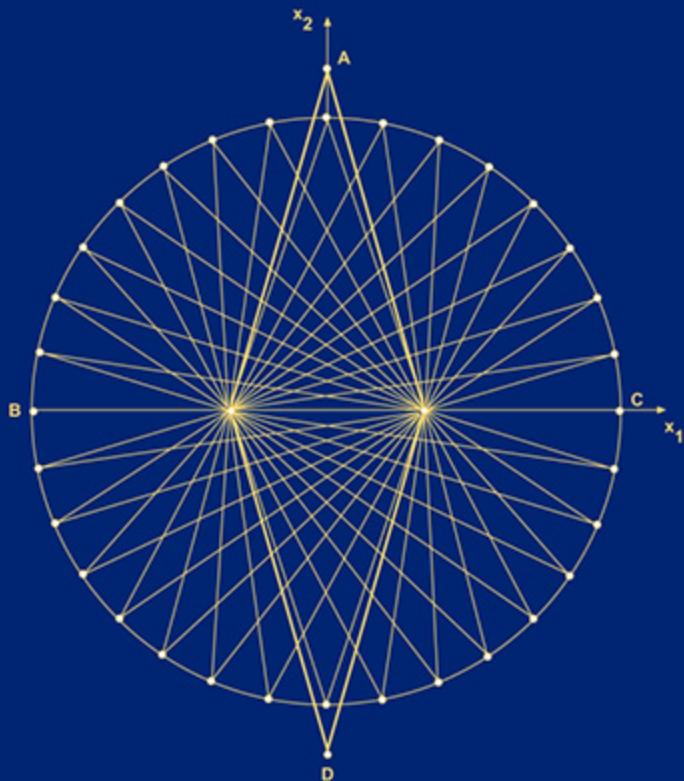


GEOMETRICALLY NONLINEAR ANALYSIS OF PLANE TRUSSES AND FRAMES



GALISHNIKOVA • DUNAISKI • PAHL

GEOMETRICALLY NONLINEAR ANALYSIS
of
PLANE TRUSSES AND FRAMES

Vera Galishnikova

Volgograd State University of Architecture and Civil Engineering

Peter Dunaiski

Stellenbosch University

Peter Jan Pahl

Stellenbosch University



Geometrically Nonlinear Analysis of Plane Trusses and Frames

Published by SUN MeDIA Stellenbosch 7600

www.africansunmedia.co.za

www.sun-e-shop.co.za

All rights reserved.

Copyright © 2009 Vera Galishnikova, Peter Dunaiski, Peter Jan Pahl

This publication presents results of research supported by the Southern African Institute of Steel Construction.

No part of this book may be reproduced or transmitted in any form or by any electronic, photographic or mechanical means, including photocopying and recording on record, tape or laser disk, on microfilm, via the Internet, by e-mail, or by any other information storage and retrieval system, without prior written permission from the publisher and the authors.

First edition 2009

ISBN: 978-1-920109-48-6

e-ISBN: 978-1-920109-99-8

DOI: 10.18820/9781920109998

Cover design by Ilse Roelofse

Typesetting: Authors

SUN PRess is an imprint of SUN MeDIA Stellenbosch. Academic, professional and reference works are published under this imprint in print and electronic format. This publication may be ordered directly from www.sun-e-shop.co.za

Contents

Preface	i
1 State of the Art of Nonlinear Structural Analysis	
1.1 Nonlinear Structural Behaviour	1
1.2 Nonlinear Structural Analysis	2
1.3 Guides to the State of the Art	4
1.4 State of the Art in the Formulation of the Theory	8
1.5 State of the Art in Numerical Solution Techniques	11
1.5.1 Test Case	11
1.5.2 Direct Iteration Method	11
1.5.3 Newton-Raphson Iteration Method	13
1.5.4 Modified Newton-Raphson Iteration Method	14
1.5.5 Stepwise Solution with the Basic Arc Increment Method	15
1.5.6 Refined Arc Increment Method	17
1.5.7 Constant Arc Increment Method	19
1.6 State of the Art in Stability Analysis	22
1.6.1 Concepts of Stability	22
1.6.2 Determination of Singular Points	23
1.6.3 Continuation of Load Paths	24
1.7 Research on Nonlinear Structural Analysis	25
1.7.1 Research Needs	25
1.7.2 Research Topics	28
2 Nonlinear Behaviour of Plane Trusses and Frames	
2.1 Nonlinear Behaviour of a 2-Bar Truss	31
2.2 Large Displacements of a Cantilever Beam	40
2.3 Tension Stiffening of a Pinned Beam	42
2.4 Bifurcation Instability of a Column	45
2.5 Contact of Two Rods	48
2.6 Tension-only Member	49
3 Nonlinear Theory of Elasticity	
3.1 Coordinate Systems	51
3.1.1 Global Coordinate System	51
3.1.2 Local Coordinate System	52
3.1.3 Coordinate Transformations	53
3.2 Configurations of a Body	54
3.2.1 Body	54
3.2.2 Configurations	55
3.3 State of Displacement	56
3.3.1 Displacement	56
3.3.2 Rigid Body Translation and Rotation	56
3.4 State of Strain	58
3.4.1 Material Elements (Method of Lagrange)	58
3.4.2 State of Deformation of Material Elements	59
3.4.3 Spatial Elements (Method of Euler)	62
3.4.4 State of Deformation of Spatial Elements	63
3.5 State of Stress	66
3.5.1 Stress Vector and Stress Tensor	66
3.5.2 Stress Vector on a Plane	67
3.5.3 Equilibrium at a Point	68
3.5.4 Piola-Kirchhoff Stress Tensors	69
3.6 Material Properties	70
3.6.1 Significance of the Strain-Stress Relationships	70
3.6.2 Voigt Notation for Strain and Stress	70
3.6.3 Constitutive Equations	71
3.7 Boundary Conditions	74
3.8 Governing Equations	76
3.8.1 Differential Governing Equations	76

3.8.2	Weighted Residual Method	77
3.8.3	Spatial Integral Form	78
3.8.4	Material Integral Form	79
3.9	Incremental Governing Equations	81
3.9.1	Incremental Solution Concept	81
3.9.2	Linearisation of the Governing Equations	83
3.10	Algebraic Governing Equations	85
3.10.1	Finite Elements	85
3.10.2	System and Element Vectors	86
3.10.3	Interpolation of Variables	87
3.10.4	Interpolation of a Configuration	88
3.10.5	Error Vector for the Incremental Governing Equations	89
3.10.6	Algebraic Incremental Equations	90
4	Plane Trusses	
4.1	Configurations of Plane Trusses	91
4.1.1	Properties of a Plane Truss	91
4.1.2	Coordinate Systems	92
4.1.3	Plane Truss Hypothesis	96
4.1.4	State of Displacement	97
4.1.5	State of Strain	98
4.1.6	Axial Bar Force	99
4.2	Analytical Governing Equations for Plane Trusses	100
4.2.1	System Vectors	100
4.2.2	Governing Equations for a State of the Truss	101
4.2.3	Increments of the State Variables	102
4.2.4	Variation of the Incremental Variables	103
4.2.5	Governing Equations for Displacement Increments	103
4.3	Algebraic Governing Equations for Plane Trusses	106
4.3.1	Interpolation of the State Variables for the Error Vector	106
4.3.2	Interpolation of the Displacement Increments	107
4.3.3	Error Vector in the Governing Equations	109
4.3.4	Algebraic Incremental Equations	109
5	Plane Frames	
5.1	Configurations of Plane Frames	113
5.1.1	Properties of a Plane Frame	113
5.1.2	Coordinate Systems	114
5.1.3	Plane Frame Hypothesis	119
5.1.4	State of Displacement	120
5.1.5	State of Strain	124
5.1.6	Small Deformation Theory	129
5.1.7	Stress Resultants	133
5.2	Analytical Governing Equations for Plane Frames	135
5.2.1	System Vectors	135
5.2.2	Governing Equations for a State of the Frame	136
5.2.3	Increments of the State Variables	137
5.2.4	Variation of the Incremental Variables	140
5.2.5	Governing Equations for the Displacement Increments	141
5.2.6	Governing Equations for a Member	144
5.2.7	Member Loads	145
5.3	Algebraic Governing Equations for Plane Frames	149
5.3.1	Interpolation Functions for the Finite Element Method	149
5.3.2	Interpolation of the Displacements and Strains	150
5.3.3	Interpolation of the Displacement and Strain Increments	153
5.3.4	Error Vector in the Governing Equations	157
5.3.5	Algebraic Incremental Equations	160
5.4	Stress Resultants in a Member	164
5.4.1	Governing Equations	164
5.4.2	Stress Resultants due to Node Displacements	164
5.4.3	Displacements and Stress Resultants due to Axial Load	165
5.4.4	Displacements and Stress Resultants due to Transverse Load	166

6	Solution of the Governing Equations	
6.1	Stepwise Solution Procedure	169
6.1.1	Concept of the Numerical Algorithm	169
6.1.2	Normalised Load Paths	170
6.1.3	Load Paths of a 2-Bar Truss	173
6.2	Assembly of the Incremental Governing Equations	177
6.3	Treatment of the Error Vector	181
6.4	Solution of the Incremental Governing Equations	183
6.5	Load Factor Increment	186
6.5.1	Path-Dependent Norms with Equal Signs	186
6.5.2	Change in Sign of the Path-Dependent Force Norm	189
6.5.3	Load Step for a 2-Bar Truss	190
6.6	Linear Equations with Profile Structure and Status	192
7	Stability Analysis	
7.1	Stiffness of Structures	197
7.1.1	Concepts of Stiffness	197
7.1.2	Total Stiffness of a 2-Bar Truss	199
7.1.3	Tangent Stiffness of a 2-Bar Truss	199
7.1.4	Secant Stiffness of a 2-Bar Truss	200
7.2	Singular Configurations of Structures	202
7.2.1	Concept of a Singular Configuration	202
7.2.2	Singular Configurations of a 2-Bar Truss	203
7.3	Nearly Singular Configurations of Structures	206
7.3.1	Concept of a Nearly Singular Configuration	206
7.3.2	Nearly Singular Configurations of a 2-Bar Truss	208
7.4	Determination of Singular Configurations	209
7.4.1	Determination of a Limit Load Factor	209
7.4.2	Singular Points for Trusses	210
7.4.3	Singular Points for Frames	214
7.5	Bifurcation Theory	221
7.5.1	Scope of the Theory	221
7.5.2	Classification of Singular Points	221
7.5.3	Load Path at a Turning Point	223
7.5.4	Load Path at a Bifurcation Point	226
7.5.5	Bifurcation Theory for Finite Element Formulations	233
7.5.6	Classification of the Singular Points of a 2-Bar Truss	234
7.6	Continuation of Load Paths at Turning Points	237
7.6.1	Continuation Concepts	237
7.6.2	Continuation at Turning Points of a 2-Bar Truss	237
7.6.3	Nearly Singular Systems of Equations	239
7.6.4	First Cycle of Iteration in the First Step at a Turning Point	244
7.6.5	General Cycle of Iteration in the First Step at a Turning Point	245
7.6.6	General Load Step of the Continuation at a Turning Point	247
7.7	Continuation of Load Paths at Bifurcation Points	248
7.7.1	Approximation of the Governing Equations	248
7.7.2	Properties at Bifurcation Points of a 2-Bar Truss	248
7.7.3	First Step beyond the Bifurcation Point of a 2-Bar Truss	258
7.7.4	First Step beyond a Bifurcation Point	261
7.7.5	General Step of the Continuation at a Bifurcation Point	263
7.7.6	Total Stiffness Matrix of a Truss	264
7.7.7	Total Stiffness Matrix of a Frame	265
7.8	Solution of the General Eigenvalue Problem	268
8	Software for Nonlinear Structural Analysis	
8.1	Scope of the Implementation	275
8.2	Object Structure of Information	276
8.2.1	Objects and Classes of Objects	276
8.2.2	Identification of Objects	277
8.2.3	Attributes	278
8.2.4	Classes of Objects for Trusses and Frames	279

8.2.5	Aggregation of Objects	280
8.2.6	Aggregations in Java	283
8.2.7	Named Objects	285
8.2.8	Data Bases	287
8.3	Structure of Frame Models	290
8.3.1	Contents of Frame Models	290
8.3.2	Class Structure of Frame Models	291
8.4	Data Structure of a Frame Model	293
8.4.1	Classes Containing the Data Structure	293
8.4.2	Data Structure in Class Frame	293
8.4.3	Data Structure in Class Node	295
8.4.4	Data Structure in Class Support	296
8.4.5	Data Structure in Class Bar	296
8.4.6	Data Structure in Class Member	297
8.4.7	Data Structure in Class Stability	299
8.4.8	Data Structure in Class Graphic Panel	299
8.5	Method Structure of a Frame Model	301
8.5.1	Class Structure for Nonlinear Analysis	301
8.5.2	Methods for Deformation Analysis	302
8.5.3	Methods for Stability Analysis	306
8.5.4	System Indices of the Node Displacements	314
8.5.5	System Stiffness Matrix, Load Vector and Error Vector	316
8.5.6	Tension-only Analysis	317
8.6	Graphical User Interface	319
8.6.1	Functionality of the Interface	319
8.6.2	Structure of the Interface	321
8.6.3	Coordinate Systems	322
8.6.4	Coordinate Grid	324
8.6.5	Paint Algorithm for the Graphics Panel	326
8.6.6	Action Algorithm for the Graphics Panel	327
8.6.7	Algorithm for the Result Panel	329
8.6.8	Algorithm for the Frame Panel	330
8.6.9	Construction of the Editors	331
8.6.10	Screen Shots of Editors	334
8.6.11	Method Structure of the Graphical Interface	336
9	Examples of Geometrically Nonlinear Structural Analysis	
9.1	Scope and Purpose of the Investigation	341
9.2	2-Bar Truss with Turning Point	342
9.3	2-Bar Truss with Bifurcation Point	352
9.4	Cantilever Beam with End Moment	360
9.5	General Conclusions	371
	References	373
	Subject Index	379

Preface

This book is an outcome of academic cooperation between the Volgograd State University of Architecture and Civil Engineering in Russia, Stellenbosch University in South Africa and the Technische Universität Berlin in Germany. The authors performed coordinated and cooperative research on nonlinear structural analysis and on computer-supported civil engineering over a period of several years. Many of the innovative aspects of this book were invented and developed in the course of the research effort. It is hoped that the results illustrate the potential of global interaction in university research.

The increasing size of the world population and the necessity to raise the standard of living in developing areas place new demands on the civil engineering profession. Since the available amount of material, energy and labour is limited, their use must be optimised. One of the strategies to achieve this goal is to reduce the amount of material that is required for structural systems. This strategy reduces the mass of the buildings as well as the energy that is required for their construction and for the production of building materials.

As the amount of material in a structure is reduced, the stiffness of the structure decreases and its deformation under load increases. The behaviour of light and slender structures cannot be analysed adequately with traditional linear theories of structural analysis. Instead, the displacements and forces must be predicted with nonlinear theories which take into account the influence of the displacements on the forces in the structure and on its stability.

Architects are designing an increasing number of light space enclosures. Wind, snow and temperature changes cause large deflections of these structures at lower load levels than in conventional structures. In particular, these space enclosures are more susceptible to buckling than heavier traditional structures. This behaviour places new demands on the reliable analysis of the stability of such structures. Deterioration of a structure in an aggressive city environment enhances the nonlinear phenomena and can trigger the collapse of the structure.

Nonlinear structural behaviour is considerably more complex than linear behaviour. For example, the principle of superposition does not hold in nonlinear analysis. While linear structural analysis is fairly standardised and therefore readily supported with computers, nonlinear analysis is not standardised because it is not a prominent part of the traditional education of civil engineers. It is difficult to evaluate the theoretical foundation of much of the commercial software for nonlinear analysis.

The authors aim to develop a generalised approach for the nonlinear analysis of structures containing all types of structural components. This book covers a first part of this topic: the geometrically nonlinear analysis of plane trusses and frames. The theory is derived by specialisation of the general theory of elasticity. This approach can be extended to other types of structural components. The numerical solution of the governing equations is treated as an initial value problem. A special theory is developed for the detection of singular configurations and for the continuation of load paths at singular points. The method of implementation of the theory by means of software with a suitable data base, algorithmic structure and graphical user interface is presented as an integral part of the contents.

1 STATE OF THE ART IN NONLINEAR STRUCTURAL ANALYSIS

1.1 NONLINEAR STRUCTURAL BEHAVIOUR

The nonlinear analysis of structures goes back to Leonard Euler (1707–1783). In the appendix "De curvis elasticis" to his *Methodus inveniendi*, Euler uses variation methods to determine the buckling load of a straight column. This stability problem could only be solved by considering equilibrium in the deformed state of the column. The equilibrium conditions in the deformed state contain the displacements of the structure as variables. In equilibrium equations, the displacements are multiplied with forces which depend on derivatives of the displacements. These products lead to governing equations which are nonlinear in the unknown displacements.

Many other sources of nonlinearity in the mechanical behaviour of solids and fluids were discovered in later times. The main nonlinear effects that are considered in the analysis of civil engineering structures today can be grouped as follows:

Geometric nonlinearity: The relationship between the strain and the displacement of a structure is nonlinear. The influence of large translations and rotations on the strains in structural components such as beams and plates is taken into account.

Large deformations: A slender structure can undergo large displacements and large rotations without experiencing large deformations. Such behaviour is observed in steel structures. Other structures are subjected to large deformations. This is, for example, the case during sheet metal forming processes. Rubber membrane structures are also subjected to large strains. These strains are taken into account in the formulation of the constitutive equations for the materials.

Material nonlinearity: The linear elastic constitutive laws of elementary structural analysis are approximations of the real material behaviour. A material such as concrete or rubber is elastic but nonlinear: the stresses are not proportional to the strains. A material can also become plastic if a limit strain is exceeded: a part of the strain remains after the stress is removed. Some materials undergo time-dependent deformations. There are specific constitutive models for elasto-plastic, visco-elastic and visco-plastic behaviour of different types of materials.

Stability: Consider an equilibrium configuration A of a loaded structure. Let the structure displace incrementally to a neighbouring equilibrium configuration B. If the structure can displace from configuration A to configuration B without a change in its loading, A is called a singular configuration of the structure. The equilibrium at A is unstable if a finite displacement leads to a reduction in the applied load. Instability can be due to geometric and to material nonlinearity. If the singularity at A is geometric, the determinant of the tangent stiffness matrix at A is null. If there is a single continuation of the load path at A to a neighbouring equilibrium configuration B, the instability is called a snap-through (turning point). If there is more than one possible continuation of the load path at A, the singularity is called a bifurcation. In the case of material instability, the material in a zone of the structure (for instance on a shear plane) is unable to resist the stresses which are required for the equilibrium of the structure, so that relative displacements occur between initially neighbouring material points of the zone.

Displacement-dependent loads: Consider a structure whose loading depends on its displacements. An inflatable membrane structure whose stability depends on the maintenance of an internal pressure in excess of the atmospheric pressure is an example of such structures. This type of loading is not conservative and causes non-linear behaviour.

Displacement-dependent supports: Consider a structure whose support conditions depend on its displacements. Typical examples of such supports are contact planes, where the stresses are null as long as there is a gap between the structure and its potential support. If the structure displaces so that it comes into contact with the support, the support becomes effective and stresses are built up in the contact plane. The change in support conditions due to contact causes nonlinear structural behaviour.

Displacement-dependent stiffness: Consider a bar or a membrane that is able to carry tensile stresses, but not compressive stresses. These structural elements are stressed if the displacements in the structure cause tensile strains, and stress free if the displacements cause compressive strains. The stiffness change of the elements due to their deformations causes nonlinear structural behaviour.

1.2 NONLINEAR STRUCTURAL ANALYSIS

The aims of the mathematical treatment of nonlinear structural phenomena are the prediction of the displacements, stresses and stability of a structure. It is assumed that the reference configuration and the history of the external influences, which act on the structure, are known. The mathematical formulations are not unique. They can, for instance, differ in the approximations introduced in the governing equations and in the numerical algorithms for the solution of the governing equations. The quality of a specific mathematical treatment of nonlinear behaviour is judged by the following criteria:

Accuracy: The behaviour of nonlinear structures that is predicted by mathematical analysis is an approximation of their real behaviour in nature. The difference between predicted and real behaviour is called the accuracy of the analysis. This accuracy depends on the formulation of the governing equations and on the methods used to solve them. The laws of physics that lead to the governing equations are verified by experiment. Exact solutions to the governing equations for nonlinear behaviour are scarce. For most analyses in civil engineering, the governing equations are therefore cast in algebraic form and solved numerically. The consequences of mathematical approximations, such as the intentional neglect of higher order terms and the errors in the numerical solution technique, can be studied theoretically. Mathematical treatment can provide tools for evaluating the accuracy of results, such as methods to determine the unbalanced forces in an instant configuration of the structure.

Reliability: Consider a structure whose deformations before bifurcation are small and do not announce the impending singularity. Let the structure remain stable on the primary branch of the continuation of the load path beyond the bifurcation, but become unstable on the secondary branches. The stability analysis is called reliable if it indicates the existence of the bifurcation and of all branches at the singularity.

Robustness: Small errors in the input data of a linear structural analysis usually lead to small errors in the predicted structural behaviour, unless the structure is unstable in nature. Such a method of analysis is called robust. Nonlinear methods of analysis can lack robustness. This is, for instance, the case if the results are sensitive to the step size of the numerical algorithm or if they depend on the trial functions as well as the method of integration used in the finite element formulation (for example bending and shear lock in nonlinear beam theory). The general use of nonlinear methods of analysis in civil engineering practice (where they are appropriate) depends on the availability of adequately robust algorithms.

Efficiency: The nonlinear analysis of a structure is an initial value problem. Stepwise solution techniques for such problems require that the incremental system equations are assembled and that their stiffness matrix is decomposed a significant number of times. The history of the behaviour is stored during the stepwise solution for a large number of configurations. Since the principle of superposition is not valid for nonlinear analysis, a separate analysis is required for each load combination. The total effort for a nonlinear analysis of a structure is therefore significantly larger than the effort for a linear analysis of the same structure. Under these conditions, algorithms that require low storage and processor capacities are highly desirable.

Clarity: Engineering intuition for nonlinear structural behaviour is considerably less developed than that for linear behaviour. This is not only due to deficiencies in engineering education and practice. The nonlinearity of the behaviour makes it more difficult to extrapolate from the known behaviour of a reference structure to the real behaviour of a similar structure. Good intuition for structural behaviour is, however, a prerequisite for good structural design. Intuition benefits from familiarity with the concepts and theories on which the nonlinear method of analysis is based. The effort required to become familiar with a theory depends to a significant extent on the clarity of the mathematical formulation of that theory. Complex derivations tend to make it more difficult to evaluate the influence of specific variables on the overall behaviour. The clarity with which a nonlinear theory is derived and with which the influence of various aspects of the formulation on the results of an analysis can be presented therefore influences the suitability of the theory for engineering practice.

Consistency: Modern structures are usually composed of many types of structural components. The method of nonlinear analysis for such structures must be capable of handling all these component types in a common load-bearing model. This is only possible if the theories for the different types of components, such as cables, beams, membranes, slabs, plates, folded plates and shells, are consistent generalisations of the common nonlinear theory of elasticity. These theories must be derived by a common general procedure using compatible behavioural hypotheses for the different types of components. The theories should be implemented on the computer with a common data base and a common engineering surface. Consistency of the mathematical formulations for different types of structural components is a prerequisite for the development of a general software package for nonlinear structural design.

1.3 GUIDES TO THE STATE OF THE ART

The practising engineer is responsible for his structural design. He cannot shift this responsibility to the authors of the software he employs in making the design. The engineer must therefore convince himself that the software he uses is appropriate for his design, and that the behaviour predicted by the software is adequately close to the real behaviour of the structure he is designing. In order to be able to perform this task, the engineer must familiarise himself with the state of the art of nonlinear structural analysis.

There are two guides to the state of the art in nonlinear structural analysis. One guide consists of papers and monographs published on the subject. An extract is presented in the list of references. The other guide consists of the software available for nonlinear structural analysis.

The publications present the concepts and mathematical derivations on which nonlinear structural analysis is based. The properties of the resulting methods of analysis are studied. The nonlinear behaviour of different types of structural components is illustrated with typical examples. This knowledge is essential for the construction of good software and for a proper understanding of nonlinear behaviour.

Since most of the solution techniques are numerical, the theoretical background by itself is not sufficient for the evaluation of the state of the art in nonlinear structural analysis. A theoretically correct approach may well lack the accuracy, reliability, robustness and efficiency required of nonlinear algorithms.

The accuracy, reliability, robustness and efficiency of a theory for nonlinear structural analysis are evaluated by means of the software in which the algorithm is implemented. The functionality of the software documents the scope of the nonlinear analyses that can be performed on the basis of the theory. The output of the software is used to study the characteristics of the solutions obtained with the theory, in particular their accuracy as well as the reliability, robustness and efficiency of the algorithms.

Unfortunately, the link between the available publications and the available software for nonlinear structural analysis is not always very explicit. In practice, it proves to be difficult to determine reliably which theories and algorithms are in fact implemented in the software. This situation severely hampers nonlinear analyses in civil engineering practice.

Literature: The literature on the nonlinear analysis of structures in civil engineering encompasses general publications on structural mechanics and specific publications on the finite element method for structural analysis. The number of publications is very large, and in many of them, much of the material is treated repeatedly with slight modifications.

The publications on structural mechanics can be grouped into topics related to the nonlinear theory of elastic bodies, to the constitutive models for different types of materials, to the theories for specific types of structural components (such as cables, beams, membranes, slabs, plates, folded plates and shells) and to the theory for the solution of nonlinear algebraic governing equations.

Monographs on structural analysis with finite elements usually contain chapters on the theory of elasticity, the structural mechanics of specific types of structural components and the matrix formulation of different kinds of finite elements. In addition, there are chapters on the constitutive laws of materials and the methods of numerical analysis of nonlinear governing equations. Special chapters may be devoted to structural stability, contact problems and nonlinear dynamic analysis.

Of particular interest is a publication in the *Journal of Aeronautical Sciences* by Turner, Clough, Martin and Topp [63] on "Stiffness and deflection analysis of complex structures" which dates back to 1956. This publication sparked a broad and well-financed research programme on the finite element method in the United States of America, which was strongly supported by both government and the aerospace industry. While this research was not dedicated specifically to nonlinear structural analysis, it did lead to significant advances in the area, which were summarised by Oden [42] in his monograph *Finite elements for nonlinear continua* in 1972.

Research on nonlinear structural analysis in Russia started in the 1940s. The first classification of nonlinear problems was presented by V. Novozhilov [40]. The nonlinear theory of elasticity was further developed in the fundamental work of I. Goldenblatt [17] and A. Lurie [36]. A. Ilyushin [22, 23] developed the theory of plasticity for small deformations. Physically nonlinear problems and related questions were first investigated in the works of G. Geniev [16] and A.A. Tolokonnikov [60].

Research in geometrically nonlinear structural behaviour was mostly concentrated on geometrically nonlinear problems of plates and shells. Best known are the works of V. Vlasov [64], A. Volmir [65], Kh. Mushtari and K. Galimov [39], V. Feodosiev [14], M. Kornishin [28], P.M. Ogibalov and M. Koltunov [43] as well as V. Petrov [47, 48]. Nonlinear behaviour of rod systems was investigated in the work of A. Gemmerling [15] and P. Lukash [35].

D. Davidenko [12, 13] introduced the parameter variation method for solving nonlinear equations that is now known as the parameter continuation method. V. Vlasov and V. Petrov [47] proposed a load step method that was applied to the analysis of plates and shells in the subsequent works of V. Petrov and his progeny [48]. This method has been used intensively for solving a wide range of problems in nonlinear mechanics.

Since the future significance of nonlinear structural analysis was clearly recognised in the 1970s, a number of ground-breaking publications appeared at the time. Well-known are papers by Wood and Zienkiewicz [70], on the geometrically nonlinear finite element analysis of beams, frames, arches and axisymmetric shells; Wunderlich and Beverungen [74], on the geometrically nonlinear theory for plane curved rods; Argyris, Dunne and Scharpf [2], on the large displacement-small strain analysis of structures; and Bathe and Bolourchi [4], on large displacement analysis of three-dimensional beam structures. The Europe-USA Workshop at Bochum [75] in 1980 and at Trondheim [6] in 1985 were devoted to nonlinear finite element analysis in structural mechanics and treated topics such as the general formulation of the theory of elasticity, geometrically nonlinear rod structures and shells, structures made of physically nonlinear materials, bifurcation, nonlinear dynamics, methods of numerical solution of nonlinear governing equations and numerical algorithms.

In 1981, the German Research Association (Deutsche Forschungsgemeinschaft DFG) established a priority programme (Schwerpunktprogramm) “Nonlinear computations in structural engineering” with E. Stein as speaker. Most of the German technical universities participated in the research projects. At the colloquium that concluded the focal programme in 1989, 72 authors presented 30 papers [58] dealing mainly with the geometrically and physically nonlinear behaviour of frames and thin-walled surface structures and the physically nonlinear behaviour of concrete structures.

Of particular interest for nonlinear stability analysis are the publications of the research group at Hannover University in the period 1988 to 1992. Basing their work on general mathematical publications by Jepson and Spence [24,57] on bifurcation and fold theory, Wagner, Wriggers and their co-workers [66,67,68,73] developed theories and algorithms for the determination of singular points and for the continuation of load paths beyond turning as well as bifurcation points.

In Russia, the stepwise and parameter continuation methods have been revised and classified by E. Grigolyuk and V. Shalashilin [19]. Their book contains an extensive list of references on the application of these methods to nonlinear problems of mechanics. V. Shalashilin and E. Kuznetsov [55] have further developed the parameter continuation method. Different forms of the parameter continuation method in combination with the finite element method were used in the works of A. Darkov and N. Shaposhnikov [11], V. Ananyan [1], F. Rekach [54], and R. Hechumov and A. Pokrovsky [20,51] for geometrically and physically nonlinear analysis of rod structures. Two books on nonlinear analysis [18,46] describing the main principles used in their commercial software products have recently been published by the scientific leaders of the main software developing companies, LIRA Soft and SCAD Soft.

In their monograph *Stability of the equilibrium of structures and related problems*, Perelmuter and Slivker [46] give a systematic presentation of the foundations of the mathematical analysis of structures consisting of straight rods, curvilinear plane rods, thin-walled rods and 3D-curvilinear bars. They present methods of solution for the governing equations and show the application of their methods and algorithms to problems in engineering practice.

From 1980 up to the present, a large number of papers and books have been published worldwide in which the theory and numerical algorithms for geometrically and physically nonlinear behaviour of many types of structural components are treated in great detail. The current state of the art is contained in a set of monographs of which the following are typical (see references):

Belytschko	<i>Nonlinear finite element analysis for continua and structures</i>
Bonet, Wood	<i>Nonlinear continuum mechanics for finite element analysis</i>
Crisfield	<i>Nonlinear finite element analysis of solids and structures</i>
Kleiber	<i>Incremental finite element modelling in nonlinear solid mechanics</i>
Kyther	<i>An introduction to linear and nonlinear finite element analysis</i>
Reddy	<i>An introduction to nonlinear finite element analysis</i>
Simo, Hughes	<i>Computational inelasticity</i>
Wriggers	<i>Nichtlineare Finite-Element-Methoden</i>
Zhong	<i>Finite element procedures for contact-impact problems</i>

Software: A large number of software packages with nonlinear structural analysis capabilities are available commercially. The list includes products such as ABACUS, ADINA, ALADDIN, ANSYS, COSMOS, DYNA, FRAME3D, GT STRUDL, LIRA, LUSAS, MSC.Marc, NASTRAN, NISA, SAP2000.Nonlinear, SCAD and SOFISTIK. The nonlinear analysis functions are frequently embedded in more general packages that also offer linear and dynamic analysis as well as extensive capabilities for the computer-aided specification of structural models and for the presentation of computed structural behaviour.

Application of the finite element method to nonlinear analysis led to the development of software products with nonlinear functions in Russia and the Ukraine. These include MicroFE, ProFET&STARK, LIRA and SCAD. Nevertheless the leading software developers in Russia acknowledge that the issue of reliable nonlinear structural analysis is still unresolved [45].

Commercial software packages for nonlinear structural analysis contain user manuals. These manuals explain the steps of nonlinear analysis, but generally assume that the user is familiar with the theory on which the analysis is based. This assumption is too optimistic. Nonlinear structural analysis is far less standardised than linear analysis, where the assumption that the engineer is familiar with the theory is usually justified. In addition, many software packages have individual properties that are sometimes due to their academic origins. It can be difficult to understand these peculiarities if the theory is not completely documented in the user manuals, and if the vendor does not guarantee full agreement between the described theory and the software implementation.

It is well known that unsuitable numerical model and solution parameters can lead to inappropriate predictions of nonlinear structural behaviour. The user is unable to choose suitable parameter values if the information necessary to guide this choice is not made available. Black box software cannot be tuned appropriately for reliable analyses, for which the practising engineer must accept responsibility. The testing of commercial software in order to establish its properties has therefore become a research topic. This approach cannot replace adequate documentation of the theory and verification of the properties of the software by its authors.

1.4 STATE OF THE ART IN THE FORMULATION OF THE THEORY

Nonlinear theory of elasticity: The scientific fundamentals of nonlinear structural analysis are well developed. It will be shown in chapter 3 that the continuum mechanics of elastic bodies are readily formulated to yield a three-dimensional theory of elasticity. The kinematics of the body is described by defining a state of displacement and a state of deformation of the body and establishing their relationship. The dynamics of the body is described by defining a stress vector on a cutting plane with a specified normal vector and a state of stress, and establishing their relationship. The governing equations are set up by formulating the equilibrium conditions for the instant configuration of the structure. The boundary conditions are specified in local coordinate spaces at the points on the surface of the body.

Since the number of unknowns in the equations of kinematics and statics exceeds the number of equations, constitutive equations are established for different models of material behaviour. These relate the stresses to the strains in the body. The total number of equations now equals the number of unknowns, so that the governing equations can be solved with suitable boundary conditions for the unknown stresses and displacements.

The formulation of the three-dimensional theory of elasticity includes the material-oriented approach of Lagrange as well as the space-oriented approach of Euler. The equilibrium equations are formulated with the approach of Euler by considering volume elements of the instant configuration of the loaded body whose edges are parallel to the coordinate axes. Since the displacements of the material points in the instant configuration are not known, the equilibrium equations must be transformed to the elements whose edges are parallel to the coordinate axes in the known reference configuration of the body. This transformation requires the definition of additional stress tensors and the derivation of relationships between the infinitesimal volume and area elements which a set of material points occupy in the reference and in the instant configuration of the body.

The differential form of the governing equations of three-dimensional elasticity is not suited for the finite element method. The weighted residual method is therefore used to obtain an integral form of the equations (weak equilibrium conditions). The most useful form for numerical treatment is obtained by means of partial integration.

The stepwise solution of the governing equations in the numerical algorithm requires their linearisation at the instant configuration. Experience shows that a good rate of convergence of the numerical algorithm strongly depends on consistent linearisation of the equations. The method of linearisation is a central feature of the nonlinear theory and is described below. It leads to the incremental governing equations that form the basis for the algebraic governing equations of the stepwise solution.

The analysis of the stability of a structure is not a part of the basic nonlinear theory of elasticity. The determination of singular configurations and the continuation of load paths beyond these configurations require additional concepts and formulations, which are introduced in later sections of this chapter.

Consistent linearisation: Consider a vector function $\mathbf{f}(\mathbf{u})$ of a variable vector \mathbf{u} . Let the vector \mathbf{u} be a function of a parameter ε for constant vectors \mathbf{u}_0 and $\Delta\mathbf{u}$:

$$\mathbf{u} = \mathbf{u}_0 + \varepsilon\Delta\mathbf{u} \quad (1.1)$$

$$\mathbf{f}(\varepsilon) = \mathbf{f}(\mathbf{u}_0 + \varepsilon\Delta\mathbf{u}) \quad (1.2)$$

The Taylor series expansion of function \mathbf{f} about the point $\varepsilon = 0$ is:

$$\mathbf{f}(\varepsilon) = [\mathbf{f}]_{\varepsilon=0} + \varepsilon \left[\frac{d\mathbf{f}}{d\varepsilon} \right]_{\varepsilon=0} + \frac{1}{2} \varepsilon^2 \left[\frac{d^2\mathbf{f}}{d\varepsilon^2} \right]_{\varepsilon=0} + \dots \quad (1.3)$$

This series is truncated after the linear term in ε and ε is set equal to 1:

$$\mathbf{f}(\mathbf{u}_0 + \Delta\mathbf{u}) = \mathbf{f}(\mathbf{u}_0) + \left[\frac{d\mathbf{f}(\mathbf{u}_0 + \varepsilon\Delta\mathbf{u})}{d\varepsilon} \right]_{\varepsilon=0} \quad (1.4)$$

The derivative of \mathbf{f} with respect to ε contains $\Delta\mathbf{u}$ as a factor. The equation therefore yields the following linearisation of the vector function $\mathbf{f}(\mathbf{u})$:

$$\mathbf{f}(\mathbf{u}_0 + \Delta\mathbf{u}) = \mathbf{f}(\mathbf{u}_0) + \mathbf{K}_T(\mathbf{u}_0) \Delta\mathbf{u} \quad (1.5)$$

$\mathbf{K}_T(\mathbf{u}_0)$ matrix of the partial derivatives of function \mathbf{f} with respect to u_i for $\varepsilon = 0$

Example: Assume that governing equation (2.20) for the 2-bar truss in chapter 2 is to be linearised. Since there is a single variable w_2 , a scalar function $f(u)$ is defined:

$$f(u) = q - u(1+u)(2+u)$$

$$u := w_2$$

$$q := p$$

The value $f(\varepsilon)$ of the function at $u = u_0 + \varepsilon\Delta u$ is given by:

$$f(\varepsilon) = f(u_0 + \varepsilon\Delta u) = q - (u_0 + \varepsilon\Delta u)(1+u_0 + \varepsilon\Delta u)(2+u_0 + \varepsilon\Delta u)$$

The function $f(\varepsilon)$ is expanded in a Taylor series at $u = u_0$ and truncated:

$$f(\varepsilon) = f(u_0) + \varepsilon \left\{ \Delta u(1+u_0 + \varepsilon\Delta u)(2+u_0 + \varepsilon\Delta u) + \right. \\ \left. \Delta u(u_0 + \varepsilon\Delta u)(2+u_0 + \varepsilon\Delta u) + \right. \\ \left. \Delta u(u_0 + \varepsilon\Delta u)(1+u_0 + \varepsilon\Delta u) \right\}_{\varepsilon=0}$$

The linearisation of (2.29) at u_0 for $\varepsilon = 1$ is thus:

$$f(u_0 + \varepsilon\Delta u) = f(u_0) + \Delta u(2+6u_0+3u_0^2) \quad (1.6)$$

Nonlinear theories for structural components: Most structural components have geometric properties which significantly influence their mechanical behaviour:

– Since the dimensions of the cross-section of a beam are much smaller than the length of the axis of the beam, it is reasonable to assume that the cross-section remains plane and normal to the beam axis under load.

