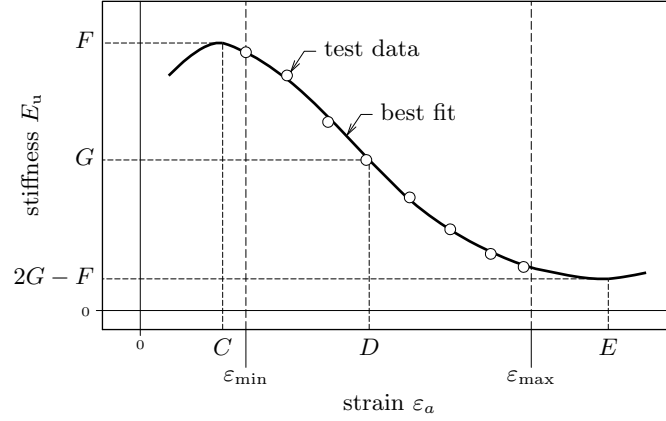


Figure 11.15: Jardine parameters in stiffness-log(strain) diagram

which are considered to be material constants. These parameters can be read directly from the stiffness-strain diagram. Parameters D and E are intermediate and can be applied to derive α and γ using the following formulae, assuming that the angular part in Eq. (11.47) has to be equal to $\frac{1}{2}\pi$ at the medium stiffness and equal to π at the minimum stiffness:

$$\gamma = \frac{\log 2}{\log\left(\frac{\log(E/C)}{\log(D/C)}\right)} \quad ; \quad \alpha = \frac{\frac{1}{2}\pi}{\left(\log(D/C)\right)^\gamma} \quad (11.48)$$

F and G are not necessarily equal to the maximum and medium measured stiffness. F is the maximum of the curve that fits the test data best; this may be a projected maximum outside the range of data. The projected minimum stiffness may even have a negative value, as long as E_u is positive in the specified range over which Eq. (11.47) is valid. This range is bounded by ε_{\min} and ε_{\max} . Beyond these boundaries the tangent Young's modulus is assumed to be constant. For strain below ε_{\min} this implies a constant secant Young's modulus, which is not the case for strain beyond ε_{\max} . A practical value for ε_{\min} is the smallest strain for which test data is available. For ε_{\max} care is required to ensure compatibility with the onset of plastic yield. With a high value for ε_{\max} , a negative elastic tangential stiffness may occur, causing numerical instability, even when ε_{\max} is chosen less than E .

In the DIANA implementation the Jardine model is generalized by substitution of the deviatoric strain invariant:

$$\varepsilon_{eq} = \sqrt{\frac{2}{3} \left((\varepsilon_1 - \varepsilon_2)^2 + (\varepsilon_2 - \varepsilon_3)^2 + (\varepsilon_3 - \varepsilon_1)^2 \right)} \quad (11.49)$$

for $\varepsilon_a \sqrt{3}$, with ε_1 , ε_2 , and ε_3 defined as the principal elastic strains. The origin of the factor $\sqrt{3}$ can be traced by substituting the stress state of the undrained triaxial test ($\varepsilon_1 = \varepsilon_a$, $\varepsilon_2 = \varepsilon_3 = -\frac{1}{2}\varepsilon_a$) into Eq. (11.49). From Eq. (11.47) an expression for the tangent Young's modulus E_{ut} can be derived. After substitution of ε_{eq} these expressions can be written as:

$$E_u = f_1(\varepsilon_{eq}) = G + (F - G) \cos(\alpha I^\gamma) \quad (11.50)$$

$$E_{ut} = f_2(\varepsilon_{eq}) = G + (F - G) \cos(\alpha I^\gamma) - \frac{(F - G) \alpha \gamma I^{\gamma-1}}{2.303} \sin(\alpha I^\gamma) \quad (11.51)$$

with

$$I = \log \left(\frac{\varepsilon_{eq}}{\sqrt{3} C} \right) \quad (11.52)$$

The equivalent elastic strains corresponding to the boundaries $\varepsilon_{eq;min}$ and $\varepsilon_{eq;max}$ are:

$$\varepsilon_{eq;min} = \varepsilon_{min} \sqrt{3} \quad \text{and} \quad \varepsilon_{eq;max} = \varepsilon_{max} \sqrt{3} \quad (11.53)$$

With the assumption of a constant tangent Young's modulus outside the boundaries, the general expression for E_u becomes:

$$E_u = \begin{cases} f_1(\varepsilon_{eq;min}) & \text{if } \varepsilon_{eq} \leq \varepsilon_{eq;min} \\ f_1(\varepsilon_{eq}) & \text{if } \varepsilon_{eq;min} < \varepsilon_{eq} \leq \varepsilon_{eq;max} \\ f_2(\varepsilon_{eq;max}) + \left(f_1(\varepsilon_{eq;max}) - f_2(\varepsilon_{eq;max}) \right) \frac{\varepsilon_{eq;max}}{\varepsilon_{eq}} & \text{if } \varepsilon_{eq} > \varepsilon_{eq;max} \end{cases} \quad (11.54)$$

For each iteration, in each integration point, Eq. (11.54) is used to calculate the updated stresses. To be able to execute analyses with initial stresses unequal to zero, the stresses have to be updated by addition of stress increments. The stress increment is found by calculating the averaged tangent stiffness over $\Delta \varepsilon_{eq}$, which can be done exactly with the given definition of the secant stiffness Eq. (11.46). We will explain this first for the triaxial test and then generalized.

The relation between the tangent and secant Young's modulus for the triaxial test is:

$$E_{ut} = \frac{d\sigma_a}{d\varepsilon_a} = \frac{d(E_u \varepsilon_a)}{d\varepsilon_a} \quad (11.55)$$

With the given relation between axial strain and the secant Young's modulus, the increment in axial stress $\Delta \sigma_a$, caused by a prescribed increment in axial strain $\Delta \varepsilon_a$, is:

$$\Delta \sigma_a = \int_{\varepsilon_a^0}^{\varepsilon_a^0 + \Delta \varepsilon_a} E_{ut} d\varepsilon_a = \left[E_u \varepsilon_a \right]_{\varepsilon_a^0}^{\varepsilon_a^0 + \Delta \varepsilon_a} = E_u (\varepsilon_a^0 + \Delta \varepsilon_a) - E_u^0 \varepsilon_a^0 \quad (11.56)$$

This can be written as:

$$\Delta \sigma_a = \frac{E_u (\varepsilon_a^0 + \Delta \varepsilon_a) - E_u^0 \varepsilon_a^0}{\Delta \varepsilon_a} = \bar{E}_{ut} \Delta \varepsilon_a \quad (11.57)$$

This linear expression relating stress increment to strain increment is suitable for generalization and usage in DIANA. The averaged Young's modulus \bar{E}_{ut} is found for the generalized case by substitution of the deviatoric strain invariant:

$$\bar{E}_{ut} = \frac{E_u \varepsilon_{eq} - E_u^0 \varepsilon_{eq}^0}{\varepsilon_{eq} - \varepsilon_{eq}^0} \quad (11.58)$$

In which ε_{eq} is the updated equivalent strain and E_u the secant stiffness for this strain, obtained with Eq. (11.54). All variables in the right hand side of Eq. (11.58) are known in DIANA before the stress is updated. The average Young's modulus is substituted into the material stiffness matrix \mathbf{D} , which in the general three-dimensional case results in:

$$\mathbf{D} = \frac{\bar{E}_{ut}}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} \end{bmatrix} \quad (11.59)$$

In which, for the Jardine model, Poisson's ratio ν is set to 0.49 by default. The stress increment in the generalized case is calculated with:

$$\Delta \boldsymbol{\sigma} = \mathbf{D} \Delta \boldsymbol{\varepsilon} \quad (11.60)$$

11.2.2 Hyperelasticity

To model rubbery material, DIANA uses the concept of *hyperelasticity*. Instead of the usual stress-strain-like descriptions, a hyperelastic material description is based on a strain energy density function that relates stresses and strains in an implicit manner. The strains are fully recoverable even if they are large.

The hyperelasticity models are based on the Total Lagrange description of large deformations. In this description the Green-Lagrange strain \mathbf{E} is defined as

$$\mathbf{E} = \frac{1}{2} (\mathbf{F}^T \mathbf{F} - \mathbf{I}) \quad (11.61)$$

with \mathbf{F} the deformation gradient. A variation of this strain measure is the *Right Cauchy-Green* stretch tensor \mathbf{C} .

$$\mathbf{C} = \mathbf{F}^T \mathbf{F} \quad (11.62)$$

From Eq. (11.62) it follows that \mathbf{C} is symmetric. All models available in DIANA are based on the invariants of the Right Cauchy-Green stretch or the principal stretches of \mathbf{C} .

The stresses can be calculated, based on the derivatives of the strain energy density function W to the strains. Because a total Lagrange approach is used, the 2nd Piola-Kirchhoff stress is related to the Green-Lagrange strain via the virtual energy

$$\delta W = \int_{V_0} \mathbf{S} \delta \mathbf{E} \, dV_0 \quad (11.63)$$

By combination of Eq. (11.62), Eq. (11.61), and Eq. (11.63) the stresses are calculated from

$$\mathbf{S} = \frac{\partial W}{\partial \mathbf{E}} = 2 \frac{\partial W}{\partial \mathbf{C}} \quad (11.64)$$

The invariants of \mathbf{C} are

$$I_1 = \text{tr } \mathbf{C} = C_{11} + C_{22} + C_{33} \quad (11.65)$$

$$I_2 = C_{11}C_{22} + C_{22}C_{33} + C_{33}C_{11} - C_{12}C_{21} - C_{23}C_{32} - C_{31}C_{13} \quad (11.66)$$

$$\begin{aligned} I_3 &= \det(\mathbf{C}) \\ &= C_{11}C_{22}C_{33} + 2C_{12}C_{23}C_{31} - C_{11}C_{23}^2 - C_{22}C_{31}^2 - C_{33}C_{12}^2 \end{aligned} \quad (11.67)$$

The eigenvalues of \mathbf{C} are the squares of the length ratios in principal directions λ_1 , λ_2 and λ_3 . An initially isotropic hyperelastic material can now be described by a strain energy density function of the invariants $W(I_1, I_2, I_3)$ or of the principal stretches $W(\lambda_1, \lambda_2, \lambda_3)$. Because the behaviour of rubbery materials is usually considered incompressible or nearly incompressible, the function W is split into a *deviatoric* part W_d and a *hydrostatic* part W_h , see for instance Van Den Bogert [10]. The deviatoric part describes a constant volume deformation, the hydrostatic part describes a uniform compression or expansion. To fulfill the separation of the deviatoric and hydrostatic part, we introduce a modified set of deformation measures:

$$J_1 = I_1 I_3^{-\frac{1}{3}} \quad J_2 = I_2 I_3^{-\frac{1}{3}} \quad J = \sqrt{I_3} \quad (11.68)$$

or alternatively for the principal stretches:

$$\tilde{\lambda}_i = \lambda_i I_3^{-\frac{1}{6}} \quad J = \lambda_1 \lambda_2 \lambda_3 \quad (11.69)$$

Note that for incompressible materials $I_3 = 1$ and the new measures do not differ from the originals. The strain energy density functions for the deviatoric part can now be written as $W_d(J_1, J_2)$ or $W_d(\tilde{\lambda}_1, \tilde{\lambda}_2, \tilde{\lambda}_3)$ and the hydrostatic part as $W_h(J)$.

11.2.2.1 Deviatoric Energy Function

A possible way to construct any type of strain energy density function for an incompressible, initially isotropic material is the *Rivlin formulation*, see for instance Treloar [78]:

$$W_d = \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} K_{ij} (I_1 - 3)^i (I_2 - 3)^j \quad (11.70)$$

This formulation can also be used for nearly incompressible materials if the invariants I_1 and I_2 are substituted by the modified invariants J_1 and J_2 .

Mooney–Rivlin. A well known model, derived from the Rivlin formulation, is the Mooney–Rivlin model [51, 62]. Here only the first order terms in Eq. (11.70) are maintained:

$$W_d = K_1 (J_1 - 3) + K_2 (J_2 - 3) \quad (11.71)$$

A subset of the Mooney–Rivlin model is the *Neo-Hookean model*, in which case $K_2 = 0$. The Neo-Hookean model can be derived for polymer-like materials from statistical thermodynamics.

Besseling. An alternative way to adapt the material model to experimental data is to use a non-integer power α instead of the Rivlin formulation. This model was proposed by Besseling [7]:

$$W_d = K_1 (J_1 - 3)^\alpha + K_2 (J_2 - 3) \quad (11.72)$$

Because of the strict separation of the deviatoric and hydrostatic part, the original model was adapted by using the modified invariants. The model is capable of describing the ascending branch at high strains in a uniaxial test, that cannot be matched by the Mooney–Rivlin model.

11.2.2.2 Hydrostatic Energy Function

The hydrostatic part of the strain energy density function, determines the compressibility of the material. Usually this *dilatation* behaviour is much more stiff than the deviatoric behaviour. In an analysis, the calculated stresses are a summation of the contribution due to the deviatoric and the hydrostatic part of the strain energy density function.

Linear compressibility. For nearly incompressible rubbery materials a linear relation between volume changes and hydrostatic stresses is valid for pressures up to hundreds of atmospheres, see for instance Van Den Bogert [10]. A linear hydrostatic stress–strain relation is a result of a quadratic strain energy density function

$$W_h = \frac{\kappa}{2} (J - 1)^2 \quad (11.73)$$

Nonlinear compressibility. For very high compressive stresses in rubber, or for other types of material, nonlinear hydrostatic strain energy density functions have been developed. The two models, implemented in DIANA are presented below.

$$\text{Simo–Taylor [73]: } W_h = \frac{\kappa}{2} \left((J - 1)^2 + (\ln J)^2 \right) \quad (11.74)$$

$$\text{Murnaghan [52]: } W_h = \frac{\kappa}{\beta} \left(\frac{1}{\beta - 1} J^{-\beta} + 1 \right) J \quad (11.74a)$$

11.2.3 Plasticity

Comparing elastic and plastic material behaviour, a fundamental observation is that in elastic behaviour no permanent deformations occur in the structure, whereas in plastic behaviour permanent, or irreversible deformations can be observed. Although the phenomenon of these irreversible deformations can be related to processes inside the material

like crystal-slip in metals, in a broader context the mathematical formulation of plasticity can be applied to all materials showing irreversible deformations, like soil and concrete. In the context of small strains, the basic assumption of an additive strain decomposition into an elastic part and an irreversible, or plastic part will be made:

$$\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}^e + \boldsymbol{\varepsilon}^p \quad (11.75)$$

Using the usual approach of the flow theory of plasticity to describe the elastoplastic material behaviour, we can not only model the total stress $\boldsymbol{\sigma}$ at time t as a function of the total strain $\boldsymbol{\varepsilon}$ at time t , but also as a function of the stress and strain history.

Assumptions. The stress and strain history of the material are usually taken into account implicitly by introducing an internal parameter, say κ , which is governed by a specific evolution law. The elastoplastic material behaviour can then be described with the following assumptions:

- The *elastic stress-strain relation*, which specifies the relation between the total stress and the elastic strain. Without loss of generality we can assume that this relation is given by

$$\boldsymbol{\sigma} = \mathbf{D}\boldsymbol{\varepsilon}^e \quad (11.76)$$

with \mathbf{D} the material stiffness matrix.

- The *yield condition*, which specifies the state of stress at which the plastic flow is initiated. This yield condition can be written as a function of the stress vector and the internal state parameter

$$f(\boldsymbol{\sigma}, \kappa) = 0 \quad (11.77)$$

The value of the yield function is less than zero, the state is assumed to be elastic and no plastic flow will occur. A state at which the yield function becomes greater than zero, is not admissible for rate-independent plasticity.

- The *flow rule*, which specifies the inelastic, or plastic strain rate vector as a function of the state of stress. According to the flow theory of plasticity, assuming Koiter's rule [44], the plastic strain rate vector is given by

$$\dot{\boldsymbol{\varepsilon}}^p = \sum_{j=1}^n \dot{\lambda}_j \frac{\partial g_j}{\partial \boldsymbol{\sigma}} \quad (11.78)$$

with the n plastic potential functions g_j which can also be considered as a function of the stress vector and the internal state parameter, i.e., $g_j(\boldsymbol{\sigma}, \kappa)$. The plastic multipliers $\dot{\lambda}_j$ are restricted by the standard Kuhn-Tucker conditions

$$f \leq 0 \quad ; \quad \dot{\lambda}_j \geq 0 \quad ; \quad \dot{\lambda}_j f = 0 \quad (11.79)$$

These conditions are actually a reformulation of the admissible states at plastic flow, i.e., no plastic flow will occur ($\dot{\lambda}_j = 0$) if the yield function is less than zero.