– Since the thickness of a plate is much smaller than the dimensions of its middle surface, it is reasonable to assume that a fibre which is straight and normal to the middle surface remains straight and normal to the middle surface under load.

It is usually advantageous for both the structural analysis of these components and for the engineering interpretation of their behaviour if these mechanical assumptions are utilised to develop a one- or two-dimensional theory for the component type.

The development of special theories for component types is a well-known approach in linear structural analysis. Linear theory also shows some of the difficulties that can be associated with this approach. Since the equations of elasticity are sufficient to solve for the displacements and stresses in a loaded body, the additional equations introduced by the hypotheses for component behaviour destroy the uniqueness of the solution and lead to contradictions. This is, for instance, well known for the determination of the shear forces in beams and for the formulation of the boundary conditions of plates. Similar phenomena are observed in the nonlinear theory for structural components.

A considerable amount of research has been devoted to the development of nonlinear theories for structural components such as cables, beams, membranes, slabs, plates, folded plates and shells. The theories are well documented in the cited monographs. In the current literature, the derivations are usually based on the three-dimensional nonlinear theory of elasticity, which is specialised for the component type by means of hypotheses on their mechanical behaviour. The trial functions for finite elements and the theoretical basis for the determination of stiffness matrices as well as load vectors are well documented.

In spite of these advances in the theory of nonlinear analysis, significant difficulties remain. A typical example is membrane locking in beams. The axial strain in a beam that undergoes large deflections depends on the first derivative $u_{1,1}$ of the axial displacement u_1 and on the square $u_{2,1}^2$ of the first derivative of the transverse displacement u_2 of the points on the axis of the beam. If the boundary conditions on the beam are such that the axial stress must be zero in order to satisfy the equilibrium conditions, the variation of $u_{1,1}$ on the axis of the beam must be proportional to the variation of $u_{2,1}^2$. In finite element formulations, u_1 is usually assumed to vary linearly whereas u_2 is assumed to vary cubically along the axis of the beam. The required proportionality of $u_{1,1}$ and $u_{2,1}^2$ is therefore not achieved. This incompatibility of the trial functions leads to beam elements that are exceedingly stiff in the axial direction and thus to inappropriate predictions of structural behaviour. In the software codes, the effect is reduced empirically by making $u_{2,1}$ constant on the axis in contradiction to the assumed displacement trial functions. Similar corrections are required for plate bending elements to alleviate shear locking, for instance through selective reduced integration.

1.5 STATE OF THE ART IN NUMERICAL SOLUTION TECHNIQUES

The equations that describe the equilibrium of the structure in a specific instant configuration are nonlinear. The solution of a set of nonlinear algebraic equations is a mathematical problem that has been investigated extensively. Initial research on nonlinear structural analysis used the classical mathematical methods of solution. These will now be summarised. It will then be shown how the classical methods were modified in order to make them adequate for nonlinear structural analysis.

1.5.1 Test Case

The numerical solution techniques for the nonlinear governing equations of structures are illustrated with the governing equation for a 2-bar truss, which is derived in section 2.1. By replacing the normalised displacement w_3 with u and the normalised load p_3 with q , the governing equation (2.20) is transformed to a simplified notation that will be used in this section. The transformed equation is:

$$u(1+u)(2+u) = q \quad (1.7)$$

The left-hand side of equation (1.7) is treated as the product of a displacement u with a stiffness $k(u)$ given by

$$k(u) = (1+u)(2+u) \quad (1.8)$$

The residual r in the governing equation for a test value $u^{(i)}$ of the displacement is defined as

$$r^{(i)} = q - u^{(i)}(1+u^{(i)})(2+u^{(i)}) \quad (1.9)$$

The tangent stiffness k_t of the truss at displacement u is obtained by differentiation of equation (1.7):

$$\begin{aligned} k_t \Delta u &= \Delta q \\ k_t &= 3u^2 + 6u + 2 \end{aligned} \quad (1.10)$$

The displacement u_{\min} corresponding to the minimum algebraic value q_{\min} of the load is obtained by setting the tangent stiffness in (1.10) to null. The value of u_{\min} is substituted into (1.7) to compute q_{\min} :

$$u_{\min} = -0.422650 \quad q_{\min} = -0.384900 \quad (1.11)$$

The solutions in the examples presented in the following sections are obtained for the pattern load $q_t = q_{\min}$.

1.5.2 Direct Iteration Method

Consider a nonlinear system of algebraic equations with a coefficient matrix \mathbf{K} , an absolute term \mathbf{q} and a solution vector \mathbf{u} . Let the system be nonlinear because the coefficients in \mathbf{K} are functions of \mathbf{u} .

$$\mathbf{K}(\mathbf{u}) \mathbf{u} = \mathbf{q} \quad (1.12)$$

An estimate $\mathbf{u}^{(0)}$ of the solution vector is determined by inspection, the matrix $\mathbf{K}(\mathbf{u}^{(0)})$ is computed and (1.12) is solved to obtain an iterated solution vector $\mathbf{u}^{(1)}$:

$$\mathbf{u}^{(1)} = (\mathbf{K}(\mathbf{u}^{(0)}))^{-1} \mathbf{q} \quad (1.13)$$

The iterated solution vector $\mathbf{u}^{(1)}$ is used to compute a new coefficient matrix $\mathbf{K}(\mathbf{u}^{(1)})$ and a new solution vector $\mathbf{u}^{(2)}$. The iterated solution $\mathbf{u}^{(s)}$ in step s of the iteration is obtained with equation (1.14):

$$\mathbf{u}^{(s)} = (\mathbf{K}(\mathbf{u}^{(s-1)}))^{-1} \mathbf{q} \quad (1.14)$$

The iteration is terminated if the following convergence criterion is satisfied:

$$\frac{(\mathbf{u}^{(s)} - \mathbf{u}^{(s-1)})^T (\mathbf{u}^{(s)} - \mathbf{u}^{(s-1)})}{\mathbf{u}^{(s)T} \mathbf{u}^{(s)}} < \delta^2 \quad (1.15)$$

δ limit of convergence

Example: Iteration for the test case starts with the initial displacement $u^{(0)} = 0$. The stiffness $k(u^{(0)})$ for the first step of iteration is determined with (1.8), the iterated displacement $u^{(1)}$ with (1.13) and the load for the iterated displacement with (1.7):

$$\begin{aligned} \text{step 0: } \quad k(u^{(0)}) &= (1+u^{(0)})(2+u^{(0)}) = 2.000000 \\ u^{(1)} &= \frac{q_i}{k(u^{(0)})} = \frac{-0.384900}{2.000000} = -0.192450 \\ q^{(1)} &= u^{(1)}(1+u^{(1)})(2+u^{(1)}) = -0.280917 \end{aligned}$$

The second step of the iteration starts with the displacement $u^{(1)}$:

$$\begin{aligned} \text{step 1: } \quad k(u^{(1)}) &= (1+u^{(1)})(2+u^{(1)}) = 1.459687 \\ u^{(2)} &= \frac{q_i}{k(u^{(1)})} = \frac{-0.384900}{1.459687} = -0.263687 \\ q^{(2)} &= u^{(2)}(1+u^{(2)})(2+u^{(2)}) = -0.337116 \end{aligned}$$

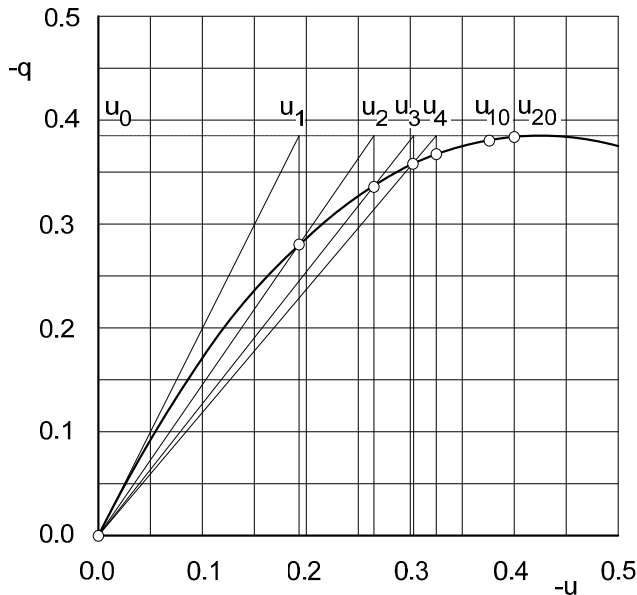


Figure 1.1 Solution of the test case by direct iteration

The procedure is repeated in the following steps and leads to the iterated displacements $u^{(3)}, u^{(4)}, u^{(10)}$ and $u^{(20)}$ in figure 1.1. The results show that the rate of convergence is high in the first few steps, but becomes low as the solution $u = -0.422650$ is approached. Fifty steps of iteration yield $u^{(50)} = -0.412627$, while one hundred steps lead to $u^{(100)} = -0.417514$. This displacement still deviates 1.2 per cent from the exact solution.

1.5.3 Newton-Raphson Iteration Method

Consider the linearised form (1.5) of a vector function $\mathbf{f}(\mathbf{u})$. A solution $\Delta\mathbf{u}$ of the equation $\mathbf{f}(\mathbf{u}) = \mathbf{0}$ is found by setting $\mathbf{f}(\mathbf{u}_0 + \Delta\mathbf{u})$ to $\mathbf{0}$:

$$\mathbf{K}_t(\mathbf{u}^{(0)})\Delta\mathbf{u} + \mathbf{f}(\mathbf{u}^{(0)}) = \mathbf{0} \quad (1.16)$$

The term $\mathbf{f}(\mathbf{u}^{(0)})$ is the residual vector $\mathbf{r}^{(0)}$ in the governing equations at point $\mathbf{u}^{(0)}$:

$$\mathbf{f}(\mathbf{u}^{(0)}) := \mathbf{r}^{(0)} = \mathbf{q}_t - \mathbf{K}(\mathbf{u}^{(0)})\mathbf{u}^{(0)} \quad (1.17)$$

At a given point $\mathbf{u}^{(0)}$, the residual vector $\mathbf{r}^{(0)}$ is computed with (1.17) and the linear equations (1.16) are solved for the displacement increment $\Delta\mathbf{u}^{(0)}$. The next cycle of iteration is performed at point $\mathbf{u}^{(1)} = \mathbf{u}^{(0)} + \Delta\mathbf{u}^{(0)}$. The iteration is terminated when the convergence criterion (1.15) is satisfied.

Example: The linearised form (1.16) of the governing equation (1.7) for the test case is given by equation (1.10). The iteration starts with the initial displacement $u^{(0)} = 0$. The residual $r^{(0)}$ is computed with equation (1.9), the tangent stiffness k_t with (1.10) and the displacement increment with (1.16):

$$\begin{aligned} \text{step 0:} \quad r^{(0)} &= q_t - u^{(0)}(1+u^{(0)})(2+u^{(0)}) = -0.384900 \\ k_t^{(0)} &= 3(u^{(0)})^2 + 6u^{(0)} + 2 = 2.000000 \\ \Delta u^{(0)} &= \frac{r^{(0)}}{k_t} = \frac{-0.384900}{2.000000} = -0.192450 \\ u^{(1)} &= u^{(0)} + \Delta u^{(0)} = -0.192450 \\ q^{(1)} &= u^{(1)}(1+u^{(1)})(2+u^{(1)}) = -0.280917 \end{aligned}$$

The second step of the iteration starts with the displacement $u^{(1)}$:

$$\begin{aligned} \text{step 1:} \quad r^{(1)} &= q_t - u^{(1)}(1+u^{(1)})(2+u^{(1)}) = -0.103983 \\ k_t^{(1)} &= 3(u^{(1)})^2 + 6u^{(1)} + 2 = 0.956411 \\ \Delta u^{(1)} &= \frac{r^{(1)}}{k_t} = \frac{-0.103983}{0.956411} = -0.108722 \\ u^{(2)} &= u^{(1)} + \Delta u^{(1)} = -0.301172 \\ q^{(2)} &= u^{(2)}(1+u^{(2)})(2+u^{(2)}) = -0.357548 \end{aligned}$$

Figure 1.2 shows that the rate of convergence of the Newton-Raphson method is faster than that of the direct iteration method. The displacement after five steps of iteration equals that of the direct method after twenty steps of iteration.

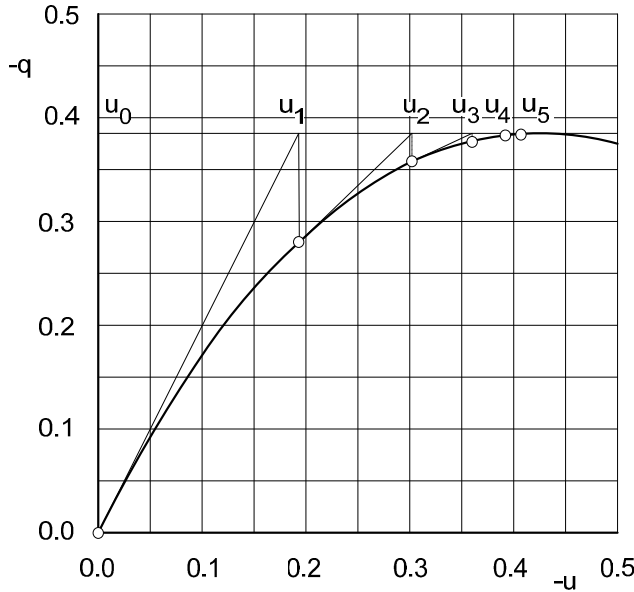


Figure 1.2 Solution of the test case by Newton-Raphson iteration

1.5.4 Modified Newton-Raphson Iteration Method

Each cycle of the original Newton-Raphson method requires the assembly and decomposition of a new incremental stiffness matrix \mathbf{K}_t . In the modified Newton-Raphson method, the matrix \mathbf{K}_t is kept constant for several cycles of iteration. Only the residual vector \mathbf{r} in (1.17) is recomputed in each cycle, so that the computational effort per cycle of iteration is reduced. The number of iteration cycles increases since the rate of convergence of the iteration decreases due to the approximation of \mathbf{K}_t .

Example: Consider the linearised form (1.10) of the governing equation (1.7) for the test case. The iteration starts with the initial displacement $u^{(0)} = 0$. The residual $r^{(0)}$ is computed with (1.9) and the displacement increment with (1.16). The tangent stiffness k_t is kept constant at the value 2.0 for the initial displacement 0.0. The first step of the iteration is identical to the original iteration in the example in section 1.5.3. The second step is modified as follows:

$$\begin{aligned}
 \text{step 1:} \quad r^{(1)} &= q - u^{(1)}(1 + u^{(1)})(2 + u^{(1)}) = -0.103983 \\
 k_t^{(1)} &= 3(u^{(0)})^2 + 6u^{(0)} + 2 = 2.000000 \\
 \Delta u^{(1)} &= \frac{r}{k_T} = \frac{-0.103983}{2.000000} = -0.056991 \\
 u^{(2)} &= u^{(1)} + \Delta u^{(1)} = -0.244442 \\
 q^{(2)} &= u^{(2)}(1 + u^{(2)})(2 + u^{(2)}) = -0.324234
 \end{aligned}$$

Figure 1.3 shows that the rate of convergence of the modified Newton-Raphson method is lower than that of the original method. After five cycles of iteration, the modified method yields $u = -0.3104$, whereas the original method yields $u = -0.4066$ and the solution is $u = -0.4226$.

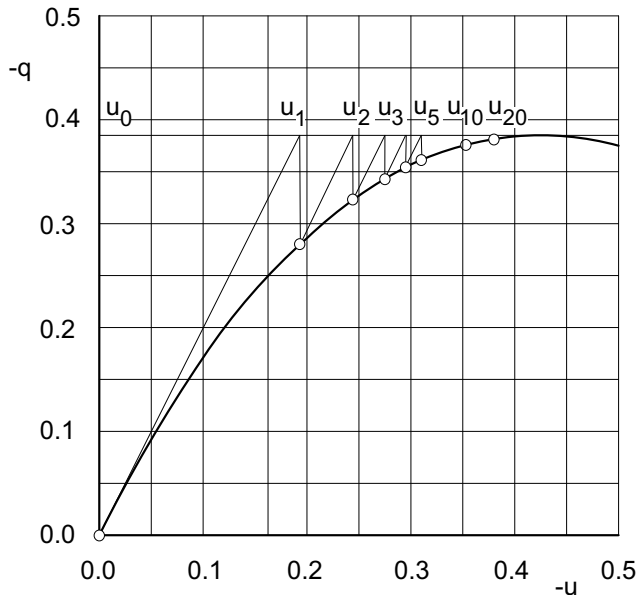


Figure 1.3 Solution of the test case by modified Newton-Raphson iteration

1.5.5 Stepwise Solution with the Basic Arc Increment Method

The state of displacement of a slender structure for a given load is not always unique. The 2-bar truss in section 2.1, for example, has three equilibrium configurations for the same loading.

If there is more than one instant configuration in which the structure is in equilibrium for a given loading, the start vector of the iteration methods in sections 1.5.2 to 1.5.4 determines which of these configurations is determined by the algorithm. A better insight into the behaviour of the structure is obtained if the analysis is formulated as an initial value problem. The complete load-deflection curve is determined so that the dependence of the states of the truss on the load history is correctly taken into account. This approach is described in detail in chapter 6.

In the stepwise algorithms for nonlinear structural analysis, the reference state of the structure is known. The load is applied in increments. The governing equations are linearised in each load step. The state of displacement at the end of a load step is estimated by solving linearised incremental governing equations.

The stiffness of a structure can change significantly along the load path. The a-priori specification of a constant magnitude of the load increments for the load steps is therefore not always a suitable approach. A load increment that leads to small displacements if it is applied to the reference configuration can cause very large displacements or even singularity if it is applied when the structure is nearly singular. In the case of the 2-bar truss in section 2.1, there are parts of the load path where the load decreases while the displacement increases.

Basic arc increment method: Experience shows that it is advantageous to control the nonlinear analysis algorithms by choosing equal arc length increments for all load steps in the load-displacement diagram. If the structure is stiff, the arc length increment is dominated by the load increment. If the structure is nearly singular and very flexible, the arc length increment is dominated by the displacement increment. This approach is called the basic arc increment method.

Example: Consider the linearised form (1.10) of the governing equation (1.7) for the test case. Let the arc length increment be $a = 0.1$. Express the load increment in a step of the analysis as $\Delta\lambda q_t$, where $\Delta\lambda$ is the load factor increment in the load step. The displacement increment in the step is:

$$\Delta u = \frac{\Delta\lambda q_t}{k_t} = \Delta\lambda u_t \quad \text{with} \quad u_t = \frac{q_t}{k_t}$$

The load factor increment is determined so that the arc length increment equals the specified value $a = 0.1$:

$$\Delta u^2 + \Delta q^2 = a^2$$

$$\Delta\lambda^2 (u_t^2 + q_t^2) = a^2$$

$$\Delta\lambda = \frac{a}{\sqrt{u_t^2 + q_t^2}}$$

$$\begin{aligned} \text{step 0:} \quad k_t^{(0)} &= 3(u^{(0)})^2 + 6u^{(0)} + 2 = 2.000000 \\ u_t^{(0)} &= -\frac{0.384900}{2.000000} = -0.192450 \\ \Delta\lambda^{(0)} &= \frac{0.100000}{\sqrt{0.192450^2 + 0.384900^2}} = 0.232379 \end{aligned}$$

The displacement and load at the end of step 0 are:

$$\begin{aligned} \Delta u^{(0)} &= \Delta\lambda^{(0)} u_t^{(0)} = -0.044721 & u^{(1)} &= u^{(0)} + \Delta u^{(0)} = -0.044721 \\ \Delta q^{(0)} &= \Delta\lambda^{(0)} q_t^{(0)} = -0.089443 & q^{(1)} &= q^{(0)} + \Delta q^{(0)} = -0.089443 \end{aligned}$$

$$\begin{aligned} \text{step 1:} \quad k_t^{(1)} &= 3(u^{(1)})^2 + 6u^{(1)} + 2 = 1.737672 \\ u_t^{(1)} &= -\frac{0.384900}{1.737674} = -0.221503 \\ \Delta\lambda^{(1)} &= \frac{0.100000}{\sqrt{0.221503^2 + 0.384900^2}} = 0.225182 \end{aligned}$$

The displacement and load at the end of step 1 are:

$$\begin{aligned} \Delta u^{(1)} &= \Delta\lambda^{(1)} u_t^{(1)} = -0.049879 & u^{(2)} &= u^{(1)} + \Delta u^{(1)} = -0.094600 \\ \Delta q^{(1)} &= \Delta\lambda^{(1)} q_t^{(1)} = -0.086673 & q^{(2)} &= q^{(1)} + \Delta q^{(1)} = -0.176115 \end{aligned}$$

The procedure is repeated until the points in figure 1.4 have been computed. The diagram shows that the solution deviates significantly from the correct solution. This effect is due to the linearisation of the governing equations in the steps of the procedure. It will be shown in section 1.5.6 that the deviation can be removed by refinement of the basic arc increment method.

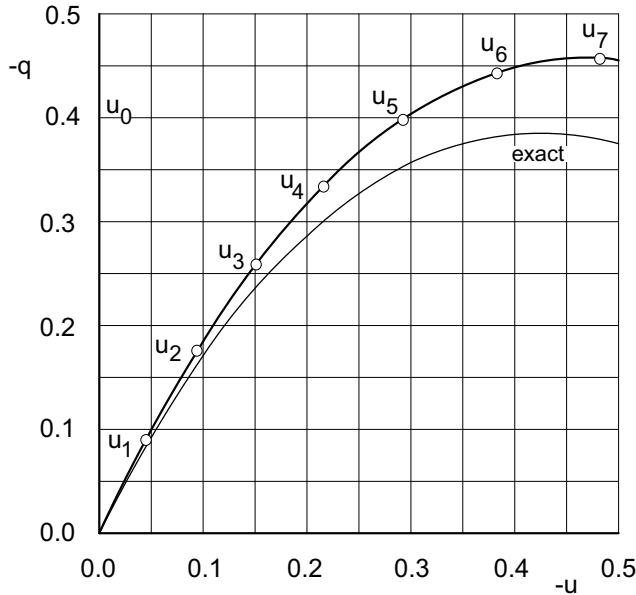


Figure 1.4 Solution of the test case with the basic arc increment method

1.5.6 Refined Arc Increment Method

The errors in the solution of the nonlinear governing equations with the basic arc increment method are primarily due to linearisation. The displacements and loads computed at the end of a load step with the linearised governing equations (as in the example of figure 1.4) are therefore treated as a trial solution in the refined arc increment method. The unbalanced forces of the trial solution are computed with the full governing equations of the nonlinear theory. These unbalanced forces are treated as loads in the next cycle of iteration for the same load step, which leads to a new trial state. The value of variable x in cycle m of the iteration in step s is denoted by $x^{(s,m)}$. The process is repeated until the unbalanced forces are sufficiently small. The analysis is then continued with the next load increment.

Example: Consider the linearised form (1.10) of the governing equation (1.7) for the test case treated in the previous example. Iteration 0 of step 0 is the same as step 0 for the basic arc increment method. The trial displacement at the end of iteration 0 is $\hat{u}^{(1,0)} = -0.044721$. In iteration 1, the unbalanced force r for the trial displacement is computed and applied as a load on the truss. The displacement correction due to this load is computed with the stiffness $k_t^{(0)} = 2.0$ at the beginning of the load step:

step 0: iteration 1

$$r^{(0,1)} = q^{(1)} - \hat{u}^{(1,0)}(1 + \hat{u}^{(1,0)})(2 + \hat{u}^{(1,0)}) = -0.005911$$

$$\Delta \hat{u}^{(0,1)} = \frac{r^{(0,1)}}{k_t^{(0)}} = \frac{-0.005911}{2.000000} = -0.002951$$

$$\hat{u}^{(1,1)} = \hat{u}^{(1,0)} + \Delta \hat{u}^{(0,1)} = -0.047672$$

The unbalanced force for iteration 2 in step 0 is computed with the corrected trial displacement $\hat{u}^{(1,1)}$:

$$\begin{aligned} \text{step 0:} \quad & \text{iteration 2} \\ r^{(0,2)} &= q^{(1)} - \hat{u}^{(1,1)}(1 + \hat{u}^{(1,1)})(2 + \hat{u}^{(1,1)}) = -0.000809 \\ \Delta \hat{u}^{(0,2)} &= \frac{r}{k_t^{(0)}} = \frac{-0.000809}{2.000000} = -0.000404 \\ \hat{u}^{(1,2)} &= \hat{u}^{(1,1)} + \Delta \hat{u}^{(0,2)} = -0.048077 \end{aligned}$$

The iteration in step 0 is terminated when the absolute value of the rest r is less than 0.000001. The corresponding displacement is $u^{(1)} = -0.048142$. The first iteration in load step 1 is performed with the initial displacement $u^{(1)}$:

$$\begin{aligned} \text{step 1:} \quad k_t &= 3(u^{(1)})^2 + 6u^{(1)} + 2 = 1.718101 \\ u_t^{(1)} &= -\frac{0.384900}{1.718101} = -0.224026 \\ \Delta \lambda^{(1)} &= \frac{0.100000}{\sqrt{0.224026^2 + 0.384900^2}} = 0.224543 \\ \Delta \hat{u}^{(1,0)} &= \Delta \lambda^{(1)} u_t^{(1)} = -0.050304 \quad \hat{u}^{(2,0)} = u^{(1)} + \Delta \hat{u}^{(1,0)} = -0.098446 \\ \Delta q^{(1)} &= \Delta \lambda^{(1)} q_t = -0.086427 \quad q^{(2)} = q^{(1)} + \Delta q^{(1)} = -0.175869 \end{aligned}$$

The trial displacement $\hat{u}^{(2,0)}$ is corrected by iteration similar to the iteration in step 1. The procedure is repeated for load steps 3 and 4. In load step 5, the iteration diverges because the arc length increment increases significantly. The results are shown in figure 1.5. The trial displacement after ten iterations in load step 5 is marked with a square.

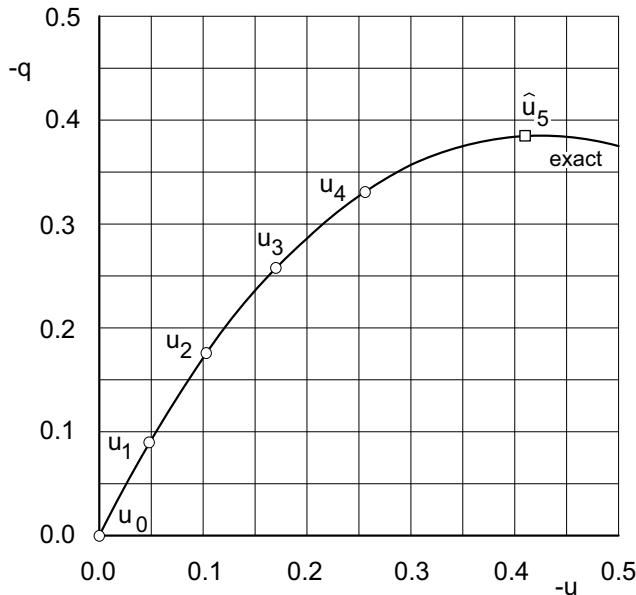


Figure 1.5 Solution of the test case with the refined arc increment method

1.5.7 Constant Arc Increment Method

The refined arc method is now modified in such a way that the arc length increment after iteration is the same in all load steps of the procedure. For this purpose, the load factor increment is recomputed in each cycle of the iteration of a load step.

The load $q^{(s)}$ and the displacement $u^{(s)}$ at the beginning of load step s are known. The tangent stiffness $k_t^{(s)}$ and the pattern displacement $u_t^{(s)}$ due to the pattern load q_t are computed. The load factor increment, the trial displacement and the load at the end of the first iteration of step s are computed as for the refined arc method:

$$\begin{aligned}\Delta\hat{\lambda}^{(s,0)} &= \frac{a}{\sqrt{(u_t^{(s)})^2 + q_t^2}} \\ \Delta\hat{u}^{(s,0)} &= \Delta\hat{\lambda}^{(s,0)} u_t^{(s)} & \hat{u}^{(s+1,0)} &= u^{(s)} + \Delta\hat{u}^{(s,0)} \\ \Delta\hat{q}^{(s,0)} &= \Delta\hat{\lambda}^{(s,0)} q_t & \hat{q}^{(s+1,0)} &= q^{(s)} + \Delta\hat{q}^{(s,0)}\end{aligned}$$

At the beginning of iteration 1 in step s , the corrective load q_c is set equal to the rest term in the trial state:

$$q_c^{(s)} = \hat{q}^{(s+1,0)} - \hat{u}^{(s+1,0)}(1 + \hat{u}^{(s+1,0)})(2 + \hat{u}^{(s+1,0)})$$

The load factor increment is recomputed so that the arc length increment is a . The corrective displacement due to q_c is denoted by u_c and determined with the tangent stiffness. The indices denoting step and iteration are dropped to improve the readability of the equation:

$$\begin{aligned}(\Delta\lambda u_t + u_c)^2 + (\Delta\lambda q_t)^2 &= a^2 \\ (u_t^2 + q_t^2)\Delta\lambda^2 + 2u_t u_c \Delta\lambda + (u_c^2 - a^2) &= 0\end{aligned}\tag{1.18}$$

The quadratic equation (1.18) is solved for $\Delta\lambda$. The trial values Δu and Δq of the load and the displacement increments in the load step as well as the displacement, load and unbalanced force in the trial state are computed. If the unbalanced force is larger than a specified limit, its value is added to q_c and the iteration is repeated. If the unbalanced force does not exceed the specified limit, the iteration is terminated and the procedure is continued with the next load step, using the computed displacement as the new initial displacement.

Example: Consider the linearised form (1.10) of the governing equation (1.7) for the test case. Let the arc length increment be $a = 0.1$, the initial displacement $u^{(0)} = 0.0$ and the initial load $q^{(0)} = 0.0$. The trial displacement and trial load at the end of iteration 0 in step 0 are computed:

step 0: iteration 0

$$\begin{aligned}k_t^{(0)} &= 3(u^{(0)})^2 + 6u^{(0)} + 2 = 2.000000 \\ u_t^{(0)} &= \frac{q_t}{k_t^{(0)}} = \frac{-0.384900}{2.000000} = -0.192450 \\ \Delta\hat{\lambda}^{(0,0)} &= \frac{a}{\sqrt{(0.19245)^2 + (0.384900)^2}} = 0.232379 \\ \hat{u}^{(1,0)} &= \Delta\hat{\lambda}^{(0,0)} u_t^{(0)} = -0.0447214 \\ \hat{q}^{(1,0)} &= \Delta\hat{\lambda}^{(0,0)} q_t = -0.0894427\end{aligned}$$

The corrective load q_c is set equal to the unbalanced force r in the trial state and the corrective displacement u_c is computed with the tangent stiffness:

step 0: iteration 1

$$r = \hat{q}^{(1)} - \hat{u}^{(1)}(1.0 - \hat{u}^{(1)})(2 - \hat{u}^{(1)}) = -0.0059106 := q_c^{(1)}$$

$$u_c^{(0,1)} = \frac{q_c}{k_t^{(0)}} = -0.002955$$

$$(u_t^2 + q_t^2)\Delta\lambda^2 + 2u_c u_t \Delta\lambda + (u_c^2 - a^2) = 0$$

$$0.1851\Delta\lambda^2 + 0.0113749\Delta\lambda - 0.00999132 = 0$$

$$\Delta\hat{\lambda}^{(0,1)} = 0.229227$$

$$\hat{u}^{(1)} = u_c^{(0,1)} + \Delta\hat{\lambda}^{(0,1)} u_t^{(0)} = -0.0470700$$

$$\hat{q}^{(1)} = \Delta\hat{\lambda}^{(0,1)} q_t = -0.0882294$$

In iteration 2 of load step 0, the unbalanced forces at the end of iteration 1 are added to the corrective load q_c . The computations in iteration 2 are similar to those in iteration 1.

step 0: iteration 2

$$r = \hat{q}^{(1)} - \hat{u}^{(1)}(1.0 - \hat{u}^{(1)})(2 - \hat{u}^{(1)}) = -0.0006319$$

$$q_c^{(2)} = q_c^{(1)} + r = -0.0059106 - 0.0006319 = -0.0065425$$

$$\Delta\hat{\lambda}^{(0,2)} = 0.228838$$

The iteration in step 0 is continued until the absolute value of the unbalanced force r is less than 0.000001. The displacement, the load and the load factor increment at termination of the iteration in step 0 are:

$$\text{step 0: } u^{(1)} = -0.0473493$$

$$q^{(1)} = -0.0880797$$

$$\Delta\lambda^{(1)} = 0.2288380$$

The tangent stiffness $k_t^{(1)}$ and pattern displacement $u_t^{(1)}$ for load step 1 are computed with the initial displacement $u^{(1)}$ which was computed at the end of the first load step:

step 1: iteration 0

$$k_t^{(1)} = u^{(1)}(1 + u^{(1)})(2 + u^{(1)}) = 1.722630$$

$$u_t^{(1)} = \frac{q_t}{k_t^{(1)}} = \frac{-0.384900}{1.722630} = -0.223437$$

$$q_c^{(1,0)} = 0 \quad u_c^{(1,0)} = 0$$

$$\Delta\hat{\lambda}^{(1,0)} = 0.224692$$

$$\hat{u}^{(1,0)} = u^{(1)} + \Delta\hat{\lambda}^{(1,0)} u_t^{(1)} = -0.0975504$$

$$\hat{q}^{(1,0)} = q^{(1)} + \Delta\hat{\lambda}^{(1,0)} q_t = -0.1745638$$

The iteration in step 1 is similar to the iteration in step 0. The procedure is repeated for the succeeding load steps until the desired load factor is reached. The results of the computation are shown in figure 1.6. The iteration in step 4, which diverged in the refined arc method, now converges in the constant arc increment method. The displacement at the end of step 5 is close to a singular point.

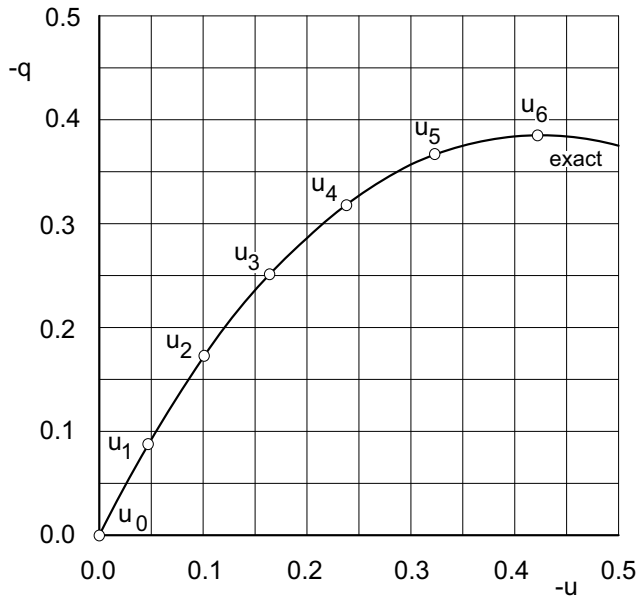


Figure 1.6 Solution of equation (1.11) with the constant arc increment method

The constant arc increment method is the state of the art for the solution of the governing equations for the geometrically nonlinear behaviour of trusses and frames. The method is sometimes varied further, for instance by replacing the tangent matrix by the secant stiffness matrix, as shown in chapters 4 to 6.

1.6 STATE OF THE ART IN STABILITY ANALYSIS

In some configurations of a structure, its shape can change significantly while there is little change in the loading and the strains remain small. This type of behaviour is considered to be a failure of the structure, even though the material does not rupture. The behaviour is called instability of the structure and must be prevented in engineering practice.

1.6.1 Concepts of Stability

A mathematical definition of stability was proposed by A.M. Lyapunov for processes that involve imperfections. He considers initial conditions \mathbf{d}_A , which lead to a state \mathbf{u}_A of the process, and slightly perturbed initial conditions \mathbf{d}_B , which lead to a state \mathbf{u}_B of the process. If a real number t exists so that every initial condition \mathbf{d}_B that lies in a sphere with centre \mathbf{d}_A and radius r leads to a state \mathbf{u}_B that lies in a sphere with centre \mathbf{u}_A and radius $t*r$, the process is stable.

The processes considered in the definition by Lyapunov are generally not in a state of equilibrium. In structural analysis, the stability of the equilibrium states of structural systems is investigated. If an equilibrium state is perturbed and the cause of the perturbation is then removed, the structure may either return to the equilibrium state, or it may remain in the perturbed state, or the perturbation may increase. If the structure returns to the equilibrium state, the structure is stable. If the perturbations increase, the structure is unstable.