- The *hardening hypothesis*, which specifies the evolution of the internal state parameter. In general the evolution is given as a function of the stress vector and the plastic strain rate vector, i.e., $\dot{\kappa} = h(\boldsymbol{\sigma}, \dot{\boldsymbol{\varepsilon}}^p)$.

Strain and stress rate. The additive decomposition of the strain rate vector is the starting point for the derivation of the (infinitesimal) stiffness relation. The stress rate vector is determined by the elastic part of the strain rate vector

$$\dot{\boldsymbol{\sigma}} = \mathbf{D} \{ \dot{\boldsymbol{\varepsilon}} - \dot{\boldsymbol{\varepsilon}}^p \} = \mathbf{D} \left\{ \dot{\boldsymbol{\varepsilon}} - \dot{\lambda} \frac{\partial g}{\partial \boldsymbol{\sigma}} \right\} \quad (11.80)$$

The consistency condition $\dot{f} = 0$, or fully

$$\dot{f} = \frac{\partial f}{\partial \boldsymbol{\sigma}} \dot{\boldsymbol{\sigma}} + \frac{\partial f}{\partial \kappa} \dot{\kappa} = 0 \quad (11.81)$$

gives the necessary expression for $\dot{\lambda}$

$$\dot{\lambda} = \frac{1}{E_p} \frac{\partial f^T}{\partial \boldsymbol{\sigma}} \dot{\boldsymbol{\sigma}} \quad (11.82)$$

with the plastic hardening modulus

$$E_p = -\frac{\partial f}{\partial \kappa} \frac{\partial \kappa}{\partial \lambda} \quad (11.83)$$

Substituting this expression in the stress rate equation and after applying the Sherman–Morrison formula, the continuum tangent matrix is obtained

$$\dot{\boldsymbol{\sigma}} = \left[\mathbf{D} - \frac{\mathbf{D} \frac{\partial g}{\partial \boldsymbol{\sigma}} \frac{\partial f^T}{\partial \boldsymbol{\sigma}} \mathbf{D}}{E_p + \frac{\partial f^T}{\partial \boldsymbol{\sigma}} \mathbf{D} \frac{\partial g}{\partial \boldsymbol{\sigma}}} \right] \dot{\boldsymbol{\varepsilon}} \quad (11.84)$$

Note that the tangent stiffness matrix becomes asymmetric if the plastic potential function is not equal to the yield function, i.e., $f \neq g$.

The evolution equations given above can be regarded as *strain driven* in the sense that the total strain vector, the inelastic strain vector and the internal state parameter are known at time t and that the incremental strain vector $\Delta \boldsymbol{\varepsilon}_{i+1}$ follows from the loading regime.

Euler Backward. The basic problem in computational elastoplasticity is that the elastoplastic constitutive equations must be updated in a consistent manner:

$$({}^t \boldsymbol{\sigma}, {}^t \boldsymbol{\varepsilon}, {}^t \kappa; \Delta \boldsymbol{\varepsilon}_{i+1}) \Rightarrow ({}^{t+\Delta t} \boldsymbol{\sigma}_{i+1}, {}^{t+\Delta t} \boldsymbol{\varepsilon}_{i+1}, {}^{t+\Delta t} \kappa_{i+1}) \quad (11.85)$$

By applying the fully implicit Euler Backward algorithm, this problem is transformed into a constrained optimization problem governed by discrete Kuhn–Tucker conditions as shown by Simo et al. [72]. It has been shown in various studies (Krieg & Krieg [45]; Schreyer et al. [71]; Ortiz & Popov [59]; Simo & Taylor [75]) that the implicit Euler Backward algorithm is stable and accurate for J_2 -plasticity. But even when the yield surface is highly distorted, the Euler Backward algorithm is unconditionally stable (Ortiz & Popov [59]) and accurate (De Borst & Feenstra [12]; Schellekens & De Borst [68]). Application of the Euler Backward algorithm results in a discrete set of equations:

$$\begin{aligned} {}^{t+\Delta t} \boldsymbol{\varepsilon}_{i+1} &= {}^t \boldsymbol{\varepsilon} + \Delta \boldsymbol{\varepsilon}_{i+1} \\ {}^{t+\Delta t} \boldsymbol{\sigma}_{i+1} &= \mathbf{D} \{ {}^{t+\Delta t} \boldsymbol{\varepsilon}_{i+1} - {}^{t+\Delta t} \boldsymbol{\varepsilon}_{i+1}^p \} \\ {}^{t+\Delta t} \boldsymbol{\varepsilon}_{i+1}^p &= {}^t \boldsymbol{\varepsilon}^p + \Delta \lambda \left. \frac{\partial g}{\partial \boldsymbol{\sigma}} \right|_{i+1} \\ {}^{t+\Delta t} \kappa_{i+1} &= {}^t \kappa + \Delta \lambda h(\boldsymbol{\sigma}, \dot{\boldsymbol{\varepsilon}}^p) \end{aligned} \quad (11.86)$$

Because the algorithm is considered within an elastic predictor–plastic corrector algorithm an elastic trial state is introduced as

$$\begin{aligned} \boldsymbol{\sigma}_E &= \mathbf{D} \{ {}^t \boldsymbol{\varepsilon}^e + \Delta \boldsymbol{\varepsilon}_{i+1} \} \\ \kappa_E &= {}^t \kappa \end{aligned} \quad (11.87)$$

which can be obtained by freezing inelastic flow during the time step.

Return-mapping. The consistent linearization of the nonlinear equations following from the discretization results in a tangent stiffness matrix which plays a crucial role in the performance and robustness of the iteration method [Vol. *Analysis Procedures*]. It has been emphasized by Simo & Taylor [74] in the classical paper about consistent tangent operators for mathematical plasticity that the crucial point is that the tangent stiffness matrix must be obtained by consistent linearization of the stress resulting from

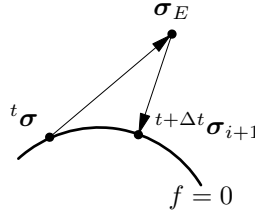


Figure 11.16: Return-mapping algorithm

the return-mapping algorithm [Fig. 11.16]. The consistent tangent stiffness matrix must be derived from the updated stress at the end of iteration $i + 1$ of Eq. (11.86):

$$\begin{aligned} d\sigma_{i+1} &= \mathbf{D} \left\{ d\varepsilon_{i+1} - d\lambda \frac{\partial g}{\partial \sigma} - \Delta\lambda \frac{\partial^2 g}{\partial \sigma^2} d\sigma_{i+1} \right\} \\ &= \mathbf{H} \left\{ d\varepsilon_{i+1} - d\lambda \frac{\partial g}{\partial \sigma} \right\} \end{aligned} \quad (11.88)$$

where the superscript $t + \Delta t$ has been dropped for convenience. The modified elastic stiffness matrix \mathbf{H} is defined by

$$\mathbf{H} = \left[\mathbf{I} + \Delta\lambda \mathbf{D} \frac{\partial^2 g}{\partial \sigma^2} \right]^{-1} \mathbf{D} \quad (11.89)$$

Differentiation of the consistency condition $df = 0$, at time $t + \Delta t$ yields the following expression

$$df_{i+1} = \frac{\partial f^T}{\partial \sigma} d\sigma_{i+1} + \frac{\partial f}{\partial \kappa} \frac{\partial \kappa}{\partial \lambda} d\lambda = 0 \quad (11.90)$$

which can be solved for $d\lambda$

$$d\lambda = \frac{1}{E_p} \frac{\partial f^T}{\partial \sigma} d\sigma_{i+1} \quad (11.91)$$

Substitution of Eq. (11.91) into Eq. (11.88) results after some algebraic manipulations into the consistent tangent stiffness relation

$$d\sigma_{i+1} = \left[\mathbf{H} - \frac{\mathbf{H} \frac{\partial g}{\partial \sigma} \frac{\partial f^T}{\partial \sigma} \mathbf{H}}{E_p + \frac{\partial f^T}{\partial \sigma} \mathbf{H} \frac{\partial g}{\partial \sigma}} \right] d\varepsilon_{i+1} \quad (11.92)$$

Note that the difference between the continuum tangent stiffness matrix given in Eq. (11.84) and the consistent tangent stiffness matrix of Eq. (11.92) is only determined by the modification of the elastic stiffness matrix \mathbf{D} with the matrix $\mathbf{I} + \Delta\lambda \mathbf{D} \frac{\partial^2 g}{\partial \sigma^2}$.

11.2.3.1 Tresca

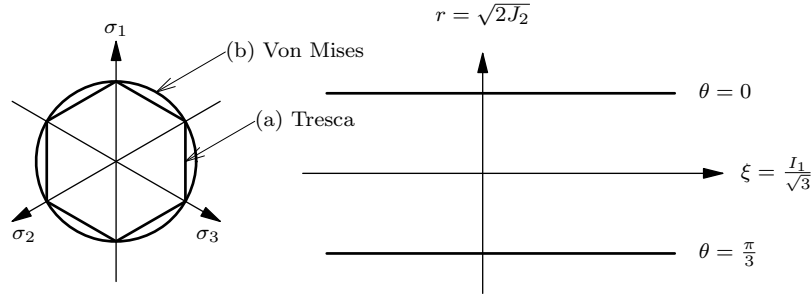
The yield condition of Tresca is a maximum shear stress condition which can be expressed in the principal stress space ($\sigma_1 \geq \sigma_2 \geq \sigma_3$) [Fig. 11.17a]:

$$f(\sigma, \kappa) = |\sigma_1 - \sigma_3| - \bar{\sigma}(\kappa) \quad (11.93)$$

with $\bar{\sigma}(\kappa)$ the uniaxial yield strength as a function of the internal state variable κ . The flow rule is in general given by the associated flow rule $g \equiv f$, which results for the plastic strain rate vector in the principal strain space

$$\dot{\varepsilon}^p = \dot{\lambda} \begin{Bmatrix} 1 \\ 0 \\ -1 \end{Bmatrix} \quad (11.94)$$

Hardening. The relation between the internal state variable κ and the plastic process is given by the hardening hypothesis. For the Tresca yield condition we consider two different hypotheses: strain hardening and work hardening.

Figure 11.17: Tresca and Von Mises yield condition (in π -and deviatoric plane)

Strain hardening. In the case of strain hardening the relation is given in the principal space by

$$\dot{\kappa} = \sqrt{\frac{2}{3} (\dot{\varepsilon}_1^p \dot{\varepsilon}_1^p + \dot{\varepsilon}_2^p \dot{\varepsilon}_2^p + \dot{\varepsilon}_3^p \dot{\varepsilon}_3^p)} \quad (11.95)$$

which can be elaborated to

$$\dot{\kappa} = \frac{2}{\sqrt{3}} \dot{\lambda} \quad (11.96)$$

Work hardening. For work hardening the basic assumption is

$$\dot{W}^p = \boldsymbol{\sigma}^T \dot{\boldsymbol{\varepsilon}}^p \equiv \bar{\sigma}(\kappa) \dot{\kappa} \quad (11.97)$$

which can be elaborated to

$$\dot{\kappa} = \dot{\lambda} \quad (11.98)$$

Relation $\bar{\sigma}$ - κ . For the Tresca yield condition the translation of uniaxial experimental data to the equivalent stress-internal state variable, the $\bar{\sigma}$ - κ relation, is independent upon the hardening hypothesis as shown in the example of Figure 11.18 on the following page.

Consider the uniaxial stress-strain diagram of Figure 11.18a. The plastic strain ε_1^p is assumed to be given by $\varepsilon_1 - \varepsilon_1^e$. Figure 11.18b shows the uniaxial stress-plastic strain diagram. For uniaxial stressing, $(\sigma_1, \sigma_2, \sigma_3) = (\sigma_1, 0, 0)$, plastic flow occurs at a vertex of the yield surface. Symmetry conditions dictate that the two possible yield directions contribute equally to the plastic strain rate vector

$$\dot{\boldsymbol{\varepsilon}}^p = \begin{Bmatrix} \dot{\varepsilon}_1^p \\ \dot{\varepsilon}_2^p \\ \dot{\varepsilon}_3^p \end{Bmatrix} = \dot{\lambda} \begin{Bmatrix} 1 \\ -\frac{1}{2} \\ -\frac{1}{2} \end{Bmatrix} \quad (11.99)$$

With the relation derived previously, we find for the relation between the uniaxial plastic strain and the internal state variable

$$\dot{\kappa} = \dot{\varepsilon}_1^p \quad (11.100)$$

for both a strain hardening and a work hardening hypothesis. The relation between the uniaxial stress and the equivalent stress is simply given by

$$\bar{\sigma} = \sigma_1 \quad (11.101)$$

Ambient influence. DIANA can handle the influence of temperature, concentration (e.g. moisture content in concrete) or maturity on the Tresca yield condition. For temperature dependency, the yield condition is given by

$$f(\boldsymbol{\sigma}, \kappa) = |\sigma_1 - \sigma_3| - f(T) \frac{\bar{\sigma}(\kappa)}{\bar{\sigma}(0)} \quad (11.102)$$

with $f(T)$ the temperature dependent tensile strength.

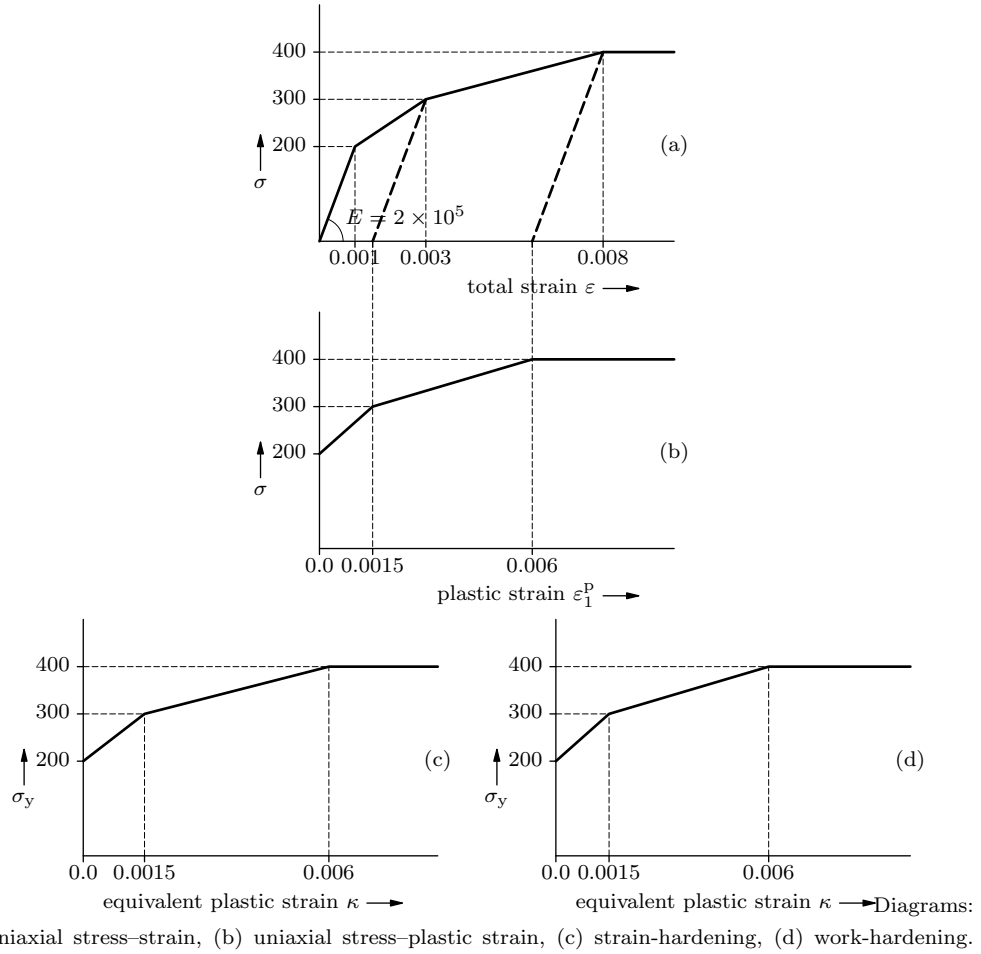


Figure 11.18: Derivation of hardening diagram for Tresca

11.2.3.2 Von Mises

The yield condition of Von Mises is a smooth approximation of the Tresca yield condition: a circular cylinder in the principal stress space [Fig. 11.17b]. The yield function of Von Mises is given by the square root formulation

$$f(\boldsymbol{\sigma}, \boldsymbol{\eta}, \kappa) = \sqrt{3J_2} - \bar{\sigma}(\kappa) = \sqrt{\frac{1}{2}(\boldsymbol{\sigma} - \boldsymbol{\eta})^T \mathbf{P}(\boldsymbol{\sigma} - \boldsymbol{\eta})} - \bar{\sigma}(\kappa) \quad (11.103)$$

where $\bar{\sigma}(\kappa)$ is the uniaxial yield strength as a function of the internal state variable κ , and $\boldsymbol{\eta}$ is the back stress or the center of the yield circle on the π plane [Fig. 11.17b], which moves in the direction of the plastic flow if kinematic hardening effect is taken into account. The projection matrix \mathbf{P} is given by

$$\mathbf{P} = \begin{bmatrix} 2 & -1 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 \\ -1 & -1 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 6 & 0 & 0 \\ 0 & 0 & 0 & 0 & 6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 6 \end{bmatrix} \quad (11.104)$$

The flow rule is generally given by the associated flow rule $g \equiv f$, which results for the plastic strain rate vector in

$$\dot{\boldsymbol{\varepsilon}}^P = \dot{\lambda} \frac{\mathbf{P}(\boldsymbol{\sigma} - \boldsymbol{\eta})}{2\bar{\sigma}} \quad (11.105)$$

The evolution of back stress $\boldsymbol{\eta}$ is given as

$$\dot{\boldsymbol{\eta}} = \frac{2}{3}(1 - \gamma) \frac{\partial \bar{\sigma}}{\partial \kappa} \dot{\lambda} \frac{\mathbf{P}(\boldsymbol{\sigma} - \boldsymbol{\eta})}{2\bar{\sigma}} \quad (11.106)$$

where γ is a scalar parameter ($0 \leq \gamma \leq 1$). The parameter γ controls the contribution of the isotropic and kinematic hardening effect. If $\gamma = 1$ then only the isotropic hardening effect is taken into account i.e. $\dot{\boldsymbol{\eta}} = 0$. If $\gamma = 0$ then only the kinematic hardening effect is taken into account. For intermediate value of γ , e.g. $\gamma = 0.5$, mixed isotropic and kinematic hardening effects are considered. DIANA also offers the possibility to specify a constant $\boldsymbol{\eta}$ i.e. the yield surface is moved by a constant stress shift. In this case $\boldsymbol{\eta}$ does not evolve according to equation 11.106, but remains constant as defined by the user.

Hardening. The relation between the internal state variable κ and the plastic process is given by the hardening hypothesis. For the Von Mises yield condition we consider two different hypotheses: strain hardening and work hardening.

Strain hardening. In the case of strain hardening the relation is given in the principal space by

$$\dot{\kappa} = \sqrt{\frac{2}{3} (\dot{\varepsilon}_1^p \dot{\varepsilon}_1^p + \dot{\varepsilon}_2^p \dot{\varepsilon}_2^p + \dot{\varepsilon}_3^p \dot{\varepsilon}_3^p)} \quad (11.107)$$

which can be elaborated to

$$\dot{\kappa} = \dot{\lambda} \quad (11.108)$$

Work hardening. For work hardening the basic assumption is

$$\dot{W}^p = \boldsymbol{\sigma}^T \dot{\boldsymbol{\varepsilon}}^p \equiv \bar{\sigma}(\kappa) \dot{\kappa} \quad (11.109)$$

with

$$\dot{\boldsymbol{\varepsilon}}^p = \begin{Bmatrix} \dot{\varepsilon}_1^p \\ \dot{\varepsilon}_2^p \\ \dot{\varepsilon}_3^p \end{Bmatrix} = \dot{\lambda} \frac{1}{2\bar{\sigma}} \begin{Bmatrix} 2\sigma_1 - \sigma_2 - \sigma_3 \\ -\sigma_1 + 2\sigma_2 - \sigma_3 \\ -\sigma_1 - \sigma_2 + 2\sigma_3 \end{Bmatrix} \quad (11.110)$$

Eq. (11.109) can be elaborated to

$$\dot{\kappa} = \dot{\lambda} \quad (11.111)$$

Relation $\bar{\sigma}$ - κ . For the Von Mises yield condition the translation of uniaxial experimental data to the equivalent stress-internal state variable, the $\bar{\sigma}$ - κ relation, is independent upon the hardening hypothesis as shown in the example of Figure 11.19 on the next page.

Consider the uniaxial stress-strain diagram of Figure 11.19a. The plastic strain ε_1^p is assumed to be given by $\varepsilon_1 - \varepsilon_1^e$. Figure 11.19b shows the uniaxial stress-plastic strain diagram. The uniaxial plastic strain rate is given by

$$\dot{\varepsilon}_1^p = \dot{\lambda} \frac{\sigma_1}{\bar{\sigma}} \quad (11.112)$$

The relation between the uniaxial stress and the equivalent stress is simply

$$\bar{\sigma} = \sigma_1 \quad (11.113)$$

The following relation can be derived

$$\dot{\varepsilon}_1^p = \dot{\lambda} \quad (11.114)$$

With the relation derived previously, we find for the relation between the uniaxial plastic strain and the internal state variable

$$\dot{\kappa} = \dot{\varepsilon}_1^p \quad (11.115)$$

for both a strain hardening and a work hardening hypothesis.

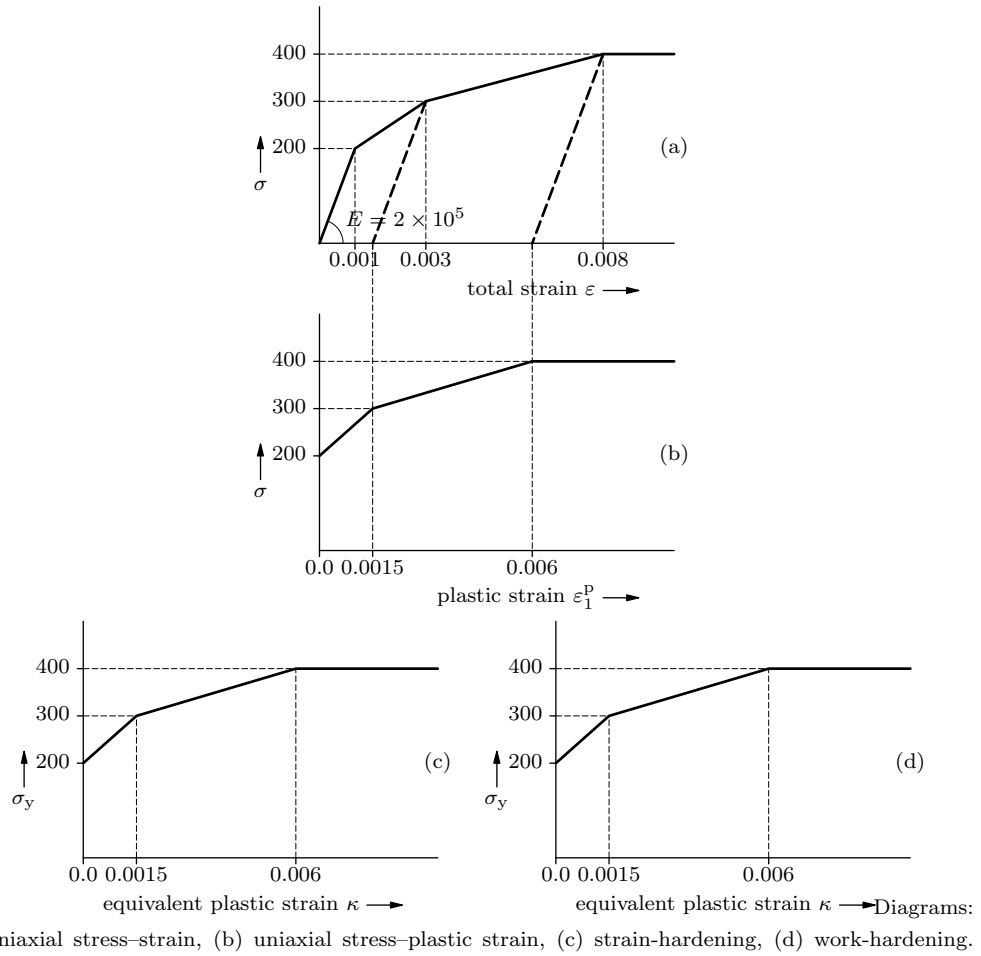


Figure 11.19: Derivation of hardening diagram for Von Mises

Ambient influence. DIANA can handle the influence of temperature, concentration (e.g. moisture content in concrete) or maturity on the Von Mises yield condition. For temperature dependency, the yield condition is given by

$$f(\boldsymbol{\sigma}, \boldsymbol{\eta}, \kappa) = \sqrt{3J_2} - f(T) \frac{\bar{\sigma}(\kappa)}{\bar{\sigma}(0)} = \sqrt{\frac{1}{2}(\boldsymbol{\sigma} - \boldsymbol{\eta})^T \mathbf{P}(\boldsymbol{\sigma} - \boldsymbol{\eta})} - f(T) \frac{\bar{\sigma}(\kappa)}{\bar{\sigma}(0)} \quad (11.116)$$

with $f(T)$ the temperature dependent tensile strength.

11.2.3.3 Mohr–Coulomb

The yield condition of Mohr–Coulomb [Fig. 11.20a], displayed on the left side in the π plane and on the right side in the rendulic plane, is an extension of the Tresca yield condition to a pressure dependent behaviour. The formulation of the yield function can be expressed in the principal stress space ($\sigma_1 \geq \sigma_2 \geq \sigma_3$) as

$$f(\boldsymbol{\sigma}, \kappa) = \frac{1}{2}(\sigma_1 - \sigma_3) + \frac{1}{2}(\sigma_1 + \sigma_3) \sin \phi(\kappa) - \bar{c}(\kappa) \cos \phi_0 \quad (11.117)$$

with $\bar{c}(\kappa)$ the cohesion as a function of the internal state variable κ , and ϕ the angle of internal friction which is also a function of the internal state variable. See also Vermeer & De Borst [80]. The initial angle of internal friction is given by ϕ_0 . The flow rule is given by a general non-associated flow rule $g \neq f$, but with the plastic potential given by

$$g(\boldsymbol{\sigma}, \kappa) = \frac{1}{2}(\sigma_1 - \sigma_3) + \frac{1}{2}(\sigma_1 + \sigma_3) \sin \psi(\kappa) \quad (11.118)$$

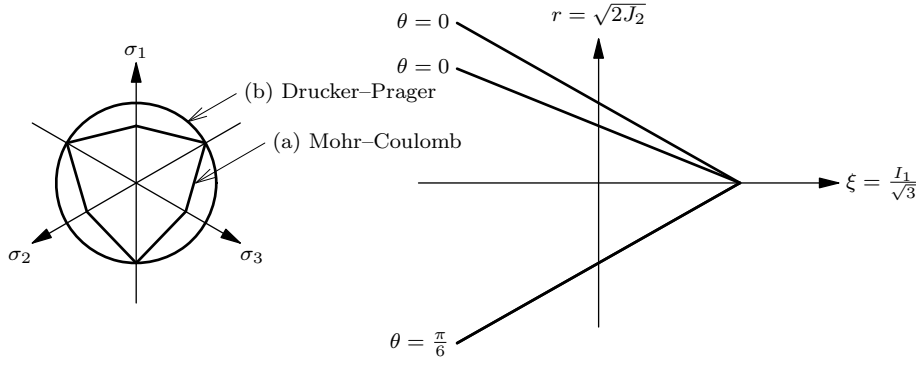


Figure 11.20: Mohr-Coulomb and Drucker-Prager yield condition

which results for the plastic strain rate vector

$$\dot{\epsilon}^p = \dot{\lambda} \begin{Bmatrix} \frac{1}{2}(1 + \sin \psi) \\ 0 \\ -\frac{1}{2}(1 - \sin \psi) \end{Bmatrix} \quad (11.119)$$

Hardening. The relation between the internal state variable κ and the plastic process is given by the hardening hypothesis. For the Mohr-Coulomb yield condition we consider only the strain hardening hypothesis.

Strain hardening. In the case of strain hardening the relation is given in the principal space by

$$\dot{\kappa} = \sqrt{\frac{2}{3} (\dot{\epsilon}_1^p \dot{\epsilon}_1^p + \dot{\epsilon}_2^p \dot{\epsilon}_2^p + \dot{\epsilon}_3^p \dot{\epsilon}_3^p)} \quad (11.120)$$

which can be elaborated to

$$\dot{\kappa} = \dot{\lambda} \sqrt{\frac{1}{3} (1 + \sin^2 \psi)} \quad (11.121)$$

Relation \bar{c} - κ . The translation of uniaxial experimental data to the equivalent cohesion-internal state variable, the \bar{c} - κ relation, depends on the hardening hypothesis. In the following example we will give the derivation for a cohesion hardening material with constant friction and dilatation angle, i.e., $\phi(\kappa) = \phi_0$ and $\psi(\kappa) = \psi_0$, and a strain hardening hypothesis.

Consider the uniaxial stress-strain diagram σ_3 - ϵ_3 of Figure 11.21a. The plastic strain ϵ_3^p is assumed to be given by $\epsilon_3 - \epsilon_3^o$. Figure 11.21b shows the uniaxial stress-plastic strain diagram. For uniaxial stressing, $(\sigma_1, \sigma_2, \sigma_3) = (0, 0, \sigma_3)$, plastic flow occurs at a vertex of the yield surface. Symmetry conditions dictate that the two possible yield directions contribute equally to the plastic strain rate vector

$$\dot{\epsilon}^p = \begin{Bmatrix} \dot{\epsilon}_1^p \\ \dot{\epsilon}_2^p \\ \dot{\epsilon}_3^p \end{Bmatrix} = \dot{\lambda} \begin{Bmatrix} \frac{1}{4}(1 + \sin \psi_0) \\ \frac{1}{4}(1 + \sin \psi_0) \\ -\frac{1}{2}(1 - \sin \psi_0) \end{Bmatrix} \quad (11.122)$$

With the relation derived previously, we find for the relation between the uniaxial plastic strain and the internal state variable for a strain hardening hypothesis

$$\dot{\kappa} = - \frac{\sqrt{1 + \sin^2 \psi_0 - \frac{2}{3} \sin \psi_0}}{1 - \sin \psi_0} \dot{\epsilon}_3^p \quad (11.123)$$

The relation between the uniaxial stress $\sigma_3 = -f_c$ and the equivalent cohesion \bar{c} is given by

$$\bar{c} = f_c \frac{1 - \sin \phi_0}{2 \cos \phi_0} \quad (11.124)$$

Figure 11.21 illustrates the procedure for $\phi_0 = \psi_0 = 30^\circ$.