Singularity of the tangential stiffness matrix is a necessary condition for the instability of an instant configuration of a structure. The singularity implies that the structure can undergo an infinitesimal displacement without a change in load. This condition is, however, not sufficient for instability. The structure is only unstable if, in addition, the load decreases due to a finite incremental displacement in the direction of the eigenvector which is associated with the eigenvalue null at the singular point. The example of bifurcation of a column in section 2.4 shows three continuations of the load path at the bifurcation A. The load path will follow path AB or AD, since small perturbations on path AC cause the column to displace to paths AB or AD.

The detection of configurations in which the tangent stiffness matrix of a structure is singular, is not difficult in principle: the determinant of the tangent stiffness matrix of an unstable structure is null. In practice, significant difficulties arise.

Since the determinant of a matrix is equal to the product of its eigenvalues, its value for large matrices can lie outside the range of data type double in computers. The determinant therefore cannot be used directly as a criterion for singularity. Near singular points of the tangential stiffness matrix, the product of the eigenvalues contains at least one value that nearly equals null. The determinant is nearly null and the governing incremental equations may become ill conditioned. As a result, the methods for the solution of the nonlinear equations described in section 1.5, become unreliable near singular points.

The determinant of the tangent stiffness matrix will usually change sign at a singular point. If, however, two eigenvalues of the tangent matrix change sign at the same singular point, for instance due to symmetry of the structure, the determinant does

not change sign. The sign of the determinant is therefore not a reliable indicator for singular points.

Besides the algebraic difficulties of stability analysis, there are inherent physical characteristics of the structural behaviour that cannot be removed by means of mathematical formulations. It is well known that buckling loads of structures can be very sensitive to imperfections, for example in the case of cylindrical shells subjected to axial compressive stress. Theories have been developed which make it possible to account for imperfections in stability analysis. Consider given patterns of load and imperfection, multiplied by a load factor and an imperfection factor that are specific for each configuration. The load and imperfection factors for singular configurations can be determined by fold theory [24, 57, 66]. Fold theory is not treated in this book. Since the imperfection pattern is usually chosen a priori, it is not certain that fold theory leads to the critical imperfections of a specific structure.

In order to be able to treat the deterministic aspects of structural stability reliably, the following three main problems must be solved:

Nearly singular configurations: A configuration of a structure is called a nearly singular configuration if the tangential stiffness matrix in that state is nearly singular. The corresponding point on the load path is called a nearly singular point. Reliable indicators that permit identification of nearly singular configurations of structures are essential for reliable stability analysis. The difficulties encountered in establishing suitable indicators are discussed in the literature. A generally applicable and widely accepted solution to the problem has not been reported.

Singular configurations: A point on a load-deflection diagram that corresponds to a configuration whose tangent stiffness matrix is singular is called a singular point. The singular point can be a snap-through point or a bifurcation point. At a snap-through point, the continuation of the load path is unique. At a bifurcation point, the load path of the structure in the continuation can follow one of several branches. The configuration of the structure at the singular point must be determined with high accuracy if the behaviour after a bifurcation is to be studied. Available methods are described in section 1.6.2.

Continuation of load paths: Methods used to extend a load path beyond a singular point are called continuation methods. While reliable methods are available for the continuation of load paths at snap-through points, continuation at bifurcation points is numerically difficult. It will be shown in chapter 7 that the difficulties are due to the singularity of the stiffness matrix in the vicinity of the continuation path. According to Belytschko [5], robust automatic procedures for continuation are not yet available.

1.6.2 Determination of Singular Points

Exact methods for the determination of the configurations of a structure in which the tangent stiffness matrix is singular are not treated extensively in the monographs given in section 1.3, which represent the state of the art. A method by Wriggers [71] based on an expanded system of equations, and a method by Belytschko [5] based on linear interpolation between the tangent matrices of neighbouring configurations, are summarised in this section.

Expanded equations: Let the nonlinear governing equations for an equilibrium state of the structure be (1.19) and let the necessary condition for its singularity be (1.20), where \mathbf{u} is the displacement in the equilibrium state, λ the load factor and \mathbf{x} an eigenvector of the tangential stiffness matrix in the equilibrium state. The function L in (1.21) sets the norm of \mathbf{x} to 1.

$$\text{equilibrium} \quad \mathbf{G}(\mathbf{u}, \lambda) = \mathbf{0} \quad (1.19)$$

$$\text{singularity} \quad \mathbf{K}_T(\mathbf{u}, \lambda) \mathbf{x} = \mathbf{0} \quad (1.20)$$

$$\text{normalisation} \quad L(\mathbf{x}) = 0 \quad (1.21)$$

Consistent linearisation is used to transform these equations, as follows:

\mathbf{K}_T	$\mathbf{0}$	$-\mathbf{P}$	$\Delta \mathbf{u}$	$= -$	$\mathbf{G}(\mathbf{u}, \lambda)$
$\frac{\partial}{\partial \mathbf{u}}(\mathbf{K}_T \mathbf{x})$	\mathbf{K}_T	$\frac{\partial}{\partial \lambda}(\mathbf{K}_T \mathbf{x})$	$\Delta \mathbf{x}$		$\mathbf{K}_T(\mathbf{u}, \lambda) \mathbf{x}$
$\mathbf{0}$	$\frac{\partial L}{\partial \mathbf{x}}$	$\mathbf{0}$	$\Delta \lambda$		$L(\mathbf{x})$

This set of equations is solved iteratively by decomposition, starting with a chosen approximate eigenvector \mathbf{x} which is then improved during the iteration.

Interpolation of tangent matrices: Let the tangent stiffness matrices of a structure in two neighbouring configurations A and B be \mathbf{K}_A and \mathbf{K}_B . The stiffness matrices are interpolated linearly with a parameter z . It is assumed that the load factor λ can be interpolated with the same parameter:

$$\mathbf{K}_T(z) = (1-z) \mathbf{K}_A + z \mathbf{K}_B \quad (1.22)$$

$$\lambda = (1-z) \lambda_A + z \lambda_B \quad (1.23)$$

At the limit point z_c , the determinant of matrix $\mathbf{K}_T(z_c)$ is null and equation (1.22) has a solution \mathbf{y} which is nontrivial:

$$(1-z) \mathbf{K}_A \mathbf{y} + z \mathbf{K}_B \mathbf{y} = \mathbf{0}$$

$$\mathbf{K}_A \mathbf{y} = z(\mathbf{K}_A - \mathbf{K}_B) \mathbf{y} \quad (1.24)$$

Equation (1.24) is a general eigenvalue problem with eigenvalue z . The structure has a singular configuration between configurations A and B if the solution z of equation (1.24) falls into the range $0 \leq z \leq 1$.

1.6.3 Continuation of Load Paths

The continuation of load paths is treated systematically by Wagner [66]. He presents several methods of deflation that permit reliable solution of the nearly singular governing equations near singular points if the eigenstate with the smallest eigenvalue is known. The numerical difficulties associated with continuation are presented. Details of the step of analysis directly following the singular point (where the methods for nearly singular governing equations cannot be applied) are not treated. The topic is considered in more detail in chapter 7 of this book.

1.7 RESEARCH ON NONLINEAR STRUCTURAL ANALYSIS

1.7.1 Research Needs

The increasing size of the world population and the necessity to raise the standard of living in disadvantaged areas place new demands on the civil engineering profession. Since the available amount of material, energy and labour is limited, their use must be optimised. One of the strategies that may contribute to this goal is a reduction in the amount of material contained in structural systems. This strategy reduces the self-weight of structures (a very significant part of the total load), thus further reducing the amount of material required to support the load. In addition, the behaviour of the structure during earthquakes can be improved by mass reduction, and the amount of energy required for the production of building materials and for the construction of buildings can be reduced.

The strategy used to reduce the amount of material for a structure with a given shape and function can require knowledge and skills that are not treated in depth in traditional civil engineering education and practice. As the amount of material in a structure is reduced, its stiffness may also be reduced and therefore its deformation under load will increase. It is no longer sufficient to compute the forces and the displacements of structural members with the traditional linear theories of structural analysis which are emphasised in the present education of civil engineers. Instead, structural behaviour must be predicted with nonlinear theories which take into account the influence of displacements on the forces in the structure and on its stability. These theories are considerably more complex than the corresponding theories for geometrically linear behaviour. For example, the principle of superposition for loading conditions, which is extensively applied in linear analysis, is no longer applicable in nonlinear analysis. While uniform computer-based methods are available for linear structural analysis, the methods for nonlinear analysis are not yet uniform. Some practising engineers do not have complete insight into the theoretical basis of the software they use for nonlinear analysis.

The review of the state of the art has shown that the theoretical foundation and the numerical procedures for the prediction of displacements and stresses in structures with nonlinear behaviour are available. Significant theoretical and numerical problems remain in the analysis of the stability of slender structures. The present degree of standardisation of the methods of analysis and of the treatment of different types of structural components is not yet satisfactory. This hampers the education of civil engineers in nonlinear analysis. There is also a regrettable gap between the algorithms used in commercial software applications for nonlinear structural analysis and the common engineering background of software users. Since the practising engineer cannot shift the responsibility for his design to the authors of the software, this dangerous gap severely restricts the application of the available methods of nonlinear analysis in civil engineering practice.

The research needs of civil engineering with respect to nonlinear structural analysis lie in three main areas: reliable methods of stability analysis, a unified approach to nonlinear analysis which relates the common theoretical background to the available commercial application software, and systematic categorisation of the main aspects of the nonlinear behaviour of typical structural systems in civil engineering.

Stability: Stability analysis in civil engineering is traditionally centred on the stability of individual components of a structure, rather than on the stability of the assemblage of structural components. This may be explained by the lack of adequate tools for stability analyses of complete structures in the past. Important aspects of behaviour, for example the horizontal stabilisation of high-rise buildings, are addressed in our building codes by indirect empirical methods such as fictitious horizontal loads that must be taken into account in addition to wind and earthquake loads. General rational methods of stability analysis with a model of the structure as a whole should be developed, which reliably predict the overall stability of the structure, the interaction between the components of the structure in providing restraint against instability of individual members, and the local stability of each individual member.

One of the areas that require additional research is the reliable determination of singular points. It is not known for the general case how many elements are required per member of a frame in order to predict singular points accurately, and how this number depends on the simplifications and interpolation functions used in the derivation of the secant stiffness matrices of the elements. A rather large number of elements per member is required if polynomials that do not satisfy the nonlinear governing equations of the individual elements exactly are used as interpolation functions. It is not known how the number of elements per member is affected if other interpolation functions are used.

The predicted nonlinear behaviour of slender structures, in particular of domes and thin-walled constructions, can depend strongly on imperfections in the geometry of the structure and on states of eigenstress of the unloaded structure. While the state of eigenstress can be influenced by the method of construction, the specification of geometric imperfections is empirical. There is no guarantee that the conventional imperfections are adequate for the investigation of innovative structures, for which an empirical background does not yet exist. Fold theory, which considers both force loads and imperfections in stability analysis, is not yet simple enough for general application. Methods remain to be developed to make fold analysis more independent of the assumed imperfection pattern.

The continuation of load paths beyond singular points is not essential in typical civil engineering construction. It does, however, become important if the load can be increased on the continuation of a load path. Continuation analysis, particularly the study of the displacements and deformations following a singular point, may contribute to the understanding of the behaviour of the structure in the vicinity of the singular point and may, for instance, yield information on desirable stiffeners. The numerical difficulties in the algorithms for the continuation of secondary branches at bifurcation points have not yet been thoroughly investigated.

Unified approach: At present, the effort required to become familiar with the scientific background of nonlinear structural analysis is unnecessarily large. The degree of standardisation of the methods of analysis is considerably less than that which has, for example, been achieved by introduction of the matrix formulation of the finite element method for linear structural analysis. Lack of standardisation is partly due to the complexity of the matter, and partly due to complicated formulations in the

literature. Scientists from different countries differ significantly in their approaches and notations.

The theory of nonlinear structural analysis should be unified in the sense that most steps of the formulation are analogous for all types of structural components. The basic approach could be to develop the formulation for every type of component as a specialisation of the common three-dimensional theory of elasticity. This will depend on a careful and complete formulation of the hypotheses on the behaviour of various types of structural components, and on a careful investigation of the unavoidable contradictions that these hypotheses cause, since the nonlinear theory of elasticity already contains the required number of equations.

The unified approach will reduce the amount of repetition in the formulations. It will also reduce the effort required by a person who is familiar with the theory for one type of component to become familiar with the theory for another type of component. It will therefore become easier to teach nonlinear analysis at university level. The aim of education is to improve the ability of young engineers to anticipate significant nonlinear effects, to map them to appropriate models and to use these models to develop a good understanding of nonlinear behaviour. Such education will provide a sound foundation for an extensive and beneficial use of nonlinear methods of structural analysis in civil engineering practice.

A unified approach is likely to generalise nonlinear structural analysis. The theories for the different types of components are more likely to be compatible, so that the numerical behaviour of the resulting algorithms will be similar. The resulting component types can be mixed freely in realistic models of complex real structures. The governing equations of these models can be solved with a common numerical algorithm. The unification of the theory of nonlinear analysis is therefore likely to contribute to its generalisation and applicability.

The development and application of commercial software packages for nonlinear structural analysis will benefit greatly from a unified theoretical approach. Clients of software houses may, after some time, have a common theoretical background that encourages the application and counteracts indiscriminate use of the software. The effort for software development is likely to decrease due to a common data base and common interactive user interfaces for the complete assemblage of different types of structural components, due to the similarities in the algorithms for the various types of components, and due to the common numerical algorithm for the solution of the nonlinear governing equations.

Categorisation of behaviour: The building industry is expected to produce specific types of structures to satisfy the needs of its clients. Examples of such structures are portal frames and parallel chord trusses for industrial buildings. The general aspects of the linear behaviour of such structures are well understood and represent the know-how on which structural design is based. It is expected that an improved understanding of the nonlinear behaviour and of the stability of such structures can lead to improved designs.

The scope of the investigations necessary to categorise nonlinear behaviour for several types of structures is very large. Adequate insight requires variation of the main design parameters such as span, height, purlin spacing, profiles of the

structural members, loading conditions and imperfections. The objective function of the designs must reflect the real goals of practical design and their relative significance, such as saving on the weight of material, reducing labour cost and reducing construction height without excessive deflections or instability.

1.7.2 Research Topics

The authors concentrate their research activities on the geometrically nonlinear analysis of structures. This does not imply that the material nonlinearities are less important. Elasto-plastic behaviour is of major significance for the design and behaviour of steel structures. The book is focussed on geometric nonlinearity so that a foundation can be laid for a unified approach to the nonlinear analysis of structures, and that advances can be made in the promising area of stability analysis. All software developments are designed so that the data structure, algorithms and user interfaces can be extended to material nonlinearity.

The research programme focuses on the following research topics: formulation of a unified theory, derivation of consistent trial functions for nonlinear finite elements, reliable detection of nearly singular configurations of a structure, exact determination of singular points, continuation of load paths beyond singular points, and categorisation of nonlinear behaviour. In all areas of this research, theoretical developments are immediately cast into object-oriented software with a common data base and a common user interface, which serves as a numerical test bed for the accuracy, reliability, efficiency and robustness of the developed theories and algorithms. The investigation of some of the research topics is well under way. The results are described in other chapters of this book.

Unified theory: The groundwork for the concept of a unified theory has been laid with research in the following areas:

- formulation of the geometrically nonlinear theory of elasticity (chapter 3)
- formulation and software implementation of the nonlinear theory for plane trusses (chapter 4)
- formulation and software implementation of the nonlinear theory for plane frames (chapter 5)
- formulation and software implementation of theory for the constant arc increment method for the solution of nonlinear governing equations (chapter 6)
- formulation of the theory of structural stability and software implementation of the algorithms for stability analysis (chapter 7).
- evaluation of the accuracy, efficiency and robustness of the algorithms based on these theories (chapter 9)
- formulation and software implementation of the nonlinear theory of space trusses
- formulation and software implementation of the nonlinear theory of space frames

Research on the unified theory can be extended to include fold analysis and related methods for the study of imperfections, as well as formulations and evaluations for additional types of structural components.

Consistent trial functions: The literature on structural mechanics appropriately emphasises the importance of consistent linearisation. The consistent approach is, however, confined to the derivation of the analytical governing equations. When the finite elements for structural components are derived, consistency is not observed with equal care. Since the suitability of the trial functions affects the quality of a finite element solution significantly, consistent linearisation should be extended to the trial functions of the finite element formulations. In chapter 5 of this book it is shown that for finite frame elements, consistency in the trial functions requires separate trial functions for the total equilibrium equations and the incremental governing equations in a load step. Similar approaches should be investigated for other types of structural elements such as plates.

Detection of nearly singular states: The state of the art shows that the sign and magnitude of the determinant of the incremental stiffness matrix are not suitable as indicators for the detection of nearly singular configurations. The coefficients of diagonal matrix \mathbf{D} in the decomposition $\mathbf{K}_T = \mathbf{LDL}^T$ of the tangential stiffness matrix \mathbf{K}_T seem to be more suitable as indicators of nearly singular states. The approach is described in chapter 6 of this book. A systematic study of suitable limit values of the coefficients in diagonal matrix \mathbf{D} for different types of structural elements seems to be required. These limit values may well differ in different phases of the analysis, such as steps of the constant arc increment method and steps of the continuation at a bifurcation point.

Exact singular states: The study of the state of the art led to two methods for the exact determination of singular tangent stiffness matrices, as described in section 1.6.2. The method of Wriggers [71] offers the advantage that the derivation is consistent. The formulation has been modified in this book to avoid unnecessary expansion of the set of governing equations. The method of Belyschko [5] does not linearise the tangent stiffness matrix consistently, since the coefficients of the matrix and not the displacement increments are interpolated linearly. The method requires the determination of a nearly singular point that lies beyond the singular point on the load path. The method is not robust because the governing equations are ill-conditioned in the vicinity of singular points, particularly bifurcation points. The consistent linearisation of the tangent stiffness matrix for a nearly singular state is presented in chapter 7 of this book. It does not require a second point on the load path. The consistent linearisation does not, however, remove the ill-conditioning of the linear algebraic system of governing equations. Therefore, an efficient deflation method which removes the eigenstate with the lowest eigenvalue from the solution of the linear equations has been developed and tested successfully.

Continuation of load paths: The continuation of load paths at singular points, particularly at bifurcation points, is the least well-developed part of geometrically nonlinear structural analysis. The necessity of a particularly careful choice of the step size on continuations is mentioned in the literature, but the causes for the observed numerical phenomena are not always clarified. The primary cause of the numerical difficulties, namely the nearly singular character of the governing equations in the entire vicinity of a load path and its influence on the iteration in the solution procedures for a load step, is not treated systematically. It seems essential to treat

the first load step following a singular point separately, since the singularity of the tangent matrix at the start of this step poses special difficulties. Additional research is required to achieve robustness in the continuation of load paths at bifurcation points.

Space trusses and space frames: The theory and methods of geometrically nonlinear analysis, which are described in this book for plane trusses and frames, have also been applied to space trusses. It proved possible to follow the unified approach in all the extensions of the theory and algorithms required for space trusses. The algorithms that compute the load paths and the singular points for plane trusses were applied without change to space trusses. The extension of the analysis from plane frames to space frames will be treated next. This extension increases the range of applicability of the method considerably, since medium and large single layer space structures are frequently designed as frames and not as trusses. In addition, many medium and high-rise steel structures are designed as steel frames. The most significant extension to the plane frame theory presented in this book is the treatment of torsion in the members of a space frame.

Besides the research topics of the authors, there are many other aspects of the nonlinear behaviour of structures that may benefit from additional research. Future research and development should be oriented so that they will eventually lead to a generalised software platform for nonlinear analysis that rests on a uniform set of theories for the different types of structural components, a uniform data base and a uniform graphical user interface.

Chapter 2 is devoted to simple structures whose behaviour is affected significantly by geometrical nonlinearity. The first two examples are distinguished by the fact that exact solutions for the behaviour of the structures are derived. They are thus useful in guiding the development of suitable numerical algorithms and in giving some insight into geometrically nonlinear behaviour. These examples are used in chapter 9 to test the software implementation of the theory.

2 NONLINEAR BEHAVIOUR OF PLANE TRUSSES AND FRAMES

2.1 NONLINEAR BEHAVIOUR OF A 2-BAR TRUSS

Consider the geometrically nonlinear behaviour of the 2-bar truss shown in figure 2.1. The bars of the truss have equal cross-sectional area A , modulus of elasticity E and length L_0 in the reference configuration. The supports A and B of the truss are pinned. The loads P_1 and P_2 at the apex C act in the direction of the global axes x_1 and x_2 respectively. Let the global coordinates of the resulting displacement of the apex be u_1 and u_2 . The load P_1 is set to null. The load P_2 as well as the reactions R_1 and R_2 are to be determined as functions of the displacements u_1 and u_2 .

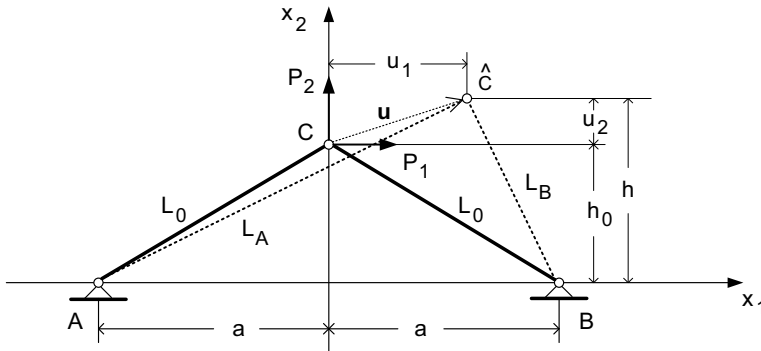


Figure 2.1 Displacement of a 2-bar truss

Strain of the bars: Figure 2.2 shows a typical bar ST of the truss in its local coordinate system y_1, y_2 . The local base vectors \mathbf{e}_1 and \mathbf{e}_2 are orthogonal unit vectors. Let the local coordinates of the displacement vector \mathbf{v} of node T relative to node S be v_1, v_2 and denote the derivative of v_m with respect to y_1 by $v_{m,1}$. Let the length of the bar in the instant configuration be L .

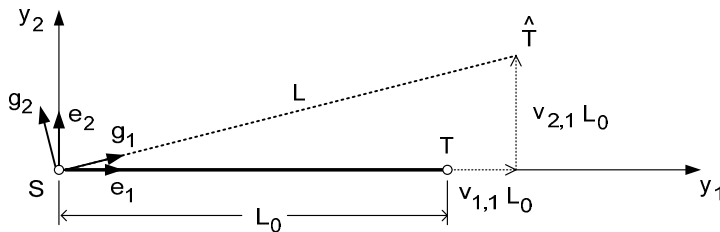


Figure 2.2 Reference and instant configurations of a bar

The base vectors $\mathbf{g}_1, \mathbf{g}_2$ of the instant configuration of the bar are the derivatives of the location vector \mathbf{y} with respect to the coordinates y_1, y_2 :

$$\mathbf{e}_1 = \frac{d\mathbf{y}}{dy_1}$$

$$\mathbf{g}_1 = \frac{d(\mathbf{y} + \mathbf{v})}{dy_1} = \mathbf{e}_1 + v_{1,1}\mathbf{e}_1 + v_{2,1}\mathbf{e}_2 \quad (2.1)$$

The axial strain of the bar follows from the nonlinear theory of elasticity:

$$\varepsilon_1 = \frac{1}{2}(\mathbf{g}_1^T \mathbf{g}_1 - \mathbf{e}_1^T \mathbf{e}_1) = \frac{1}{2}((1 + v_{1,1})^2 + v_{2,1}^2 - 1)$$

$$\varepsilon_1 = \frac{1}{2} \left(\left(\frac{L}{L_0} \right)^2 - 1 \right) = \frac{1}{2L_0^2} (L^2 - L_0^2) \quad (2.2)$$

Expression (2.2) is used to compute the strain ε_A in bar AC of the truss:

$$L_0^2 = a^2 + h_0^2$$

$$L_A^2 = (a + u_1)^2 + (h_0 + u_2)^2$$

$$\varepsilon_A = \frac{1}{2L_0^2} ((a + u_1)^2 + (h_0 + u_2)^2 - (a^2 + h_0^2))$$

$$= \frac{1}{2L_0^2} (2au_1 + u_1^2 + 2h_0u_2 + u_2^2)$$

$$\varepsilon_A = \frac{1}{2} \left(\frac{h_0}{L_0} \right)^2 \left(\frac{u_1}{h_0} \left(2 \frac{a}{h_0} + \frac{u_1}{h_0} \right) + \frac{u_2}{h_0} \left(2 + \frac{u_2}{h_0} \right) \right) \quad (2.3)$$

Expression (2.2) is also used to compute the strain ε_B in bar BC of the truss:

$$L_B^2 = (a - u_1)^2 + (h_0 + u_2)^2$$

$$\varepsilon_B = \frac{1}{2L_0^2} ((a - u_1)^2 + (h_0 + u_2)^2 - (a^2 + h_0^2))$$

$$= \frac{1}{2L_0^2} (-2au_1 + u_1^2 + 2h_0u_2 + u_2^2)$$

$$\varepsilon_B = \frac{1}{2} \left(\frac{h_0}{L_0} \right)^2 \left(\frac{u_1}{h_0} \left(-2 \frac{a}{h_0} + \frac{u_1}{h_0} \right) + \frac{u_2}{h_0} \left(2 + \frac{u_2}{h_0} \right) \right) \quad (2.4)$$

Equilibrium: Let the forces in bars AC and BC in the instant configuration of the truss be N_A and N_B . The bar forces are computed with the physical stresses that act in the bar. Since the base vector \mathbf{g}_1 of the instant configuration is not a unit vector, the stresses computed from the strains in the bar must be multiplied with the length of the base vector \mathbf{g}_1 :

$$N_A = A E \frac{L_A}{L_0} \varepsilon_A \quad (2.5)$$

$$N_B = A E \frac{L_B}{L_0} \varepsilon_B \quad (2.6)$$

The bar forces are decomposed into their components in the global space:

$$\text{bar AC: } N_{1A} = \frac{a + u_1}{L_A} N_A = \frac{AE}{L_0} (a + u_1) \varepsilon_A \quad (2.7)$$

$$N_{2A} = \frac{h_0 + u_2}{L_A} N_A = \frac{AE}{L_0} (h_0 + u_2) \varepsilon_A$$

$$\text{bar BC: } N_{1B} = \frac{-a + u_1}{L_A} N_B = \frac{AE}{L_0} (-a + u_1) \varepsilon_B \quad (2.8)$$

$$N_{2B} = \frac{h_0 + u_2}{L_A} N_B = \frac{AE}{L_0} (h_0 + u_2) \varepsilon_B$$

The global components of the bar forces at apex C are in equilibrium with the applied loads P_1 and P_2 :

$$(a + u_1) \varepsilon_A + (-a + u_1) \varepsilon_B = \frac{P_1 L_0}{AE} \quad (2.9)$$

$$(h_0 + u_2) (\varepsilon_A + \varepsilon_B) = \frac{P_2 L_0}{AE} \quad (2.10)$$

The strains of the bars are substituted from (2.3) and (2.4) into (2.9) and (2.10) to yield the governing equations for the displacement of the apex of the truss:

$$\frac{u_1}{h_0} \left(2 \left(\frac{a}{h_0} \right)^2 + \left(\frac{u_1}{h_0} \right)^2 + \frac{u_2}{h_0} \left(2 + \frac{u_2}{h_0} \right) \right) = \frac{P_1}{AE} \left(\frac{L_0}{h_0} \right)^3 \quad (2.11)$$

$$\left(1 + \frac{u_2}{h_0} \right) \left(\left(\frac{u_1}{h_0} \right)^2 + \frac{u_2}{h_0} \left(2 + \frac{u_2}{h_0} \right) \right) = \frac{P_2}{AE} \left(\frac{L_0}{h_0} \right)^3 \quad (2.12)$$

Normalised variables: The physical variables that describe the behaviour of the truss are normalised as follows:

$$\text{normalised displacements: } w_i := \frac{u_i}{h_0} \quad i \in \{1, 2\} \quad (2.13)$$

$$\text{normalised loads: } p_i := \frac{P_i}{AE} \left(\frac{L_0}{h_0} \right)^3 \quad (2.14)$$

$$\text{normalised reactions: } r_i := \frac{2R_i}{AE} \left(\frac{L_0}{h_0} \right)^3 \quad (2.15)$$

$$\text{normalised bar forces: } n_i := \frac{2N_i}{AE} \left(\frac{L_0}{h_0} \right)^3 \quad (2.16)$$

$$\text{shape factor: } m := \frac{a}{h_0} \quad (2.17)$$

The normalised variables (2.13) to (2.17) are substituted into equations (2.11) and (2.12), which acquire the following normalised form:

$$p_1 = w_1 (2m^2 + w_1^2 + w_2 (2 + w_2)) \quad (2.18)$$

$$p_2 = (1 + w_2) (w_1^2 + w_2 (2 + w_2)) \quad (2.19)$$

Solution 1: $p_1 = 0$ and $w_1 = 0$

Equation (2.18) is satisfied because the factor w_1 is null. The vertical load p_2 at the apex follows from (2.19):

$$p_2 = w_2(1+w_2)(2+w_2) \quad (2.20)$$

Equation (2.20) shows that there are three displacement states $w_2 = 0, -1$ and -2 for which the load p_2 is null. These states are the reference configuration, the instant configuration in which the bars are collinear and the mirror image of the reference configuration with respect to axis x_1 . The configurations are shown in figure 2.3:

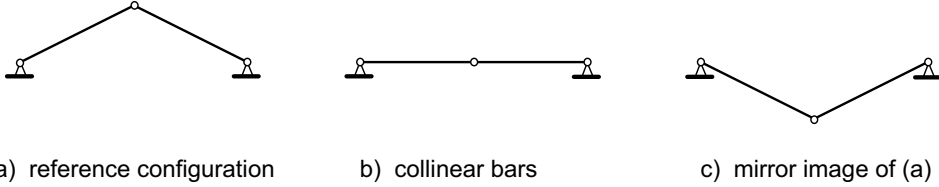


Figure 2.3 Equilibrium states with load $p_2 = 0$

The load reaches its extreme value at the displacements for which the derivative of p_2 with respect to w_2 is null:

$$\frac{dp_2}{dw_2} = 3w_2^2 + 6w_2 + 2 = 0$$

$$w_{2c} = -0.422649731 \quad \rightarrow \quad p_{2c} = -0.384900179 \quad (2.21)$$

$$w_{2c} = -1.577350269 \quad \rightarrow \quad p_{2c} = 0.384900179 \quad (2.22)$$

The strain in bars AC and BC follows from (2.3) and (2.4):

$$\varepsilon_A = \varepsilon_B = \frac{1}{2} \left(\frac{h_0}{L_0} \right)^2 w_2(1+w_2) \quad (2.23)$$

The reactions at supports A and B follow from the components (2.7) and (2.8) of the bar forces in global space by substituting (2.23). The normalised reactions (2.15) are:

$$r_{1A} = -m w_2(2+w_2) \quad r_{2A} = -w_2(1+w_2)(2+w_2) \quad (2.24)$$

$$r_{1B} = m w_2(2+w_2) \quad r_{2B} = -w_2(1+w_2)(2+w_2) \quad (2.25)$$

The horizontal reactions at nodes A and B reach extreme values at the displacements for which the derivative of r_1 with respect to w_2 is null:

$$\frac{dr_{1A}}{dw_2} = -2m(1+w_2) = 0 \quad \rightarrow \quad w_2 = -1 \quad (2.26)$$

$$r_{1A} = -r_{1B} = m \quad (2.27)$$

The maximum horizontal reaction occurs when the bars AC and BC are collinear. The load in this configuration is null, as shown in diagram (b) of figure 2.3.

The variation of the load p_2 with the displacement w_2 of the apex is shown in figure 2.4. The truss has three equilibrium configurations for each load in the range $-0.3849 \leq p_2 \leq 0.3849$. Figure 2.5 shows the reaction r_1 as a function of the load p_2 .

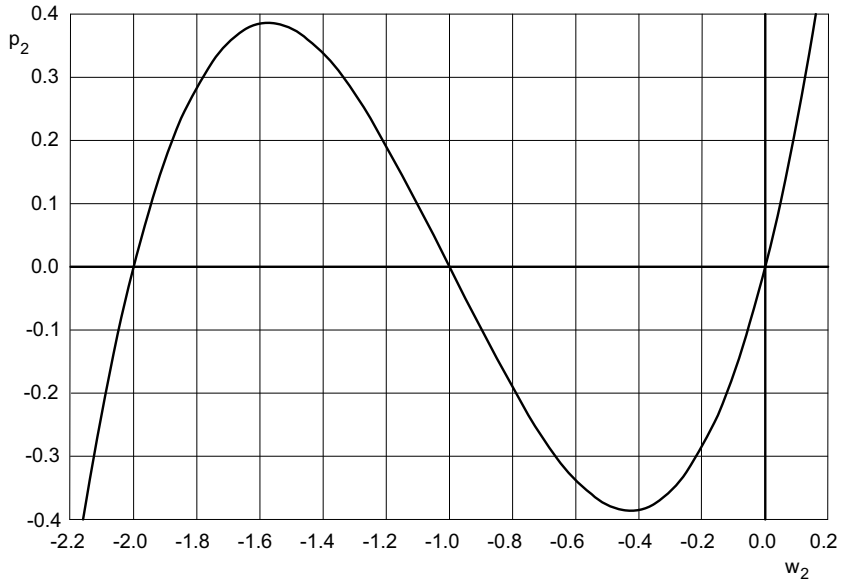


Figure 2.4 Variation of load p_2 with displacement w_2 (solution 1)

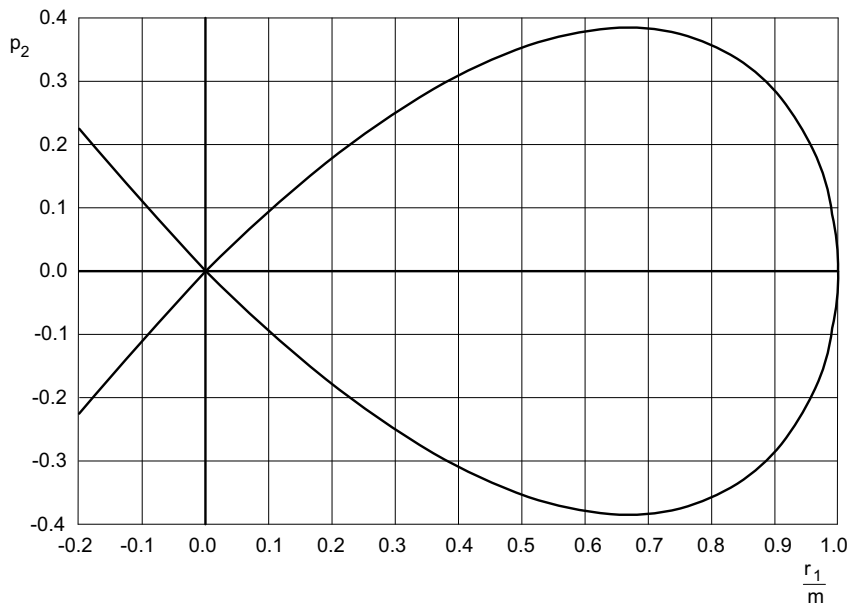


Figure 2.5 Variation of reaction r_1 with load p_2 (solution 1, shape factor m)

Solution 2: $p_1 = 0$ and $2m^2 + w_1^2 + w_2(2 + w_2) = 0$

Equation (2.18) is satisfied because the second factor is null:

$$2m^2 + w_1^2 + w_2(2 + w_2) = 0 \quad (2.28)$$

The vertical load p_2 follows from (2.19):

$$p_2 = -2m^2(1 + w_2) \quad (2.29)$$

The strain in bars AC and BC follows from (2.3) and (2.4):

$$\varepsilon_A = -\left(\frac{h_0}{L_0}\right)^2 m(m - w_1) \quad (2.30)$$

$$\varepsilon_B = -\left(\frac{h_0}{L_0}\right)^2 m(m + w_1) \quad (2.31)$$

The reactions at nodes A and B follow from components (2.7) and (2.8) of the bar forces in global space by substituting the strains from (2.30) and (2.31):

$$r_{1A} = 2m(m^2 - w_1^2) \quad (2.32)$$

$$r_{2A} = 2m(m - w_1)(1 + w_2) \quad (2.33)$$

$$r_{1B} = -2m(m^2 - w_1^2) \quad (2.34)$$

$$r_{2B} = 2m(m + w_1)(1 + w_2) \quad (2.35)$$

Substitution of w_2 from (2.29) into (2.28) yields a quadratic equation for w_1 :

$$w_1^2 = 1 - 2m^2 - \left(\frac{p_2}{2m^2}\right)^2 \quad (2.36)$$

Equation (2.36) has real roots for w_1 if load p_2 satisfies condition (2.37):

$$p_2^2 < 4m^4(1 - 2m^2) \quad (2.37)$$

Equation (2.37) has real roots for p_2 if the shape factor m satisfies condition (2.38):

$$m \leq m_c = \frac{1}{\sqrt{2}} = 0.707107 \quad (2.38)$$

For shape factors $m \leq m_c$, equation (2.37) yields two real roots $\pm p_{2c}$ for which the truss is in equilibrium:

$$p_{2c} = -2m^2\sqrt{1 - 2m^2} \quad (2.39)$$

Substitution of (2.39) into (2.29) yields the critical displacement w_{2c} :

$$w_{2c} = -1 + \sqrt{1 - 2m^2} \quad (2.40)$$

The displacement state (w_1, w_2) for the load range $-p_{2c} \leq p_2 \leq p_{2c}$ is given by (2.28):

$$w_1^2 + (1 + w_2)^2 = 1 - 2m^2 \quad m \leq m_c \quad (2.41)$$

The locus of apex C as it displaces for the load range $-p_{2c} \leq p_2 \leq p_{2c}$ is a circle with midpoint at the origin and radius $\sqrt{1 - 2m^2}$. The displacement history of apex C is shown in figure 2.6 for the shape factor $m = 0.3$.