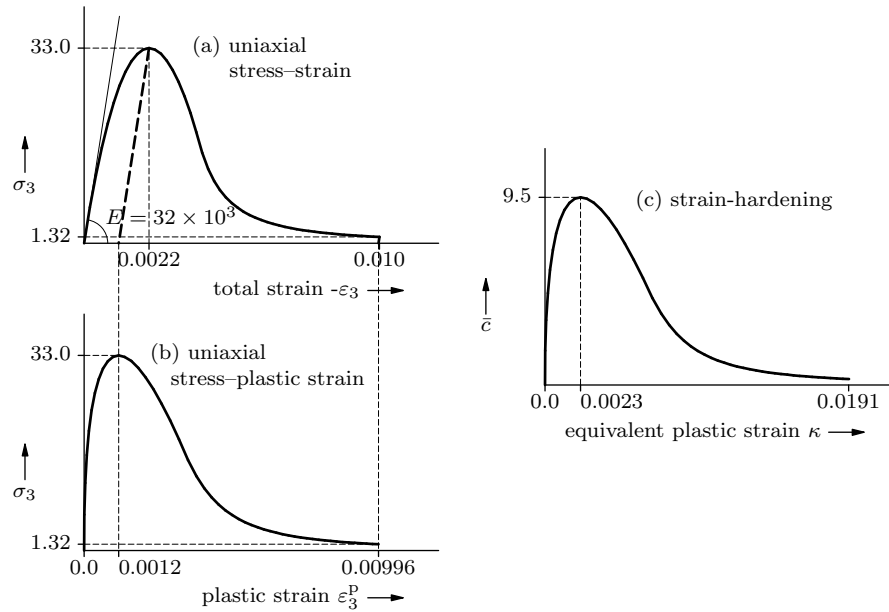


Figure 11.21: Derivation of hardening diagram for Mohr-Coulomb

Tension cut-off. In case of the stress return to the tensile regime an optional tension cut-off value can be specified. In Figure 11.22 the Mohr-Coulomb failure surface is shown

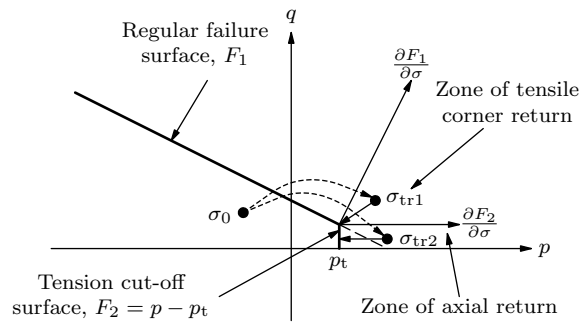


Figure 11.22: Tension cut-off surface for Mohr-Coulomb plasticity model

in $p - q$ plane ($p = I_1/3$, $q = \sqrt{3J_2}$) where the regular Mohr-Coulomb failure surface F_1 is intersected by a tension cut-off surface F_2 . The tension cut-off surface F_2 limits the tension to a cut-off value of p_t . The flow directions from the regular failure surface $\frac{\partial F_1}{\partial \sigma}$ and the tension cut-off surface $\frac{\partial F_2}{\partial \sigma}$ creates two zones for the stress return to the tension cut-off surface, namely the zone of tensile corner return and the zone of axial return. As shown in Figure 11.22, when the trial state σ_{tr1} goes to the zone of tensile corner return, it returns to the corner formed by the intersection of the tension cut-off surface and the regular failure surface. On the other hand when the trial stress σ_{tr2} goes to the zone of axial return, it returns to the tension cut-off surface in a direction parallel to the pressure axis p .

In principal stress space the intersection between the tension cut-off surface and the regular Mohr-Coulomb failure surface form an irregular hexagon with six corners. Returning to such a corner would involve three active yield surfaces (two regular surfaces and the tension cut-off surface) which is very complicated. In order to avoid this situation the boundary of the intersection surface is approximated by a smooth elliptical function such that it goes through all six corners.

11.2.3.4 Drucker–Prager

The yield condition of Drucker–Prager is a smooth approximation of the Mohr–Coulomb yield surface, which is a conical surface in the principal stress space [Fig. 11.20b]. The formulation is given by

$$f(\boldsymbol{\sigma}, \boldsymbol{\eta}, \kappa) = \sqrt{\frac{1}{2}(\boldsymbol{\sigma} - \boldsymbol{\eta})^T \mathbf{P}(\boldsymbol{\sigma} - \boldsymbol{\eta})} + \alpha_f \boldsymbol{\pi}^T (\boldsymbol{\sigma} - \boldsymbol{\eta}) - \beta \bar{c}(\kappa) \quad (11.125)$$

with $\bar{c}(\kappa)$ the cohesion as a function of the internal state variable κ , and $\boldsymbol{\eta}$ is the back stress or the center of the yield circle on the π plane [Fig. 11.17b], which moves in the direction of the plastic flow if kinematic hardening effect is taken into account. The projection matrix is equal to the projection matrix of the Von Mises yield condition defined in Eq. (11.104). The projection vector $\boldsymbol{\pi}$ is given by

$$\boldsymbol{\pi} = \begin{Bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{Bmatrix} \quad (11.126)$$

The scalar quantities α_f and β are given by

$$\alpha_f = \frac{2 \sin \phi(\kappa)}{3 - \sin \phi(\kappa)} \quad \text{and} \quad \beta = \frac{6 \cos \phi_0}{3 - \sin \phi} \quad (11.127)$$

The angle of internal friction ϕ is also a function of the internal state variable. The initial angle of internal friction is given by ϕ_0 . The flow rule is given by a general non-associated flow rule $g \neq f$, with the plastic potential given by

$$g(\boldsymbol{\sigma}, \boldsymbol{\eta}, \kappa) = \sqrt{\frac{1}{2}(\boldsymbol{\sigma} - \boldsymbol{\eta})^T \mathbf{P}(\boldsymbol{\sigma} - \boldsymbol{\eta})} + \alpha_g \boldsymbol{\pi}^T (\boldsymbol{\sigma} - \boldsymbol{\eta}) \quad (11.128)$$

with the scalar α_g defined by the dilatancy angle ψ

$$\alpha_g = \frac{2 \sin \psi(\kappa)}{3 - \sin \psi(\kappa)} \quad (11.129)$$

Which results for the plastic strain rate vector in

$$\dot{\boldsymbol{\epsilon}}^p = \dot{\lambda} \left\{ \frac{\mathbf{P}(\boldsymbol{\sigma} - \boldsymbol{\eta})}{2\Psi} + \alpha_g \boldsymbol{\pi} \right\} \quad (11.130)$$

with the scalar Ψ defined by

$$\Psi = \sqrt{\frac{1}{2}(\boldsymbol{\sigma} - \boldsymbol{\eta})^T \mathbf{P}(\boldsymbol{\sigma} - \boldsymbol{\eta})} \quad (11.131)$$

The evolution of back stress $\boldsymbol{\eta}$ is given as

$$\dot{\boldsymbol{\eta}} = \frac{2}{3}(1 - \gamma) \frac{\beta}{1 + 2\alpha_f \alpha_g} \frac{\partial \bar{c}}{\partial \kappa} \dot{\kappa} \left\{ \frac{\mathbf{P}(\boldsymbol{\sigma} - \boldsymbol{\eta})}{2\Psi} + \alpha_g \boldsymbol{\pi} \right\} \quad (11.132)$$

where γ is a scalar parameter ($0 \leq \gamma \leq 1$), defined in the same way as the Von Mises plasticity, which acts only on the cohesion hardening.

Hardening. The relation between the internal state variable κ and the plastic process is given by the hardening hypothesis. For the Drucker–Prager yield condition we consider only the strain hardening hypothesis.

Strain hardening. In the case of strain hardening the relation is given in the principal space by

$$\dot{\kappa} = \sqrt{\frac{2}{3} (\dot{\varepsilon}_1^p \dot{\varepsilon}_1^p + \dot{\varepsilon}_2^p \dot{\varepsilon}_2^p + \dot{\varepsilon}_3^p \dot{\varepsilon}_3^p)} \quad (11.133)$$

With

$$\begin{Bmatrix} \dot{\varepsilon}_1^p \\ \dot{\varepsilon}_2^p \\ \dot{\varepsilon}_3^p \end{Bmatrix} = \dot{\lambda} \left(\frac{1}{2\Psi} \begin{Bmatrix} 2\sigma_1 - \sigma_2 - \sigma_3 \\ -\sigma_1 + 2\sigma_2 - \sigma_3 \\ -\sigma_1 - \sigma_2 + 2\sigma_3 \end{Bmatrix} + \alpha_g \begin{Bmatrix} 1 \\ 1 \\ 1 \end{Bmatrix} \right) \quad (11.134)$$

Eq. (11.133) can be elaborated to

$$\dot{\kappa} = \dot{\lambda} \sqrt{1 + 2\alpha_g^2} \quad (11.135)$$

Experimental derivation of plasticity parameters.

Uniaxial fit. The translation of uniaxial experimental data to the equivalent cohesion–internal state variable, the \bar{c} – κ relation, depends on the hardening hypothesis. In the following example we will give the derivation for a cohesion hardening material with constant friction and dilatation angle, i.e., $\phi(\kappa) = \phi_0$ and $\psi(\kappa) = \psi_0$, and a strain hardening hypothesis.

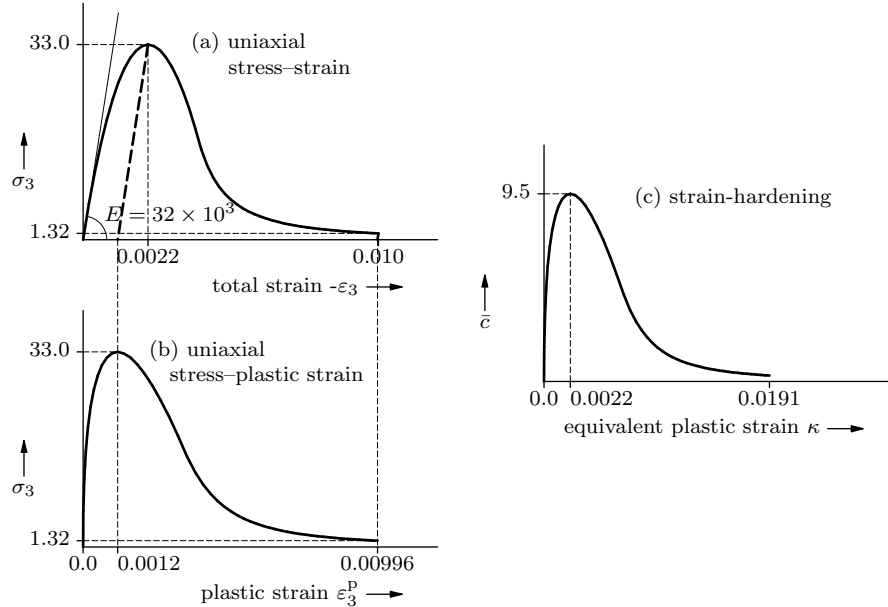


Figure 11.23: Derivation of hardening diagram for Drucker–Prager

Consider the uniaxial stress–strain diagram σ_3 – ε_3 of Figure 11.23a. The plastic strain ε_3^p is assumed to be given by $\varepsilon_3 - \varepsilon_3^s$. Figure 11.23b shows the uniaxial stress–plastic strain diagram. With the assumption $\sigma_3 \leq 0$, the uniaxial plastic strain rate is given by

$$\dot{\varepsilon}_3^p = -\dot{\lambda} (1 - \alpha_g) \quad (11.136)$$

With the relation derived previously, we find for the relation between the uniaxial plastic strain and the internal state variable for a strain hardening hypothesis

$$\dot{\kappa} = -\frac{\sqrt{1 + 2\alpha_g^2}}{1 - \alpha_g} \dot{\varepsilon}_3^p \quad (11.137)$$

The relation between the uniaxial stress $\sigma_3 = -f_c$ and the equivalent cohesion \bar{c} is given by

$$\bar{c} = f_c \frac{1 - \alpha_f}{\beta} = f_c \frac{1 - \sin \phi_0}{2 \cos \phi_0} \quad (11.138)$$

if the friction angle is constant. Figure 11.23 illustrates the procedure for $\phi_0 = \psi_0 = 30^\circ$.

Biaxial fit. The constitutive behaviour of materials like concrete under biaxial states of stress is in general different from the constitutive behaviour under uniaxial loading conditions. The experimental data of concrete subjected to proportional biaxial loading shows the influence of the lateral compressive stress on the strength of the material. Experiments by Kupfer & Gerstle [47] produced the data as shown in Figure 11.24 with the biaxial fit of the Drucker–Prager failure surface. The maximum compressive strength increases approximately 16% under conditions of equal biaxial compression and about 25% increase is achieved at a stress ratio of $\sigma_1/\sigma_2 = 0.5$. The parameters of the Drucker–Prager failure surface, the friction angle ϕ and the cohesion \bar{c} , are calibrated with the following procedure. The uniaxial fit is given in Eq. (11.138) as

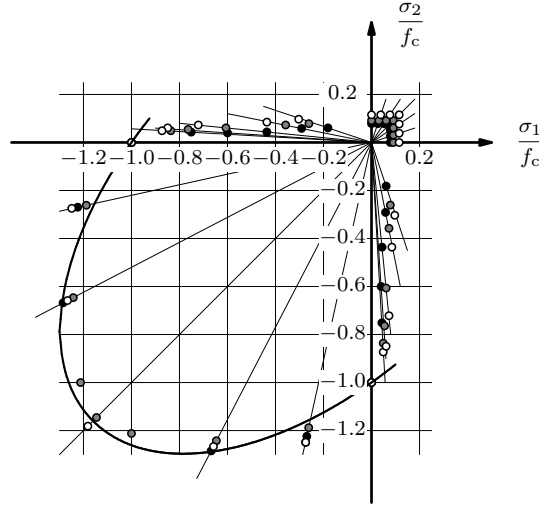


Figure 11.24: Biaxial strength of plain concrete, Kupfer and Gerstle

$$\bar{c} = f_c \frac{1 - \alpha_f}{\beta}$$

The biaxial fit is calculated by substituting the stress vector in case of a plane stress state

$$\boldsymbol{\sigma} = \begin{Bmatrix} -a f_c \\ -a f_c \\ 0 \end{Bmatrix} \quad (11.139)$$

with a the multiplication factor for the biaxial strength. Substituting the stress vector into the equation of the failure surface Eq. (11.125) yields the following condition

$$a f_c - 2 \alpha_f a f_c - \beta \bar{c} = 0 \quad \therefore \quad \bar{c} = a f_c \frac{1 - 2 \alpha_f}{\beta} \quad (11.140)$$

Solving Eq. (11.138) and Eq. (11.140) for α_f , given the factor a , results in

$$\alpha_f = \frac{a - 1}{2a - 1} = \frac{2 \sin \phi_0}{3 - \sin \phi_0} \quad (11.141)$$

which is solved for $\sin \phi_0$

$$\sin \phi_0 = \frac{3 \alpha_f}{2 + \alpha_f} = \frac{3a - 3}{5a - 3} \quad (11.142)$$

Finally, the cohesion is derived from the uniaxial compressive strength and the friction angle ϕ_0 according to

$$\bar{c} = f_c \frac{1 - \sin \phi_0}{2 \cos \phi_0} \quad (11.143)$$

For a normal strength quality concrete, the ratio between the uniaxial compressive strength and the biaxial compressive strength is approximately 1.16 which results in a friction angle $\phi_0 \approx 10^\circ$ and a cohesion $\bar{c} \approx 0.42 f_c$.

Tension cut-off. In case of the stress return to the tensile regime an optional tension cut-off value can be specified. The additional tension cut-off surface is defined in the same way as in the case of Mohr–Coulomb model ([§ 11.2.3.3 p. 238]). However, because Drucker–Prager failure surface has a smooth conical shape in the principal stress space, its intersection with the tension cut-off plane forms a circle. Therefore, unlike the Mohr–Coulomb model, approximation by an additional elliptical function is not necessary.

11.3 Nonlinear Solution Procedures

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 §11.1, 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 section 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 section 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.

$$\mathbf{f}_{\text{int}} = \mathbf{f}_{\text{ext}} \quad (11.144)$$

$$\mathbf{u}_i = \mathbf{u}_i^0 \quad (i \text{ prescribed}) \quad (11.145)$$

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

$$\mathbf{f}_{\text{int}}(\mathbf{u}, \text{history}) = \mathbf{f}_{\text{ext}}(\mathbf{u}) \quad (11.146)$$

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. (11.146) 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

$${}^{t+\Delta t}\mathbf{u} = {}^t\mathbf{u} + \Delta\mathbf{u} \quad (11.147)$$

and, with \mathbf{g} as the *out-of-balance* force vector (the residual forces).

$$\mathbf{g}(\Delta \mathbf{u}) = \mathbf{f}_{\text{ext}}(\Delta \mathbf{u}) - \mathbf{f}_{\text{int}}(\Delta \mathbf{u}) = \mathbf{0} \quad (11.148)$$

Acceleration forces (dynamic analysis) are considered in Chapter 10 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.

11.3.1 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. 11.25]. In all procedures, the

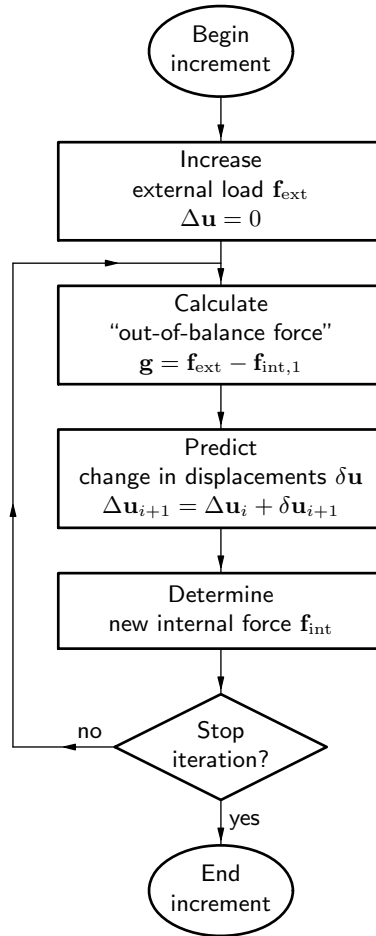


Figure 11.25: Iteration process

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

$$\Delta \mathbf{u}_{i+1} = \Delta \mathbf{u}_i + \delta \mathbf{u}_{i+1} \quad (11.149)$$

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

$$\delta \mathbf{u}_i = \mathbf{K}_i^{-1} \mathbf{g}_i \quad (11.150)$$

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 [§ 11.3.5.2 p. 251]. This method adapts the increment size.

11.3.1.1 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. (11.150) 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:

$$\mathbf{K}_i = \frac{\partial \mathbf{g}}{\partial \Delta \mathbf{u}} \quad (11.151)$$

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. (11.151) is evaluated every iteration [Fig. 11.26]. This means that the prediction of Eq. (11.150) 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

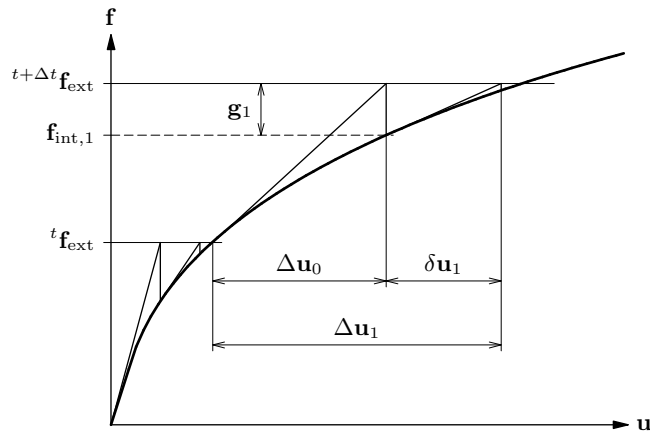


Figure 11.26: Regular Newton–Raphson iteration

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.

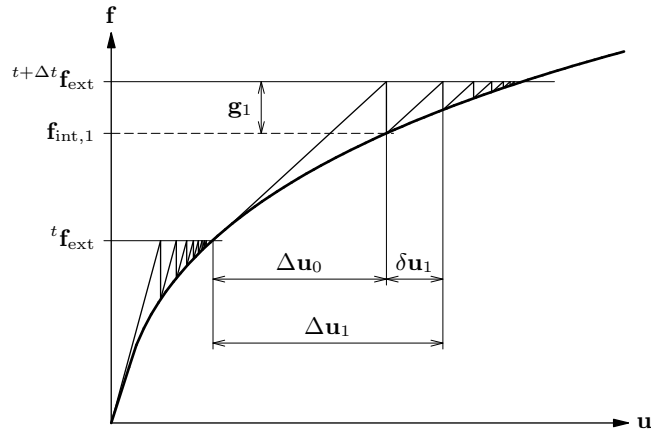


Figure 11.27: Modified Newton–Raphson iteration

Modified Newton–Raphson. The Modified Newton–Raphson method only evaluates the stiffness relation Eq. (11.151) at the start of the increment [Fig. 11.27]. This means that the prediction is always based on a converged equilibrium state. Usually, Modified 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.

11.3.1.2 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. 11.28]. Unlike Regular Newton–Raphson, the Quasi-Newton method does not set up a completely new stiffness matrix every iteration. In

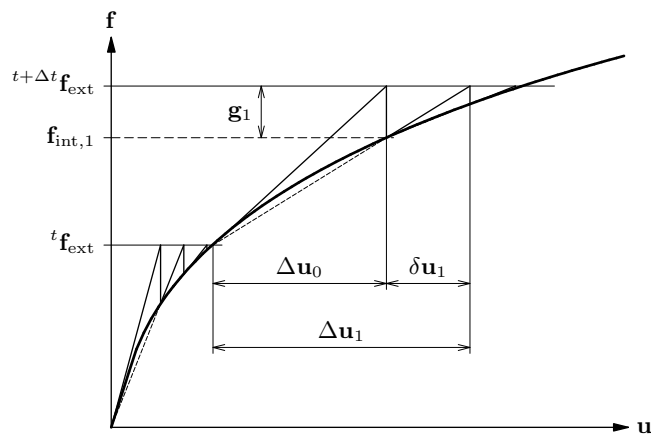


Figure 11.28: Quasi-Newton iteration

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

$$\mathbf{K}_{i+1} \delta \mathbf{u}_i = \delta \mathbf{g}_i \quad (11.152)$$

With a matrix \mathbf{K}_i that fulfills Eq. (11.152), the next iterative increment is calculated from Eq. (11.150). 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. (11.152).

$$\mathbf{K}_{i+1} = \mathbf{K}_i + \frac{(\delta \mathbf{g}_i - \mathbf{K}_i \delta \mathbf{u}_i) \mathbf{c}^T}{\mathbf{c}^T \delta \mathbf{u}_i} \quad (11.153)$$

$$\begin{aligned} \mathbf{K}_{i+1} = \mathbf{K}_i + & \frac{(\delta \mathbf{g}_i - \mathbf{K}_i \delta \mathbf{u}_i) \mathbf{c}^T + \mathbf{c} (\delta \mathbf{g}_i - \mathbf{K}_i \delta \mathbf{u}_i)^T}{\mathbf{c}^T \delta \mathbf{u}_i} \\ & - \frac{(\delta \mathbf{g}_i - \mathbf{K}_i \delta \mathbf{u}_i)^T \delta \mathbf{u}_i \mathbf{c} \mathbf{c}^T}{(\mathbf{c}^T \delta \mathbf{u}_i)^2} \end{aligned} \quad (11.154)$$

In Eq. (11.153) and Eq. (11.154) 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. (11.153) \mathbf{c} is substituted by $\delta \mathbf{u}$ and \mathbf{K}_{i+1} is inverted, the Broyden method results:

$$\mathbf{K}_{i+1}^{-1} = \mathbf{K}_i^{-1} + \frac{(\delta \mathbf{u}_i - \mathbf{K}_i^{-1} \delta \mathbf{g}_i) \delta \mathbf{u}_i^T \mathbf{K}_i^{-1}}{\delta \mathbf{u}_i^T \mathbf{K}_i^{-1} \delta \mathbf{g}_i} \quad (11.155)$$

BFGS. More elaborative, Eq. (11.154) can yield the relation attributed to Broyden, Fletcher, Goldfarb and Shanno, and therefore known as the BFGS method:

$$\mathbf{K}_{i+1}^{-1} = \left(\mathbf{I} + \frac{\delta \mathbf{u}_i \delta \mathbf{g}_i^T}{\delta \mathbf{u}_i^T \delta \mathbf{g}_i} \right) \mathbf{K}_i^{-1} \left(\mathbf{I} - \frac{\delta \mathbf{g}_i \delta \mathbf{u}_i^T}{\delta \mathbf{u}_i^T \delta \mathbf{g}_i} \right) + \frac{\delta \mathbf{u}_i \delta \mathbf{u}_i^T}{\delta \mathbf{u}_i^T \delta \mathbf{g}_i} \quad (11.156)$$

The inverse secant stiffness matrices are not calculated explicitly, but the iterative displacements $\delta \mathbf{u}$ are calculated directly by substitution of Eq. (11.155) or Eq. (11.156) in Eq. (11.150). By successive application of Eq. (11.155) or Eq. (11.156), 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 11.28, but the Quasi-Newton relation Eq. (11.152) 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 11.28, 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.

11.3.1.3 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. 11.29]. This method potentially has the slowest convergence, but it

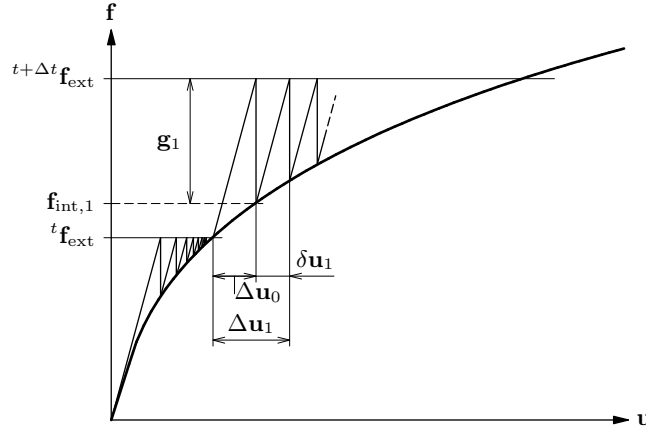


Figure 11.29: Linear Stiffness iteration

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.

11.3.2 Continuation

If a deformation process is relatively continuous, then the displacements of the previous increment can be used as a first prediction for the current increment. The displacements are scaled by the ratio between the current and the previous loading factor or by the ratio of time increments in case of time stepping. Figure 11.30 shows the Continuation method combined with Linear Stiffness iteration. Compared to Figure 11.29, this example needs fewer iterations. But be careful:

If the equilibrium path shows irregularities, it is not advisable to use the Continuation method. An irregularity occurs for instance at loading–unloading points. In that case the displacement increments of the previous step are not a good prediction for the increments in the current step.

In the Continuation method, the first prediction of $\delta \mathbf{u}$ involves only a scalar multiplication on vector $\Delta \mathbf{u}$. This hardly uses any computation time.

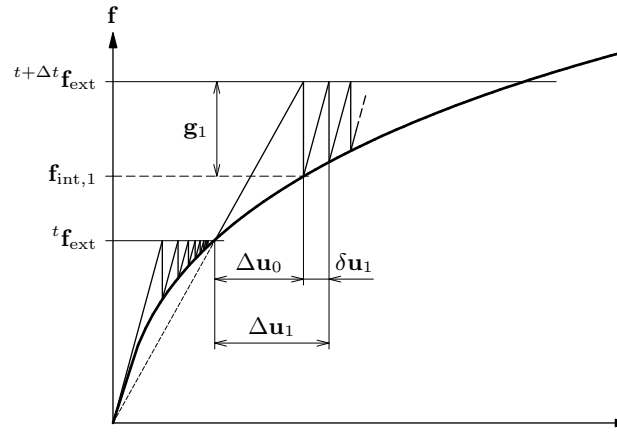


Figure 11.30: Continuation method (Linear Stiffness)

11.3.3 Line Search

All iteration methods described so far are based on a *reasonable prediction*, so that the iteration process converges to the ‘exact’ numerical solution. If the prediction is too far from equilibrium the iteration process will not converge! This easily takes place in structures with strong nonlinearities, for instance cracking. Line Search algorithms can increase the convergence rate and are especially useful if the ordinary iteration process fails.

The Line Search algorithm uses a prediction of the iterative displacement increment $\delta \mathbf{u}$ as obtained by one of the ordinary iteration algorithms and scales this vector by a value to minimize the energy potential Π . Although ‘energy potential’ may not be the correct terminology for the physical behaviour, e.g. in case of plasticity, this poses no problem for the algorithmic implementation within an increment. While the local minimum of the energy potential represents the equilibrium, the minimum in the line search direction can be regarded as the best solution in the predicted direction.

To determine an optimal magnification factor, the incremental displacement vector $\Delta \mathbf{u}_{i+1}$ is calculated from a scaled iterative increment

$$\Delta \mathbf{u}_{i+1} = \Delta \mathbf{u}_i + \eta \delta \mathbf{u}_{i+1} \quad (11.157)$$

where $\delta \mathbf{u}_{i+1}$ is derived from the selected iteration method, see Crisfield [22, §9.2]. A minimum of Π in the line search direction requires that the derivative of Π to η must be zero.

$$s(\eta) = \frac{\partial \Pi}{\partial \eta} = \frac{\partial \Pi}{\partial \mathbf{u}} \frac{\partial \mathbf{u}}{\partial \eta} = \mathbf{g}(\eta)^\top \delta \mathbf{u} = 0 \quad (11.158)$$

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. 11.31]. In most cases a value $\Psi = 0.8$ is sufficient to 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.

11.3.4 Convergence Criteria

The iteration process must be stopped if the results are satisfactory [Fig. 11.25 p. 243]. For this purpose, DIANA offers several convergence norms. Besides stopping the iteration in

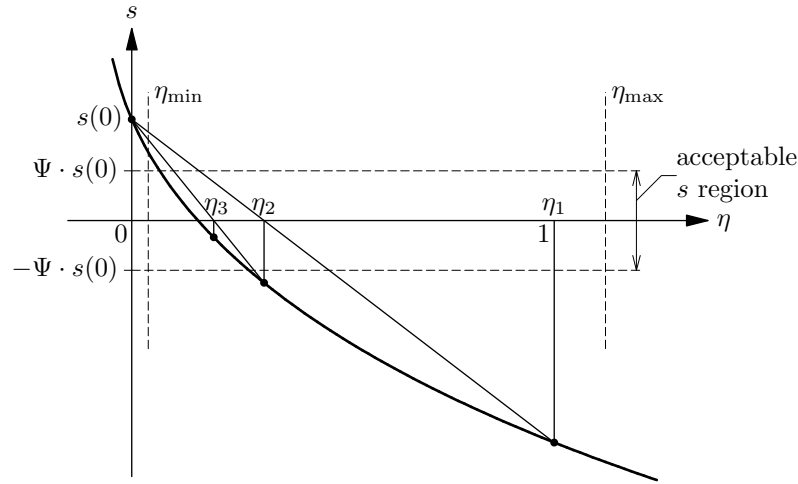


Figure 11.31: Line Search iteration

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 11.32 specifies the items used to set up the various norms.

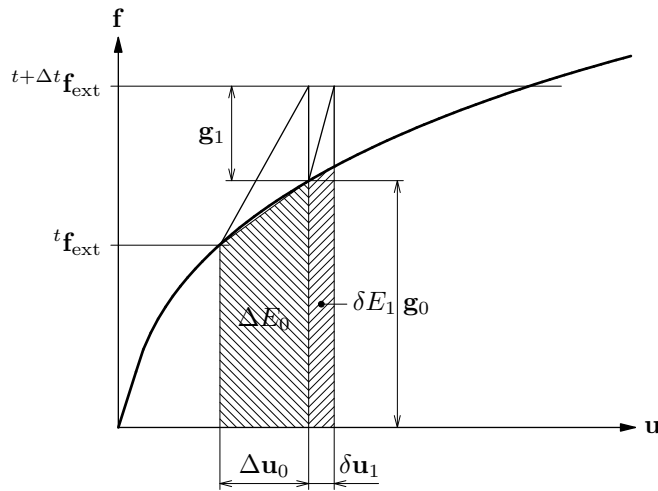


Figure 11.32: Norm items

11.3.4.1 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

$$\text{Force norm ratio} = \frac{\sqrt{\mathbf{g}_i^T \mathbf{g}_i}}{\sqrt{\mathbf{g}_0^T \mathbf{g}_0}} \quad (11.159)$$

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. (11.159). 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.

11.3.4.2 Displacement Norm

The displacement norm 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.

$$\text{Displacement norm ratio} = \frac{\sqrt{\delta \mathbf{u}_i^T \delta \mathbf{u}_i}}{\sqrt{\Delta \mathbf{u}_0^T \Delta \mathbf{u}_0}} \quad (11.160)$$

From Eq. (11.160) 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.

11.3.4.3 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 11.32 with ΔE_0 and δE_1 . To determine convergence, the energy ratio is calculated as

$$\text{Energy norm ratio} = \left| \frac{\delta \mathbf{u}_i^T (\mathbf{f}_{\text{int},i+1} + \mathbf{f}_{\text{int},i})}{\Delta \mathbf{u}_0^T (\mathbf{f}_{\text{int},1} + \mathbf{f}_{\text{int},0})} \right| \quad (11.161)$$

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. (11.158), 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 too large.

11.3.4.4 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.

$$\text{Residual norm ratio} = \frac{|\sqrt{\mathbf{g}_i^T \mathbf{g}_i} - \sqrt{\mathbf{g}_{i-1}^T \mathbf{g}_{i-1}}|}{|\sqrt{\mathbf{g}_0^T \mathbf{g}_0} - \sqrt{\mathbf{g}_n^T \mathbf{g}_n}|} \quad (11.162)$$

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.