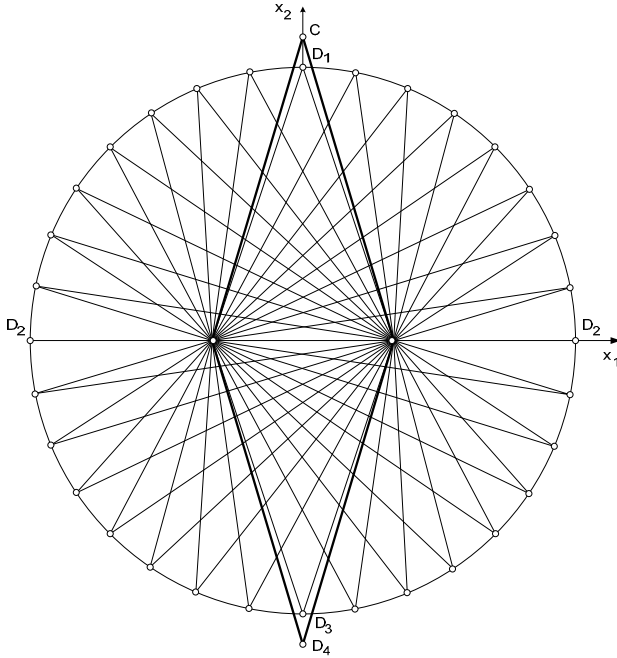


Figure 2.6 Displacement states of a truss with shape factor 0.3 (solution 2)

Tangent stiffness matrix: The incremental equations for a load step of the truss are:

$$\mathbf{K} \Delta \mathbf{u} = \Delta \mathbf{P} \quad (2.42)$$

- \mathbf{K} tangent stiffness matrix of the 2-bar truss
- $\Delta \mathbf{u}$ displacement increment in a load step
- $\Delta \mathbf{P}$ load increment in a load step

The coefficient k_{im} of the tangent stiffness matrix \mathbf{K} of the truss is the partial derivative of the load coordinate P_i with respect to the displacement coordinate u_m . Definitions (2.13) and (2.14) are used to replace the physical variables P_i and u_m by the normalised variables p_i and w_m :

$$k_{im} = \frac{\partial P_i}{\partial u_m} = \frac{AE}{h_0} \left(\frac{h_0}{L_0} \right)^3 \frac{\partial p_i}{\partial w_m} \quad (2.43)$$

The tangent stiffness matrix of the 2-bar truss follows from (2.18) and (2.19):

$$\mathbf{K} = \frac{AE}{h_0} \left(\frac{h_0}{L_0} \right)^3 \begin{bmatrix} 2m^2 + 3w_1^2 + w_2(2 + w_2) & 2w_1(1 + w_2) \\ 2w_1(1 + w_2) & w_1^2 + 3w_2^2 + 6w_2 + 2 \end{bmatrix} \quad (2.44)$$

The linear elastic stiffness matrix of the 2-bar truss follows from (2.44) by setting the displacement coordinates to null:

$$\mathbf{K}_{\text{linear}} = \frac{AE}{h_0} \left(\frac{h_0}{L_0} \right)^3 \begin{bmatrix} 2m^2 & 0 \\ 0 & 2 \end{bmatrix} \quad (2.45)$$

The displacement state (w_1, w_2) for which the tangent stiffness matrix \mathbf{K} is computed must be an equilibrium state of the truss. Solutions 1 and 2 are therefore substituted into expression (2.44) in order to obtain the tangent stiffness matrices for the two possible load paths.

Solution 1: $w_1 = 0$

Expression (2.44) leads to the following equation:

$$\frac{AE}{h_0} \begin{pmatrix} h_0 \\ L_0 \end{pmatrix}^3 \begin{array}{|c|c|c|} \hline 2m^2 + w_2(2 + w_2) & 0 & \Delta u_1 \\ \hline 0 & 3w_2^2 + 6w_2 + 2 & \Delta u_2 \\ \hline \end{array} = \begin{array}{|c|} \hline 0 \\ \hline \Delta P_2 \\ \hline \end{array} \quad (2.46)$$

Solution 2: $2m^2 + w_1^2 + w_2(2 + w_2) = 0$

Expression (2.44) leads to the following equation:

$$\frac{AE}{h_0} \begin{pmatrix} h_0 \\ L_0 \end{pmatrix}^3 \begin{array}{|c|c|c|} \hline 2w_1^2 & 2w_1(1 + w_2) & \Delta u_1 \\ \hline 2w_1(1 + w_2) & 2(1 + w_2)^2 - 2m^2 & \Delta u_2 \\ \hline \end{array} = \begin{array}{|c|} \hline 0 \\ \hline \Delta P_2 \\ \hline \end{array} \quad (2.47)$$

Stability: The truss is in a singular configuration if the tangent stiffness matrix \mathbf{K} is singular so that a nontrivial solution $\Delta \mathbf{u}$ exists for the homogeneous equation (2.42). Consider solution 1. The tangent matrix in expression (2.46) is singular if at least one of its diagonal coefficients is null. The displacement w_2 and the diagonal coefficients of solution 1 have the following values in a singular configuration:

$$\text{Case 1:} \quad k_{11} = 2m^2 + w_2(2 + w_2) = 0 \quad (2.48)$$

$$w_{2c} = -1 \pm \sqrt{1 - 2m^2}$$

$$k_{22} = 2(1 - 3m^2)$$

$$\text{Case 2:} \quad k_{11} = 3w_2^2 + 6w_2 + 2 = 0 \quad (2.49)$$

$$w_{2c} = -1 \pm \frac{1}{\sqrt{3}} = -0.422650 \text{ or } -1.577350$$

$$k_{22} = 2\left(m^2 - \frac{1}{3}\right)$$

Consider solution 2. The determinant of the tangent matrix (2.47) is:

$$\det \mathbf{K} = -4m^2 w_1^2 \quad (2.50)$$

The determinant of the tangent stiffness matrix for solution 2 equals null only for the displacement $w_1 = 0$.

The load path of a truss determines which of the potential singular configurations is reached first. This is called the critical configuration, since it determines the load at which the truss first becomes unstable. It will be shown in chapter 7 that the critical configuration is determined by the shape factor m defined in (2.17).

Yield strain: The strain in the bars at the singular point for snap-through in solution 1 is determined with (2.23) by setting $w_2 = w_{2c}$ from (2.21):

$$\varepsilon_c = -0.1220 \left(\frac{h_0}{L_0} \right)^2 \quad m > m_c \quad (2.51)$$

The yield strain for steel is of the order of 0.0012. The truss will thus remain elastic until it reaches the snap-through point if the aspect ratio of the truss satisfies condition (2.52). This aspect ratio corresponds to an angle change of 11.5 degrees between the bars at the apex C. If the truss height h_0 exceeds the limit (2.52), the bars of the truss yield before the limit is reached (material nonlinearity).

$$h_0 \leq 0.100 L_0 \quad m > m_c \quad (2.52)$$

The strain in the bars at the limit point for bifurcation in solution 2 is determined with (2.30) by setting $w_1 = 0$:

$$\varepsilon_c = - \left(\frac{a}{L_0} \right)^2 \quad m \leq m_c \quad (2.53)$$

The truss will remain elastic until it reaches the bifurcation point if the shape of the truss satisfies condition (2.54). This shape has an angle of 4.0 degrees between the bars at the apex C. If the half-width a of the base of the truss exceeds the limit (2.54), the bars of the truss yield before the bifurcation point is reached (material nonlinearity).

$$a \leq 0.0346 L_0 \quad m \leq m_c \quad (2.54)$$

Conclusions: The geometrically nonlinear behaviour of a 2-bar truss illustrates several aspects of the nonlinear behaviour of slender structures. The relationship between the applied load and the resulting displacements of the truss is strongly nonlinear. There can be more than one state of displacement corresponding to a given state of loading. The principle of superposition of load cases, which depends on a linear relationship between loads and displacements, cannot be applied if the behaviour is nonlinear.

The behaviour of the 2-bar truss also illustrates the significance of singular configurations for the behaviour of slender structures. A truss in a singular configuration can undergo infinitesimal displacements at constant load. There are two types of singular configurations: snap-through points and bifurcation points. The continuation of the load path at a snap-through point is unique. At a bifurcation point, the load path of a 2-bar truss can be continued in more than one direction. The shape factor m determines which type of singular configuration is reached first when the structure is loaded.

2.2 LARGE DISPLACEMENTS OF A CANTILEVER BEAM

Figure 2.7 shows a prismatic cantilever beam with span L and modulus of elasticity E , whose cross-section has a moment of inertia J . A moment M is applied at the tip of the cantilever. The displacements of the points on the axis of the beam are to be determined as a function of the end moment M .

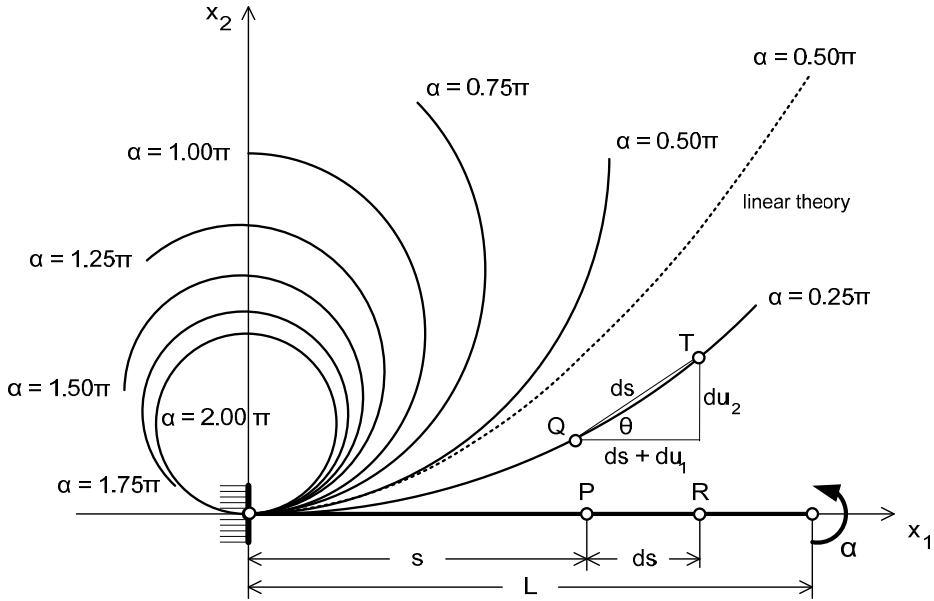


Figure 2.7 Displacements of a cantilever beam subjected to an end moment

Consider points P and R with coordinates s and $s + ds$ on the axis of the reference configuration of the beam. Let these material points of the axis displace to Q and T in an instant configuration of the beam. Let the coordinates of the displacement of point P be (u_1, u_2) in the global space, and let the corresponding displacements of point R be $(u_1 + du_1, u_2 + du_2)$. Denote the rotation of the tangent to the axis of the beam at P from the reference configuration to the instant configuration by θ . Let the value of θ be positive in the anti-clockwise direction.

The forces at point Q are in equilibrium if the resultants of the stresses acting on the cross-section at Q are null. The moments about point Q are in equilibrium if the moment due to the stresses acting on the cross-section equals the end moment M . Since the axial stress resultant is null, the axis of the beam does not stretch. The distance of point Q from the support of the beam, measured on the axis, is thus equal to s . Assume that the deformation of the beam satisfies the Kirchhoff hypothesis, so that the curvature of the axis is equal to the bending stiffness:

$$\frac{d\theta}{ds} = \frac{M}{EJ} \quad (2.55)$$

The rotation θ at support $s = 0$ is null. The curvature (2.55) is integrated from the support to point Q:

$$\theta = \frac{M}{EJ} s \quad (2.56)$$

The trigonometric functions of angle θ are determined in the triangle with base QT:

$$\sin\theta = \frac{du_2}{ds} \quad (2.57)$$

$$\cos\theta = 1 + \frac{du_1}{ds} \quad (2.58)$$

The moment M , the displacements u_i and the distance s are replaced by the normalised moment α , the normalised displacements v_i and the normalised distance t :

$$\alpha = \frac{ML}{EJ} \quad (2.59)$$

$$v_i = \frac{u_i}{L} \quad (2.60)$$

$$t = \frac{s}{L} \quad (2.61)$$

Substitution of (2.55) and (2.59) to (2.61) into (2.57) and (2.58) yields:

$$\frac{dv_1}{dt} = \cos \alpha t - 1 \quad (2.62)$$

$$\frac{dv_2}{dt} = \sin \alpha t \quad (2.63)$$

Expressions (2.62) and (2.63) are integrated with respect to t :

$$v_1 = \frac{1}{\alpha} \sin \alpha t - t + c_1$$

$$v_2 = -\frac{1}{\alpha} \cos \alpha t + c_2$$

The constants c_1 and c_2 are determined with the boundary conditions $v_1 = 0$ and $v_2 = 0$ at $t = 0$:

$$v_1 = \frac{1}{\alpha} \sin \alpha t - t \quad (2.64)$$

$$v_2 = \frac{1}{\alpha} (1 - \cos \alpha t) \quad (2.65)$$

The instant configurations of the beam for $\alpha = 0.25\pi, 0.50\pi, \dots, 2.00\pi$ are shown in figure 2.7. The maximum rotation at the tip of the beam is 2π . The displacements computed with the linear beam theory for $\alpha = 0.50\pi$ are shown as a dotted line. The significant deviation between linear and nonlinear theory is evident. It becomes larger as the end moment is increased. The example shows that nonlinear frame theory must account for large rotations.

2.3 TENSION STIFFENING OF A PINNED BEAM

Figure 2.8 shows a beam with span L , modulus of elasticity E , cross-sectional area A and moment of inertia J . The beam is pinned at both ends so that displacements in the direction of both axes are prevented at both ends of the beam. Both ends of the beam can rotate freely. The beam is subjected to a sinusoidal load q per unit length. The displacement u , normal to the axis of the beam, is to be determined.

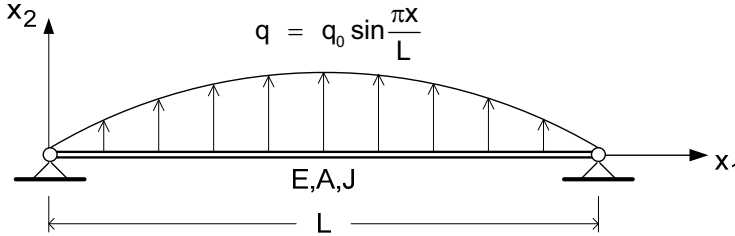


Figure 2.8 Pinned beam subjected to sinusoidal load

As the beam deflects, its axis is stretched. The stretching of the axis leads to an axial force in the beam. Due to the curvature of the deflected axis, the axial force balances part of the applied load, as shown in figure 2.9:

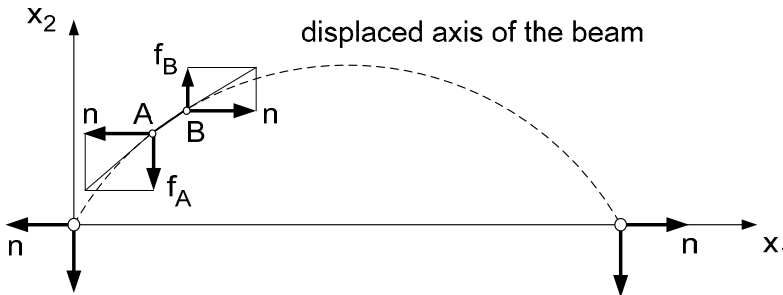


Figure 2.9 Forces acting on beam element AB due to stretching of the axis

Equilibrium in the horizontal direction requires the horizontal component n of the axial force to be constant. The vertical component of the axial force is determined so that the axial force is tangential to the displaced axis of the beam. The following vertical forces act at the end points of beam element AB in figure 2.9:

$$f_A = -n \frac{du_{2A}}{dx_1}$$

$$f_B = n \left(\frac{du_{2A}}{dx_1} + \frac{d^2u_{2A}}{dx_1^2} dx_1 \right)$$

The resultant vertical force $f_A + f_B$ is in equilibrium with a part q_n of the applied load:

$$n \frac{d^2u_2}{dx_1^2} + q_n = 0 \tag{2.66}$$

The part q_b of the applied load that is carried by bending is determined by the usual beam theory:

$$EJ \frac{d^4 u_2}{dx_1^4} - q_b = 0 \quad (2.67)$$

The total applied load is $q = q_n + q_b$. The governing equation for the pinned beam subjected to the total applied load follows by subtracting (2.66) from (2.67):

$$EJ \frac{d^4 u_2}{dx_1^4} - n \frac{d^2 u_2}{dx_1^2} = q \quad (2.68)$$

where n is the constant horizontal component of the axial force, tension positive. The trial solution for u_2 is chosen by inspection:

$$u_2 = u_0 \sin \frac{\pi x}{L} \quad (2.69)$$

Substitution of (2.69) into (2.68) yields the expression for the applied load q_0 :

$$q_0 = u_0 \left(EJ \left(\frac{\pi}{L} \right)^4 + n \left(\frac{\pi}{L} \right)^2 \right) \quad (2.70)$$

The horizontal component of the axial force in the beam is computed from the axial strain in the instant configuration of the beam. The horizontal component of the axial strain in element AB of figure 2.9 is given by:

$$\varepsilon_h = \left(\sqrt{1+u_{2,1}^2} - 1 \right) \frac{1}{\sqrt{1+u_{2,1}^2}} = 1 - \frac{1}{\sqrt{1+u_{2,1}^2}} \cong \frac{1}{2} \left(\frac{du_2}{dx_1} \right)^2 \quad (2.71)$$

The average horizontal component of the strain in the beam is:

$$\bar{\varepsilon}_h = \frac{1}{L} \int_0^L \frac{1}{2} u_0^2 \left(\frac{\pi}{L} \right)^2 \cos^2 \frac{\pi x_1}{L} dx_1 = \frac{\pi}{4} \left(\frac{u_0}{L} \right)^2 \quad (2.72)$$

The horizontal component of the axial force in the beam is:

$$n = AE \bar{\varepsilon}_h = AE \frac{\pi}{4} \left(\frac{u_0}{L} \right)^2 \quad (2.73)$$

Result (2.73) is substituted into expression (2.70) for the load amplitude:

$$\begin{aligned} q_0 &= JEL \left(\frac{\pi}{L} \right)^4 \left(\frac{u_0}{L} \right) + AEL \left(\frac{\pi}{4} \right) \left(\frac{\pi}{L} \right)^2 \left(\frac{u_0}{L} \right)^3 \\ &= \frac{\pi^4 JE}{L^3} \left(\frac{u_0}{L} \right) \left(1 + \frac{L^3}{\pi^4 EJ} \frac{\pi}{4} AEL \left(\frac{\pi}{L} \right)^2 \left(\frac{u_0}{L} \right)^2 \right) \end{aligned}$$

The load q_0 and displacement u_2 are replaced by normalised variables \bar{q} and \bar{u} :

$$\bar{q} = \frac{q_0 L^3}{\pi^4 EJ} \quad (2.74)$$

$$\bar{u} = \frac{u_0}{L} \quad (2.75)$$

$$c = \frac{L^3}{\pi^4 EJ} \frac{\pi}{4} EAL \left(\frac{\pi}{L} \right)^2 = \frac{1}{4\pi} \left(\frac{L}{r} \right)^2 \quad \text{with } r^2 = \frac{J}{A} \quad (2.76)$$

The normalised load on the pinned beam is thus:

$$\bar{q} = \bar{u}(1 + c\bar{u}^2) \quad (2.77)$$

The normalised stiffness \bar{k} of the beam is defined as follows:

$$\bar{k} = \frac{\bar{q}}{\bar{u}} = 1 + c\bar{u}^2 \quad (2.78)$$

The normalised axial force \bar{n} in the beam is defined as follows:

$$\bar{n} = \frac{n}{q_0 L} = \frac{c\bar{u}}{\pi^2(1 + c\bar{u}^2)} \quad (2.79)$$

Figure 2.10 shows the influence of the normalised displacement \bar{u} on the normalised stiffness \bar{k} for different values of the slenderness ratio L/r :

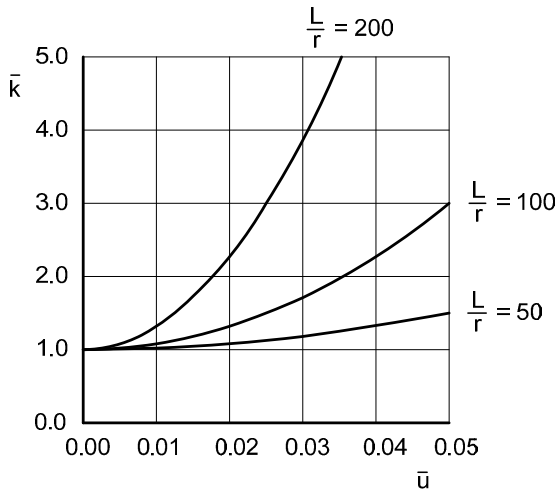


Figure 2.10 Influence of the displacement \bar{u} on the stiffness \bar{k} of a pinned beam

Figure 2.11 shows the influence of the normalised displacement \bar{u} on the normalised axial force \bar{n} for different values of the slenderness ratio L/r :

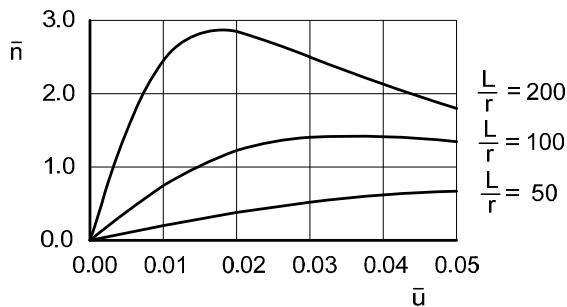


Figure 2.11 Influence of the displacement \bar{u} on the axial force \bar{n} of a pinned beam

Conclusions: The linear beam theory does not predict any difference between the behaviour of a pinned beam and a simply supported beam. Nonlinear analysis of the pinned beam shows that there are significant errors in the behaviour predicted by linear analysis. The errors are due to the stretching of the axis of the pinned beam, which is ignored in the linear theory. This stretching affects both the stiffness and the axial force of the pinned beam. An analogous effect does not exist in simply supported beams. The real behaviour of a pinned beam differs strongly from that of a simply supported beam.

2.4 BIFURCATION INSTABILITY OF A COLUMN

Consider a rigid column of length a , supported by a spring with rotational stiffness k at one end and free at the other end. In the reference configuration the axis of the column lies on the global axis x_1 . The column is subjected to an axial load P_1 in the direction of the negative axis x_1 , as shown in figure 2.12. Let the angle of rotation ω of the column about its support be positive in the direction from axis x_1 to axis x_2 . In addition to the load P_1 , the column carries a perturbation load P_2 that acts in the positive direction of axis x_2 for positive rotation ω , and in the negative direction of axis x_2 for negative rotation ω .

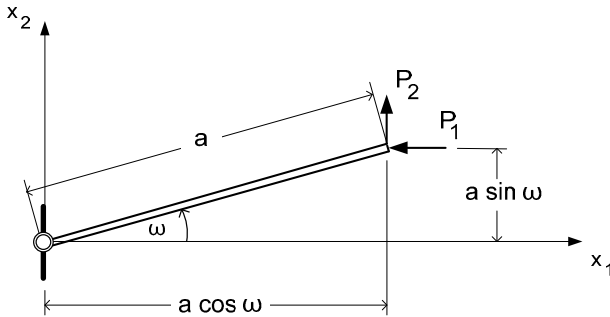


Figure 2.12 Rigid cantilever column with a rotation spring support

The moment about the support due to the loads P_1 and P_2 is in equilibrium with the moment the rotation causes in the spring:

$$P_1 a \sin \omega + P_2 a \cos \omega = k \omega \quad (2.80)$$

Let P_2 be a fraction m of P_1 and define the dimensionless load p . Equation (2.80) is transformed into (2.81):

$$p(\sin \omega + m \cos \omega) = 0 \quad (2.81)$$

$$p = \frac{P_1 a}{k} \quad (2.82)$$

$$P_2 = m P_1 \quad (2.83)$$

The solution of the nonlinear governing equation (2.81) for a given load p depends on the value of the parameter m , as follows:

Case 1: unperturbed column with $m = 0$

The character of the solutions of (2.81) is determined by substitution of the power series for $\sin \omega$:

$$p \left(\omega - \frac{1}{3!} \omega^3 + \frac{1}{5!} \omega^5 - \dots \right) = \omega$$

$$\omega \left(p \left(1 - \frac{1}{6} \omega^2 + \frac{1}{120} \omega^4 - \dots \right) - 1 \right) = 0 \quad (2.84)$$

Solution 1.1: $\omega = 0$:

The column is in equilibrium for every value of load p .

Solution 1.2: $\omega \neq 0$:

The second factor on the left hand side of (2.84) is set to null. The equation is solved for ω^2 :

$$\frac{1}{6} \omega^2 - \frac{1}{120} \omega^4 = 1 - \frac{1}{p} \quad (2.85)$$

The left-hand side of (2.85) is not negative for real values of ω . The smallest value of the load p for which equation (2.85) has real roots is therefore $p = 1.0$. This value is called the critical load of the column and denoted by p_c .

$$p_c = 1.0 \quad (2.86)$$

For loads $p < p_c$ there is no real value $\omega \neq 0$ of the rotation for which the column is in equilibrium. For $p \geq p_c$, the column has three equilibrium configurations. The rotations are $\omega = 0$ and the values of equal magnitude but opposite sign, which satisfy (2.87):

$$p = \frac{\omega}{\sin \omega} \quad (2.87)$$

The solution for $m = 0$ is shown by curve "d" in figure 2.13. The limit point $(0.0, p_c)$ is a bifurcation point, since there are three infinitesimal rotation increments that lead to equilibrium configurations of the column at constant load. The figure shows that the load that the column can carry on the load branches BC, BD and BE increases as the rotation ω increases, so that the limit point is not a point of failure.

At the limit point B, the derivative of the load p with respect to the rotation ω is null. The second derivative of p with respect to ω must be positive at the limit point since the load increases on all three branches. Differentiation of (2.87) yields:

$$\frac{dp}{d\omega} = \frac{\sin \omega - \omega \cos \omega}{\sin^2 \omega} \quad (2.88)$$

$$\begin{aligned} \frac{d^2p}{d\omega^2} &= \frac{\omega \sin^3 \omega - 2 \sin \omega \cos \omega (\sin \omega - \omega \cos \omega)}{\sin^4 \omega} \\ &= \frac{\omega}{\sin \omega} - \frac{2 \cos \omega}{\sin^2 \omega} + \frac{2 \omega \cos^2 \omega}{\sin^3 \omega} \end{aligned} \quad (2.89)$$

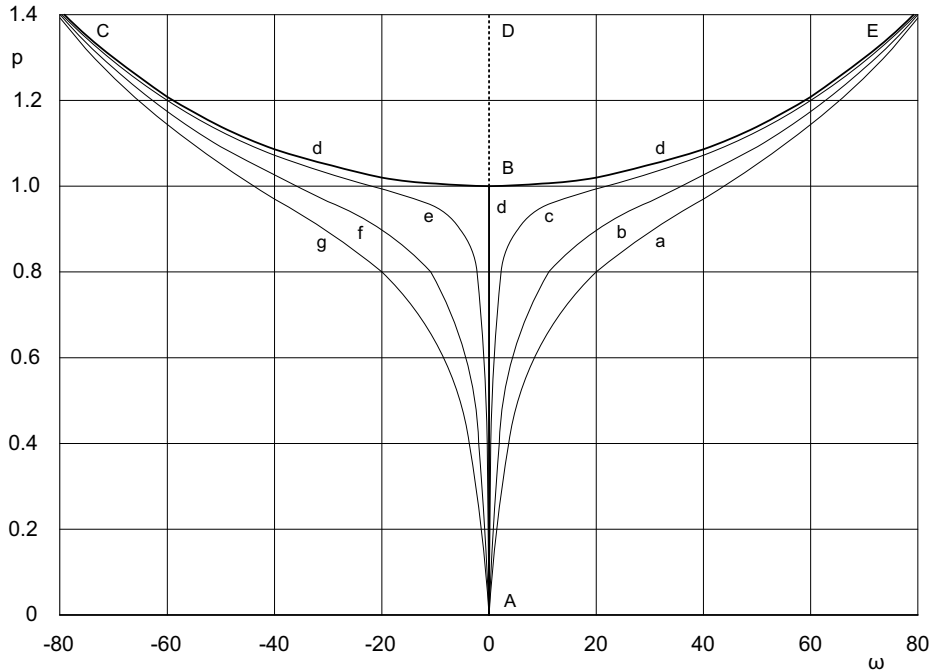


Figure 2.13 Bifurcation of a rigid cantilever column with spring support
Variation of the rotation ω with the load p for load ratio $m = P_1/P_2$

a	$m = 0.10$	e	$m = -0.01$
b	$m = 0.05$	f	$m = -0.05$
c	$m = 0.01$	g	$m = -0.10$
d	$m = 0.0$		

The limits of the load p in (2.87) and its derivatives in (2.88) and (2.89) as the rotation ω goes to null are determined with the rule of de l'Hospital:

$$\lim_{\omega \rightarrow 0} p = \lim_{\omega \rightarrow 0} \frac{1}{\cos \omega} = 1.0$$

$$\lim_{\omega \rightarrow 0} \frac{dp}{d\omega} = \lim_{\omega \rightarrow 0} \frac{\omega}{2 \cos \omega} = 0.0$$

$$\lim_{\omega \rightarrow 0} \frac{2 \cos \omega}{\sin^2 \omega} = \lim_{\omega \rightarrow 0} \frac{-1}{\cos \omega} = -1.0$$

$$\lim_{\omega \rightarrow 0} \frac{\omega \cos^2 \omega}{\sin^3 \omega} = \lim_{\omega \rightarrow 0} \frac{-5}{6 \cos \omega} = -\frac{5}{6}$$

$$\lim_{\omega \rightarrow 0} \frac{d^2 p}{d\omega^2} = 1 + 1 - \frac{5}{3} = \frac{1}{3}$$

The second derivative of p with respect to ω is positive as required. The question now is which of the branches of the load path the column will follow after it has reached the singular point B. This question is studied in case 2 by determining the influence of a small transverse perturbation load P_2 on the behaviour of the column.

Case 2: perturbed column with $m \neq 0$

Equation (2.81) is solved directly for the load p as a function of the rotation ω :

$$p = \frac{\omega}{\sin \omega + m \cos \omega} \tag{2.90}$$

The perturbation load P_2 has been specified in such a way that the sign of the variables ω , $\sin \omega$ and m is always the same. The trigonometric ratio $\cos \omega$ is positive for the angles in figure 2.13. The sign of the rotation can therefore be ignored in expression (2.90). The denominator of (2.90) differs from null. The value of p is positive for $\omega > 0$ and equals null for $\omega = 0$. The variation of p with ω for different values of parameter m is shown in figure 2.13.

It should be noted that for $m > 0$, there is no solution corresponding to branch BD of diagram "d" for $m = 0$. If the column is thus loaded with $P_2 = 0$ to a point on BD and a small perturbation load such as $P_2 = 0.01 * P_1$ is applied, the column undergoes a large rotation until an equilibrium configuration is reached on curve "c". Branch BD of the load diagram is thus unstable and will not be followed in nature.

The example shows that the load on a structure need not necessarily decrease after bifurcation. In the case of the 2-bar truss, the load can be decreased after bifurcation. In the case of the cantilever column, the load can only be increased after bifurcation.

2.5 CONTACT OF TWO RODS

Figure 2.14 shows collinear rods AB and CD of equal length. Their free ends B and C are separated by a gap of width s . Rod AB with axial stiffness AE is fixed at node A and carries an axial load P at node B. Rod CD with axial stiffness $3AE$ is fixed at node D. The displacement u of node B in the axial direction is to be determined as a function of the applied load P .

Consider bar AB while the gap is open. The displacement of node B is given by:

$$u = \frac{Pa}{AE} \tag{2.91}$$

The gap closes when the load reaches the value P_0 which makes $u = s$:

$$P_0 = \frac{AEs}{a} \tag{2.92}$$

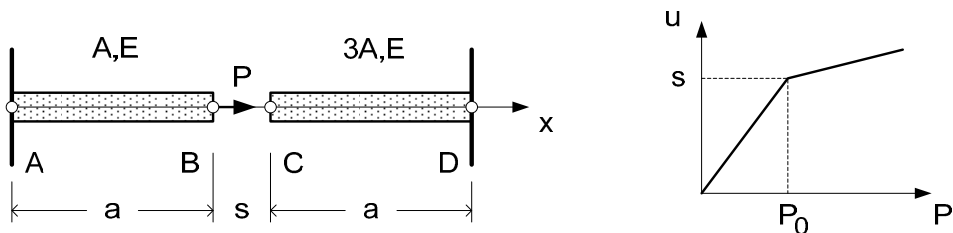


Figure 2.14 Displacement u of node B due to axial load P

If the load is increased further, the displacement of node B is given by:

$$u = s + \frac{(P - P_0)a}{4EA} \quad (2.93)$$

The nonlinear load - displacement diagram is shown in figure 2.14.

2.6 TENSION - ONLY MEMBER

Figure 2.15 shows bars AB and BC that are connected at node B. Rod AB has length a and axial stiffness EA_1 . Node A is fixed. The tension-only member BC has length b and is fixed at node C. The axial stiffness of BC is null if the length of the member is less than b and EA_2 otherwise. The displacement of node B is to be determined as a function of the axial force P that acts at node B.

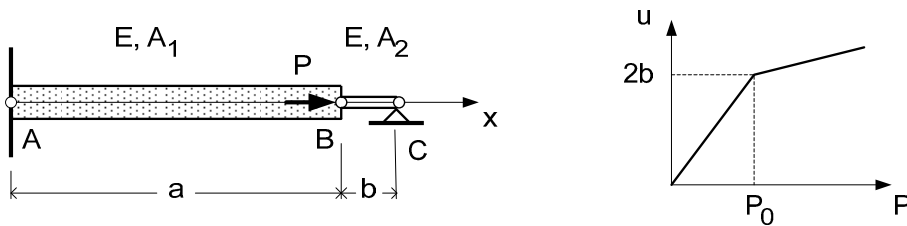


Figure 2.15 Displacement u of node B due to axial load P

If the displacement of node B is less than $2b$, the length of member BC is less than b and the force in BC is null. The displacement of node B is given by:

$$u = \frac{Pa}{EA_1} \quad (2.94)$$

Bar BC regains its original length when the load reaches a value P_0 so that $u = 2b$:

$$P_0 = 2EA_1 \frac{b}{a} \quad (2.95)$$

If the load is increased further, both members are in tension and the displacement is:

$$u = 2b + \frac{P - P_0}{E(A_1 + A_2)} \quad (2.96)$$

The nonlinear load - displacement diagram is shown in figure 2.15.

3 NONLINEAR THEORY OF ELASTICITY

3.1 COORDINATE SYSTEMS

3.1.1 Global Coordinate System

Consider a three-dimensional Euclidian space with orthonormal base vectors $\mathbf{i}_1, \mathbf{i}_2, \mathbf{i}_3$. The location of a point P in the space is described by means of a Cartesian coordinate system with a fixed origin O, as shown in figure 3.1.

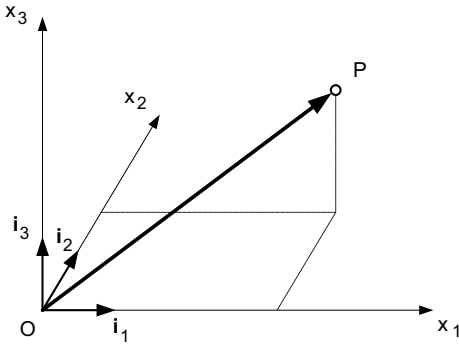


Figure 3.1 Cartesian global coordinate system

The vector from the origin O to point P is called the location vector of point P and is denoted by \mathbf{x} . The location vector is expressed as a linear combination of the base vectors \mathbf{i}_k . The coefficients of the linear combination are called the global coordinates of point P and denoted by x_1, x_2, x_3 .

$$\mathbf{x} = x_1 \mathbf{i}_1 + x_2 \mathbf{i}_2 + x_3 \mathbf{i}_3 \quad (3.1)$$

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = x_1 \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + x_2 \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + x_3 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

- \mathbf{x} location vector of point P
- x_k global coordinate of point P for base vector \mathbf{i}_k
- \mathbf{i}_k base vector of the Euclidian space

The base vectors are the partial derivatives of the location vector \mathbf{x} with respect to the coordinates x_k :

$$\mathbf{i}_k = \frac{\partial \mathbf{x}}{\partial x_k} := \mathbf{x}_{,k} \quad (3.2)$$

The entity consisting of the origin O and the base vectors $\mathbf{i}_1, \mathbf{i}_2, \mathbf{i}_3$ is called a global coordinate system.