11.3.5 Stepping Schemes and Arc-length Method

The incremental-iterative solution procedure consists of two parts: the *increment* part and the *iteration* part. The iteration part was discussed in §11.3.1, 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 [§11.3.5.4]. This method requires only the final loading to be specified. From this, DIANA automatically determines the intermediate step size.

11.3.5.1 Load and Displacement Control

In §11.3.1 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. 11.33a]. Another way to put an external load on a structure is to prescribe certain displacements \mathbf{u}^c . This is called ‘displacement control’ [Fig. 11.33b]. In case of displacement control the external force vector is not increased di-

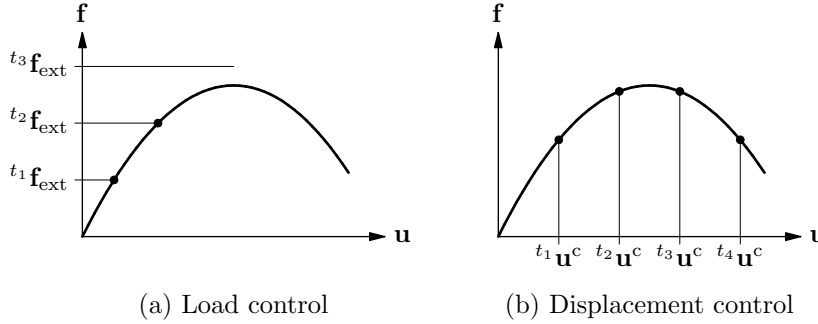


Figure 11.33: Load and displacement control

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. (11.150) 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:

$$\begin{bmatrix} \mathbf{K}^{uu} & \mathbf{K}^{uc} \\ \mathbf{K}^{cu} & \mathbf{K}^{cc} \end{bmatrix}_0 \begin{Bmatrix} \Delta \mathbf{u}^u \\ \Delta \mathbf{u}^c \end{Bmatrix} = \begin{Bmatrix} \mathbf{g}^u \\ \mathbf{g}^c \end{Bmatrix}_0 \quad (11.163)$$

The unknown displacement increments $\Delta \mathbf{u}^u$ can be calculated from the first row in Eq. (11.163) as

$$\Delta \mathbf{u}_0^u = (\mathbf{K}^{uu})^{-1} \{-\mathbf{K}_0^{uc} \Delta \mathbf{u}^c + \mathbf{g}_0^u\} \quad (11.164)$$

Comparing Eq. (11.150) and Eq. (11.164) indicates that $-\mathbf{K}_0^{uc} \Delta \mathbf{u}^c$ 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 a 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.

11.3.5.2 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 11.34a¹ can

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.

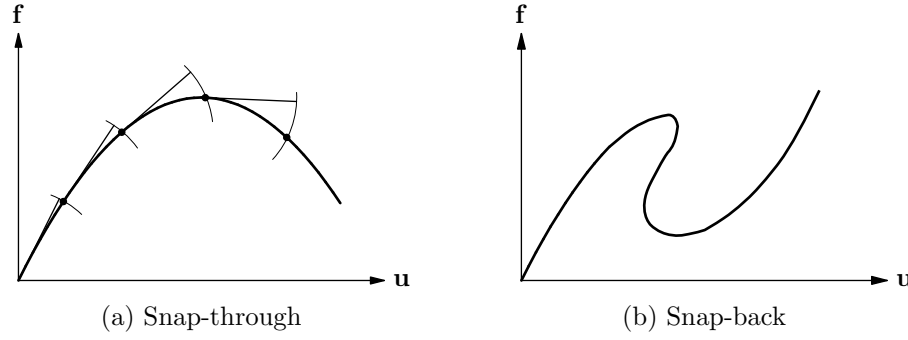


Figure 11.34: Arc-length control

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. 11.34b], 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. (11.150) results in

$$\delta\mathbf{u}_i = \mathbf{K}_i^{-1} \left(\Delta\lambda_i \hat{\mathbf{f}} + {}^t\mathbf{f}_{\text{int}} - \mathbf{f}_{\text{int},i} \right) \quad (11.165)$$

The solution $\delta\mathbf{u}_i$ is now split in two parts:

$$\delta\mathbf{u}_i^{\text{I}} = \mathbf{K}_i^{-1} ({}^t\mathbf{f}_{\text{int}} - \mathbf{f}_{\text{int},i}) \quad \text{and} \quad \delta\mathbf{u}_i^{\text{II}} = \mathbf{K}_i^{-1} \hat{\mathbf{f}} \quad (11.166)$$

The total iterative increment is then derived from

$$\delta\mathbf{u}_i = \delta\mathbf{u}_i^{\text{I}} + \Delta\lambda_i \delta\mathbf{u}_i^{\text{II}} \quad (11.167)$$

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

$$\Delta\mathbf{u}_i^{\text{T}} \Delta\mathbf{u}_i = \Delta l^2 \quad (11.168)$$

where Δl is the required arc length. Substitution of Eq. (11.149) and Eq. (11.167) in Eq. (11.168) gives the value for $\Delta\lambda$

$$\Delta\lambda_i = \frac{-a_2 \pm \sqrt{a_2^2 - 4a_1a_3}}{2a_1} \quad (11.169)$$

with

$$\begin{aligned} a_1 &= (\delta\mathbf{u}_i^{\text{II}})^{\text{T}} \delta\mathbf{u}_i^{\text{II}} \\ a_2 &= 2(\delta\mathbf{u}_i^{\text{I}})^{\text{T}} \delta\mathbf{u}_i^{\text{II}} + 2(\Delta\mathbf{u})^{\text{T}} \delta\mathbf{u}_i^{\text{II}} \\ a_3 &= 2(\Delta\mathbf{u})^{\text{T}} \delta\mathbf{u}_i^{\text{I}} + (\delta\mathbf{u}_i^{\text{I}})^{\text{T}} \delta\mathbf{u}_i^{\text{I}} + (\Delta\mathbf{u})^{\text{T}} \Delta\mathbf{u} - \Delta l^2 \end{aligned}$$

Normally, two solutions for $\Delta\lambda$ fulfill Eq. (11.168) 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 [27]. 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

$$\cos \theta = \frac{(\Delta \mathbf{u}_{i-1})^T \delta \mathbf{u}_i}{\|\Delta \mathbf{u}_{i-1}\| \|\delta \mathbf{u}_i\|} \quad (11.171)$$

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. (11.168) 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

$$(\Delta \mathbf{u}_{i-1})^T \delta \mathbf{u}_i = 0 \quad (11.172)$$

where the quadratic term in $\delta \mathbf{u}_i$ is ignored. Substituting Eq. (11.167) into Eq. (11.172) leads to the expression for $\Delta \lambda_i$

$$\Delta \lambda_i = - \frac{(\Delta \mathbf{u}_{i-1})^T \delta \mathbf{u}_i^I}{(\Delta \mathbf{u}_{i-1})^T \delta \mathbf{u}_i^{II}} \quad (11.173)$$

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. (11.168) and Eq. (11.172), 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

$$\mathbf{v} = \begin{Bmatrix} \alpha_1 u_1 \\ \alpha_2 u_2 \\ \alpha_3 u_3 \\ \dots \\ \alpha_n u_n \end{Bmatrix} \quad (11.174)$$

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.

$$\mathbf{v} = \begin{Bmatrix} \dots \\ \dots \\ \alpha_1 u_p + \alpha_2 u_q \\ \dots \end{Bmatrix} \quad (11.175)$$

This vector is used in the constraint equations Eq. (11.168) or Eq. (11.172).

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.

11.3.5.3 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 [§ 11.3.5.2 p. 251], and the *cutback based automatic incremental loading* method [§ 11.3.5.4]. 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

$${}^{t+\Delta t}\Delta\lambda_0 = \frac{{}^t\Delta l}{\sqrt{{}^\delta\mathbf{u}_0^T {}^\delta\mathbf{u}_0}} \left(\frac{N^d}{{}^tN} \right)^\gamma \quad (11.176)$$

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

$${}^{t+\Delta t}\Delta\lambda_0 = \frac{{}^t\Delta l}{\sqrt{{}^\delta\mathbf{u}_0^T {}^\delta\mathbf{u}_0}} \quad \text{and} \quad {}^{t+\Delta t}\Delta l = {}^{t+\Delta t}\Delta\lambda_0 \sqrt{{}^\delta\mathbf{u}_0^T {}^\delta\mathbf{u}_0} \quad (11.177)$$

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. 11.35]. 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.

The new loading factor is derived from

$${}^{t+\Delta t}\Delta\lambda_0 = \sqrt{\frac{{}^t\Delta\lambda_n ({}^t\Delta\mathbf{u}_n)^T \hat{\mathbf{f}}}{|{}^\delta\mathbf{u}_0^T \hat{\mathbf{f}}|}} \quad (11.178)$$

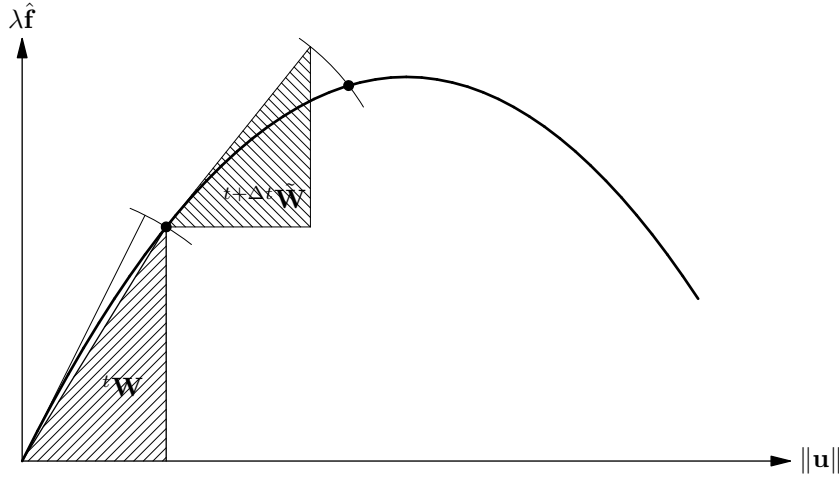


Figure 11.35: Work increment

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

$${}^{t+\Delta t}\Delta\lambda = \begin{cases} +|{}^{t+\Delta t}\Delta\lambda| & \text{if } {}^t\Delta\mathbf{u}_n^T \delta\mathbf{u}_0^{II} \geq 0 \\ -|{}^{t+\Delta t}\Delta\lambda| & \text{if } {}^t\Delta\mathbf{u}_n^T \delta\mathbf{u}_0^{II} < 0 \end{cases} \quad (11.179)$$

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.

11.3.5.4 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.

$${}^{t+\Delta t}\lambda_0 \times \text{minsize} \leq {}^{t+\Delta t}\lambda \leq {}^{t+\Delta t}\lambda_0 \times \text{maxsize} \quad (11.180)$$

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 and the calculation is restarted. If, after successive failures, the load step becomes smaller than a user-specified part of the full loading, 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 the step size is increased. Optionally, the maximum load step size can be limited to a user specified part of the full loading 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

$$\frac{d\mathbf{u}}{dt} = \mathbf{f}(\mathbf{u}) \quad (11.181)$$

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. (11.181) with initial condition $\mathbf{u}(t_n) = \mathbf{u}_n$. The time step error is then

$$\mathbf{u}(t_{n+1}) - \mathbf{u}_{n+1} = \mathcal{O}(\Delta t^{m+1}) \quad (11.182)$$

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 [§ 10.5.5 p. 199], 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

$$\|\mathbf{u}_{n+1} - \mathbf{u}(t_{n+1})\| \leq \|\hat{\mathbf{u}}_{n+1} - \mathbf{u}_{n+1}\| \leq \epsilon_{\text{rel}} \times \|\mathbf{u}_{n+1}\| + \epsilon_{\text{abs}} \quad (11.183)$$

This criterion is both used to reject time steps and estimate the next time step (Hairer and Warner [33, § IV.8]):

$$\Delta t_{n+1} = \Delta t_n \times \frac{\|\hat{\mathbf{u}}_{n+1} - \mathbf{u}_{n+1}\|^{\frac{1}{2}}}{\epsilon_{\text{rel}} \times \|\mathbf{u}_n\| + \epsilon_{\text{abs}}} \quad (11.184)$$

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 [§ 11.3.5.4 p. 255]. Conversely, if SDIRK2 is used while there 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.

11.4 User–supplied Subroutines

DIANA offers you, as an end-user, the opportunity to supply Fortran source code of some predefined subroutines. The user-supplied subroutines should be coded with great care. The routine should perform the intended function without influencing other parts of DIANA.

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 [42].
- 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.

Stress and strain vectors. Many of the user-supplied subroutines have strain or stress vectors in their argument list. The length and contents of these vectors depend on the stress situation, c.q. the element family for which the routine is called. Unless otherwise specified, the contents of the strain and stress vectors are as indicated in Table 11.1 on the next page.

Table 11.1: STRESS AND STRAIN VECTORS IN USER-SUPPLIED SUBROUTINES

Element family	ns	sig(ns)	dsig(ns)	eps(ns)	eps0(ns)
Truss elements	1	σ_{xx}		ε_{xx}	
	3	$\sigma_{xx}, \sigma_{yy}, \sigma_{zz}$		$\varepsilon_{xx}, \varepsilon_{yy}, \varepsilon_{zz}$	
Two-dimensional class-II & -III beams	2	σ_{xx}, σ_{xy}		$\varepsilon_{xx}, \gamma_{xy}$	
	4	$\sigma_{xx}, \sigma_{yy}, \sigma_{zz}, \sigma_{xy}$		$\varepsilon_{xx}, \varepsilon_{yy}, \varepsilon_{zz}, \gamma_{xy}$	
Three-dimensional class-II & -III beams	3	$\sigma_{xx}, \sigma_{xy}, \sigma_{zx}$		$\varepsilon_{xx}, \gamma_{xy}, \gamma_{zx}$	
	5	$\sigma_{xx}, \sigma_{yy}, \sigma_{zz}, \sigma_{xy}, \sigma_{yz}$		$\varepsilon_{xx}, \varepsilon_{yy}, \varepsilon_{zz}, \gamma_{xy}, \gamma_{yz}$	
Plane stress elements	3	$\sigma_{xx}, \sigma_{yy}, \sigma_{xy}$		$\varepsilon_{xx}, \varepsilon_{yy}, \gamma_{xy}$	
	4	$\sigma_{xx}, \sigma_{yy}, \sigma_{zz}, \sigma_{xy}$		$\varepsilon_{xx}, \varepsilon_{yy}, \varepsilon_{zz}, \gamma_{xy}$	
Plane strain & axisymmetric elements	4	$\sigma_{xx}, \sigma_{yy}, \sigma_{zz}, \sigma_{xy}$		$\varepsilon_{xx}, \varepsilon_{yy}, \varepsilon_{zz}, \gamma_{xy}$	
Curved shell elements	6	$\sigma_{xx}, \sigma_{yy}, \sigma_{zz} = 0, \sigma_{xy}, \sigma_{yz}, \sigma_{zx}$		$\varepsilon_{xx}, \varepsilon_{yy}, \varepsilon_{zz}, \gamma_{xy}, \gamma_{yz}, \gamma_{zx}$	
Solid elements	6	$\sigma_{xx}, \sigma_{yy}, \sigma_{zz}, \sigma_{xy}, \sigma_{yz}, \sigma_{zx}$		$\varepsilon_{xx}, \varepsilon_{yy}, \varepsilon_{zz}, \gamma_{xy}, \gamma_{yz}, \gamma_{zx}$	
Embedded bar reinforcements	1	σ_{xx}		ε_{xx}	
Embedded bar reinforcements	1	σ_{xx}		ε_{xx}	
Embedded grid reinforcements	2	σ_{xx}, σ_{yy}		$\varepsilon_{xx}, \varepsilon_{yy}$	

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] ALLAART, A. P. *Design Principles for Flexible Pavements – a Computational Model for Granular Bases*. PhD thesis, Delft University of Technology, 1992.
- [3] ALLMAN, D. J. A simple cubic displacement element for plate bending. *Int. J. Num. Meth. Eng.* 10 (1976), 263–281.
- [4] ALLMAN, D. J. Evaluation of the constant strain triangle with drilling rotations. *Int. J. Num. Meth. Eng.* 26 (1988), 2645–2655.
- [5] BATHE, K.-J. *Finite Element Procedures in Engineering Analysis*. Prentice-Hall, 1982.
- [6] BATHE, K.-J. *Finite Element Procedures*. Prentice-Hall, 1996.
- [7] BESSELING, J. F. Finite element properties, based upon elastic potential interpolation. In *Hybrid and Mixed Finite Element Methods*. John Wiley & Sons, 1983, pp. 253–266.
- [8] 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.
- [9] BOER, DE, A. *Design strategy structural concrete in 3D*. PhD thesis, Delft University of Technology, 2010.
- [10] BOGERT, VAN DEN, P. A. J. *Computational Modelling of Rubberlike Materials*. PhD thesis, Delft University of Technology, 1991.
- [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] BORST, DE, R., AND FEENSTRA, P. H. Studies in anisotropic plasticity with reference to the Hill criterion. *Int. J. Num. Meth. Eng.* 29 (1990), 315–336.
- [13] BOYCE, H. R. A non-linear model for the elastic behaviour of granular materials under repeated loading. In *Proc. Int. Symposium on Soils under Cyclic and Transient Loading* (Swansea, 1980).
- [14] 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.
- [15] BUILDING SEISMIC SAFETY COUNCIL. NEHRP Guidelines for the Seismic Rehabilitation of Buildings. Tech. Rep. FEMA-273, Federal Emergency Management Agency, Washington DC, 1997.
- [16] BYSKOV, E., AND HUTCHINSON, J. W. Mode interaction in axially stiffened cylindrical shells. *AIAA Journal* 15, 7 (1977), 941–948.