3.1.2 Local Coordinate System

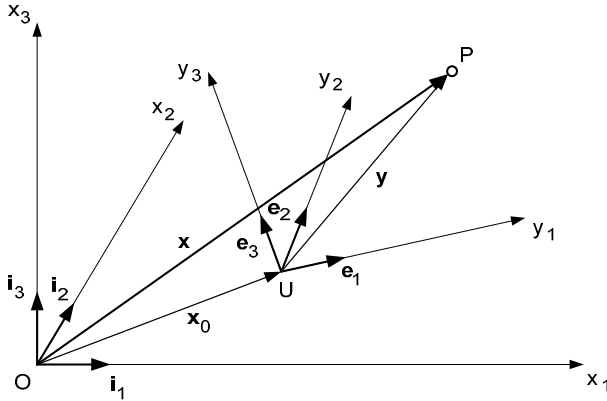


Figure 3.2 Cartesian local coordinate system

Let $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$ be orthonormal vectors that form a basis for the Euclidian space in figure 3.1, and let U be a point in the Euclidian space whose location vector in the global coordinate system is \mathbf{x}_0 . The Cartesian coordinate system with origin U and base vectors \mathbf{e}_k is called a local coordinate system. The coordinates e_{ik} of the base vectors \mathbf{e}_k are specified in the global space:

$$\mathbf{e}_k = e_{1k}\mathbf{i}_1 + e_{2k}\mathbf{i}_2 + e_{3k}\mathbf{i}_3 \quad k \in \{1,2,3\} \quad (3.3)$$

e_{ik} coordinates of the local base vector \mathbf{e}_k

The location vector of a point P in the local coordinate system is denoted by \mathbf{y} . The location vector is expressed as a linear combination of unit vectors $\mathbf{j}_1, \mathbf{j}_2, \mathbf{j}_3$, which are identical to $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$, but whose coordinates are referred to the local coordinate system. The coefficients of the linear combination are called the local coordinates of point P and denoted by y_1, y_2, y_3 .

$$\mathbf{y} = y_1\mathbf{j}_1 + y_2\mathbf{j}_2 + y_3\mathbf{j}_3 \quad (3.4)$$

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = y_1 \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + y_2 \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + y_3 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

\mathbf{y} location vector of point P
 y_k local coordinate of point P for base vector \mathbf{j}_k
 \mathbf{j}_k local base vector of the Euclidian space

The location vector \mathbf{x} is the vector sum of the location vectors \mathbf{x}_0 and \mathbf{y} . The coordinates of the vectors can be added if they are referred to the same basis. The vectors $\mathbf{j}_1, \mathbf{j}_2, \mathbf{j}_3$ are therefore mapped to $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$:

$$\mathbf{x} = \mathbf{x}_0 + y_1 \mathbf{e}_1 + y_2 \mathbf{e}_2 + y_3 \mathbf{e}_3 \quad (3.5)$$

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} x_{10} \\ x_{20} \\ x_{30} \end{bmatrix} + y_1 \begin{bmatrix} e_{11} \\ e_{21} \\ e_{31} \end{bmatrix} + y_2 \begin{bmatrix} e_{12} \\ e_{22} \\ e_{32} \end{bmatrix} + y_3 \begin{bmatrix} e_{13} \\ e_{23} \\ e_{33} \end{bmatrix}$$

x_{k0} global coordinates of the local origin

The base vectors \mathbf{e}_k are arranged in the columns of a matrix which is called rotation matrix and denoted by \mathbf{R} . Equation (3.5) is rewritten using \mathbf{R} :

$$\mathbf{x} = \mathbf{x}_0 + \mathbf{R} \mathbf{y} \quad (3.6)$$

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} x_{10} \\ x_{20} \\ x_{30} \end{bmatrix} + \begin{bmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \\ e_{31} & e_{32} & e_{33} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$$

\mathbf{x} global location vector of point P

\mathbf{y} local location vector of point P

\mathbf{x}_0 global location vector of the local origin

\mathbf{R} rotation matrix for the coordinate transformation

Since the base vectors \mathbf{e}_k are orthonormal, the rotation matrix is orthonormal and its inverse is equal to its transpose:

$$\mathbf{e}_i^T \mathbf{e}_m = \delta_{im} \quad (3.7)$$

$$\mathbf{R}^T \mathbf{R} = \mathbf{I} \quad (3.8)$$

$$\mathbf{R}^{-1} = \mathbf{R}^T \quad (3.9)$$

$$\begin{bmatrix} \mathbf{e}_1^T \\ \mathbf{e}_2^T \\ \mathbf{e}_3^T \end{bmatrix} * \begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 & \mathbf{e}_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

3.1.3 Coordinate Transformations

Physical quantities such as displacements, which are tensors of the first order, are expressed as linear combinations of base vectors. Other physical quantities such as stresses, which are tensors of the second order, are expressed as sums of vector products of base vectors. The coordinates in these expressions change if the basis is changed. The laws of transformation for the coordinates will now be determined.

Consider a vector denoted by \mathbf{u} if it is expressed as a linear combination of the global base vectors \mathbf{i}_k , and by \mathbf{v} if it is expressed as a linear combination of the local base vectors \mathbf{j}_k . The coefficients of the linear combinations are called the coordinates of the vector and denoted by u_k and v_k .

$$\mathbf{u} = u_1 \mathbf{i}_1 + u_2 \mathbf{i}_2 + u_3 \mathbf{i}_3 \quad (3.10)$$

$$\mathbf{v} = v_1 \mathbf{j}_1 + v_2 \mathbf{j}_2 + v_3 \mathbf{j}_3$$

The local base vectors are mapped to the global coordinate system with (3.3). This yields the transformation between the global and the local coordinates of the vector:

$$\mathbf{u} = v_1 \mathbf{e}_1 + v_2 \mathbf{e}_2 + v_3 \mathbf{e}_3 \quad (3.11)$$

$$\mathbf{u} = \mathbf{R} \mathbf{v} \quad (3.12)$$

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} | & | & | \\ \mathbf{e}_1 & \mathbf{e}_2 & \mathbf{e}_3 \\ | & | & | \end{bmatrix} * \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}$$

Consider a matrix denoted by \mathbf{U} if it is expressed as a linear combination of vector products of the global base vectors \mathbf{i}_k , and by \mathbf{V} if it is expressed as a linear combination of vector products of the local base vectors \mathbf{j}_k . The coefficients of the linear combinations of the vector products are called the coordinates of the matrix and denoted by u_{km} and v_{km} .

$$\mathbf{U} = \sum_{k=1}^3 \sum_{m=1}^3 u_{km} \mathbf{i}_k \mathbf{i}_m^T \quad (3.13)$$

$$\mathbf{V} = \sum_{k=1}^3 \sum_{m=1}^3 v_{km} \mathbf{j}_k \mathbf{j}_m^T$$

The local base vectors are mapped to the global coordinate system with (3.3). This yields the transformation between the global and the local coordinates of the matrix:

$$\mathbf{U} = \sum_{k=1}^3 \sum_{m=1}^3 v_{km} \mathbf{e}_k \mathbf{e}_m^T$$

$$\mathbf{U} = \mathbf{R} \mathbf{V} \mathbf{R}^T \quad (3.14)$$

- \mathbf{U} matrix containing the global tensor coordinates u_{km}
- \mathbf{V} matrix containing the local tensor coordinates v_{km}

3.2 CONFIGURATIONS OF A BODY

3.2.1 Body

Continuum: A point of Euclidian space occupied by material is called a material point. A set of material points that occupy a subspace of the Euclidian space is called a body. A subspace that is compact and connected is called a continuum. In the following, only bodies that occupy a continuum are considered.

Solid body: A body can be solid or fluid. The two body states differ in the change of topology with time. Neighbouring material points of a solid body remain neighbouring points independent of time. Points of a fluid body change their neighbourhood with time. In the following, only solid bodies are considered.

Rigid body: A body is called rigid if the distance between any two points of the body remains constant as the body is displaced in space. It will be shown in section 3.3.2 that the rigid body motion of a body can be decomposed into a translation and a rotation.

Deformable body: A body is called deformable if the distance between any two points of the body can change as the body is displaced in space. It will be shown in section 3.4.2 that the deformation of a body can be decomposed into the stretching of fibres and angle changes between fibres of the body. In the following, only deformable bodies are considered.

Surface of a body: The compact subspace a body occupies in Euclidian space is closed. The boundary points of the subspace are considered to be points of the body. The set containing the boundary points of the body is called the surface of the body.

3.2.2 Configurations

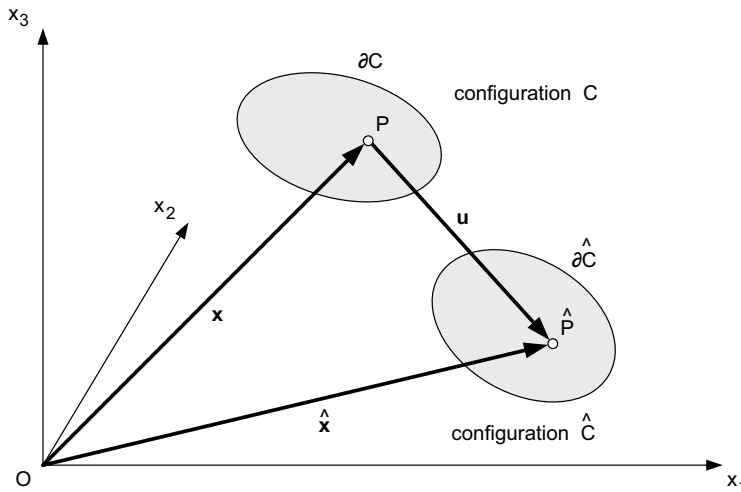


Figure 3.3 Configurations of a body

Reference configuration: Consider a body at a freely chosen point in time, which is conventionally denoted by $t = 0$. The subspace that the body occupies at this point in time, as shown in figure 3.3, is called the reference configuration of the body and denoted by C . The surface of the body in the reference configuration is denoted by ∂C . The location vector of a material point P in the reference configuration is denoted by \mathbf{x} , and its coordinates in global space are denoted by x_1, x_2, x_3 :

$$\mathbf{x} = x_1 \mathbf{i}_1 + x_2 \mathbf{i}_2 + x_3 \mathbf{i}_3 \quad (3.15)$$

x_k coordinates of a material point P in the reference configuration C

Instant configuration: The location of a material point P of a body in a Euclidian space changes with time if the body is subjected to prescribed displacements and forces at its surface. The location of a material point at an arbitrary point in time is called the instant location of P and denoted by \hat{P} . The location vector of the displaced point \hat{P} is denoted by $\hat{\mathbf{x}}$:

$$\hat{\mathbf{x}} = \hat{x}_1 \hat{\mathbf{i}}_1 + \hat{x}_2 \hat{\mathbf{i}}_2 + \hat{x}_3 \hat{\mathbf{i}}_3 \quad (3.16)$$

\hat{x}_k coordinates of a material point \hat{P} in the instant configuration \hat{C}

The set of instant locations of the points of a body at a specific point in time t is called an instant configuration of the body and denoted by \hat{C} . The boundary of an instant configuration of a body is denoted by $\partial\hat{C}$.

3.3 STATE OF DISPLACEMENT

3.3.1 Displacement

The difference between the location vector $\hat{\mathbf{x}}$ of a material point in an instant configuration \hat{C} and its location vector \mathbf{x} in the reference configuration C is called the displacement vector of the material point. It is denoted by \mathbf{u} . Different material points of the body have different displacement vectors. The set of displacement vectors for all points of the body in the instant configuration \hat{C} is called the state of displacement in configuration \hat{C} .

$$\mathbf{u} = \hat{\mathbf{x}} - \mathbf{x} \quad (3.17)$$

$$\mathbf{u} = u_1 \hat{\mathbf{i}}_1 + u_2 \hat{\mathbf{i}}_2 + u_3 \hat{\mathbf{i}}_3 \quad (3.18)$$

$$u_k = \hat{x}_k - x_k \quad (3.19)$$

\mathbf{u} displacement vector of material point P
 u_k global coordinates of the displacement vector \mathbf{u}

3.3.2 Rigid Body Translation and Rotation

Translation: A body translates as a rigid body if all points of the body experience the same displacement \mathbf{u} from the reference configuration C to the instant configuration \hat{C} . Let the location vectors of any two points A and B in C be \mathbf{x}_A and \mathbf{x}_B . The square of the distance between points A and B in the reference configuration is given by:

$$s^2 = (\mathbf{x}_B - \mathbf{x}_A)^T (\mathbf{x}_B - \mathbf{x}_A)$$

The location vectors of points A and B in \hat{C} are $\mathbf{x}_A + \mathbf{u}$ and $\mathbf{x}_B + \mathbf{u}$. The square of the distance between points A and B in the instant configuration is given by:

$$\begin{aligned} \hat{s}^2 &= ((\mathbf{x}_B + \mathbf{u}) - (\mathbf{x}_A + \mathbf{u}))^T ((\mathbf{x}_B + \mathbf{u}) - (\mathbf{x}_A + \mathbf{u})) \\ &= (\mathbf{x}_B - \mathbf{x}_A)^T (\mathbf{x}_B - \mathbf{x}_A) = s^2 \end{aligned} \quad (3.20)$$

Since $s = \hat{s}$, the translation is a rigid body motion.

Rotation matrix: Consider the rotation of a material point P about a straight axis defined by the unit vector \mathbf{a} shown in figure 3.4. Let U be an arbitrary point on the axis and let the perpendicular from P cut the axis in point M . The length of \overline{PM} is called the radius of rotation and denoted by r .

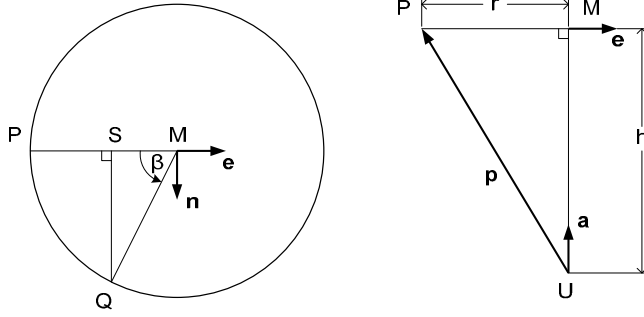


Figure 3.4 Rotation of point P about axis **a**

Let P rotate anti-clockwise about M through an angle β to position Q. Construct the perpendicular \overline{QS} to \overline{PM} . Let \mathbf{e} be a unit vector in the direction from P to M and \mathbf{n} a unit vector normal to \mathbf{e} . Vector \mathbf{n} is determined with the cross-product of vectors \overline{UM} and \overline{UP} , as follows:

$$h\mathbf{a} \times \mathbf{p} = hr\mathbf{n} \quad \rightarrow \quad \mathbf{n} = \frac{1}{r}\mathbf{a} \times \mathbf{p} \quad (3.21)$$

Unit vector \mathbf{e} is the cross-product of vectors \mathbf{a} and \mathbf{n} . Vector \overline{UQ} is the sum of the vectors $\overline{UP}, \overline{PS}, \overline{SQ}$ and denoted by $\hat{\mathbf{p}}$.

$$\begin{aligned} \mathbf{e} &= \mathbf{a} \times \mathbf{n} = \frac{1}{r}\mathbf{a} \times (\mathbf{a} \times \mathbf{p}) \\ \hat{\mathbf{p}} &= \mathbf{p} + r\mathbf{e}(1 - \cos\beta) + r\mathbf{n}\sin\beta \\ \hat{\mathbf{p}} &= \mathbf{p} + (1 - \cos\beta)\mathbf{a} \times (\mathbf{a} \times \mathbf{p}) + \sin\beta \mathbf{a} \times \mathbf{p} \end{aligned} \quad (3.22)$$

The cross-product $\mathbf{a} \times \mathbf{p}$ is written in matrix form:

$$\mathbf{a} \times \mathbf{p} = \mathbf{A}\mathbf{p} \quad (3.23)$$

$$\begin{array}{|c|} \hline a_2p_3 - a_3p_2 \\ \hline a_3p_1 - a_1p_3 \\ \hline a_1p_2 - a_2p_1 \\ \hline \end{array} = \begin{array}{|ccc|} \hline 0 & -a_3 & a_2 \\ \hline a_3 & 0 & -a_1 \\ \hline -a_2 & a_1 & 0 \\ \hline \end{array} * \begin{array}{|c|} \hline p_1 \\ \hline p_2 \\ \hline p_3 \\ \hline \end{array}$$

Substitution of (3.23) into (3.22) yields the rotation matrix \mathbf{R} for axis \mathbf{a} and angle β :

$$\hat{\mathbf{p}} = \mathbf{R}\mathbf{p} \quad (3.24)$$

$$\mathbf{R} = \mathbf{I} + \mathbf{A}\sin\beta + \mathbf{A}^2(1 - \cos\beta) \quad (3.25)$$

The direction vector \mathbf{a} and the associated matrix A have the following properties:

$$\begin{aligned} \mathbf{a}^T \mathbf{a} &= 1 & \mathbf{A} &= -\mathbf{A}^T \\ \mathbf{A}^T \mathbf{A} &= \mathbf{I} - \mathbf{a}\mathbf{a}^T & (\mathbf{A}^T \mathbf{A})^2 &= \mathbf{A}^T \mathbf{A} \end{aligned}$$

These expressions are substituted into the product $\mathbf{R}^T \mathbf{R}$ formed with (3.25):

$$\mathbf{R}^T \mathbf{R} = \mathbf{I} + 2\mathbf{A}^2(1 - \cos\beta) + \mathbf{A}^T \mathbf{A}(\sin^2\beta + (1 - \cos\beta)^2) = \mathbf{I} \quad (3.26)$$

The rotation matrix \mathbf{R} is orthonormal.

Rotation: The displacement vector of material point P due to rotation β about axis \mathbf{a} is the difference of the location vectors of points P and Q:

$$\mathbf{u} = \hat{\mathbf{p}} - \mathbf{p} = (\mathbf{R} - \mathbf{I})\mathbf{p} \quad (3.27)$$

Let all material points of a body rotate about the same axis \mathbf{a} through the same angle β . Consider material points A and B with location vectors \mathbf{x}_A and \mathbf{x}_B in the reference configuration. The square of their distance in the reference configuration is:

$$s^2 = (\mathbf{x}_B - \mathbf{x}_A)^T (\mathbf{x}_B - \mathbf{x}_A)$$

The square of the distance between points A and B in the instant configuration is:

$$\begin{aligned} \hat{s}^2 &= ((\mathbf{x}_B + \mathbf{u}_B) - (\mathbf{x}_A + \mathbf{u}_A))^T ((\mathbf{x}_B + \mathbf{u}_B) - (\mathbf{x}_A + \mathbf{u}_A)) \\ &= (\mathbf{x}_B - \mathbf{x}_A)^T \mathbf{R}^T \mathbf{R} (\mathbf{x}_B - \mathbf{x}_A) \\ &= (\mathbf{x}_B - \mathbf{x}_A)^T (\mathbf{x}_B - \mathbf{x}_A) \end{aligned}$$

Since $s = \hat{s}$, the rotation is a rigid body motion.

3.4 STATE OF STRAIN

3.4.1 Material Elements (Method of Lagrange)

Concept: Each material point of the body is identified by its location in the reference configuration of the body. A material element is defined by the infinitesimal volume that its material points occupy in the reference configuration. The edges of the volume are parallel to the global coordinate axes, as shown in figure 3.5. The diagonal $d\mathbf{x}$ of the volume is the difference between the location vectors of opposite corner points of the brick shape.

$$d\mathbf{x} = dx_1 \mathbf{i}_1 + dx_2 \mathbf{i}_2 + dx_3 \mathbf{i}_3 \quad (3.28)$$

$$dx_k = x_{k(Q)} - x_{k(P)}$$

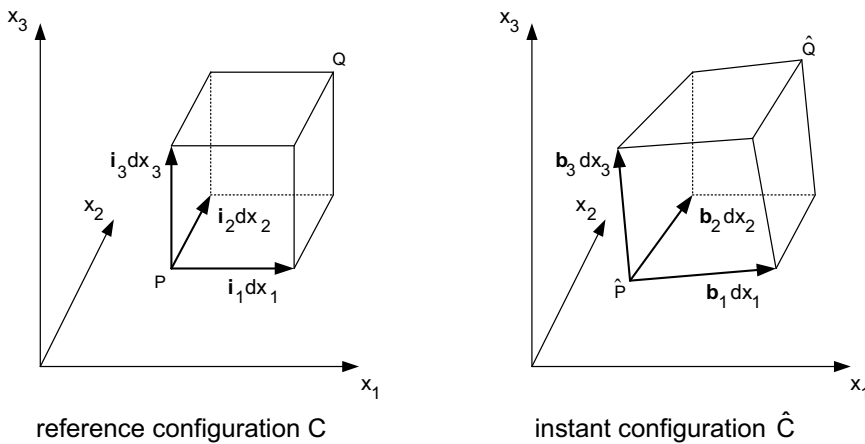


Figure 3.5 Material elements of Lagrange

The material element is displaced in the instant configuration to another volume in Euclidian space that is no longer brick-shaped. The diagonal vector $d\hat{\mathbf{x}}$ of this new volume is given by:

$$d\hat{\mathbf{x}} = dx_1 \mathbf{b}_1 + dx_2 \mathbf{b}_2 + dx_3 \mathbf{b}_3 \quad (3.29)$$

\mathbf{b}_k instant base vector

Instant base vectors: In general, the vector \mathbf{b}_k is not a unit vector and is not parallel to the base vector \mathbf{i}_k . Since the material points are identified by their location in the reference configuration, \mathbf{b}_k at a point is the derivative of the instant location vector $\hat{\mathbf{x}}$ with respect to the reference coordinate x_k of the point. For a given state of displacement $\mathbf{u}(\mathbf{x})$ of the body, the instant base vectors follow from (3.17):

$$\begin{aligned} \mathbf{b}_k &= \frac{\partial \hat{\mathbf{x}}}{\partial x_k} = \frac{\partial (\mathbf{x} + \mathbf{u})}{\partial x_k} = \frac{\partial \mathbf{x}}{\partial x_k} + \frac{\partial \mathbf{u}}{\partial x_k} \\ \mathbf{b}_k &= \mathbf{i}_k + \frac{\partial u_1}{\partial x_k} \mathbf{i}_1 + \frac{\partial u_2}{\partial x_k} \mathbf{i}_2 + \frac{\partial u_3}{\partial x_k} \mathbf{i}_3 \end{aligned} \quad (3.30)$$

3.4.2 State of Deformation of Material Elements

Material deformation gradient: The instant coordinates of a material point are functions $\hat{x}_k(x_1, x_2, x_3)$ of the reference coordinates of the point. The diagonal $d\hat{\mathbf{x}}$ of a material element of the body in the instant configuration can therefore be expressed as a function of the diagonal $d\mathbf{x}$ in the reference configuration:

$$d\hat{\mathbf{x}} = \mathbf{F} d\mathbf{x} \quad (3.31)$$

$$\begin{array}{|c|} \hline d\hat{x}_1 \\ \hline d\hat{x}_2 \\ \hline d\hat{x}_3 \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline \frac{\partial \hat{x}_1}{\partial x_1} & \frac{\partial \hat{x}_1}{\partial x_2} & \frac{\partial \hat{x}_1}{\partial x_3} \\ \hline \frac{\partial \hat{x}_2}{\partial x_1} & \frac{\partial \hat{x}_2}{\partial x_2} & \frac{\partial \hat{x}_2}{\partial x_3} \\ \hline \frac{\partial \hat{x}_3}{\partial x_1} & \frac{\partial \hat{x}_3}{\partial x_2} & \frac{\partial \hat{x}_3}{\partial x_3} \\ \hline \end{array} * \begin{array}{|c|} \hline dx_1 \\ \hline dx_2 \\ \hline dx_3 \\ \hline \end{array}$$

The matrix \mathbf{F} is called the material deformation gradient of the displacement field. The columns of \mathbf{F} contain the instant base vectors \mathbf{b}_k as defined in (3.30).

$$\mathbf{F} = \begin{array}{|c|c|c|} \hline \mathbf{b}_1 & \mathbf{b}_2 & \mathbf{b}_3 \\ \hline \end{array} \quad (3.32)$$

\mathbf{F} deformation gradient of the instant configuration

Change of volume: The volume of a material element in the reference configuration in figure 3.5 is computed with the vector and scalar products of its edge vectors:

$$dv = (\mathbf{i}_1 dx_1) \times (\mathbf{i}_2 dx_2) \cdot (\mathbf{i}_3 dx_3) = dx_1 dx_2 dx_3$$

The volume of the same material element in the instant configuration is determined analogously with the edge vectors of the volume it occupies in the instant configuration:

$$d\hat{v} = (\mathbf{b}_1 dx_1) \times (\mathbf{b}_2 dx_2) \cdot (\mathbf{b}_3 dx_3) = \det \mathbf{F} dv \quad (3.33)$$

Change of area: Consider an infinitesimal material surface whose points occupy an infinitesimal area with normal \mathbf{n} and size da in the reference configuration. Let the same material points occupy an infinitesimal area with normal $\hat{\mathbf{n}}$ and size $d\hat{a}$ in an instant configuration of the body. The relationship between the two area vectors is determined by forming a volume with the area vector $d\mathbf{a}$ and an arbitrary infinitesimal material vector $d\mathbf{x}$:

$$\text{reference configuration:} \quad d\mathbf{a} = \mathbf{n} da \quad dv = d\mathbf{x}^T d\mathbf{a}$$

$$\text{instant configuration:} \quad d\hat{\mathbf{a}} = \hat{\mathbf{n}} d\hat{a} \quad d\hat{v} = d\hat{\mathbf{x}}^T d\hat{\mathbf{a}}$$

$$\text{transformations:} \quad d\hat{\mathbf{x}} = \mathbf{F} d\mathbf{x} \quad d\hat{v} = \det \mathbf{F} dv$$

$$\text{substitution:} \quad d\hat{v} = \det \mathbf{F} dv = \det \mathbf{F} d\mathbf{x}^T d\mathbf{a} \quad (3.34)$$

$$d\hat{v} = d\hat{\mathbf{x}}^T d\hat{\mathbf{a}} = d\mathbf{x}^T \mathbf{F}^T d\hat{\mathbf{a}} \quad (3.35)$$

Since (3.34) and (3.35) are valid for arbitrary vectors $d\mathbf{x}$, the areas in the reference and the instant configurations are related as follows:

$$\det \mathbf{F} d\mathbf{a} = \mathbf{F}^T d\hat{\mathbf{a}} \quad (3.36)$$

Displacement gradient: It follows from (3.30) that the deformation gradient \mathbf{F} can be split up into a unit matrix \mathbf{I} and a matrix containing the partial derivatives of the displacement coordinates. This matrix is called the displacement gradient of the instant configuration and denoted by \mathbf{H} :

$$\mathbf{F} = \mathbf{I} + \mathbf{H} \quad (3.37)$$

$$\mathbf{H} = \begin{array}{|c|c|c|} \hline \frac{\partial u_1}{\partial x_1} & \frac{\partial u_1}{\partial x_2} & \frac{\partial u_1}{\partial x_3} \\ \hline \frac{\partial u_2}{\partial x_1} & \frac{\partial u_2}{\partial x_2} & \frac{\partial u_2}{\partial x_3} \\ \hline \frac{\partial u_3}{\partial x_1} & \frac{\partial u_3}{\partial x_2} & \frac{\partial u_3}{\partial x_3} \\ \hline \end{array}$$

Strain tensor of Green: Consider the change in the square of the length of the diagonal of a material element between its reference and its instant configuration:

$$\begin{aligned} d\hat{s}^2 - ds^2 &= d\hat{\mathbf{x}}^T d\hat{\mathbf{x}} - d\mathbf{x}^T d\mathbf{x} \\ &= d\mathbf{x}^T ((\mathbf{I} + \mathbf{H})^T (\mathbf{I} + \mathbf{H}) - \mathbf{I}) d\mathbf{x} \\ &= d\mathbf{x}^T (\mathbf{H} + \mathbf{H}^T + \mathbf{H}^T \mathbf{H}) d\mathbf{x} \end{aligned}$$

The expression in brackets is taken as a measure of the deformation of the body at the point and denoted by $2\mathbf{E}$. Matrix \mathbf{E} contains the coordinates of the strain tensor of Green:

$$\mathbf{E} = \frac{1}{2}(\mathbf{H} + \mathbf{H}^T + \mathbf{H}^T\mathbf{H}) \quad (3.38)$$

$$e_{km} = \frac{1}{2}\left(\frac{\partial u_k}{\partial x_m} + \frac{\partial u_m}{\partial x_k} + \sum_i \frac{\partial u_i}{\partial x_k} \frac{\partial u_i}{\partial x_m}\right)$$

Interpretation of the strain coordinates: The base vector \mathbf{i}_1 is deformed into the instant base vector \mathbf{b}_1 . The square of the length of \mathbf{b}_1 is:

$$\begin{aligned} \mathbf{b}_1^T\mathbf{b}_1 &= \left(\mathbf{i}_1 + \frac{\partial u_1}{\partial x_1}\mathbf{i}_1 + \frac{\partial u_2}{\partial x_1}\mathbf{i}_2 + \frac{\partial u_3}{\partial x_1}\mathbf{i}_3\right)^T \left(\mathbf{i}_1 + \frac{\partial u_1}{\partial x_1}\mathbf{i}_1 + \frac{\partial u_2}{\partial x_1}\mathbf{i}_2 + \frac{\partial u_3}{\partial x_1}\mathbf{i}_3\right) \\ &= 1 + 2\frac{\partial u_1}{\partial x_1} + \left(\frac{\partial u_1}{\partial x_1}\right)^2 + \left(\frac{\partial u_2}{\partial x_1}\right)^2 + \left(\frac{\partial u_3}{\partial x_1}\right)^2 = 1 + 2e_{11} \end{aligned} \quad (3.39)$$

The strain e_{11} is a measure of the change in length of the base vector \mathbf{i}_1 and thus a measure of the stretching of a material fibre in the direction of \mathbf{i}_1 :

$$|\mathbf{b}_1| - |\mathbf{i}_1| \cong \sqrt{1 + 2e_{11}} - 1 \cong e_{11} \quad (3.40)$$

The base vectors \mathbf{i}_1 and \mathbf{i}_2 are orthogonal. They are deformed into the instant base vectors \mathbf{b}_1 and \mathbf{b}_2 . The angle θ between \mathbf{b}_1 and \mathbf{b}_2 is given by:

$$\cos \theta = \frac{\mathbf{b}_1^T\mathbf{b}_2}{|\mathbf{b}_1||\mathbf{b}_2|} \cong \mathbf{b}_1^T\mathbf{b}_2 \quad (3.41)$$

$$\begin{aligned} \mathbf{b}_1^T\mathbf{b}_2 &= \left(\mathbf{i}_1 + \frac{\partial u_1}{\partial x_1}\mathbf{i}_1 + \frac{\partial u_2}{\partial x_1}\mathbf{i}_2 + \frac{\partial u_3}{\partial x_1}\mathbf{i}_3\right)^T \left(\mathbf{i}_2 + \frac{\partial u_1}{\partial x_2}\mathbf{i}_1 + \frac{\partial u_2}{\partial x_2}\mathbf{i}_2 + \frac{\partial u_3}{\partial x_2}\mathbf{i}_3\right) \\ &= \frac{\partial u_1}{\partial x_2} + \frac{\partial u_2}{\partial x_1} + \frac{\partial u_1}{\partial x_1} \frac{\partial u_1}{\partial x_2} + \frac{\partial u_2}{\partial x_1} \frac{\partial u_2}{\partial x_2} + \frac{\partial u_3}{\partial x_1} \frac{\partial u_3}{\partial x_2} \\ &\cong 2e_{12} \end{aligned}$$

The strain e_{12} is a measure of the angle change θ between vectors \mathbf{i}_1 and \mathbf{i}_2 :

$$\begin{aligned} \cos \theta &= \sin\left(\frac{\pi}{2} - \theta\right) = 2e_{12} \\ \frac{\pi}{2} - \theta &\cong 2e_{12} \end{aligned} \quad (3.42)$$

The axial strains e_{kk} are thus a measure of the relative change in length of fibres that are parallel to the coordinate axes in the reference configuration. The shear strains e_{km} are a measure of the change in angle between fibres parallel to the axes x_k and x_m in the reference configuration.

Principal strain: The strain tensor \mathbf{E} is transformed according to (3.14) with the rotation matrix \mathbf{R} of the coordinate space. Consider a special rotation matrix \mathbf{N} for which the strain tensor becomes a diagonal matrix \mathbf{S} because all shear strains go to null. For this case (3.14) leads to a special eigenvalue problem:

$$\mathbf{E} \mathbf{N} = \mathbf{N} \mathbf{S}$$

$$\mathbf{E} \mathbf{n}_k = s_k \mathbf{n}_k \quad k \in \{1,2,3\} \quad (3.43)$$

$$\mathbf{N} = \begin{bmatrix} | & | & | \\ \mathbf{n}_1 & \mathbf{n}_2 & \mathbf{n}_3 \\ | & | & | \end{bmatrix} \quad \mathbf{S} = \begin{bmatrix} s_1 & 0 & 0 \\ 0 & s_2 & 0 \\ 0 & 0 & s_3 \end{bmatrix}$$

\mathbf{n}_k unit vector in the principal direction k
 s_k principal axial strain for the direction \mathbf{n}_k

The diagonal elements s_k of \mathbf{S} are called principal strains of the material element; the columns of \mathbf{N} are called principal directions of the state of strain.

Since the shear strains are null if the state of strain is referred to the principal directions, (3.43) shows that the angle between the principal fibres does not change between configurations C and \hat{C} . Expression (3.39) shows that the diagonal coefficients of \mathbf{S} are a measure of the stretching of the principal fibres.