- [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] 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.
- [24] ERP, VAN, E. M. *Advanced Buckling Analysis of Beams with Arbitrary Cross-sections*. PhD thesis, Eindhoven University of Technology, 1989.
- [25] FENVES, G., AND CHOPRA, A. K. Earthquake Analysis and Response of Concrete Gravity Dams. Tech. Rep. UCBEERC-8410, University of California, Berkeley, 1984.
- [26] FLANAGAN, D. P., AND BELYTSCHKO, T. A uniform strain hexahedron and quadrilateral with orthogonal hourglass control. *Int. J. Num. Meth. Eng.* 17 (1981), 679–706.
- [27] 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.
- [28] GEORGE, A., AND LIU, J. W. *Computer Solution of Large Sparse Positive Definite Systems*. Prentice-Hall, 1981.
- [29] GÉRADIN, M., AND RIXEN, D. *Mechanical Vibrations – Theory and Application to Structural Dynamics*, 2nd ed. John Wiley & Sons, 1997.
- [30] GIJZEN, VAN, M. B. Conjugate Gradient-like Solution Algorithms for the Mixed Finite Element Approximation of the Biharmonic Equation Applied to Plate Bending Problems. Tech. Rep. BI-92-154, TNO Building and Construction Research, Rijswijk, The Netherlands, 1992.
- [31] 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.
- [32] GUPTA, A. K. *Response Spectrum Method In Seismic Analysis and Design of Structures*. CRC Press, Inc., 1992.
- [33] HAIRER, E., AND WANNER, G. *Solving Ordinary Differential Equations II*, 2nd ed. Springer-Verlag, 1996.
- [34] HESTENES, M. R., AND STIEFEL, E. Methods of conjugate gradients for solving linear systems. *J. Res. Nat. Bur. Standards* 49, 6 (1952), 409–436.
- [35] 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.

- [36] 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.
- [37] HUÉTINK, J. *On the Simulation of Thermo-mechanical Forming Processes*. PhD thesis, University of Twente, 1986.
- [38] HUGHES, T. J. R. *The Finite Element Method – Linear Static and Dynamic Finite Element Analysis*. Prentice-Hall, 1987.
- [39] JARDINE, R. J., POTTS, D. M., FOURIE, A. B., AND BURLAND, J. B. Studies of the influence of non-linear stress–strain characteristics in soil–structure interaction. *Géotechnique* 36, 3 (1986), 377–396.
- [40] JARDINE, R. J., SYMES, M. J., AND BURLAND, J. B. The measurement of soil stiffness in the triaxial apparatus. *Géotechnique* 34, 3 (1984), 323–340.
- [41] KARYPIS, G., AND KUMAR, V. A fast and highly quality multilevel scheme for partitioning irregular graphs. *SIAM J. Sc. Comp.* 20, 1 (1999).
- [42] KERNIGHAN, B. W., AND PLAUGER, P. J. *The Elements of Programming Style*, 2nd ed. McGraw-Hill, 1978.
- [43] KOITER, W. T. *Over de Stabiliteit van Elastisch Evenwicht*. PhD thesis, Delft University of Technology, 1945.
- [44] KOITER, W. T. Stress–strain relations, uniqueness and variational theorems for elastic-plastic materials with a singular yield surface. *Q. Appl. Mech.* 11 (1953), 350–354.
- [45] KRIEG, R. D., AND KRIEG, D. B. Accuracies of numerical solution methods for the elastic-perfectly plastic model. *J. Pressure Vessel Techn.* 99 (1977), 510–515.
- [46] 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.
- [47] KUPFER, H. B., AND GERSTLE, K. H. Behavior of concrete under biaxial stresses. *J. Eng. Mech. Div., ASCE* 99, 4 (1973), 853–866.
- [48] LINGEN, F., BONNIER, P., BRINGREVE, 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.
- [49] 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.
- [50] MERKS, P. J. G. Wapeningsgrootheden bij Plaatconstructies. Tech. Rep. BSW 88-05, Dutch Public Works, Rijswijk, The Netherlands, 1988.
- [51] MOONEY, M. A theory of elastic deformations. *J. Appl. Physics* 11 (1940), 582.
- [52] MURNAGHAN, F. D. *Finite Deformation of an Elastic Solid*. John Wiley & Sons, 1951.
- [53] NAFEMS. *Guidelines to Finite Element Practice*. National Agency for Finite Element Methods & Standards (NAFEMS), Glasgow, 1984.
- [54] NAFEMS. *A Finite Element Primer*. National Agency for Finite Element Methods & Standards (NAFEMS), Glasgow, 1992.
- [55] NEN. Concrete Bridges (VBB 1995) (Structural requirements and calculation methods). Tech. Rep. NEN 6723, Nederlands Normalisatie-instituut, 1995.

- [56] NEN. The Design of Steel Bridges (VOSB 1995) (Basic requirements and simple rules). Tech. Rep. NEN 6788, Nederlands Normalisatie-instituut, 1995.
- [57] 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.
- [58] 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.
- [59] ORTIZ, M., AND POPOV, E. P. Accuracy and stability of integration algorithms for elastoplastic constitutive relations. *Int. J. Num. Meth. Eng.* 21 (1985), 1561–1576.
- [60] POLIZZI, E. FEAST Eigenvalue Solver. <http://www.ecs.umass.edu/~polizzi/feast/index.htm>. Accessed: 2014/02/19.
- [61] RAHMAN, T. *A Perturbation Approach for Geometrically Nonlinear Structural Analysis Using a General Purpose Finite Element Code*. PhD thesis, Delft University of Technology, 2009.
- [62] RIVLIN, R. S. Large elastic deformations of isotropic materials, fundamental concepts. *Phyl. Trans. Roy. Soc. London* 240 (1948), 459–490.
- [63] 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.
- [64] SAAD, Y. ILUT: a dual threshold incomplete ILU factorization. *Numerical Linear Algebra with Applications* 1 (1994), 387–402.
- [65] SAAD, Y. *Iterative Methods for Sparse Linear Systems*, 2nd ed. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2003.
- [66] 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.
- [67] SANGERS, A. Enhancing iterative solution methods for general FEM computations using rigid body modes. Master’s thesis, Delft University of Technology, 2014.
- [68] SCHELLEKENS, J. C. J., AND BORST, DE, R. The use of the Hoffmann yield criterion in Finite Element Analysis of anisotropic composites. *Comp. & Struct.* 37, 6 (1990), 1087–1096.
- [69] SCHENK, O. *Scalable Parallel Sparse LU Factorization Methods on Shared Memory Multiprocessors*. PhD thesis, Eidgenössische Technische Hochschule Zürich.
- [70] 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.
- [71] SCHREYER, H. L., KULAK, R. F., AND KRAMER, J. M. Accurate numerical solutions for elastic-plastic models. *J. Pressure Vessel Techn.* 101 (1979), 226–234.
- [72] SIMO, J. C., KENNEDY, J. G., AND GOVINDJEE, S. Non-smooth multisurface plasticity and viscoplasticity. Loading/unloading conditions and numerical algorithms. *Int. J. Num. Meth. Eng.* 26 (1988), 2161–2185.
- [73] SIMO, J. C., AND TAYLOR, R. L. Penalty function formulations for incompressible nonlinear elastostatics. *Comp. Meth. Appl. Mech. Eng.* 35 (1982), 107–118.
- [74] SIMO, J. C., AND TAYLOR, R. L. Consistent tangent operators for rate-independent elastoplasticity. *Comp. Meth. Appl. Mech. Eng.* 48 (1985), 101–118.
- [75] SIMO, J. C., AND TAYLOR, R. L. A return-mapping algorithm for plane stress elastoplasticity. *Int. J. Num. Meth. Eng.* 22 (1986), 649–670.

- [76] 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.
- [77] TISO, P. *Finite Element Based Reduction Methods for Static and Dynamic Analysis of Thin-Walled Structures*. PhD thesis, Delft University of Technology, 2006.
- [78] TRELOAR, L. R. G. *The Physics of Rubber Elasticity*, 3rd ed. Oxford University Press, 1975.
- [79] UNIVERSITY OF COLORADO. Advanced Finite Element Methods - Pyramid Solid Elements. <http://www.colorado.edu/engineering/CAS/courses.d/AFEM.d/AFEM.Ch12.d/AFEM.Ch12.pdf>. Accessed: 2015/02/23.
- [80] VERMEER, P. A., AND BORST, DE, R. Non-associated plasticity for soils, concrete and rock. *Heron* 29, 3 (1984), 3–64.
- [81] 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.
- [82] 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.
- [83] 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

ABS rule, [196](#)
Accuracy, eigenvalue analysis, **149**
Adaptive loading, [254](#)
Adaptive loading, cutback based, [255](#)
Adaptive loading, energy based, [254](#)
Adaptive loading, iteration based, [254](#)
Adaptive time increments, [254](#)
Adaptive time increments, cutback based, [255](#)
Adaptive time increments, iteration based, [254](#)
Aircraft hull, **48**
Arc-length control, [251](#)
Automatic load increments, [254](#)
Automatic load increments, cutback based, [255](#)
Automatic load increments, energy based, [254](#)
Automatic load increments, iteration based, [254](#)
Automatic time increments, [254](#)
Automatic time increments, cutback based, [255](#)
Automatic time increments, iteration based, [254](#)
Axisymmetric elements, bar reinforcement, **105**
Axisymmetric elements, grid reinforcement, **108**
Axisymmetric elements, options, **153**

B

Backward substitution, [181](#)
Bar reinforcement, **103**
Base excitation, Response Spectrum Analysis, [195](#)
BCL6S2 element, **211**
BCL6S3 element, **211**
Beam elements, **71**
Beam elements, bar reinforcement, **104**
Beam elements, stability analysis, **152**, **156**
Bending moments, flat shell elements, [51](#)
Bending moments, plate bending elements, [50](#)
Bernoulli beams, **71**
Besseling hyperelasticity, [229](#)
BFGS iteration, [246](#)

Biaxial failure envelope, **142**
Biaxial stress state, [241](#)
Bifurcation, [255](#)
BL4S2 element, **210**
Bond-slip reinforcements, **110**, **110**
Bond-slip, interface elements, **89**, **90**
Bottom absorption, [201](#)
Boundary conditions, essential, [17](#)
Boundary conditions, natural, [17](#)
Boundary surface elements, [204](#)
Box girder, **47**
Boyce nonlinear elasticity, [225](#)
BQ12S4 element, **211**
BQ24S4 element, **212**
BQ24S8 element, **212**
Broyden iteration, [246](#)
BT18S3 element, **213**
BT18S6 element, **213**
BT9S3 element, **213**
Buckling values, **149**
Bulk modulus, nonlinear elasticity, [224](#)

C

Cauchy stress, [217](#)
CG, [183](#)
CHX60 element, **42**
CHX96 element, **43**
CL12I element, **86**
CL18B element, **79**
CL18I element, **90**
CL24I element, **89**
CL32I element, **89**
CL3CM element, **93**
CL6TM element, **206**
CL6TR element, **70**
CL9BE element, **78**
CL9TM element, **206**
CL9TR element, **70**
Class-I beams, cross-section, **80**, **81**
Class-I beams, eccentricity, **81**
Class-I beams, profile library, **80**
Class-I beams, stability analysis, **152**, **156**
Class-II beams, cross-section, **81**
Class-II beams, stability analysis, **156**
Class-III beams, automatic tying, **120**
Class-III beams, cross-section, **81**
Class-III beams, eccentricity, **81**
Class-III beams, stability analysis, **152**
CMOD control, [253](#)
Cohesion, Drucker–Prager plasticity, [240](#), [241](#)
Cohesion, Mohr–Coulomb plasticity, [236](#), [237](#)
Cohesion, vs. internal state, [237](#), [240](#)

- Composed elements, [90](#)
 - Composed line elements, [90](#)
 - Composed surface elements, [90](#)
 - Compressibility, [229](#)
 - Compressive strength, concrete plasticity, [241](#)
 - Concrete biaxial failure envelope, [142](#)
 - Concrete safety factors, [142](#)
 - Condition number, [183](#)
 - Conjugate Gradient, [183](#)
 - Conservative loading, [152](#)
 - Consistent mass analysis, [146](#)
 - Constant Stiffness iteration, [247](#)
 - Constrained Minimization, [220](#)
 - Contact analysis, [220](#)
 - Contact elements, [220](#)
 - Continuation analysis, [158](#), [160](#)
 - Continuation iteration, [247](#)
 - Convergence criteria, [248](#)
 - Convergence rate, [244](#), [246](#), [248](#)
 - Coulomb friction, [220](#)
 - CPY39 element, [40](#)
 - CQ12C element, [95](#)
 - CQ16A element, [33](#)
 - CQ16E element, [30](#)
 - CQ16M element, [26](#)
 - CQ16O element, [29](#)
 - CQ18M element, [27](#)
 - CQ24P element, [53](#)
 - CQ24TM element, [208](#)
 - CQ40F element, [55](#)
 - CQ40L element, [64](#)
 - CQ40S element, [59](#)
 - CQ48F element, [57](#)
 - CQ48I element, [87](#)
 - CQ48L element, [65](#)
 - CQ48S element, [63](#)
 - CQ60S element, [60](#)
 - CQ8CM element, [94](#)
 - CQC rule, [196](#)
 - Crack tip, [215](#)
 - Crisfield iteration, [246](#)
 - Critical damping factor, [192](#)
 - Cross-section, beam elements, [80](#)
 - CT12A element, [32](#)
 - CT12E element, [30](#)
 - CT12M element, [26](#)
 - CT12O element, [29](#)
 - CT18P element, [53](#)
 - CT18TM element, [208](#)
 - CT30A element, [33](#)
 - CT30E element, [31](#)
 - CT30F element, [54](#)
 - CT30L element, [63](#)
 - CT30S element, [59](#)
 - CT36F element, [57](#)
 - CT36I element, [87](#)
 - CT36L element, [64](#)
 - CT36S element, [62](#)
 - CT45S element, [60](#)
 - CT6CM element, [94](#)
 - CT9CM element, [94](#)
 - CTE30 element, [40](#)
 - CTE48 element, [42](#)
 - CTP45 element, [41](#)
 - CTP72 element, [43](#)
 - Curved shell elements, automatic tying, [120](#)
 - Curved shell elements, bar reinforcement, [105](#), [111](#)
 - Curved shell elements, eccentricity, [66](#)
 - Curved shell elements, grid reinforcement, [108](#)
 - Curved shell elements, stability analysis, [153](#)
- ## D
- Damping, [192](#)
 - Damping, added, [203](#)
 - Damping, continuous, [96](#), [194](#)
 - Damping, modal, [192](#)
 - Damping, Rayleigh, [147](#), [151](#), [192](#)
 - Damping, strain energy based, [151](#)
 - Damping, structural, [192](#)
 - Damping, viscous, [192](#)
 - Decomposition, [181](#)
 - Degrees of freedom, [18](#)
 - Der Kiureghian, A., [196](#)
 - Deviatoric strain energy, [229](#)
 - Deviatoric strains, [135](#)
 - Deviatoric stresses, [136](#)
 - Direct solution, [181](#), [194](#)
 - Direct time integration, [197](#)
 - Direction dependent participation factors, [149](#)
 - Displacement control, [251](#)
 - Displacement norm, [249](#)
 - Displacement, phased analysis, [176](#)
 - Distributed mass elements, [204](#)
 - Distributed translational mass elements, [204](#)
 - Divergence, [244](#), [249](#)
 - Domain decomposition, [184](#)
 - Drag coefficient, [126](#)
 - Drilling rotation, curved shell elements, [47](#)
 - Drilling rotation, flat shell elements, [47](#)
 - Drilling rotation, plane stress elements, [23](#)
 - Drucker–Prager plasticity, [239](#)
 - Dynamic analysis, rotation inertia, [102](#)
 - Dynamic concrete biaxial failure envelope, [142](#)
- ## E
- Eccentricity, [118](#), [120](#)
 - Eccentricity, beam elements, [81](#)
 - Effective force vector, [251](#)
 - Effective mass, [150](#)
 - Eigenfrequency, [148](#), [149](#)
 - Eigenvalue, [148](#), [148](#)
 - Element internal forces, [144](#)
 - Element interpolation matrix, [17](#)
 - Element rigidity matrix, [18](#)
 - Element transformation matrix, [18](#)
 - ELF modal distribution, [193](#)
 - Embedded reinforcements, *see* Reinforcements
 - Energy load increments, [254](#)
 - Energy norm, [250](#)
 - Engineering strains, [135](#)
 - Equality tying, [117](#)
 - Equivalent lateral force modal distribution, [193](#)
 - Euler Backward integration, [229](#)

Euler Backward time integration, [198](#)
 Explicit time integration, transient analysis, [197](#)

F

FEMA 273, [193](#)
 First order displacement, [153](#)
 Flat shell elements, analytically integrated, [56](#)
 Flat shell elements, stability analysis, [153](#)
 Floor, [46](#)
 Flow rule, [229](#)
 Fluid density, fluid–structure interaction, [209](#)
 Fluid–structure interaction, background theory, [200](#)
 Fluid–structure interaction, eigenvalue analysis, [145](#)
 Fluid–structure interaction, fluid density, [209](#)
 Fluid–structure interaction, interface elements, [209](#)
 Force norm, [249](#)
 Force, internal vector, [19](#)
 Force–elongation diagram, spring elements, [96](#)
 Forces, flat shell elements, [51](#)
 Forces, plate bending elements, [50](#)
 Fortran compiler, [257](#)
 Forward substitution, [181](#)
 Foundation, [35](#)
 Frame, [71](#)
 Free field, boundary elements, [204](#)
 Free vibration, [146](#), [149](#)
 Frequency motion, [148](#)
 Frequency response analysis, [193](#)
 Friction, Coulomb, [220](#)

G

General connections, [119](#)
 Generalized mass, [149](#)
 Generalized Minimal Residual, [184](#)
 Geometric nonlinearity, [215](#)
 Geometric stress–stiffness matrix, [147](#)
 GMRES, [184](#)
 Grains nonlinear elasticity, [225](#)
 Granular material, [224](#)
 Grid reinforcement, [104](#)

H

Hardening, [229](#)
 Hardening, Drucker–Prager plasticity, [239](#)
 Hardening, Mohr–Coulomb plasticity, [237](#)
 Hardening, Tresca plasticity, [232](#)
 Hardening, Von Mises plasticity, [235](#)
 Hilber–Hughes–Taylor time integration, [198](#)
 Hinge, [67](#)
 Hooke’s law, [18](#)
 HX24L element, [39](#)
 Hydrostatic pressure capacity, [142](#)
 Hydrostatic pressure, grains, [224](#)
 Hydrostatic strain energy, [229](#)
 Hyperelasticity, [216](#), [228](#)

Hypoelasticity, [224](#)

I

Identity matrix, [147](#)
 Ill-conditioned system, [186](#)
 ILU preconditioning, [185](#)
 Imperfections, [156](#)
 Imperfections, user-specified, [157](#)
 Implicit time integration, structural dynamics, [197](#)
 Incremental procedures, [250](#)
 Indirect Displacement control, [253](#)
 Influence field, [126](#)
 Input data, dynamic analysis, [191](#)
 Interconnection tying, [117](#)
 Interface elements, contact, [220](#)
 Interface elements, fluid–structure interaction, [209](#)
 Interface elements, stability analysis, [153](#)
 Internal beam arm, [140](#), [165](#)
 Internal force vector, [19](#)
 Interpolation functions, [18](#)
 Interpolation matrix, [17](#)
 Isoparametric elements, stability analysis, [156](#)
 Iterative solution procedure, linear analysis, [182](#)
 Iterative solution procedure, nonlinear analysis, [243](#)

J

Jacobi preconditioning, [185](#)
 Jardine elastoplasticity, [225](#)
 Jaumann rate, [217](#)

K

Kirchhoff plate, [46](#)
 Koiter’s rule, [229](#)
 Krylov subspace, [183](#)
 Krylov subspace, Conjugate Gradient, [183](#)
 Krylov subspace, Generalized Minimal Residual, [184](#)
 Kuhn–Tucker conditions, [229](#)

L

L12BE element, [76](#)
 L12BEA element, [79](#)
 L12IF element, [89](#)
 L13BE element, [77](#)
 L16IF element, [88](#)
 L20IF element, [88](#)
 L2CMP element, [93](#)
 L2TRU element, [69](#)
 L4TM element, [205](#)
 L4TRU element, [69](#)
 L6BEA element, [78](#)
 L6BEN element, [75](#)
 L6TM element, [206](#)
 L6TRU element, [70](#)
 L7BEN element, [77](#)
 L8IF element, [86](#)

Lagrange, [216](#), [217](#)
 Large deformations, [155](#)
 LDU decomposition, [181](#)
 Line Search, [248](#)
 Linear Stiffness iteration, [247](#)
 Load, [120](#)
 Load control, [251](#)
 Location points of reinforcement, bar, [103](#)
 Lode's angle, [137](#)
 Lower plane, plate bending elements, [51](#)
 Lumped mass, [146](#)
 Lumping, [191](#)

M

Mass, [191](#)
 Mass density, [192](#)
 Mass elements, stability analysis, [153](#)
 Mass, added, [203](#)
 Master node, [116](#)
 Matrix spring elements, [96](#)
 Mesh refinement, [117](#)
 METIS, [184](#)
 Metis reordering, [182](#)
 Mid plane, plate bending elements, [51](#)
 Mindlin beam, [71](#)
 Mindlin plate, [46](#)
 Mindlin shell, [47](#)
 Mobile load, [122](#)
 Modal damping, [192](#)
 Modal damping factor, [151](#)
 Modal mass, [151](#)
 Modal pushover analysis, [193](#)
 Modal result combinations, [196](#)
 Mode superposition, frequency response analysis, [194](#)
 Mode superposition, Response Spectrum Analysis, [195](#)
 Modified Newton–Raphson iteration, nonlinear analysis, [245](#)
 Mohr–Coulomb plasticity, [236](#)
 Mohr–Coulomb failure criterion, [142](#)
 Mooney–Rivlin hyperelasticity, [229](#)
 Mother elements of reinforcement, [102](#), [103](#)
 Murnaghan compressibility, [229](#)

N

N4IF element, [85](#)
 N6IF element, [85](#)
 Negative pivots for (un)loading, [255](#)
 Neo-Hookean deviatoric strain energy, [229](#)
 Newmark time integration, [197](#)
 Newton–Raphson iteration, nonlinear analysis, [244](#)
 Nodal load, [144](#)
 Nonconservative loading, [215](#), [219](#)
 Nonlinear analysis, [242](#)
 Nonlinear analysis, iteration, [243](#)
 Nonlinear elasticity, [224](#)

Norms for convergence, [248](#)

O

Off-shore structures, [35](#)
 OpenMP, [184](#)
 Ordering, [185](#)
 Orthotropic thickness, [34](#)

P

Parallel direct sparse solver, [182](#)
 PARDISO, [182](#)
 Participation factors, [149](#)
 Participation factors, direction dependent, [149](#)
 Participation vectors, [149](#)
 Particles of reinforcement, bars, [103](#)
 Particles of reinforcement, grids, [104](#)
 Perturbation analysis, [148](#), [157](#), [158](#)
 Phased analysis, displacement, [176](#)
 Pile Foundations, [110](#)
 Piola–Kirchhoff stress (2nd), [217](#)
 Plane strain elements, grid reinforcement, [107](#)
 Plane strain elements, stability analysis, [152](#)
 Plane stress elements, bar reinforcement, [105](#),
[111](#)
 Plane stress elements, grid reinforcement, [107](#)
 Plane stress elements, stability analysis, [152](#)
 Plastic yield, [137](#)
 Plasticity, [229](#)
 Plate bending elements, stability analysis, [153](#)
 Polymer material, [229](#)
 Postbuckling analysis, [157](#), [158](#)
 Preconditioning, [184](#)
 Predefined shapes for beam elements, [80](#)
 Pressure, [137](#)
 Pressure supports, [210](#)
 Principal strain, [135](#)
 Principal stress, [136](#)
 Profile library, [80](#)
 Pseudo-acceleration, [195](#)
 Pushover analysis, [193](#)
 PY15L element, [38](#)

Q

Q12ME element, [27](#)
 Q12PL element, [52](#)
 Q12TM element, [207](#)
 Q20SF element, [54](#)
 Q20SH element, [58](#)
 Q24IF element, [87](#)
 Q24SF element, [56](#)
 Q24SH element, [62](#)
 Q4CMP element, [94](#)
 Q8AXI element, [32](#)
 Q8EPS element, [30](#)
 Q8MEM element, [25](#)
 Q8OME element, [28](#)

Quasi-Newton iteration, [245](#)

R

Random imperfection, [157](#)
 Rayleigh damping, [192](#)
 Reaction forces, [145](#)
 Reference norm, [249](#)
 Refinement of mesh, [117](#)
 Regular Newton–Raphson iteration, [244](#)
 Reinforcement, [67](#)
 Reinforcement bar, shear stress, [142](#)
 Reinforcement bar, stress gradient, [142](#)
 Reinforcement forces, [137](#)
 Reinforcement moments, [137](#)
 Reinforcement, bar, [103](#)
 Reinforcement, grid, [104](#)
 Reinforcement, stability analysis, [153](#)
 Reinforcements, bond-slip, [110](#)
 Reinforcements, embedded, [102](#)
 Residual forces, [145](#)
 Residual forces, nonlinear analysis, [243](#)
 Residual norm, [250](#)
 Response Spectrum Analysis, [195](#)
 Response Spectrum Analysis, modal result combinations, [196](#)
 Return-mapping, [229](#)
 Right Cauchy–Green stretch, [228](#)
 Rigidity matrix, [18](#)
 Rivlin deviatoric strain energy, [229](#)
 Rotational mass, [101](#)
 Runge–Kutta time integration, [199](#)

S

Safety factors, [142](#)
 Schur Complement, [186](#)
 Schwarz domain decomposition, [184](#)
 SDIRK2 method, [199](#), [256](#)
 Second order displacements, [153](#), [157](#)
 Sections, reinforcement, [104](#)
 Seismic Moment, [135](#)
 Shape functions, [18](#)
 Shear capacity, [142](#)
 Shear locking, curved shell elements, [59](#), [60](#), [63](#)
 Shear locking, layered shell elements, [64](#), [65](#)
 Shear locking, plate bending elements, [46](#)
 Shear reduction, curved shell elements, [66](#)
 Shear stress correction, [72](#)
 Shear stress, reinforcement bar, [142](#)
 Shell elements, automatic tying, [120](#)
 Ship hull, [48](#)
 Simo–Taylor compressibility, [229](#)
 Single Program, Multiple Data, [184](#)
 Single-point tyings, [117](#)
 Slave node, [116](#)
 Snap-back behaviour, [252](#)
 Snap-through behaviour, [251](#), [254](#), [255](#)
 Softening, [147](#)
 Soil, [31](#), [33](#)
 Soil-structure interaction, [35](#)
 Solid elements, automatic tying, [120](#)
 Solid elements, bar reinforcement, [106](#), [111](#)

Solid elements, stability analysis, [153](#)
 Solution methods, [180](#)
 Solution, iterative procedure, [242](#), [243](#)
 Sommerfeld radiation, [201](#)
 SP1RO element, [99](#)
 SP1TR element, [98](#)
 SP2RO element, [99](#)
 SP2TR element, [98](#)
 Space deck, [67](#)
 Sparse Cholesky solver, [182](#)
 Sparse matrix, [181](#)
 Spectral acceleration, [195](#)
 Spectral displacement, [195](#)
 Spherical Path, [252](#)
 Spline elements, stability analysis, [153](#)
 SPMD, [184](#)
 Spring diagrams, base spring, [97](#)
 Spring elements, stability analysis, [153](#)
 SRSS rule, [196](#)
 Stability analysis, [151](#), [153](#)
 Stability analysis, linearized buckling, [147](#), [149](#)
 Standard eigenproblem, [147](#), [148](#)
 Static concrete biaxial failure envelope, [142](#)
 Static condensation, [186](#)
 Steady-state response, [191](#)
 Stiffener, [67](#), [71](#)
 Storage tank, [48](#)
 Straight-normals hypothesis, curved shell elements, [47](#)
 Straight-normals hypothesis, flat shell elements, [46](#)
 Straight-normals hypothesis, plate bending elements, [46](#)
 Strain, [134](#)
 Strain energy based damping, [151](#)
 Strain matrix, [135](#)
 Stress, [136](#)
 Stress gradient, reinforcement bar, [142](#)
 Stress invariants, [137](#)
 Stress matrix, [136](#)
 Stress stiffening, [215](#)
 Stress–strain relation, [229](#)
 Structural dynamics, [191](#)
 Substructuring, [186](#)
 Superelement, [186](#)
 Symbols, glossary of, [xxi](#)

T

T15SF element, [53](#)
 T15SH element, [58](#)
 T18FSH element, [56](#)
 T18IF element, [86](#)
 T18SF element, [55](#)
 T18SH element, [61](#)
 T3CMP element, [93](#)
 T6AXI element, [31](#)
 T6EPS element, [29](#)
 T6MEM element, [25](#)
 T6OME element, [28](#)
 T9MEM element, [27](#)
 T9PLA element, [52](#)
 T9TM element, [207](#)

Tangent stiffness, [151](#), [153](#)
 Tapered cross-section of beam elements, [80](#)
 Tapered thickness, plane stress elements, [34](#)
 TE12L element, [38](#)
 Temperature influence, Tresca plasticity, [233](#)
 Temperature influence, Von Mises plasticity, [236](#)
 Tensile strength, Tresca plasticity, [233](#)
 Tensile strength, Von Mises plasticity, [236](#)
 Thickness, curved shell elements, [66](#)
 Time integration, [196](#)
 Total Lagrange, [217](#)
 TP18L element, [39](#)
 Traffic load, ENV 1991-3 code, [125](#)
 Transformation, element matrix, [18](#)
 Transformation, stress and strain, [137](#)
 Transient response, [191](#), [196](#)
 Translation mass, [101](#)
 Translation spring, [98](#)
 Tresca plasticity, [232](#)
 Truck load, [123](#)
 Truck load, ENV 1991-3 code, [125](#)
 Truss elements, [67](#)
 Truss elements, stability analysis, [152](#)
 Tunnel, [47](#)
 Tyings, automatic, [120](#)

U

Ultimate Limit State, [167](#)
 Uniaxial stress state, [240](#)
 Units, eigenfrequencies, [149](#)
 Updated Lagrange, [217](#)
 Updated Normal Plane, [253](#)
 Upper plane, plate bending elements, [51](#)
 User-supplied subroutines, [256](#)

V

Variables, [209](#), [210](#)
 VBB code, [123](#)
 Viscous damping, [192](#)
 Volumetric strain, [135](#)
 Von Mises plasticity, [234](#)
 Von Mises strain, [135](#)
 Von Mises stress, [136](#)
 VOSB code, [123](#)

W

Wilson time integration, [198](#)

Y

Yield condition, [229](#)
 Young's modulus, position dependent, [35](#), [44](#)

Z

Zero-normal-stress hypothesis, curved shell elements, [47](#)
 Zone in beam element, [81](#)