3.4.3 Spatial Elements (Method of Euler)

Concept: Each material point of the body is identified by its location in the instant configuration of the body. A spatial element is defined by the infinitesimal volume its material points occupy in the instant configuration. The edges of the volume are parallel to the global coordinate axes, as shown in figure 3.6. The diagonal $d\hat{\mathbf{x}}$ of the volume is the difference of the location vectors of opposite corner points.

$$d\hat{\mathbf{x}} = d\hat{x}_1 \mathbf{i}_1 + d\hat{x}_2 \mathbf{i}_2 + d\hat{x}_3 \mathbf{i}_3 \quad (3.44)$$

$$d\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k(Q)} - \hat{\mathbf{x}}_{k(P)}$$

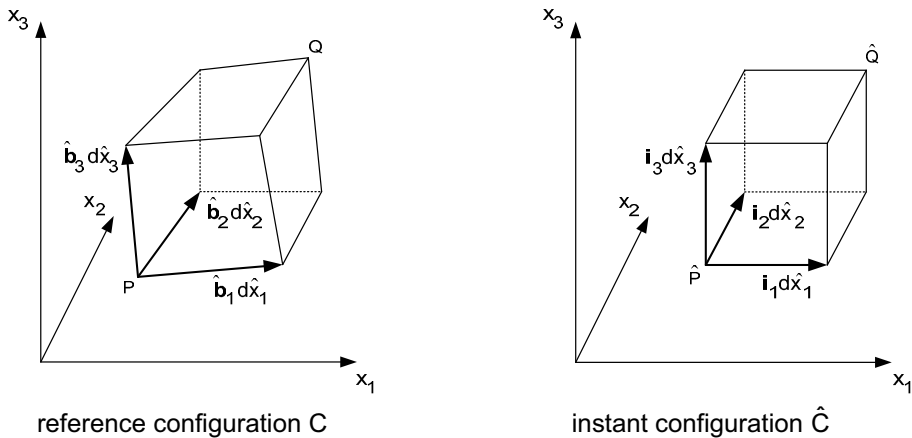


Figure 3.6 Spatial elements of Euler

The material points contained in the spatial element of the instant configuration have been displaced from their locations in the reference configuration. The volume they occupied in the reference configuration is not brick-shaped but continuous. The diagonal vector $d\mathbf{x}$ of this original volume is given by:

$$d\mathbf{x} = d\hat{x}_1 \hat{\mathbf{b}}_1 + d\hat{x}_2 \hat{\mathbf{b}}_2 + d\hat{x}_3 \hat{\mathbf{b}}_3 \quad (3.45)$$

$\hat{\mathbf{b}}_k$ reference base vector

Reference base vectors: In general, vector $\hat{\mathbf{b}}_k$ is not a unit vector and is not parallel to the base vector \mathbf{i}_k . Since the material points are identified by their location in the instant configuration, $\hat{\mathbf{b}}_k$ at a point is the derivative of the reference location vector $\hat{\mathbf{x}}$ with respect to the instant coordinate \hat{x}_k of the point. For a given state of displacement $\mathbf{u}(\hat{\mathbf{x}})$ of the body, the instant base vectors follow from (3.17):

$$\begin{aligned} \hat{\mathbf{b}}_k &= \frac{\partial \mathbf{x}}{\partial \hat{x}_k} = \frac{\partial (\hat{\mathbf{x}} - \mathbf{u})}{\partial \hat{x}_k} = \frac{\partial \hat{\mathbf{x}}}{\partial \hat{x}_k} - \frac{\partial \mathbf{u}}{\partial \hat{x}_k} \\ \hat{\mathbf{b}}_k &= \mathbf{i}_k - \frac{\partial u_1}{\partial \hat{x}_k} \mathbf{i}_1 - \frac{\partial u_2}{\partial \hat{x}_k} \mathbf{i}_2 - \frac{\partial u_3}{\partial \hat{x}_k} \mathbf{i}_3 \end{aligned} \quad (3.46)$$

3.4.4 State of Deformation of Spatial Elements

Spatial deformation gradient: The reference coordinates of a material point of a spatial element are functions $x_k(\hat{x}_1, \hat{x}_2, \hat{x}_3)$ of the instant coordinates of the point. The diagonal $d\mathbf{x}$ of the spatial element in the reference configuration can therefore be expressed as a function of the diagonal $d\hat{\mathbf{x}}$ in the instant configuration:

$$d\mathbf{x} = \hat{\mathbf{F}} d\hat{\mathbf{x}} \quad (3.47)$$

$$\begin{array}{|c|} \hline dx_1 \\ \hline dx_2 \\ \hline dx_3 \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline \frac{\partial x_1}{\partial \hat{x}_1} & \frac{\partial x_1}{\partial \hat{x}_2} & \frac{\partial x_1}{\partial \hat{x}_3} \\ \hline \frac{\partial x_2}{\partial \hat{x}_1} & \frac{\partial x_2}{\partial \hat{x}_2} & \frac{\partial x_2}{\partial \hat{x}_3} \\ \hline \frac{\partial x_3}{\partial \hat{x}_1} & \frac{\partial x_3}{\partial \hat{x}_2} & \frac{\partial x_3}{\partial \hat{x}_3} \\ \hline \end{array} * \begin{array}{|c|} \hline d\hat{x}_1 \\ \hline d\hat{x}_2 \\ \hline d\hat{x}_3 \\ \hline \end{array}$$

The matrix $\hat{\mathbf{F}}$ is called the spatial deformation gradient of the displacement field. The columns of the spatial deformation gradient contain the reference base vectors $\hat{\mathbf{b}}_k$, as defined in (3.45).

$$\hat{\mathbf{F}} = \begin{array}{|c|c|c|} \hline \hat{\mathbf{b}}_1 & \hat{\mathbf{b}}_2 & \hat{\mathbf{b}}_3 \\ \hline \end{array} \quad (3.48)$$

$\hat{\mathbf{F}}$ deformation gradient of the instant configuration

The material and spatial deformation gradients are inverse matrices:

$$d\mathbf{x} = \hat{\mathbf{F}} d\hat{\mathbf{x}} = \hat{\mathbf{F}} \mathbf{F} d\mathbf{x} \rightarrow \hat{\mathbf{F}} \mathbf{F} = \mathbf{I} \quad (3.49)$$

Change of volume and area: Expression (3.49) is used to derive the formulas for the change in volume and area of a spatial element from (3.33) and (3.36):

$$dv = \det \hat{\mathbf{F}} d\hat{v} \quad (3.50)$$

$$\det \hat{\mathbf{F}} d\hat{\mathbf{a}} = \hat{\mathbf{F}}^T d\mathbf{a} \quad (3.51)$$

- $d\mathbf{a}$ area increment in the reference configuration
- $d\hat{\mathbf{a}}$ area increment in the instant configuration
- dv volume increment in the reference configuration
- $d\hat{v}$ volume increment in the instant configuration
- \mathbf{F} material deformation gradient
- $\hat{\mathbf{F}}$ spatial deformation gradient

Displacement gradient: It follows from (3.46) that the deformation gradient $\hat{\mathbf{F}}$ can be split up into a unit matrix \mathbf{I} and a matrix containing the partial derivatives of the displacement coordinates. This matrix is called the displacement gradient of the reference configuration and denoted by $\hat{\mathbf{H}}$:

$$\hat{\mathbf{F}} = \mathbf{I} - \hat{\mathbf{H}} \quad (3.52)$$

$$\hat{\mathbf{H}} = \begin{array}{|c|c|c|} \hline \frac{\partial u_1}{\partial \hat{x}_1} & \frac{\partial u_1}{\partial \hat{x}_2} & \frac{\partial u_1}{\partial \hat{x}_3} \\ \hline \frac{\partial u_2}{\partial \hat{x}_1} & \frac{\partial u_2}{\partial \hat{x}_2} & \frac{\partial u_2}{\partial \hat{x}_3} \\ \hline \frac{\partial u_3}{\partial \hat{x}_1} & \frac{\partial u_3}{\partial \hat{x}_2} & \frac{\partial u_3}{\partial \hat{x}_3} \\ \hline \end{array}$$

Strain tensor of Euler: Consider the change in the square of the length of the diagonal of a spatial element between its instant and its reference configuration:

$$\begin{aligned} d\hat{S}^2 - ds^2 &= d\hat{\mathbf{x}}^T d\hat{\mathbf{x}} - d\mathbf{x}^T d\mathbf{x} \\ &= d\hat{\mathbf{x}}^T (\mathbf{I} - (\mathbf{I} - \hat{\mathbf{H}})^T (\mathbf{I} - \hat{\mathbf{H}})) d\hat{\mathbf{x}} \\ &= d\hat{\mathbf{x}}^T (\hat{\mathbf{H}} + \hat{\mathbf{H}}^T - \hat{\mathbf{H}}^T \hat{\mathbf{H}}) d\hat{\mathbf{x}} \end{aligned}$$

The expression in brackets is taken as a measure of the deformation of the body at the point and denoted by $2\hat{\mathbf{E}}$. Matrix $\hat{\mathbf{E}}$ contains the coordinates \hat{e}_{km} of the strain tensor of Euler:

$$\hat{\mathbf{E}} = \frac{1}{2} (\hat{\mathbf{H}} + \hat{\mathbf{H}}^T - \hat{\mathbf{H}}^T \hat{\mathbf{H}}) \quad (3.53)$$

$$\hat{e}_{km} = \frac{1}{2} \left(\frac{\partial u_k}{\partial \hat{x}_m} + \frac{\partial u_m}{\partial \hat{x}_k} - \sum_i \frac{\partial u_i}{\partial \hat{x}_k} \frac{\partial u_i}{\partial \hat{x}_m} \right)$$

Principal strain: The strain tensor $\hat{\mathbf{E}}$ is transformed according to (3.14) with the rotation matrix \mathbf{R} of the coordinate space. Consider a special rotation matrix $\hat{\mathbf{N}}$ for which the strain tensor becomes a diagonal matrix $\hat{\mathbf{S}}$ because all shear strains go to null. For this case, (3.14) leads to a special eigenvalue problem:

$$\begin{aligned} \hat{\mathbf{E}} \hat{\mathbf{N}} &= \hat{\mathbf{N}} \hat{\mathbf{S}} \\ \hat{\mathbf{E}} \hat{\mathbf{n}}_k &= \hat{s}_k \hat{\mathbf{n}}_k \quad k \in \{1,2,3\} \end{aligned} \quad (3.54)$$

$$\hat{\mathbf{N}} = \begin{array}{|c|c|c|} \hline & & \\ \hline \hat{\mathbf{n}}_1 & \hat{\mathbf{n}}_2 & \hat{\mathbf{n}}_3 \\ \hline \end{array} \quad \hat{\mathbf{S}} = \begin{array}{|c|c|c|} \hline \hat{s}_1 & 0 & 0 \\ \hline 0 & \hat{s}_2 & 0 \\ \hline 0 & 0 & \hat{s}_3 \\ \hline \end{array}$$

$\hat{\mathbf{n}}_k$ unit vector in the principal direction k
 \hat{s}_k principal axial strain for the direction $\hat{\mathbf{n}}_k$

The diagonal elements \hat{s}_k of $\hat{\mathbf{S}}$ are called principal strains of the volume element; the columns of $\hat{\mathbf{N}}$ are called principal directions of the state of strain.

Since the shear strains are null if the state of strain is referred to the principal directions, (3.54) shows that the angle between the principal fibres does not change between configurations C and \hat{C} . Expression (3.40) shows that the diagonal coefficients of $\hat{\mathbf{S}}$ are a measure of the stretching of the principal fibres.

3.5 STATE OF STRESS

3.5.1 Stress Vector and Stress Tensor

Consider a virtual plane section that cuts the instant configuration \hat{C} of a body into subconfigurations \hat{C}_1 and \hat{C}_2 , as shown in figure 3.7. Let the outer unit normal at a point P on the section be \hat{n} for \hat{C}_1 and $-\hat{n}$ for \hat{C}_2 . Consider an infinitesimal vicinity of point P with area $\Delta\hat{a}$.

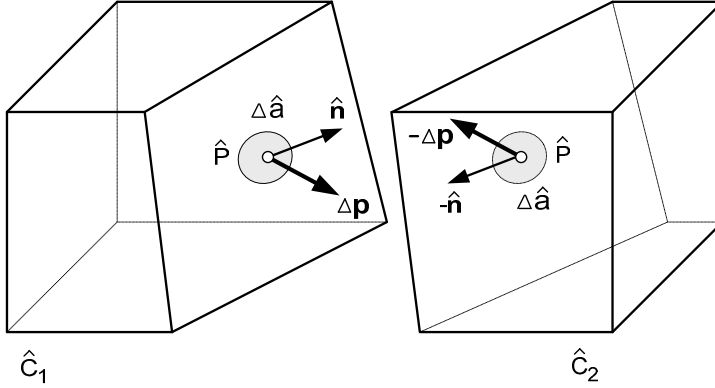


Figure 3.7 Virtual section of the instant configuration of a body

Replace the action of \hat{C}_2 on \hat{C}_1 over the area $\hat{n}\Delta\hat{a}$ by a fictitious force $\Delta\hat{p}$. The limit of the ratio of $\Delta\hat{p}$ to $\Delta\hat{a}$ when $\Delta\hat{a}$ goes to null is called the stress vector acting on the surface \mathbf{n} at point A and is denoted by $\hat{\mathbf{t}}$:

$$\hat{\mathbf{t}}(\hat{P}, \hat{\mathbf{n}}) := \lim_{\Delta\hat{a} \rightarrow 0} \frac{\Delta\hat{\mathbf{p}}}{\Delta\hat{a}} \quad (3.55)$$

- $\hat{\mathbf{t}}$ stress vector at point \hat{P}
- \hat{P} point at which the stress acts
- $\hat{\mathbf{n}}$ outer normal to the plane at point \hat{P}
- $\Delta\hat{a}$ infinitesimal area on the section which contains point \hat{P}
- $\Delta\hat{\mathbf{p}}$ infinitesimal force acting on area $\Delta\hat{a}$

The coordinates of the stress vector in global space are denoted by \hat{t}_k :

$$\hat{\mathbf{t}} = \hat{t}_1 \mathbf{i}_1 + \hat{t}_2 \mathbf{i}_2 + \hat{t}_3 \mathbf{i}_3 \quad (3.56)$$

Stress tensor of Cauchy: Consider the stress vectors acting on the positive faces of a spatial element in the instant configuration. Let the vector acting on the face with normal \mathbf{i}_m be denoted by $\hat{\mathbf{t}}_m$:

$$\hat{\mathbf{t}}_m = \hat{t}_{1m} \mathbf{i}_1 + \hat{t}_{2m} \mathbf{i}_2 + \hat{t}_{3m} \mathbf{i}_3 \quad (3.57)$$

- $\hat{\mathbf{t}}_m$ stress vector on the face with outer normal \mathbf{i}_m
- \hat{t}_{km} stress coordinate for base vector \mathbf{i}_k on face \mathbf{i}_m

The matrix $\hat{\mathbf{T}}$ whose columns contain the stress vectors acting on the three positive faces of an element is called the stress tensor of Cauchy:

$$\hat{\mathbf{T}} = \begin{bmatrix} \hat{\mathbf{t}}_{11} & \hat{\mathbf{t}}_{12} & \hat{\mathbf{t}}_{13} \\ \hat{\mathbf{t}}_{21} & \hat{\mathbf{t}}_{22} & \hat{\mathbf{t}}_{23} \\ \hat{\mathbf{t}}_{31} & \hat{\mathbf{t}}_{32} & \hat{\mathbf{t}}_{33} \end{bmatrix} \quad (3.58)$$

$\hat{\mathbf{T}}$ stress tensor of Cauchy

3.5.2 Stress Vector on a Plane

Stress vector on a section: Let the Cauchy stress tensor at a point P of an instant configuration and the normal of a plane through this point be given. The stress vector acting on the plane is determined by considering the tetrahedron in figure 3.8.

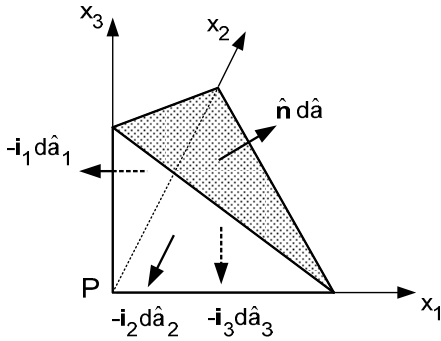


Figure 3.8 Area vectors of a tetrahedron

Since the surface of the tetrahedron is closed, the sum of the surface vectors is null:

$$\hat{\mathbf{n}}d\hat{\mathbf{a}} - \mathbf{i}_1d\hat{\mathbf{a}}_1 - \mathbf{i}_2d\hat{\mathbf{a}}_2 - \mathbf{i}_3d\hat{\mathbf{a}}_3 = \mathbf{0} \quad (3.59)$$

$d\hat{\mathbf{a}}_k$ area of the tetrahedron face with outer normal $-\mathbf{i}_k$
 $d\hat{\mathbf{a}}$ area of the tetrahedron face with outer normal $\hat{\mathbf{n}}$
 $\hat{\mathbf{n}}_k$ coordinate of unit normal vector $\hat{\mathbf{n}}$

Let the stress on the inclined face of the tetrahedron be $\hat{\mathbf{t}}$. The stresses acting on the negative coordinate faces are given by the negative columns $-\hat{\mathbf{t}}_k$ of the Cauchy stress tensor. The sum of the forces acting on the faces of the tetrahedron is null:

$$\hat{\mathbf{t}}d\hat{\mathbf{a}} - \hat{\mathbf{t}}_1d\hat{\mathbf{a}}_1 - \hat{\mathbf{t}}_2d\hat{\mathbf{a}}_2 - \hat{\mathbf{t}}_3d\hat{\mathbf{a}}_3 = \mathbf{0} \quad (3.60)$$

The expression for the stress vector $\hat{\mathbf{t}}$ follows by substitution of (3.59) in (3.60):

$$\hat{\mathbf{t}} = \hat{\mathbf{T}} \hat{\mathbf{n}} \quad (3.61)$$

$$\begin{bmatrix} \hat{t}_1 \\ \hat{t}_2 \\ \hat{t}_3 \end{bmatrix} = \begin{bmatrix} \hat{t}_{11} & \hat{t}_{12} & \hat{t}_{13} \\ \hat{t}_{21} & \hat{t}_{22} & \hat{t}_{23} \\ \hat{t}_{31} & \hat{t}_{32} & \hat{t}_{33} \end{bmatrix} * \begin{bmatrix} \hat{n}_1 \\ \hat{n}_2 \\ \hat{n}_3 \end{bmatrix}$$

$\hat{\mathbf{t}}$ stress vector on the plane with normal $\hat{\mathbf{n}}$
 $\hat{\mathbf{T}}$ Cauchy stress tensor

Principal stress: For a given state of stress $\hat{\mathbf{T}}$ at a point P of an instant configuration, the stress vector $\hat{\mathbf{t}}$ on an arbitrary plane with unit normal $\hat{\mathbf{n}}$ through P can be determined with (3.61). Consider the special case where the stress vector is normal to the plane. Since the vectors $\hat{\mathbf{t}}$ and $\hat{\mathbf{n}}$ are parallel, $\hat{\mathbf{t}}$ in (3.61) is set equal to $\hat{p} \hat{\mathbf{n}}$:

$$\hat{\mathbf{T}} \hat{\mathbf{n}}_k = \hat{p}_k \hat{\mathbf{n}}_k \quad k \in \{1,2,3\} \quad (3.62)$$

\hat{p}_k principal stress acting on plane k
 $\hat{\mathbf{n}}_k$ principal direction (unit normal of plane k)

The three eigenvalues of the special eigenvalue problem (3.62) are the principal stresses at the point. The three eigenvectors are the principal directions at the point. The shear stress on the principal planes with the normals $\hat{\mathbf{n}}_k$ is null.

3.5.3 Equilibrium at a Point

The forces acting on the faces of a spatial element with normals \mathbf{i}_1 and $-\mathbf{i}_1$ are shown in figure 3.9. Similar pairs of forces act on the other faces of the element. Let the body have mass density $\hat{\rho}$ and be subjected to a volume load \mathbf{q} per unit of mass.

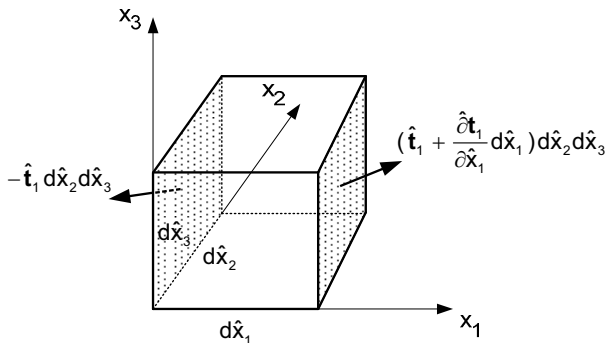


Figure 3.9 Forces acting on an infinitesimal element of the body

The sum of the forces acting on the element in the equilibrium state is null:

$$\frac{\partial \hat{t}_1}{\partial \hat{x}_1} + \frac{\partial \hat{t}_2}{\partial \hat{x}_2} + \frac{\partial \hat{t}_3}{\partial \hat{x}_3} + \hat{\rho} \mathbf{q} = \mathbf{0} \quad (3.63)$$

The equilibrium equation for the axial direction k follows from (3.63):

$$\frac{\partial \hat{\mathbf{t}}_{k1}}{\partial \hat{\mathbf{x}}_1} + \frac{\partial \hat{\mathbf{t}}_{k2}}{\partial \hat{\mathbf{x}}_2} + \frac{\partial \hat{\mathbf{t}}_{k3}}{\partial \hat{\mathbf{x}}_3} + \hat{\rho} \mathbf{q}_k = 0 \quad k \in \{1,2,3\} \quad (3.64)$$

The sum of the moments acting on the element in the equilibrium state is null. The evaluation of the vector products shows that the Cauchy stress tensor is symmetric:

$$(\mathbf{i}_1 d\hat{\mathbf{x}}_1) \times (\hat{\mathbf{t}}_1 d\hat{\mathbf{x}}_2 d\hat{\mathbf{x}}_3) + (\mathbf{i}_2 d\hat{\mathbf{x}}_2) \times (\hat{\mathbf{t}}_2 d\hat{\mathbf{x}}_3 d\hat{\mathbf{x}}_1) + (\mathbf{i}_3 d\hat{\mathbf{x}}_3) \times (\hat{\mathbf{t}}_3 d\hat{\mathbf{x}}_1 d\hat{\mathbf{x}}_2) = \mathbf{0}$$

$$\hat{\mathbf{t}}_{km} = \hat{\mathbf{t}}_{mk} \quad (3.65)$$

3.5.4 Piola-Kirchhoff Stress Tensors

1. Piola-Kirchhoff stress tensor: Consider a surface element $d\mathbf{a}$ of the reference configuration of a body, which is displaced to a surface element $d\hat{\mathbf{a}}$ in an instant configuration. The stress tensor that causes the same force $d\mathbf{f}$ on $d\mathbf{a}$ as the Cauchy stress tensor causes on $d\hat{\mathbf{a}}$ is called the 1. Piola-Kirchhoff stress tensor and denoted by \mathbf{P} . The relationship between the two stress tensors is determined with (3.51):

$$d\mathbf{f} = \mathbf{P} d\mathbf{a} = \hat{\mathbf{T}} d\hat{\mathbf{a}} = \det \mathbf{F} \hat{\mathbf{T}} \hat{\mathbf{F}}^T d\mathbf{a}$$

$$\mathbf{P} = \det \mathbf{F} \hat{\mathbf{T}} \hat{\mathbf{F}}^T \quad (3.66)$$

The 1. Piola-Kirchhoff stress tensor is not symmetric. Its coordinates are referred to the global base vectors. The stress vector on the plane with normal \mathbf{i}_k is given by:

$$\mathbf{p}_k = p_{1k} \mathbf{i}_1 + p_{2k} \mathbf{i}_2 + p_{3k} \mathbf{i}_3 \quad (3.67)$$

2. Piola-Kirchhoff stress tensor: Let the force vector defined in (3.67) be referred to the base vectors $\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3$ of the displaced material element:

$$\mathbf{p}_k = s_{1k} \mathbf{b}_1 + s_{2k} \mathbf{b}_2 + s_{3k} \mathbf{b}_3 \quad k \in \{1,2,3\} \quad (3.68)$$

Since the base vectors are the columns of the material deformation gradient \mathbf{F} , the three equations (3.68) can be combined as follows:

$$\mathbf{P} = \mathbf{F} \mathbf{S} \quad (3.69)$$

\mathbf{p}_1	\mathbf{p}_2	\mathbf{p}_3	$=$	\mathbf{b}_1	\mathbf{b}_2	\mathbf{b}_3	$*$	<table border="1" style="border-collapse: collapse; text-align: center;"> <tr> <td style="padding: 2px;">s_{11}</td> <td style="padding: 2px;">s_{12}</td> <td style="padding: 2px;">s_{13}</td> </tr> <tr> <td style="padding: 2px;">s_{21}</td> <td style="padding: 2px;">s_{22}</td> <td style="padding: 2px;">s_{23}</td> </tr> <tr> <td style="padding: 2px;">s_{31}</td> <td style="padding: 2px;">s_{32}</td> <td style="padding: 2px;">s_{33}</td> </tr> </table>	s_{11}	s_{12}	s_{13}	s_{21}	s_{22}	s_{23}	s_{31}	s_{32}	s_{33}
s_{11}	s_{12}	s_{13}															
s_{21}	s_{22}	s_{23}															
s_{31}	s_{32}	s_{33}															

The coordinates of matrix \mathbf{S} are called the 2. Piola-Kirchhoff stress tensor. Substitution of (4.32) into (4.35) yields the relationship with the Cauchy stress tensor $\hat{\mathbf{T}}$:

$$\mathbf{S} = (\det \mathbf{F}) \hat{\mathbf{T}} \hat{\mathbf{T}}^T \quad (3.70)$$

$$\hat{\mathbf{T}} = (\det \hat{\mathbf{F}}) \mathbf{F} \mathbf{S}^T$$

Since the Cauchy stress tensor is symmetric, it follows from (3.70) that the 2. Piola-Kirchhoff stress tensor is also symmetric:

$$\mathbf{S}^T = ((\det \mathbf{F}) \hat{\mathbf{T}} \hat{\mathbf{T}}^T)^T = \det \mathbf{F} \hat{\mathbf{T}} \hat{\mathbf{T}}^T = \mathbf{S} \quad (3.71)$$

3.6 MATERIAL PROPERTIES

3.6.1 Significance of Stress - Strain Relationships

The description of the kinematic and static properties of deformable bodies in the preceding sections employs 15 independent variables: 3 displacement coordinates u_k , 6 symmetric strain coordinates e_{km} and 6 symmetric stress coordinates s_{km} . Due to symmetry of the tensors used in the formulation, 3 strain coordinates and 3 stress coordinates are dependent variables.

Only 9 equations have been obtained for the 15 variables so far: the 6 strain-displacement equations (3.38) and the 3 equilibrium equations (3.64). Another 6 equations are required to make the number of equations equal to the number of independent variables, so that the set of equations can have a solution. These additional equations are obtained by means of experiments that relate the stress coordinates to the strain coordinates.

The stress-strain relationship for large strains is nonlinear. It has been stipulated in the first chapter that only structures with small strains will be considered in this book. This assumption is justified for most structural systems in civil engineering. It is now assumed that the material behaviour can be approximated with linear constitutive equations in the range of small strains.

3.6.2 Voigt Notation for Strain and Stress

Since the strain and stress tensors are of the second order, the description of their relationship requires an elasticity tensor of the fourth order. In order to avoid this tensor notation, the engineering notation for the states of stress and strain is used. The symmetric strain and stress tensors each have 6 independent coordinates. These coordinates are transformed and collected in the state of stress vector σ and the state of strain vector ϵ (Voigt notation):

$$\sigma = \begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{23} \\ \sigma_{31} \\ \sigma_{12} \end{bmatrix} \quad \epsilon = \begin{bmatrix} \epsilon_{11} \\ \epsilon_{22} \\ \epsilon_{33} \\ \epsilon_{23} \\ \epsilon_{31} \\ \epsilon_{12} \end{bmatrix} \quad (3.72)$$

σ_{km}	=	s_{km}	engineering stress
ϵ_{km}	=	$2e_{km}$	engineering shear strain, $k \neq m$
ϵ_{kk}	=	e_{kk}	engineering axial strain
s_{km}			coordinate of the 2. Piola-Kirchhoff stress tensor
e_{km}			coordinate of the strain tensor of Green

3.6.3 Constitutive Equations

If the material is subject to small strains, the relationship between the state of stress vector σ and the state of strain vector ϵ is described with the constitutive matrix \mathbf{C} :

$$\sigma = \mathbf{C}\epsilon \quad (3.73)$$

- σ state of stress in (3.72)
- ϵ state of strain in (3.72)
- \mathbf{C} constitutive matrix

The material is called linear if its constitutive matrix is independent of the states of stress and strain. The material is called homogeneous if it has the same properties at every material point. The material is called elastic if the state of stress for a given state of strain is the same for loading and for unloading of the material. The constitutive matrix is a specific property of each material.

Isotropic: A material is called isotropic if its properties are the same in every direction. An isotropic material is described with two independent parameters: the modulus of elasticity E and the Poisson ratio ν . The strain in the direction of a normal stress σ_{ii} is $\epsilon_{ii} = \sigma_{ii}/E$; the strain in the orthogonal direction is $-\nu\epsilon_{ii}$. The shear strain due to a shear stress σ_{ik} is $\epsilon_{ik} = \sigma_{ik}/G$, where the shear modulus G is a function of E and ν . The inverse of the constitutive matrix is thus:

$$\mathbf{C}^{-1} = \begin{array}{|c|c|c|c|c|c|} \hline \frac{1}{E} & -\frac{\nu}{E} & -\frac{\nu}{E} & 0 & 0 & 0 \\ \hline -\frac{\nu}{E} & \frac{1}{E} & -\frac{\nu}{E} & 0 & 0 & 0 \\ \hline -\frac{\nu}{E} & -\frac{\nu}{E} & \frac{1}{E} & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & \frac{1}{G} & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & \frac{1}{G} & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & \frac{1}{G} \\ \hline \end{array} \quad (3.74)$$

The constitutive matrix is determined by inverting (3.74):

$$\mathbf{C} = \begin{array}{|c|c|c|c|c|c|} \hline (1-\nu)N & \nu N & \nu N & 0 & 0 & 0 \\ \hline \nu N & (1-\nu)N & \nu N & 0 & 0 & 0 \\ \hline \nu N & \nu N & (1-\nu)N & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & G & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & G & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & G \\ \hline \end{array} \quad (3.75)$$

$$N = \frac{E}{(1+\nu)(1-2\nu)} \quad G = \frac{E}{2(1+\nu)}$$

Transverse isotropic: A material is called transverse isotropic if it has one principal direction. The material is isotropic in all planes that are normal to the principal direction. A Cartesian coordinate system is chosen with axes y_1, y_2 in the isotropic plane and axis y_3 pointing in the principal direction.

The stress σ_{33} causes strains $\varepsilon_{33} = \sigma_{33}/E_3$ and $\varepsilon_{11} = \varepsilon_{22} = -v_3 \sigma_{33}/E_1$. A stress σ_{kk} in the direction y_1 or y_2 causes a strain $\varepsilon_{kk} = \sigma_{kk}/E_1$ in that direction, a strain $-v_1 \varepsilon_{kk}$ normal to it in the isotropic plane and a strain $-v_3 \varepsilon_{kk}$ in the direction y_3 . The shear modulus is G_1 in the isotropic plane and G_3 in the planes normal to it.

A transverse-isotropic material is thus described with 5 independent parameters: E_1 and v_1 in the isotropic plane, and E_3, G_3 and v_3 in the planes normal to it. If the coordinates of the states of strain and stress are referred to the coordinate system y_1, y_2, y_3 as defined above, the inverse of the constitutive matrix is given by:

$$\mathbf{C}^{-1} = \begin{array}{|c|c|c|c|c|c|} \hline \frac{1}{E_1} & -\frac{v_1}{E_1} & -\frac{v_3}{E_1} & 0 & 0 & 0 \\ \hline -\frac{v_1}{E_1} & \frac{1}{E_1} & -\frac{v_3}{E_1} & 0 & 0 & 0 \\ \hline -\frac{v_3}{E_1} & -\frac{v_3}{E_1} & \frac{1}{E_3} & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & \frac{1}{G_3} & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & \frac{1}{G_3} & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & \frac{1}{G_1} \\ \hline \end{array} \quad (3.76)$$

E_i moduli of elasticity
 G_i shear moduli
 v_i Poisson numbers

The constitutive matrix is determined by inverting (3.76):

$$\mathbf{C} = \begin{array}{|c|c|c|c|c|c|} \hline c_1 N & c_2 N & c_3 N & 0 & 0 & 0 \\ \hline c_2 N & c_1 N & c_3 N & 0 & 0 & 0 \\ \hline c_3 N & c_3 N & c_4 N & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & G_3 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & G_3 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & G_1 \\ \hline \end{array} \quad (3.77)$$

$$\begin{aligned} N &= \frac{E_1}{c_0} & G_1 &= \frac{E}{2(1+v_1)} & n_3 &= \frac{E_1}{E_3} \\ c_0 &= n_3(1-v_1^2) - 2v_3^2(1+v_1) & c_1 &= n_3 - v_3^2 \\ c_2 &= v_1 n_3 + v_3^2 & c_3 &= v_3(1+v_1) & c_4 &= 1-v_1^2 \end{aligned}$$

Orthotropic: A material is called orthotropic if it has three principal directions. In order to describe the properties of the material, a Cartesian coordinate system is chosen with the axes y_1, y_2, y_3 pointing in the principal directions.

Stress σ_{11} causes strains $\varepsilon_{11} = \sigma_{11}/E_1$, $\varepsilon_{22} = -v_1 \sigma_{11}/E_1$ and $\varepsilon_{33} = -v_2 \sigma_{11}/E_1$. Stress σ_{22} causes strains $\varepsilon_{22} = \sigma_{22}/E_2$, $\varepsilon_{11} = -v_1 \sigma_{22}/E_1$ and $\varepsilon_{33} = -v_3 \sigma_{22}/E_1$. Stress σ_{33} causes strains $\varepsilon_{33} = \sigma_{33}/E_3$, $\varepsilon_{11} = -v_2 \sigma_{33}/E_1$ and $\varepsilon_{22} = -v_3 \sigma_{33}/E_1$. For each shear stress there is an independent shear modulus G_k .

An orthotropic material is thus described with 9 independent parameters E_i, G_i and v_i . The inverse of the constitutive matrix is given by:

$$\mathbf{C}^{-1} = \begin{bmatrix} \frac{1}{E_1} & -\frac{v_1}{E_1} & -\frac{v_2}{E_1} & 0 & 0 & 0 \\ -\frac{v_1}{E_1} & \frac{1}{E_2} & -\frac{v_3}{E_1} & 0 & 0 & 0 \\ -\frac{v_2}{E_1} & -\frac{v_3}{E_1} & \frac{1}{E_3} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{G_1} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{G_2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{G_3} \end{bmatrix} \quad (3.78)$$

E_i moduli of elasticity
 G_i shear moduli
 v_i Poisson numbers

The constitutive matrix is determined by inverting (3.78):

$$\mathbf{C} = \begin{bmatrix} c_{11} N & c_{12} N & c_{13} N & 0 & 0 & 0 \\ c_{12} N & c_{22} N & c_{23} N & 0 & 0 & 0 \\ c_{13} N & c_{23} N & c_{33} N & 0 & 0 & 0 \\ 0 & 0 & 0 & G_1 & 0 & 0 \\ 0 & 0 & 0 & 0 & G_2 & 0 \\ 0 & 0 & 0 & 0 & 0 & G_3 \end{bmatrix} \quad (3.79)$$

$$c_0 = \det \begin{bmatrix} 1 & -v_1 & -v_2 \\ -v_1 & n_2 & -v_3 \\ -v_2 & -v_3 & n_3 \end{bmatrix} \quad N = \frac{E_1}{c_0} \quad n_2 = \frac{E_1}{E_2} \quad n_3 = \frac{E_1}{E_3}$$

$$\begin{aligned} c_{11} &= n_2 n_3 - v_3^2 & c_{22} &= n_3 - v_2^2 & c_{33} &= n_2 - v_1^2 \\ c_{12} &= v_1 n_3 + v_2 v_3 & c_{13} &= v_2 n_2 + v_1 v_3 & c_{23} &= v_3 + v_1 v_2 \end{aligned}$$

3.7 BOUNDARY CONDITIONS

The environment influences an elastic body at each point of its surface through prescribed stresses and displacements. These external influences are called the boundary conditions of the body. A separate set of boundary conditions is prescribed for every instant configuration of the body.

Prescribed displacements: If the prescribed external influence at a point is its displacement, the point is called a support. The coordinates of the displacement vector are prescribed. Rigid motion of the body is controlled by supports that prescribe the rigid body translations in the direction of the global coordinate axes and the rigid body rotations about each of the global axes.

If the displacement vector is described in global space, the vector and its coordinates are denoted by \underline{u} and \underline{u}_k . If the displacement vector is described in a local space, the vector and its coordinates are denoted by \underline{v} and \underline{v}_k .

Tractions: If the prescribed external influence at a point P is a stress vector, the vector is called a traction vector. If a surface is stress free, the magnitude of the traction is null.

If the traction vector at P is specified for the area and outer normal $\hat{\mathbf{n}}$ of the instant configuration, the traction and its coordinates are denoted by $\hat{\mathbf{t}}$ and \hat{t}_k . The traction is related to the outer normal and the Cauchy stress tensor $\hat{\mathbf{T}}$ by (3.61):

$$\hat{\mathbf{t}} = \hat{\mathbf{T}} \hat{\mathbf{n}} \quad (3.80)$$

$\hat{\mathbf{t}}$ traction at a point with Cauchy state of stress $\hat{\mathbf{T}}$ and outer normal $\hat{\mathbf{n}}$

If the traction vector is specified for the area and the outer normal \mathbf{n} of the reference configuration, it is related to the outer normal \mathbf{n} and the 1. Piola-Kirchhoff stress tensor \mathbf{P} in (3.66) by considering the equilibrium of a tetrahedron corresponding to figure 3.8:

$$\underline{\mathbf{p}} = \mathbf{P} \mathbf{n} \quad (3.81)$$

$\underline{\mathbf{p}}$ traction at a point with 1. Piola-Kirchhoff state of stress \mathbf{P} and outer normal \mathbf{n}

Transformation (3.69) is substituted into (3.81) to express this boundary condition in terms of the 2. Piola-Kirchhoff stress tensor:

$$\underline{\mathbf{p}} = \mathbf{F} \mathbf{S} \mathbf{n} \quad (3.82)$$

\mathbf{F} material deformation gradient
 \mathbf{S} 2. Piola – Kirchhoff stress tensor

Mixed boundary conditions: The boundary conditions at point P can also be mixed: for each coordinate direction either the coordinate of the displacement or the coordinate of the traction is specified. Since mixed boundary conditions do not consist of complete vectors, the axes of the local coordinate systems are chosen so that they point in the direction of the prescribed displacement and traction coordinates, as shown in figure 3.10. None of the base vectors \mathbf{e}_k of a local coordinate system needs to be normal to the surface at P.

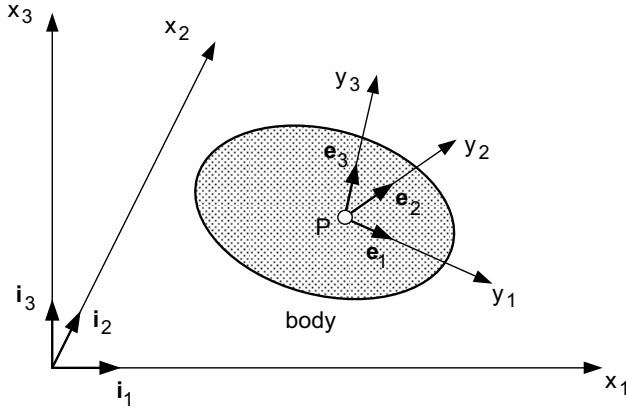


Figure 3.10 Local coordinate system at point P on the surface of the body

The transformation matrix for the local coordinate system is given by (3.6):

$$\mathbf{x} = \mathbf{x}_0 + \mathbf{R}\mathbf{y} \quad (3.83)$$

\mathbf{x}_0 global location vector of the boundary point P
 \mathbf{R} rotation matrix containing the base vectors \mathbf{e}_k

The global coordinates of the displacement vector of point P are transformed to the local space at P with (3.9) and (3.12):

$$\mathbf{v} = \mathbf{R}^T \mathbf{u} \quad (3.84)$$

\mathbf{u} vector containing the displacement coordinates of point P in global space
 \mathbf{v} vector containing the displacement coordinates of point P in local space

The determination of the global coordinates of the tractions corresponds to (3.84):

$$\hat{\underline{\mathbf{t}}}_{(y)} = \mathbf{R}^T \hat{\underline{\mathbf{t}}}_{(x)} \quad (3.85)$$

$$\underline{\mathbf{p}}_{(y)} = \mathbf{R}^T \underline{\mathbf{p}}_{(x)}$$

$\hat{\underline{\mathbf{t}}}_{(x)}, \underline{\mathbf{p}}_{(x)}$ vectors containing the traction coordinates of point P in global space

$\hat{\underline{\mathbf{t}}}_{(y)}, \underline{\mathbf{p}}_{(y)}$ vectors containing the traction coordinates of point P in local space

Unified boundary conditions: In order to avoid the separate treatment of the traction, displacement and mixed boundary conditions, a set of prescribed traction coordinates C_t and a set of prescribed displacement coordinates C_u are defined. For every local or global coordinate axis at every point on the surface, a boundary condition is contributed to either C_t or C_u .

$$C_u := \{ \underline{u}_k \mid \mathbf{x} \in \partial C \wedge u_k(\mathbf{x}) \text{ is prescribed} \} \quad (3.86)$$

$$C_t := \{ \hat{\underline{\mathbf{t}}}_k \mid \mathbf{x} \in \partial C \wedge \hat{\mathbf{t}}_k(\mathbf{x}) \text{ is prescribed} \} \quad (3.87)$$

\underline{u}_k prescribed global or local displacement coordinate

$\hat{\underline{\mathbf{t}}}_k$ prescribed global or local Cauchy or Piola-Kirchhoff traction coordinate

3.8 GOVERNING EQUATIONS

3.8.1 Differential Governing Equations

The formulation of the nonlinear theory of elasticity in the preceding sections yields the 15 equations required to determine the 3 displacement coordinates and the 6 independent coordinates of the stress and strain tensors as functions of the prescribed displacements and tractions. The following equations were obtained for an instant configuration of an elastic body:

strain - displacement equations

$$e_{km} = \frac{1}{2}(u_{k,m} + u_{m,k} + \sum_i u_{i,k}u_{i,m}) \quad i,k,m \in \{1,2,3\} \quad (3.88)$$

$u_{k,m}$ partial derivative of the displacement coordinate u_k with respect to x_m
 e_{km} coordinate of the strain tensor of Green

equilibrium equations

$$\hat{t}_{k1,1} + \hat{t}_{k2,2} + \hat{t}_{k3,3} + \hat{\rho}q_k = 0 \quad (3.89)$$

\hat{t}_{km} coordinate of the 2. Piola-Kirchhoff stress tensor
 $\hat{\rho}$ mass per unit volume in the instant configuration
 q_k load per unit mass

constitutive equations

$$\boldsymbol{\sigma} = \mathbf{C} \boldsymbol{\varepsilon} \quad (3.90)$$

$\boldsymbol{\sigma}$ Voigt vector of the 2. Piola-Kirchhoff stress tensor
 $\boldsymbol{\varepsilon}$ Voigt vector of the state of strain, derived from the Green tensor
 \mathbf{C} constitutive matrix

stress vector on a plane

$$\hat{\mathbf{t}} = \hat{\mathbf{T}} \hat{\mathbf{n}} \quad (3.91)$$

$\hat{\mathbf{t}}$ stress vector on a plane of the instant configuration with normal $\hat{\mathbf{n}}$
 $\hat{\mathbf{T}}$ Cauchy stress tensor

Equation (3.91) is required to relate the prescribed tractions to the state of stress of the body. The boundary conditions are specified with the unified method:

boundary conditions

$$\mathbf{x} \in \partial C \wedge \underline{u}_j \in C_u \rightarrow u_j = \underline{u}_j \quad (3.92)$$

$$\mathbf{x} \in \partial C \wedge \hat{\underline{t}}_j \in C_t \rightarrow \hat{t}_j = \hat{\underline{t}}_j \quad (3.93)$$

C_u prescribed displacement coordinates, see (3.86)
 C_t prescribed traction coordinates, see (3.87)

Analytical solutions for these nonlinear governing equations can only be found in a few special cases. Nonlinear analyses of structures in engineering practice require the development of numerical methods that remain applicable if the geometry and the loading of the structure are complex.

3.8.2 Weighted residual method

The approximate solution of the nonlinear governing equations of the theory of elasticity is developed by means of the weighted residual method. The governing equations in section 3.8.1 (called the strong form of the governing equations) are transformed into integral equations (called the weak form of the governing equations) which are suitable for approximation with the finite element method.

Trial and residual functions: Assume that a trial function $\mathbf{u}(\mathbf{x})$ is chosen freely as an approximate solution of the nonlinear governing equations in section 3.8.1. The substitution of the trial function into the governing equations will, in general, lead to residuals in these equations. The following residual functions are defined at every point \mathbf{x} of the instant configuration:

$$r_{1km}(\mathbf{x}) = \mathbf{e}_{km} - \frac{1}{2}(\mathbf{u}_{k,m} + \mathbf{u}_{m,k} + \sum_i \mathbf{u}_{i,k} \mathbf{u}_{i,m}) \quad i, k, m \in \{1, 2, 3\} \quad (3.94)$$

$$r_{2k}(\mathbf{x}) = \hat{\mathbf{t}}_{k1,1} + \hat{\mathbf{t}}_{k2,2} + \hat{\mathbf{t}}_{k3,3} + \hat{\rho} \mathbf{q}_k \quad k \in \{1, 2, 3\} \quad (3.95)$$

$$\mathbf{r}_3(\mathbf{x}) = \boldsymbol{\sigma} - \mathbf{C} \boldsymbol{\varepsilon} \quad (3.96)$$

The following residual functions are defined at every point \mathbf{x} of the surface of the instant configuration:

$$\mathbf{r}_4(\mathbf{x}) = \hat{\mathbf{t}} - \hat{\mathbf{T}} \hat{\mathbf{n}} \quad (3.97)$$

$$r_{5j} = \mathbf{u}_j - \underline{\mathbf{u}}_j \quad \underline{\mathbf{u}}_j \in C_u \quad (3.98)$$

$$r_{6j} = \hat{\mathbf{t}}_i - \underline{\hat{\mathbf{t}}}_i \quad \underline{\hat{\mathbf{t}}}_i \in C_t \quad (3.99)$$

A-priori conditions: Assume that one of the equations (3.94) to (3.99) is satisfied exactly for every trial function that can be chosen: its residual function is identical to null. Then this equation does not contribute to the error in the approximate solution and is called an a-priori condition. It is possible to develop approximate solutions for many combinations of a-priori conditions selected from (3.94) to (3.99). The residuals r_{1km} and \mathbf{r}_3 are readily made equal to null for a given trial displacement. The strain-displacement equations (3.94) and the constitutive equations (3.96) are therefore declared a-priori conditions. Experience with the finite element method shows this to be a suitable choice for a broad range of problems in structural mechanics.

Weighted residuals: The residual in (3.95), which is not an a-priori condition, is multiplied with a weight function and integrated over the volume of the configuration. The residuals in (3.97) to (3.99) are also multiplied with weight functions and integrated over the surface of the instant configuration. Like the trial functions, the admissible weight functions must satisfy the a-priori conditions. If the resulting integrals are null for arbitrary admissible weight functions, then the trial function is the exact solution to the problem.

Since the integrals of the weighted residuals can only be set equal to null for a finite set of chosen weight functions, the solution will, in general, be approximate. A proper choice of the weight functions is essential to the accuracy of the weighted residual method. All weighted residuals must have the same physical dimension. In order to obtain a symmetric system of algebraic governing equations, it is convenient to

choose the variations $\delta \mathbf{u}$ of the displacements and the variations $\delta \hat{\mathbf{t}}$ of the stresses at the surface of the body as weight functions. This leads to the weak spatial form of the governing equations in section 3.8.3.

3.8.3 Spatial Integral Form

The residuals r_{2k} are multiplied with displacement variations δu_k and the residuals r_3 with the displacement variation $\delta \mathbf{u}$ as weight functions. The products are integrated over the volume of the instant configuration. The residuals r_4 are multiplied with the displacement variations $\delta \mathbf{u}$, the residuals r_{5j} with the stress variations $\delta \hat{t}_j$, and the residuals r_{6j} with the displacement variations δu_j as weight functions. The products are integrated over the surface of the body. The sums of the integrals are the spatial form of the weak governing equations:

$$\int_{\hat{C}} \sum_i \sum_m (\delta u_i \frac{\partial \hat{t}_{im}}{\partial \hat{x}_m}) d\hat{v} = - \int_{\hat{C}} \sum_i \delta u_i \hat{\rho} q_i d\hat{v} + \int_{\partial \hat{C}} \sum_i \delta u_i (\sum_m \hat{t}_{im} \hat{n}_{im} - \hat{t}_i) d\hat{a} + \int_{\hat{t}_i \in C_t} \delta u_i (\hat{t}_i - \underline{t}_i) d\hat{a} + \int_{u_i \in C_u} \delta \hat{t}_i (u_i - \underline{u}_i) d\hat{a} \quad (3.100)$$

The first integral is transformed by partial integration to make the integral form more suitable for numerical approximation:

$$\int_{\hat{C}} \sum_i \sum_m \delta u_i \frac{\partial \hat{t}_{im}}{\partial \hat{x}_m} d\hat{v} = - \int_{\hat{C}} \sum_i \sum_m \hat{t}_{im} \delta (\frac{\partial u_i}{\partial \hat{x}_m}) d\hat{v} + \int_{\partial \hat{C}} \sum_i \sum_m \delta u_i \hat{t}_{im} \hat{n}_m d\hat{a} \quad (3.101)$$

Equation (3.101) is substituted on the left-hand side of equation (3.100):

$$- \int_{\hat{C}} \sum_i \sum_m \hat{t}_{im} \delta (\frac{\partial u_i}{\partial \hat{x}_m}) d\hat{v} = - \int_{\hat{C}} \sum_i \delta u_i \hat{\rho} q_i d\hat{v} - \int_{\partial \hat{C}} \sum_i \delta u_i \hat{t}_i + \int_{\hat{t}_i \in C_t} \delta u_i (\hat{t}_i - \underline{t}_i) d\hat{a} + \int_{u_i \in C_u} \delta \hat{t}_i (u_i - \underline{u}_i) d\hat{a} \quad (3.102)$$

The integral over $\partial \hat{C}$ is split into integrals for the coordinates associated with C_u and with C_t :

$$\int_{\partial \hat{C}} \sum_i \delta u_i \hat{t}_i d\hat{a} = \int_{\hat{t}_i \in C_t} \delta u_i \hat{t}_i d\hat{a} + \int_{u_i \in C_u} \delta u_i \hat{t}_i d\hat{a} \quad (3.103)$$

Equation (3.103) is substituted on the right-hand side of equation (3.102) and the integral containing $u_i - \underline{u}_i$ is set to null by choosing $u_i = \underline{u}_i$ on C_u :

$$\int_{\hat{C}} \sum_i \sum_m \hat{t}_{im} \delta (\frac{\partial u_i}{\partial \hat{x}_m}) d\hat{v} = \int_{\hat{C}} \sum_i \delta u_i \hat{\rho} q_i d\hat{v} + \int_{\hat{t}_i \in C_t} \delta u_i \hat{t}_i d\hat{a} + \int_{u_i \in C_u} \delta u_i \hat{t}_i d\hat{a} \quad (3.104)$$

$$\underline{u}_i \in C_u \rightarrow u_i = \underline{u}_i$$

Equation (3.104) has been derived for spatial elements of the body in the instant configuration, since the equilibrium conditions must be satisfied for this configuration. The instant configuration is not known a priori. Therefore equation (3.104) will be transformed to material elements in the reference configuration.

3.8.4 Material Integral Form

The instant volume element $d\hat{v}$ in the integral on the left-hand side of (3.104) is replaced by the volume element dv of the reference configuration with (3.33). The coordinates \hat{t}_{im} of the Cauchy stress tensor are replaced by the coordinates s_{jk} of the 2. Piola-Kirchhoff stress tensor with (3.70). The instant coordinate \hat{x}_i is replaced by $x_i + u_i$. The double sum then becomes a function of the Green strain tensor \mathbf{E} and of the 2. Piola-Kirchhoff stress tensor $\hat{\mathbf{T}}$:

$$\begin{aligned}
\sum_i \sum_m \hat{t}_{im} \delta\left(\frac{\partial u_i}{\partial \hat{x}_m}\right) d\hat{v} &= \sum_i \sum_m \sum_j \sum_k s_{jk} \frac{\partial \hat{x}_i}{\partial x_j} \frac{\partial \hat{x}_m}{\partial x_k} \delta\left(\frac{\partial u_i}{\partial \hat{x}_m}\right) dv \\
&= \sum_i \sum_j \sum_k s_{jk} \frac{\partial \hat{x}_i}{\partial x_j} \delta\left(\frac{\partial u_i}{\partial x_k}\right) dv \\
&= \sum_i \sum_j \sum_k s_{jk} \left(\frac{\partial x_i}{\partial x_j} + \frac{\partial u_i}{\partial x_j}\right) \delta\left(\frac{\partial u_i}{\partial x_k}\right) dv \\
&= \sum_j \sum_k s_{jk} \left[\delta\left(\frac{\partial u_i}{\partial x_k}\right) + \sum_i \frac{\partial u_i}{\partial x_j} \delta\left(\frac{\partial u_i}{\partial x_k}\right)\right] dv \\
&= \sum_j \sum_k \frac{1}{2} s_{jk} \delta\left[\frac{\partial u_i}{\partial x_k} + \frac{\partial u_k}{\partial x_j} + \sum_i \frac{\partial u_i}{\partial x_j} \frac{\partial u_i}{\partial x_k}\right] dv \\
&= \sum_j \sum_k \delta e_{jk} s_{jk} dv
\end{aligned} \tag{3.105}$$

The second and third terms on the right-hand side of result (3.104) are transformed by expressing the stress vector in terms of the state of stress with (3.61). The instant area vector $d\hat{\mathbf{a}}$ is replaced by the reference vector $d\mathbf{a}$ with (3.36) and the Cauchy stress tensor $\hat{\mathbf{T}}$ is replaced by the 2. Piola-Kirchhoff stress tensor with (3.66):

$$\begin{aligned}
\sum_i \delta u_i \hat{t}_i d\hat{\mathbf{a}} &= \delta \mathbf{u}^T \hat{\mathbf{t}} d\hat{\mathbf{a}} = \delta \mathbf{u}^T \hat{\mathbf{T}} \hat{\mathbf{n}} d\hat{\mathbf{a}} = \delta \mathbf{u}^T \hat{\mathbf{T}} d\hat{\mathbf{a}} \\
&= \delta \mathbf{u}^T \hat{\mathbf{T}} \hat{\mathbf{F}}^T d\mathbf{a} (\det \mathbf{F}) \\
&= \delta \mathbf{u}^T \mathbf{P} n d\mathbf{a} \\
&= \delta \mathbf{u}^T \mathbf{p} d\mathbf{a}
\end{aligned} \tag{3.106}$$

Since the mass of an element does not change with the configuration of the body, the force acting on an element remains constant:

$$\hat{\rho} q_i d\hat{v} = \rho q_i dv \tag{3.107}$$

Substitute (3.105) to (3.107) into (3.104) and use the Voigt notation (3.72) for stress and strain. This yields the material integral form of the governing equations:

$$\int_C \delta \boldsymbol{\varepsilon}^T \boldsymbol{\sigma} dv = \int_C \delta \mathbf{u}^T \mathbf{q} \rho dv + \int_{C_u} \delta \mathbf{u}^T \mathbf{p} d\mathbf{a} + \int_{C_t} \delta \mathbf{u}^T \underline{\mathbf{p}} d\mathbf{a} \tag{3.108}$$

$$\mathbf{u}_i \in C_u \rightarrow \mathbf{u}_i = \underline{\mathbf{u}}_i$$

Equation (3.108) is similar to the Principle of Virtual Displacements of the linear theory of elasticity. The semantics of equation (3.108) differ significantly from those of the linear principle. Important aspects of the nonlinear theory are:

- The strain coordinates are nonlinear functions of the displacement derivatives.
- The stress coordinates of the 2. Piola-Kirchhoff tensor are referred to the base vectors of both the reference and the instant space (mixed tensor).
- The coordinates of the load vector are referred to the global base vectors. The body only behaves conservatively if the direction of the loads is independent of the displacements.

Exact solutions of the governing equations (3.108) are known for a few special cases, as illustrated in chapter 2. For the complex geometry and loading of the structures usually encountered in civil engineering, approximate numerical solutions of the equations are obtained. There are two main aspects of the approximation:

- (a) The spatial variation of the state variables is approximated by means of the finite element method. The same approximation is made in linear theory.
- (b) The nonlinear equations are solved approximately by means of a stepwise numerical procedure using the constant arc increment method.

Approximation (a) of the spatial variation is specific for each component type and is treated as part of the finite element theory for the component type. Approximation (b) of the nonlinearity is based on a consistent linearisation of the governing equations (3.108). The method is the same for all component types. This linearisation is treated in the next section and leads to incremental governing equations.

3.9 INCREMENTAL GOVERNING EQUATIONS

3.9.1 Incremental Solution Concept

Configurations: The nonlinear behaviour of a body is determined stepwise. It is assumed that the state of the body at the beginning of each step is known. The equations used to determine the state of the body at the end of the step, given the state at the beginning of the step, are called incremental governing equations. These equations are derived by considering the three configurations of the body shown in figure 3.11. The unknown instant state is the direct successor of the known instant state.

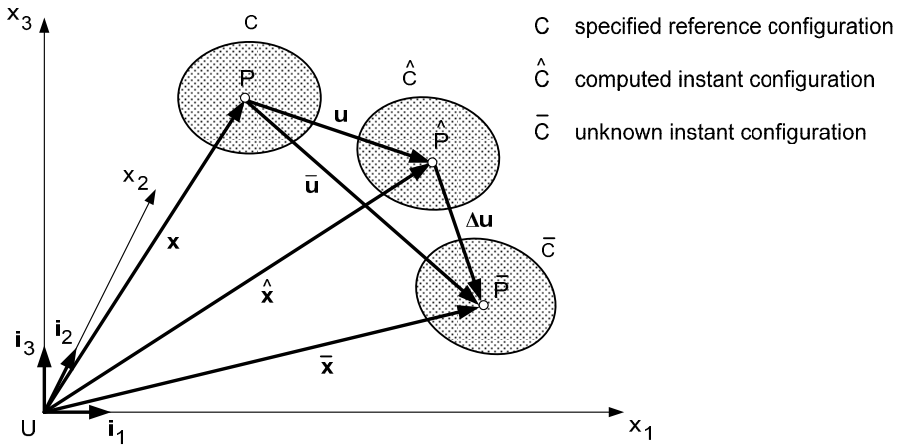


Figure 3.11 Incremental configurations of a body

The incremental governing equations are obtained by consistent linearisation of the nonlinear governing equations. The first step of consistent linearisation is the linearisation of the behavioural variables.

Increments of state variables: Material points are identified by their location in the reference configuration of the body (total Lagrangian approach). The displacement $\bar{\mathbf{u}}$ of a point P in the unknown instant state is the vector sum of the displacement \mathbf{u} of point P in the known instant configuration and a displacement increment $\Delta\mathbf{u}$ from the known to the unknown instant configuration:

$$\bar{\mathbf{u}} = \mathbf{u} + \Delta\mathbf{u} \quad (3.109)$$

$\bar{\mathbf{u}}$ displacement in the unknown instant configuration \bar{C}

\mathbf{u} displacement in the known instant configuration \hat{C}

$\Delta\mathbf{u}$ displacement increment from configuration \hat{C} to configuration \bar{C}

The increments of the other behavioural variables of the body are defined in a similar manner. The value of a variable in the known configuration \hat{C} is not marked; its value in the unknown instant configuration \bar{C} , which is the successor of \hat{C} in the stepwise procedure, is marked with a bar and the increment is marked with the symbol Δ .

$$\text{strain} \quad \bar{\mathbf{e}}_{im} = \mathbf{e}_{im} + \Delta \mathbf{e}_{im} \quad (3.110)$$

$$\text{stress} \quad \bar{\mathbf{s}}_{im} = \mathbf{s}_{im} + \Delta \mathbf{s}_{im} \quad (3.111)$$

$$\text{load} \quad \bar{\mathbf{q}}_i = \mathbf{q}_i + \Delta \mathbf{q}_i \quad (3.112)$$

$$\text{traction} \quad \hat{\mathbf{t}}_i = \hat{\mathbf{t}}_i + \Delta \hat{\mathbf{t}}_i \quad (3.113)$$

Incremental strain-displacement relations: Consider the coordinates of the Green strain tensor in the known and the unknown instant configuration:

$$\mathbf{e}_{im} = \frac{1}{2}(\mathbf{u}_{i,m} + \mathbf{u}_{m,i} + \sum_k \mathbf{u}_{k,i} \mathbf{u}_{k,m}) \quad (3.114)$$

$$\begin{aligned} \bar{\mathbf{e}}_{im} &= \frac{1}{2}(\bar{\mathbf{u}}_{i,m} + \bar{\mathbf{u}}_{m,i} + \sum_k \bar{\mathbf{u}}_{k,i} \bar{\mathbf{u}}_{k,m}) \\ &= \frac{1}{2}(\mathbf{u}_{i,m} + \mathbf{u}_{m,i} + \Delta \mathbf{u}_{i,m} + \Delta \mathbf{u}_{m,i} + \sum_k (\mathbf{u}_{k,i} + \Delta \mathbf{u}_{k,i})(\mathbf{u}_{k,m} + \Delta \mathbf{u}_{k,m})) \end{aligned}$$

The difference of the two strain tensors is split into a linear and a nonlinear part:

$$\Delta \mathbf{e}_{im} = \Delta \mathbf{e}_{im}^0 + \Delta \mathbf{e}_{im}^n \quad (3.115)$$

$$\Delta \mathbf{e}_{im}^0 = \frac{1}{2}(\Delta \mathbf{u}_{i,m} + \Delta \mathbf{u}_{m,i} + \sum_k (\mathbf{u}_{k,i} \Delta \mathbf{u}_{k,m} + \mathbf{u}_{k,m} \Delta \mathbf{u}_{k,i})) \quad (3.116)$$

$$\Delta \mathbf{e}_{im}^n = \frac{1}{2} \sum_k \Delta \mathbf{u}_{k,i} \Delta \mathbf{u}_{k,m} \quad (3.117)$$

$\Delta \mathbf{e}_{im}^0$ linear component of the strain increment from $\hat{\mathbf{C}}$ to $\bar{\mathbf{C}}$

$\Delta \mathbf{e}_{im}^n$ nonlinear component of the strain increment from $\hat{\mathbf{C}}$ to $\bar{\mathbf{C}}$

Variation of the state variables: The variation $\delta \bar{\mathbf{u}}$ of the state of displacement in the unknown instant configuration is used as weight function on the right-hand side of the integral form (3.108). Since the body is in equilibrium in the known instant configuration, the displacement variation $\delta \mathbf{u}$ for that state is set to null:

$$\delta \bar{\mathbf{u}} = \delta(\mathbf{u} + \Delta \mathbf{u}) = \delta \mathbf{u} + \delta(\Delta \mathbf{u}) = \delta(\Delta \mathbf{u}) \quad (3.118)$$

$\delta \bar{\mathbf{u}}$ variation of the displacement in the unknown instant configuration

$\delta(\Delta \mathbf{u})$ variation of the displacement increment

The variation $\delta \bar{\mathbf{e}}_{im}$ of the strain coordinates in the unknown instant configuration is used as weight function on the left-hand side of the integral form (3.108). Since the body is in equilibrium in the known instant configuration, the strain variation $\delta \bar{\mathbf{e}}_{im}$ for that state is set to null:

$$\delta \bar{\mathbf{e}}_{im} = \delta(\Delta \mathbf{e}_{im}) = \delta(\Delta \mathbf{e}_{im}^0) + \delta(\Delta \mathbf{e}_{im}^n) \quad (3.119)$$

$$\delta(\Delta \mathbf{e}_{im}^0) = \frac{1}{2} \left(\delta(\mathbf{u}_{i,m}) + \delta(\mathbf{u}_{m,i}) + \sum_k (\mathbf{u}_{k,i} \delta(\Delta \mathbf{u}_{k,m}) + \mathbf{u}_{k,m} \delta(\Delta \mathbf{u}_{k,i})) \right)$$

$$\delta(\Delta \mathbf{e}_{im}^n) = \frac{1}{2} \sum_k (\Delta \mathbf{u}_{k,i} \delta(\Delta \mathbf{u}_{k,m}) + \Delta \mathbf{u}_{k,m} \delta(\Delta \mathbf{u}_{k,i}))$$

$\delta(\Delta \mathbf{e}_{im}^0)$ variation of the linear strain increment

$\delta(\Delta \mathbf{e}_{im}^n)$ variation of the nonlinear strain increment

3.9.2 Linearisation of the governing equations

The strain variation (3.118) and the stress tensor from (3.111) are substituted into the left-hand side of the material form (3.108) of the governing equations:

$$\begin{aligned} \int_{\mathcal{C}} \sum_i \sum_m \delta \bar{\mathbf{e}}_{im} \bar{\mathbf{s}}_{im} dv &= \int_{\mathcal{C}} \sum_i \sum_m \delta (\Delta \mathbf{e}_{im}^0 + \Delta \mathbf{e}_{im}^n) (\mathbf{s}_{im} + \Delta \mathbf{s}_{im}) dv \\ &= \int_{\mathcal{C}} \sum_i \sum_m \delta (\Delta \mathbf{e}_{im}^0) \mathbf{s}_{im} dv + \int_{\mathcal{C}} \sum_i \sum_m \delta (\Delta \mathbf{e}_{im}^0) \Delta \mathbf{s}_{im} dv + \\ &\quad \int_{\mathcal{C}} \sum_i \sum_m \delta (\Delta \mathbf{e}_{im}^n) \mathbf{s}_{im} dv + \int_{\mathcal{C}} \sum_i \sum_m \delta (\Delta \mathbf{e}_{im}^n) \Delta \mathbf{s}_{im} dv \end{aligned} \quad (3.120)$$

The displacement variation (3.118) is substituted on the right-hand side of (3.108):

$$\begin{aligned} \int_{\mathcal{C}_u} \delta \bar{\mathbf{u}}_i \bar{\mathbf{q}}_i \rho dv + \int_{\mathcal{C}_u} \delta \bar{\mathbf{u}}_i \bar{\mathbf{p}}_i da + \int_{\mathcal{C}_t} \delta \bar{\mathbf{u}}_i \bar{\mathbf{p}}_i da &= \\ &\quad \int_{\mathcal{C}} \delta (\Delta \mathbf{u}_i) \mathbf{q}_i \rho dv + \int_{\mathcal{C}_u} \delta (\Delta \mathbf{u}_i) \mathbf{p}_i da + \int_{\mathcal{C}_t} \delta (\Delta \mathbf{u}_i) \underline{\mathbf{p}}_i da + \\ &\quad \int_{\mathcal{C}} \delta (\Delta \mathbf{u}_i) \Delta \mathbf{q}_i \rho dv + \int_{\mathcal{C}_u} \delta (\Delta \mathbf{u}_i) \Delta \mathbf{p}_i da + \int_{\mathcal{C}_t} \delta (\Delta \mathbf{u}_i) \Delta \underline{\mathbf{p}}_i da \end{aligned} \quad (3.121)$$

The first term on the right-hand side of equation (3.120) and the first three terms on the right-hand side of equation (3.121) describe the equilibrium of the known instant configuration. They are collected in an error term $\delta \mathbf{e}$:

$$\delta \mathbf{e} = \int_{\mathcal{C}_u} \delta (\Delta \mathbf{u}_i) \mathbf{p}_i da + \int_{\mathcal{C}_t} \delta (\Delta \mathbf{u}_i) \underline{\mathbf{p}}_i da + \int_{\mathcal{C}} \delta (\Delta \mathbf{u}_i) \mathbf{q}_i \rho dv - \int_{\mathcal{C}} \sum_i \sum_m \delta (\Delta \mathbf{e}_{im}^0) \mathbf{s}_{im} dv \quad (3.122)$$

$\delta \mathbf{e}$ error term of the known instant configuration

The error term in equation (3.122) is null if the displacements and reactions in the computed instant configuration $\hat{\mathcal{C}}$ are exact. The last term in (3.120) is neglected because it is quadratic in the displacement increments $\Delta \mathbf{u}_i$. Expressions (3.120) to (3.122) are substituted into the material governing equations (3.108):

$$\begin{aligned} \int_{\mathcal{C}} \sum_i \sum_m \delta (\Delta \mathbf{e}_{im}^0) \Delta \mathbf{s}_{im} dv + \int_{\mathcal{C}} \sum_i \sum_m \delta (\Delta \mathbf{e}_{im}^n) \mathbf{s}_{im} dv &= \delta \mathbf{e} + \\ &\quad \int_{\mathcal{C}} \delta (\Delta \mathbf{u}_i) \Delta \mathbf{q}_i \rho dv + \int_{\mathcal{C}_u} \delta (\Delta \mathbf{u}_i) \Delta \mathbf{p}_i da + \int_{\mathcal{C}_t} \delta (\Delta \mathbf{u}_i) \Delta \underline{\mathbf{p}}_i da \end{aligned} \quad (3.123)$$

The variation of the nonlinear strain increment in the second term on the left-hand side of (3.123) is replaced by (3.119):

$$\begin{aligned} \sum_i \sum_m \delta (\Delta \mathbf{e}_{im}^n) \mathbf{s}_{im} &= \sum_i \sum_m \frac{1}{2} \mathbf{s}_{im} \sum_k (\Delta \mathbf{u}_{k,i} \delta (\Delta \mathbf{u}_{k,m}) + \Delta \mathbf{u}_{k,m} \delta (\Delta \mathbf{u}_{k,i})) \\ &= \sum_k \sum_i \sum_m \delta (\Delta \mathbf{u}_{k,i}) \mathbf{s}_{im} \Delta \mathbf{u}_{k,m} \end{aligned} \quad (3.124)$$

The derivatives $\mathbf{u}_{k,m}$ of \mathbf{u}_k are collected in a vector \mathbf{h}_k so that (3.124) becomes:

$$\sum_i \sum_m \delta (\Delta \mathbf{e}_{im}^n) \mathbf{s}_{im} = \sum_k \delta (\Delta \mathbf{h}_k)^T \mathbf{S} \Delta \mathbf{h}_k \quad (3.125)$$

Expression (3.125) is substituted into (3.123), which is then written in the Voigt notation (3.72). The resulting equation is linear in the displacement increments. It is called the incremental governing equation.

$$\int_C \delta(\Delta \boldsymbol{\epsilon}^0)^T \mathbf{C} \Delta \boldsymbol{\epsilon}^0 dv + \int_C \sum_i \delta(\Delta \mathbf{h}_i)^T \mathbf{S} \Delta \mathbf{h}_i dv = \delta e + \int_C \delta(\Delta \mathbf{u})^T \Delta \mathbf{q} \rho dv + \int_{C_u} \delta(\Delta \mathbf{u})^T \Delta \mathbf{p} da + \int_{C_t} \delta(\Delta \mathbf{u})^T \Delta \underline{\mathbf{p}} da \quad (3.126)$$

The error term (3.122) is also expressed in matrix form:

$$\delta e = \int_C \delta(\Delta \mathbf{u})^T \mathbf{q} \rho dv + \int_{C_u} \delta(\Delta u_i) p_i da + \int_{C_t} \delta(\Delta u_i) \underline{p}_i da - \int_C \delta(\Delta \boldsymbol{\epsilon}^0)^T \boldsymbol{\sigma} dv \quad (3.127)$$

The incremental governing equations (3.126) and the error term (3.127) are chosen as foundation for traditional numerical solutions of the nonlinear governing equation (3.108) for elastic bodies. It will be shown in subsequent chapters that the solution of these equations can be improved by retaining the last term in (3.120). The linear form of the incremental governing equations is then achieved by approximation of the term through iteration in each load step.

3.10 ALGEBRAIC GOVERNING EQUATIONS

3.10.1 Finite Elements

The analytical governing equations of the nonlinear theory of elasticity have been developed for bodies whose displacement field can be expressed with functions which are valid in the entire volume and on the entire surface of the body. The shape and behaviour of typical structures in civil engineering are so complex that it is usually not possible to find such mathematical functions. Approximate methods of solution are therefore developed by decomposing the body into finite elements.

Nodes and elements: Consider an elastic body of general shape. This shape is mapped to the computer by decomposing the volume of the body into elements of finite size (for example into tetrahedrons) and placing nodes at selected locations (for example the corners of the tetrahedrons). The surface of the body is subdivided into elements with nodes (for instance triangles) which are compatible with the volume elements and their nodes. The location of the nodes and the topology of the elements are mapped to the computer.

Nodal variables and interpolation functions: The value which an arbitrary variable w assumes at a specific node is called a nodal variable. The field $w(\mathbf{x})$ cannot be described completely by specifying its value at all nodes of the body, since the body contains points which are not nodes. In order to describe the field $w(\mathbf{x})$ completely, an interpolation function $w_e(\mathbf{x})$ is defined for each element of the body. This function yields the value of w at an arbitrary point \mathbf{x} inside the element as a function of the coordinates of \mathbf{x} and selected nodal values of w . Outside the element, the value of the function is null. The field $w(\mathbf{x})$ is the sum of the element interpolation functions.

$$w(\mathbf{x}) = \sum_e w_e \quad (3.128)$$

$$\mathbf{x} \in E: \quad w_e(\mathbf{x}) = f(\mathbf{x}, w_{n_0}, \dots, w_{n_k})$$

$$\mathbf{x} \notin E: \quad w_e(\mathbf{x}) = 0$$

w_{n_i} nodal value of w at node i of element E

Shape: The coordinates of the nodes and the topology of the elements of a body are not adequate to describe its shape. The coordinates of the location vector for the points in an element of the body must be interpolated between the nodes. This requires the definition of an interpolation function for each location coordinate. These functions together with the nodes and the elements define the shape of the reference configuration of the body.

Equations: In order to derive the algebraic governing equations for a body which has been decomposed into a set of finite elements, the displacement field $\mathbf{u}(\mathbf{x})$ of the body is expressed as a sum of element displacement functions $\mathbf{u}_e(\mathbf{x})$ in analogy to (3.128). Some of the nodal displacement variables in $\mathbf{u}(\mathbf{x})$ are prescribed, the others are treated as unknowns. The surface tractions $\hat{\mathbf{t}}(\mathbf{x})$ which act on the body are interpolated in a similar manner. The algebraic governing equations are formulated so that they yield the unknown nodal displacement and traction variables.

3.10.2 System and Element Vectors

The nodal displacement variables of the body are collected in a displacement vector for the system. The nodal displacement variables of an element are collected in an element displacement vector. The relationship between element and system displacement vectors is described with topology matrices. Independent topology matrices are defined for the volume and surface elements:

$$\mathbf{u}_e = \mathbf{T}_e \mathbf{u}_s \quad (3.129)$$

$$\mathbf{u}_r = \mathbf{T}_r \mathbf{u}_s \quad (3.130)$$

\mathbf{u}_e volume element displacement vector

\mathbf{u}_r surface element displacement vector

\mathbf{u}_s system displacement vector

\mathbf{T}_e topology matrix for volume element e

\mathbf{T}_r topology matrix for surface element r

The set of nodal loads acting on the body is arranged in a system load vector. The set of nodal reactions of the body is arranged in a system reaction vector. The dimensions of these vectors are chosen equal to the dimension of the system displacement vector. Elements of the vectors which do not correspond to loads or reactions in the body are set to null. In the following, the term nodal force is used as common term for nodal loads and nodal reactions.

The system index of a coordinate of the force at a node is equal to the system index of the coordinate of the corresponding displacement at the node. The distinction between prescribed and computed displacement coordinates is made by means of a status vector, whose dimension equals that of the system displacement vector. The value of the status coefficient is "true" if the displacement is prescribed.

$$\mathbf{u}_s = \begin{bmatrix} \dots \\ \mathbf{x} \\ \mathbf{x} \\ \dots \\ \mathbf{x} \\ \dots \end{bmatrix} \quad \mathbf{q}_s = \begin{bmatrix} \mathbf{x} \\ \circ \\ \circ \\ \mathbf{x} \\ \circ \\ \mathbf{x} \end{bmatrix} \quad \mathbf{r}_s = \begin{bmatrix} \circ \\ \dots \\ \dots \\ \circ \\ \dots \\ \circ \end{bmatrix} \quad \mathbf{s}_s = \begin{bmatrix} \mathbf{f} \\ \mathbf{t} \\ \mathbf{t} \\ \mathbf{f} \\ \mathbf{t} \\ \mathbf{f} \end{bmatrix} \quad (3.131)$$

... computed value t true
 x prescribed value f false
 o unused coefficient

\mathbf{u}_s system displacement vector

\mathbf{q}_s system load vector

\mathbf{r}_s system reaction vector

\mathbf{s}_s system status vector

3.10.3 Interpolation of Variables

Interpolation with a constant base: Consider a vector field \mathbf{w} whose coordinates are referred to a base that is independent of the displacement of the body. Examples of such bases are the global base and the local base of the reference configuration of a member of a frame. Assume that the coordinates of the nodal values of the vector are referred to the same constant base.

Consider an element vector $\mathbf{w}_e^{(s)}$ that contains the nodal values of \mathbf{w} in element e at the start of load step s , and a corresponding vector $\mathbf{w}_e^{(s+1)}$ for step $s+1$. Let the coordinates w_k of the vector at a point \mathbf{x} be interpolated with different functions $g_k(\mathbf{x})$ that are independent of state s :

$$w_k^{(s)}(\mathbf{x}) = \mathbf{g}_k^T(\mathbf{x}) \mathbf{w}_e^{(s)} \quad (3.132)$$

$$w_k^{(s+1)}(\mathbf{x}) = \mathbf{g}_k^T(\mathbf{x}) \mathbf{w}_e^{(s+1)} \quad (3.133)$$

Since the coordinates $w_k^{(s)}$ and $w_k^{(s+1)}$ are associated with the same base vector, they can be subtracted to yield the increment $\Delta w_k^{(s)}$ in the coordinate w_k during step s :

$$\Delta w_k^{(s)} = w_k^{(s+1)} - w_k^{(s)} \quad (3.134)$$

Interpolation with a variable base: Consider a vector field \mathbf{w} whose coordinates are referred to a base that changes with the displacement of the body. An example of such a base is the local base of the instant coordinate space of a member of a frame whose axis y_1 passes through the end points of the displaced member. Let the base at the start of load step s be $\mathbf{f}_1^{(s)}, \mathbf{f}_2^{(s)}, \mathbf{f}_3^{(s)}$. Assume that the coordinates of \mathbf{w} and of the nodal values of the vector are referred to the same variable base.

Consider the same element vector $\mathbf{w}_e^{(s)}$ and the same interpolation functions $g_k(\mathbf{x})$ as in the case of the constant base. Then equations (3.132) and (3.133) remain valid. The coordinates, however, no longer refer to the same base:

$$\mathbf{w}_s = w_1^{(s)} \mathbf{f}_1^{(s)} + w_2^{(s)} \mathbf{f}_2^{(s)} + w_3^{(s)} \mathbf{f}_3^{(s)} \quad (3.135)$$

$$\mathbf{w}_{s+1} = w_1^{(s+1)} \mathbf{f}_1^{(s+1)} + w_2^{(s+1)} \mathbf{f}_2^{(s+1)} + w_3^{(s+1)} \mathbf{f}_3^{(s+1)}$$

The definition (3.134) of the coordinate increments during load step s is no longer valid. Yet the incremental governing equations for load step s require the definition of increments. Additional considerations are necessary, which will be treated for frames in chapter 5. It will be shown that different interpolation functions are required for a variable and for its increment.

Interpolation for three-dimensional bodies: The coordinates of all vector fields for three-dimensional bodies are referred to the constant global base. The coordinates of the nodal values are referred to the same base. Since the global base is constant, all coordinate increments can be computed as the difference of the values of the coordinate at the beginning and end of the step, as shown in (3.134).

3.10.4 Interpolation of a Configuration

Interpolation of the displacements: The displacement vector \mathbf{u} at a point \mathbf{x} is interpolated with form matrices in the volume and surface elements:

$$\mathbf{u} = \mathbf{S}_e \mathbf{u}_e = \mathbf{S}_e \mathbf{T}_e \mathbf{u}_s \quad \mathbf{x} \in C_e \quad (3.136)$$

$$\mathbf{u} = \mathbf{S}_r \mathbf{u}_r = \mathbf{S}_r \mathbf{T}_r \mathbf{u}_s \quad \mathbf{x} \in C_r \quad (3.137)$$

$\mathbf{S}_e(\mathbf{x})$ form matrix for volume element e
 $\mathbf{S}_r(\mathbf{x})$ form matrix for surface element r

The interpolation of the variation of the displacement follows from (3.118):

$$\delta(\Delta \mathbf{u}) = \mathbf{S}_e \mathbf{T}_e \delta(\Delta \mathbf{u}_s) \quad \mathbf{x} \in C_e \quad (3.138)$$

$$\delta(\Delta \mathbf{u}) = \mathbf{S}_r \mathbf{T}_r \delta(\Delta \mathbf{u}_s) \quad \mathbf{x} \in C_r \quad (3.139)$$

The partial derivative of a displacement coordinate follows from (3.136):

$$u_i = \mathbf{s}_i^T \mathbf{T}_e \mathbf{u}_s \quad \mathbf{x} \in C_e \quad (3.140)$$

$$\frac{\partial u_i}{\partial x_m} = \frac{\partial \mathbf{s}_i^T}{\partial x_m} \mathbf{T}_e \mathbf{u}_s := \mathbf{g}_{im}^T \mathbf{T}_e \mathbf{u}_s \quad (3.141)$$

\mathbf{s}_i^T i -th row of matrix \mathbf{S}_e
 \mathbf{g}_{im} interpolation vector for displacement derivative $u_{i,m}$

The three derivatives of a displacement coordinate are collected in a vector \mathbf{h}_k :

$$\mathbf{h}_k = \mathbf{G}_k \mathbf{T}_e \mathbf{u}_s \quad (3.142)$$

$$\begin{array}{|c|} \hline u_{k,1} \\ \hline u_{k,2} \\ \hline u_{k,3} \\ \hline \end{array} = \begin{array}{|c|} \hline \mathbf{g}_{k1}^T \\ \hline \mathbf{g}_{k2}^T \\ \hline \mathbf{g}_{k3}^T \\ \hline \end{array} \mathbf{T}_e \mathbf{u}_s$$

Interpolation of the strains: If the displacement field is known, the variations of the strain increments are determined exactly by computing the displacement derivatives with (3.141) and the variation of the strain coordinates with (3.119):

$$\Delta \boldsymbol{\varepsilon}^0 = \mathbf{B}_e \Delta \mathbf{u}_e = \mathbf{B}_e \mathbf{T}_e \Delta \mathbf{u}_s \quad (3.143)$$

$$\delta \begin{array}{|c|} \hline \Delta \mathbf{e}_{11}^0 \\ \hline \Delta \mathbf{e}_{22}^0 \\ \hline \Delta \mathbf{e}_{33}^0 \\ \hline \Delta \mathbf{e}_{23}^0 \\ \hline \Delta \mathbf{e}_{31}^0 \\ \hline \Delta \mathbf{e}_{12}^0 \\ \hline \end{array} = \begin{array}{|c|} \hline \mathbf{g}_{11}^T + \sum_k u_{k,1} \mathbf{g}_{k1}^T \\ \hline \mathbf{g}_{22}^T + \sum_k u_{k,2} \mathbf{g}_{k2}^T \\ \hline \mathbf{g}_{33}^T + \sum_k u_{k,3} \mathbf{g}_{k3}^T \\ \hline 0.5(\mathbf{g}_{23}^T + \mathbf{g}_{32}^T + \sum_k (u_{k,2} \mathbf{g}_{k3}^T + u_{k,3} \mathbf{g}_{k2}^T)) \\ \hline 0.5(\mathbf{g}_{31}^T + \mathbf{g}_{13}^T + \sum_k (u_{k,3} \mathbf{g}_{k1}^T + u_{k,1} \mathbf{g}_{k3}^T)) \\ \hline 0.5(\mathbf{g}_{12}^T + \mathbf{g}_{21}^T + \sum_k (u_{k,1} \mathbf{g}_{k2}^T + u_{k,2} \mathbf{g}_{k1}^T)) \\ \hline \end{array} \mathbf{T}_e \delta(\Delta \mathbf{u}_s)$$

Interpolation of the stresses: The stress at an interior point of the body is determined from the strain at that point with the constitutive equation (3.73). At the surface of the body, the coordinates of the 1. Piola-Kirchhoff stress tensor are interpolated in the surface elements:

$$\mathbf{p} = \mathbf{S}_p \mathbf{p}_r \quad (3.144)$$

\mathbf{p}_r stress vector for element r (intensities at the nodes)

\mathbf{S}_p stress interpolation matrix

The stresses acting on the surface of the body are replaced by nodal forces of equal residual weight:

$$\int_{\partial C} \delta(\Delta \mathbf{u})^T \mathbf{p} da = \delta(\Delta \mathbf{u}_s)^T \sum_r \mathbf{T}_r^T \int_{C_r} \mathbf{S}_r^T \mathbf{S}_p \mathbf{p}_r da = \delta(\Delta \mathbf{u}_s)^T (\mathbf{q}_s + \mathbf{r}_s) \quad (3.145)$$

3.10.5 Error Vector for the Incremental Governing Equations

The stepwise numerical solution procedure does not require the solution of the governing equations (3.108) for any configuration. It does, however, require the solution of the incremental governing equation (3.126) that contains the error term δe for the governing equations of the configuration at the beginning of the step.

The error term δe is determined with equation (3.127). The first integral is replaced by the sum of integrals over the volume elements. The variation of the displacement increment is substituted from (3.138):

$$\int_C \delta(\Delta \mathbf{u})^T \mathbf{q} \rho dv = \delta(\Delta \mathbf{u}_s)^T \sum_e \mathbf{T}_e^T \int_{C_e} \mathbf{S}_e^T \mathbf{q} \rho dv \quad (3.146)$$

The second and third integrals in (3.127) are replaced by (3.145):

$$\int_{C_u} \delta(\Delta u_i)^T p_i da + \int_{C_t} \delta(\Delta u_i)^T \underline{p}_i da = \delta(\Delta \mathbf{u}_s)^T (\mathbf{q}_s + \mathbf{r}_s) \quad (3.147)$$

In the fourth integral in (3.127), the stress is computed for the given displacement field with (3.88) and (3.90) and variation (3.143) of the strain increment is substituted:

$$\int_C \delta(\Delta \boldsymbol{\epsilon}^0)^T \boldsymbol{\sigma} dv = \delta(\Delta \mathbf{u}_s)^T \sum_e \mathbf{T}_e^T \int_{C_e} \mathbf{B}_e^T \boldsymbol{\sigma} dv \quad (3.148)$$

Results (3.146) to (3.148) are substituted into the error term (3.127). The system error vector \mathbf{e}_s is defined as follows:

$$\begin{aligned} \delta e &= \delta(\Delta \mathbf{u}_s)^T \mathbf{e}_s \\ \mathbf{e}_s &= \sum_e \mathbf{T}_e^T \int_{C_e} \mathbf{S}_e^T \mathbf{q}_e \rho dv + \mathbf{q}_s + \mathbf{r}_s - \sum_e \mathbf{T}_e^T \int_{C_e} \mathbf{B}_e^T \boldsymbol{\sigma} dv \end{aligned} \quad (3.149)$$

\mathbf{e}_s system error vector

The error vector consists of a set of unbalanced forces that act at the nodes and are in force and moment equilibrium. These forces are due to the linearisation of the incremental governing equations. They can be corrected in subsequent steps of the algorithm, as is shown in chapter 6.

3.10.6 Algebraic Incremental Equations

The algebraic incremental equations are derived from (3.126). The first three integrals are replaced by sums of integrals over the volume of the elements of the body. In the first integral on the left, the strain increment is substituted from (3.143):

$$\int_C \delta(\Delta \boldsymbol{\epsilon}^0)^T \mathbf{C} \Delta \boldsymbol{\epsilon}^0 dv = \delta(\Delta \mathbf{u}_s)^T \left(\sum_e \mathbf{T}_e^T \mathbf{K}_{ve} \mathbf{T}_e \right) \Delta \mathbf{u}_s \quad (3.150)$$

$$\mathbf{K}_{ve} = \int_{C_e} \mathbf{B}_e^T \mathbf{C} \mathbf{B}_e dv$$

\mathbf{K}_{ve} deformation stiffness matrix of element e

The displacement derivatives (3.142) are substituted into the second integral:

$$\int_C \sum_i \delta(\Delta \mathbf{h}_i)^T \mathbf{S} \Delta \mathbf{h}_i dv = \delta(\Delta \mathbf{u}_s)^T \left(\sum_e \mathbf{T}_e^T \mathbf{K}_{ge} \mathbf{T}_e \right) \Delta \mathbf{u}_s \quad (3.151)$$

$$\mathbf{K}_{ve} = \int_{C_e} \sum_i \mathbf{G}_i^T \mathbf{S} \mathbf{G}_i dv$$

\mathbf{K}_{ge} geometric stiffness matrix of element e

The displacement increment (3.136) is substituted into the first integral on the right:

$$\int_C \delta(\Delta \mathbf{u})^T \Delta \mathbf{q} \rho dv = \delta(\Delta \mathbf{u}_s)^T \left(\sum_e \mathbf{T}_e^T \mathbf{q}_e \right) \quad (3.152)$$

$$\mathbf{q}_e = \int_{C_e} \mathbf{S}_e^T \Delta \mathbf{q} \rho dv$$

\mathbf{q}_e volume load vector of element e

The second and third integrals on the right are replaced by (3.145):

$$\int_{C_u} \delta(\Delta \mathbf{u})^T \Delta \mathbf{p} da + \int_{C_t} \delta(\Delta \mathbf{u})^T \Delta \underline{\mathbf{p}} da = \delta(\Delta \mathbf{u}_s)^T (\Delta \mathbf{q}_s + \Delta \mathbf{r}_s) \quad (3.153)$$

Expressions (3.149) to (3.153) are substituted into (3.126) to yield the algebraic form of the incremental governing equations:

$$\delta(\Delta \mathbf{u}_s)^T \mathbf{K}_s \Delta \mathbf{u}_s = \delta(\Delta \mathbf{u}_s)^T (\Delta \mathbf{q}_s + \Delta \mathbf{r}_s + \mathbf{e}_s) \quad (3.154)$$

$$\mathbf{K}_s = \sum_e \mathbf{T}_e^T (\mathbf{K}_{ve} + \mathbf{K}_{ge}) \mathbf{T}_e$$

$$\Delta \mathbf{q}_s = \Delta \mathbf{q}_{s1} + \sum_e \mathbf{T}_e^T \mathbf{q}_e$$

\mathbf{K}_s incremental system stiffness matrix

$\Delta \mathbf{q}_s$ incremental system load vector

$\Delta \mathbf{r}_s$ incremental system reaction vector

\mathbf{e}_s system error vector

A specific finite element for three-dimensional nonlinear analysis, such as a tetrahedron, is derived by substituting its form matrices into the formulas of this chapter.

4 PLANE TRUSSES

4.1 CONFIGURATIONS OF PLANE TRUSSES

4.1.1 Properties of a Plane Truss

An elastic plane truss is an elastic body whose shape, external influences and internal behaviour have specific properties that are defined in this section.

Bars and nodes: Consider a body composed of bars with straight axes, as shown in figure 4.1. The end points of the axis of a bar are called nodes. It is assumed that a bar does not have internal nodes. Let the cross-section of the bar normal to its axis be constant over the length of the bar and assume that its centroid lies on the axis of the bar. Two or more bars can be connected at a common node.

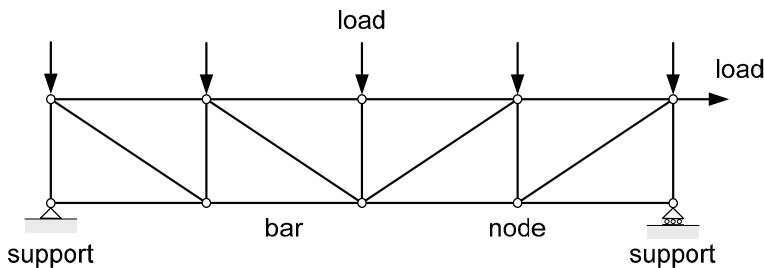


Figure 4.1 Components of a plane truss

Truss: A body consisting of bars and nodes is called a truss if the dimensions of the cross-section of each bar are much smaller than the length of its axis, and if the connections are assumed to be pinned so that they can transfer forces but not moments between the bars. The truss is called plane if the axes of all its bars are located in a common plane, and if the external influences acting on the truss are prescribed displacements and forces that satisfy the following conditions:

- The prescribed forces are called loads. The loads are applied only at nodes and act in the plane of the truss.
- The prescribed displacements are called supports. The supports are applied only at nodes and act in the plane of the truss. The displacement of all nodes of the truss normal to its plane is null.

Let the nodes and bars of the plane truss in figure 4.1 be named. The topology of the truss is described by specifying the names of the nodes at the beginning and at the end of each bar. The geometry of the truss is described by specifying the coordinates of the nodes and the area of the cross-sections of the bars. The only specified material property is the modulus of elasticity of the bars. For each load, the name of the node where it is applied, the direction in which it acts and the magnitude of the prescribed force are specified. For each support, the name of the node where it is applied, the direction in which it acts and the magnitude of the prescribed displacement are specified. The displacements of the nodes and the forces in the bars due to the external influences are to be determined.

4.1.2 Coordinate Systems

The nonlinear behaviour of trusses is described with the four Cartesian coordinate systems shown in figure 4.2 and one normalised coordinate system for bars. The dimension of the location vectors is reduced from three for the general theory of elasticity to two for plane trusses. The transformation of the coordinates of tensors between the coordinate spaces in figure 4.2 plays an important role in the nonlinear theory of plane trusses.

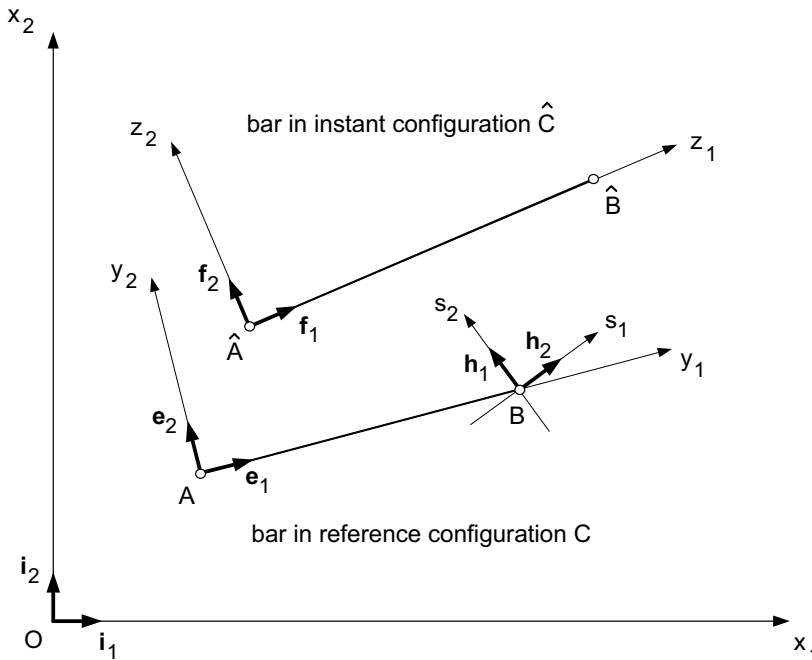


Figure 4.2 Coordinate systems for nonlinear truss analysis

- x_1, x_2 global space with base i_1, i_2
- y_1, y_2 reference space with base e_1, e_2
- z_1, z_2 instant space with base f_1, f_2
- s_1, s_2 support space with base h_1, h_2

Global space: The global coordinate system is used to describe the geometry of the truss and to specify the location of the origin as well as the coordinates of the base vectors of the other coordinate systems. The location coordinates of a point P in global space are denoted by x_1, x_2 :

$$\mathbf{x} = x_1 \mathbf{i}_1 + x_2 \mathbf{i}_2 \tag{4.1}$$

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = x_1 \begin{bmatrix} 1 \\ 0 \end{bmatrix} + x_2 \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Reference space: Each bar of the truss has a specific reference coordinate system. This coordinate system is computed in the reference configuration of the truss and does not change in the course of the analysis. The reference coordinate system is used to specify the location of points on the axis of the bar and the coordinates of their displacement. The coordinates of the stiffness matrix of the bar are referred to the reference space.

Consider bar AB in the reference configuration of the truss, as shown in figure 4.2. Let the global coordinates of points A and B be specified:

$$\mathbf{x}_A = \begin{bmatrix} x_{1A} \\ x_{2A} \end{bmatrix} \quad \mathbf{x}_B = \begin{bmatrix} x_{1B} \\ x_{2B} \end{bmatrix} \quad (4.2)$$

- \mathbf{x}_A location vector for node A in the reference configuration
- \mathbf{x}_B location vector for node B in the reference configuration
- x_{1A} global coordinates of node A in the reference configuration
- x_{1B} global coordinates of node B in the reference configuration

Node A is chosen as the origin of the reference coordinate system of the bar. The unit base vector \mathbf{e}_1 for axis y_1 points from the start node A to the end node B of the bar. The global coordinates of \mathbf{e}_1 are computed as follows:

$$L = \sqrt{(x_{1B} - x_{1A})^2 + (x_{2B} - x_{2A})^2} \quad (4.3)$$

$$\mathbf{e}_1 := \begin{bmatrix} e_{11} \\ e_{21} \end{bmatrix} = \frac{1}{L} \begin{bmatrix} (x_{1B} - x_{1A}) \\ (x_{2B} - x_{2A}) \end{bmatrix}$$

The unit base vector \mathbf{e}_2 is chosen so that it is orthogonal to \mathbf{e}_1 :

$$\mathbf{e}_2 = \begin{bmatrix} e_{12} \\ e_{22} \end{bmatrix} = \begin{bmatrix} -e_{21} \\ e_{11} \end{bmatrix} \quad (4.4)$$

The columns of the rotation matrix \mathbf{R}_{xy} for the transformation of the coordinates of a point between the global space and the reference space contain the base vectors:

$$\mathbf{x} = \mathbf{x}_A + \mathbf{R}_{xy} \mathbf{y} \quad (4.5)$$

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_{1A} \\ x_{2A} \end{bmatrix} + \begin{bmatrix} e_{11} & e_{12} \\ e_{21} & e_{22} \end{bmatrix} * \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$$

\mathbf{R}_{xy} rotation matrix global - reference space

Instant space: Each bar of the truss has a specific instant coordinate system for each instant configuration of the truss. The axial force in the bar is referred to the instant space. This coordinate system is computed in the instant configuration of the bar.

The bar of the truss in figure 4.2 has displaced from its reference location AB to an instant location $\hat{A}\hat{B}$. Let the global coordinates of points \hat{A} and \hat{B} be known:

$$\hat{\mathbf{x}}_A = \begin{bmatrix} \hat{x}_{1A} \\ \hat{x}_{2A} \end{bmatrix} \quad \hat{\mathbf{x}}_B = \begin{bmatrix} \hat{x}_{1B} \\ \hat{x}_{2B} \end{bmatrix} \quad (4.6)$$

$\hat{\mathbf{x}}_A, \hat{\mathbf{x}}_B$ location vectors for nodes A and B
 $\hat{x}_{1A}, \hat{x}_{1B}$ global coordinates of nodes A and B in reference space

The displaced node \hat{A} is chosen as the origin of the instant coordinate system of the bar in the instant configuration \hat{C} . The unit base vector $\hat{\mathbf{f}}_1$ for axis z_1 points from the start node \hat{A} to the end node \hat{B} of the bar. The global coordinates of $\hat{\mathbf{f}}_1$ are computed as follows:

$$\hat{L} = \sqrt{(\hat{x}_{1B} - \hat{x}_{1A})^2 + (\hat{x}_{2B} - \hat{x}_{2A})^2} \quad (4.7)$$

$$\hat{\mathbf{f}}_1 := \begin{bmatrix} f_{11} \\ f_{21} \end{bmatrix} = \frac{1}{\hat{L}} \begin{bmatrix} \hat{x}_{1B} - \hat{x}_{1A} \\ \hat{x}_{2B} - \hat{x}_{2A} \end{bmatrix}$$

The unit base vector $\hat{\mathbf{f}}_2$ is chosen so that it is orthogonal to $\hat{\mathbf{f}}_1$:

$$\hat{\mathbf{f}}_2 = \begin{bmatrix} f_{12} \\ f_{22} \end{bmatrix} = \begin{bmatrix} -f_{21} \\ f_{11} \end{bmatrix} \quad (4.8)$$

The columns of the rotation matrix \mathbf{R}_{xz} for the transformation of the coordinates of a point between the global space and the reference space contain the base vectors:

$$\mathbf{x} = \hat{\mathbf{x}}_A + \mathbf{R}_{xz} \mathbf{z} \quad (4.9)$$

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_{1A} \\ x_{2A} \end{bmatrix} + \begin{bmatrix} f_{11} & f_{12} \\ f_{21} & f_{22} \end{bmatrix} * \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}$$

\mathbf{R}_{xz} rotation matrix global - instant space

The transformation between reference and instant space follows from (4.5) and (4.9):

$$\mathbf{x}_A + \mathbf{R}_{xy} \mathbf{y} = \hat{\mathbf{x}}_A + \mathbf{R}_{xz} \mathbf{z} \quad (4.10)$$

$$\mathbf{y}_A = \mathbf{R}_{xy}^T (\hat{\mathbf{x}}_A - \mathbf{x}_A)$$

$$\mathbf{R}_{yz} = \mathbf{R}_{xy}^T \mathbf{R}_{xz} \quad (4.11)$$

\mathbf{R}_{yz} rotation matrix reference - instant space

Support space: The directions in which the displacements and forces at a support are prescribed may not coincide with the global coordinate axes. A Cartesian support coordinate system is therefore defined at each node of the frame. The node itself is the origin of the coordinate system. If the directions of the support coordinate axes are not specified explicitly, they coincide with the directions of the global coordinate axes. The coordinates of the support space are denoted by s_1, s_2 and the orthonormal base vectors by $\mathbf{h}_1, \mathbf{h}_2$.

$$\mathbf{x} = \mathbf{x}_N + \mathbf{R}_{xs} \mathbf{s} \quad (4.12)$$

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_{1N} \\ x_{2N} \end{bmatrix} + \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} * \begin{bmatrix} s_1 \\ s_2 \end{bmatrix}$$

\mathbf{R}_{xs} rotation matrix global - support space

h_{im} coordinates of base vector \mathbf{h}_m

The transformation from support to reference space follows from (4.5) and (4.12):

$$\begin{aligned} \mathbf{x}_A + \mathbf{R}_{xy} \mathbf{y} &= \mathbf{x}_N + \mathbf{R}_{xs} \mathbf{s} \\ \mathbf{s} &= \mathbf{s}_A + \mathbf{R}_{sy} \mathbf{y} \end{aligned} \quad (4.13)$$

$$\begin{aligned} \mathbf{s}_A &= \mathbf{R}_{xs}^T (\mathbf{x}_A - \mathbf{x}_N) \\ \mathbf{R}_{sy} &= \mathbf{R}_{xs}^T \mathbf{R}_{xy} \end{aligned} \quad (4.14)$$

\mathbf{R}_{sy} rotation matrix support - reference space

Normalised coordinate: Points on the axis of a bar are identified with a normalised coordinate z that is independent of the configuration of the frame:

$$\begin{aligned} y_1 &= \frac{1}{2}(1+z)\mathbf{a} & -1 \leq z \leq 1 & \quad (4.15) \\ z_1 &= \frac{1}{2}(1+z)\hat{\mathbf{a}} \end{aligned}$$

z normalised axial coordinate of the bar

\mathbf{a} length of the bar in the reference configuration

$\hat{\mathbf{a}}$ length of the bar in the instant configuration

Transformation of coordinates: The general rules for the transformation of tensor coordinates with changes of coordinate space were derived in section 3.1.3. Results (3.12) and (3.14) are now specialised for two-dimensional space.

Consider a vector denoted by \mathbf{u} if it is expressed as a linear combination of the global base vectors \mathbf{i}_k , and by \mathbf{v} if it is expressed as a linear combination of the local base vectors \mathbf{j}_k . The coordinates of the vector transform according to (3.12):

$$\mathbf{u} = \mathbf{R} \mathbf{v} \quad (4.16)$$

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} e_{11} & e_{12} \\ e_{21} & e_{22} \end{bmatrix} * \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$$

Consider a matrix denoted by \mathbf{U} if it is expressed as a linear combination of vector products of the global base vectors \mathbf{i}_k , and by \mathbf{V} if it is expressed as a linear combination of vector products of the local base vectors \mathbf{j}_k . The coordinates of the matrix transform according to (3.14):

$$\mathbf{U} = \mathbf{R}\mathbf{V}\mathbf{R}^T \quad (4.17)$$

$$\begin{bmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{bmatrix} = \begin{bmatrix} e_{11} & e_{12} \\ e_{21} & e_{22} \end{bmatrix} * \begin{bmatrix} v_{11} & v_{12} \\ v_{21} & v_{22} \end{bmatrix} * \begin{bmatrix} e_{11} & e_{21} \\ e_{12} & e_{22} \end{bmatrix}$$

4.1.3 Plane Truss Hypothesis

The plane truss hypothesis is formulated to reduce the three-dimensional theory of elasticity to a one-dimensional theory for a bar of a plane truss.

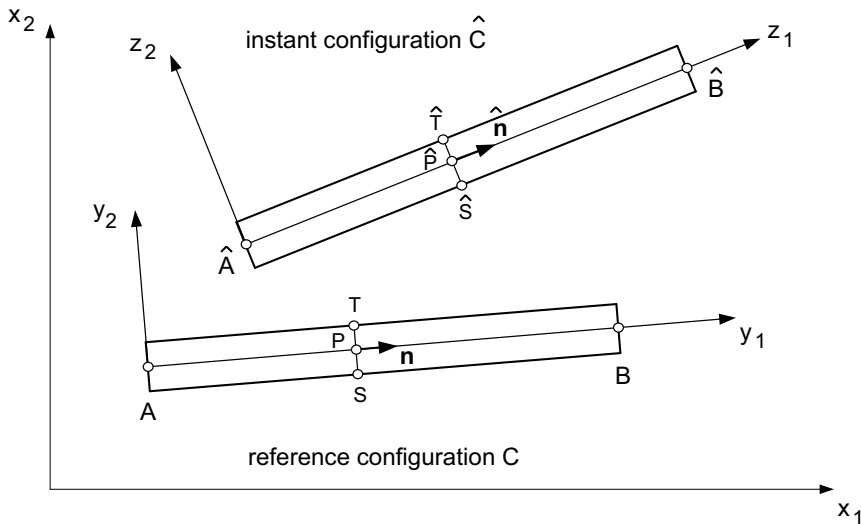


Figure 4.3 Displacement of a bar

Consider a point P on the axis AB of the reference configuration C of the bar in figure 4.3. Let SPT be a plane cross-section of the bar whose normal \mathbf{n} is parallel to the axial base vector \mathbf{e}_1 . The truss hypothesis postulates the following properties of the material points of the bar in the instant configuration \hat{C} :

- The axis $\hat{A}\hat{P}\hat{B}$ of the bar in the instant configuration is a straight line.
- The displaced cross-section $\hat{S}\hat{P}\hat{T}$ is plane.
- The shape of cross-section $\hat{S}\hat{P}\hat{T}$ equals the shape of cross-section SPT .
- The normal to cross-section $\hat{S}\hat{P}\hat{T}$ is parallel to the axial base vector \mathbf{f}_1 of the instant configuration.

4.1.4 State of Displacement

Node displacement vectors: The difference between the location vector of a node of a truss in an instant configuration of the truss and its location vector in the reference configuration is called the displacement of the node in the instant configuration:

$$\mathbf{u} := \hat{\mathbf{x}} - \mathbf{x} \quad (4.18)$$

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \end{bmatrix} - \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

\mathbf{u} node displacement vector for the instant configuration \hat{C}
 u_i global displacement coordinates of the node

The displacement vector of a node n is denoted by different symbols depending on the coordinate space to which the coordinates of the vector are referred:

$$\mathbf{u}_n = \begin{bmatrix} u_{1n} \\ u_{2n} \end{bmatrix} \quad \mathbf{v}_n = \begin{bmatrix} v_{1n} \\ v_{2n} \end{bmatrix} \quad \mathbf{w}_n = \begin{bmatrix} w_{1n} \\ w_{2n} \end{bmatrix} \quad \mathbf{d}_n = \begin{bmatrix} d_{1n} \\ d_{2n} \end{bmatrix} \quad (4.19)$$

$\mathbf{u}_n, \mathbf{v}_n, \mathbf{w}_n, \mathbf{d}_n$ displacement vectors for node n
 u_{1n}, u_{2n} displacement coordinates in global space x_1, x_2
 v_{1n}, v_{2n} displacement coordinates in reference space y_1, y_2
 w_{1n}, w_{2n} displacement coordinates in instant space z_1, z_2
 d_{1n}, d_{2n} displacement coordinates in support space s_1, s_2

The transformations between the coordinates of the vectors in (4.19) follow from the general rule of transformation (4.16) for tensor coordinates. For the transformation between global and reference space, it follows from (4.5) and (4.18) that:

$$\mathbf{u}_n = \hat{\mathbf{x}}_n - \mathbf{x}_n = \mathbf{x}_A + \mathbf{R}_{xy}(\mathbf{y}_n + \mathbf{v}_n) - (\mathbf{x}_A + \mathbf{R}_{xy}\mathbf{y}_n)$$

$$\mathbf{u}_n = \mathbf{R}_{xy}\mathbf{v}_n \quad (4.20)$$

\mathbf{R}_{xy} rotation matrix global - reference space, see (4.5)

Similar transformation rules are obtained for the other spaces:

$$\mathbf{u}_n = \mathbf{R}_{xy}\mathbf{v}_n \quad (4.21)$$

$$\mathbf{u}_n = \mathbf{R}_{xz}\mathbf{w}_n \quad (4.22)$$

$$\mathbf{v}_n = \mathbf{R}_{yz}\mathbf{w}_n \quad (4.23)$$

$$\mathbf{d}_n = \mathbf{R}_{sy}\mathbf{v}_n \quad (4.24)$$

Bar displacement vectors: The node displacement vectors for the nodes A and B of the bar shown in figure 4.3 are collected in the bar displacement vector, which is denoted by different symbols depending on the space to which its coordinates are referred:

$$\mathbf{u}_e = \begin{bmatrix} u_{1A} \\ u_{2A} \\ u_{1B} \\ u_{2B} \end{bmatrix} \quad \mathbf{v}_e = \begin{bmatrix} v_{1A} \\ v_{2A} \\ v_{1B} \\ v_{2B} \end{bmatrix} \quad \mathbf{w}_e = \begin{bmatrix} w_{1A} \\ w_{2A} \\ w_{1B} \\ w_{2B} \end{bmatrix} \quad \mathbf{d}_e = \begin{bmatrix} d_{1A} \\ d_{2A} \\ d_{1B} \\ d_{2B} \end{bmatrix} \quad (4.25)$$

The rules of transformation for the coordinates of the bar displacement vectors follow from the rules of transformation for the node displacement vectors in (4.21) to (4.24):

$$\mathbf{u}_e = \mathbf{R}_{exy} \mathbf{v}_e \quad (4.26)$$

$$\mathbf{u}_e = \mathbf{R}_{exz} \mathbf{w}_e \quad (4.27)$$

$$\mathbf{v}_e = \mathbf{R}_{eyz} \mathbf{w}_e \quad (4.28)$$

$$\mathbf{d}_e = \mathbf{R}_{esy} \mathbf{v}_e \quad (4.29)$$

\mathbf{R}_{exy} element rotation matrix global - reference space

\mathbf{R}_{exz} element rotation matrix global - instant space

\mathbf{R}_{eyz} element rotation matrix reference - instant space

\mathbf{R}_{eny} element rotation matrix support - reference space

Each of the element rotation matrices in (4.26) to (4.29) contains the coordinate rotation matrices as diagonal matrices. For example, \mathbf{R}_{exy} has the following structure:

$$\mathbf{R}_{exy} = \begin{bmatrix} \mathbf{R}_{xy} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_{xy} \end{bmatrix} \quad (4.30)$$

Since the coordinate rotation matrices are orthonormal, the element rotation matrices are also orthonormal:

$$\mathbf{R}_{exy}^T \mathbf{R}_{exy} = \mathbf{I} \quad (4.31)$$

4.1.5 State of Strain

The state of strain in a bar is simplified considerably if the truss hypothesis in section 4.1.3 is stipulated. The bar displacement can then be decomposed into three components: a rigid body motion so that the material point at node A is displaced to location \hat{A} , a rigid body rotation so that the axis of the bar points in the direction from \hat{A} to \hat{B} and a stretching of the axis so that node B is displaced to location \hat{B} .

The stretching is the only component of the displacement that leads to strain. The coordinates of the Green strain tensor (3.38) are simplified as follows:

(a) Since the truss hypothesis stipulates that the shape of the cross-section of the bar remains unchanged, the strains e_{22} , e_{33} and e_{23} of the Green tensor are null.

(b) Since the truss hypothesis stipulates that the cross-section remains plane and that the normal of the displaced section points in the direction of the displaced axis of the bar, the coordinates e_{12} and e_{13} of the strain tensor of Green are null.

The only component of the strain tensor that is not null is the axial strain ϵ_{11} , which is constant over the cross-section. It is denoted by ϵ and determined with the general formulas (3.38) of the theory of elasticity:

$$\epsilon = v_{1,1} + \frac{1}{2}(v_{1,1}^2 + v_{2,1}^2) \quad (4.32)$$

$$v_{i,1} := \frac{dv_i}{dy_1}$$

- ϵ axial strain in the bar
- v_i displacement coordinate referred to reference space
- y_1 local axial coordinate in the reference configuration of the bar

The displacement derivatives will be derived from the element displacement vector after the displacement interpolation in the bars has been specified.

4.1.6 Axial Bar Force

The axial strain ϵ in the bar is given by expression (4.32). The product of the strain ϵ and the modulus of elasticity E yields the 2. Piola-Kirchhoff stress s , which is the coefficient of the base vector \mathbf{b}_1 of the instant configuration, as defined in (3.68). The axial force vector \mathbf{a}_1 in the bar is therefore given by:

$$\begin{aligned} s &= E\epsilon = E\left(v_{1,1} + \frac{1}{2}(v_{1,1}^2 + v_{2,1}^2)\right) \\ \mathbf{a}_1 &= A s \mathbf{b}_1 = AE \epsilon \mathbf{b}_1 \end{aligned} \quad (4.33)$$

- s axial 2. Piola-Kirchhoff stress
- ϵ axial strain of Green
- E modulus of elasticity
- \mathbf{a}_1 axial force in the bar
- A area of the cross-section of the bar
- \mathbf{b}_1 axial base vector of the instant configuration

The base vector \mathbf{b}_1 is not a unit vector. Its coordinates are expressed in (3.30) as a function of the state of displacement of the bar. The engineering stress σ is the coefficient of a unit vector pointing in the direction of the axis of the bar in the instant configuration. The amount a_1 of the axial force and the engineering stress σ in the axial direction are derived from (4.33):

$$a_1 = AE\epsilon\sqrt{(1+v_{1,1})^2 + v_{2,1}^2} = A\sigma \quad (4.34)$$

$$\sigma = E\left(v_{1,1} + \frac{1}{2}(v_{1,1}^2 + v_{2,1}^2)\right)\sqrt{(1+v_{1,1})^2 + v_{2,1}^2} \quad (4.35)$$

- a_1 amount of the axial force
- σ axial engineering stress

The base vector \mathbf{b}_1 in (4.33) points in the direction of the instant axis of the bar. The bar force \mathbf{a}_1 in (4.33) therefore acts in the direction from the start node to the end node of the bar in the instant configuration.

4.2 ANALYTICAL GOVERNING EQUATIONS FOR PLANE TRUSSES

4.2.1 System Vectors

The variables defined at the nodes of a truss are collected in system vectors. The analytical governing equations for a plane truss are formulated in terms of these system vectors.

$$\mathbf{d}_s = \begin{bmatrix} \dots \\ \text{x} \\ \text{x} \\ \dots \\ \text{x} \\ \dots \end{bmatrix} \quad \mathbf{q}_s = \begin{bmatrix} \text{x} \\ \circ \\ \circ \\ \text{x} \\ \circ \\ \text{x} \end{bmatrix} \quad \mathbf{r}_s = \begin{bmatrix} \circ \\ \dots \\ \dots \\ \circ \\ \dots \\ \circ \end{bmatrix} \quad \mathbf{s}_s = \begin{bmatrix} \text{f} \\ \text{t} \\ \text{t} \\ \text{f} \\ \text{t} \\ \text{f} \end{bmatrix} \quad (4.36)$$

... computed value
 x prescribed value
 o unused coefficient

t true
 f false

\mathbf{d}_s system displacement vector
 \mathbf{q}_s system load vector
 \mathbf{r}_s system reaction vector
 \mathbf{s}_s system status vector

The nodes of the truss are numbered consecutively. The node vectors are entered into the system vectors in the order of the node numbers. The coordinates of the displacement and force vectors at a node are referred to the support space. In each coordinate direction at each node of the truss, either the coordinate of the displacement of the node, or the component of the force acting on the node is prescribed. The other value, which is not prescribed, is determined in the analysis of the truss.

A displacement coordinate at a node can be prescribed or free. Both prescribed and free coordinates are stored in a common system vector \mathbf{d}_s . The distinction between prescribed and computed displacement coordinates is made by means of a system status vector \mathbf{s}_s , whose dimension equals that of the system displacement vector. The value of the status coefficient is "true" if the displacement is prescribed.

A force coordinate at a node can be prescribed or free. A prescribed force coordinate is called a load. The loads are collected in a system load vector \mathbf{q}_s . The dimension of the system load vector equals that of the system displacement vector. Rows of the vector which do not correspond to loads are unused.

A free force coordinate at a node is called a reaction. All reactions are collected in a system reaction vector \mathbf{r}_s . The dimension of the system reaction vector equals that of the system displacement vector. Rows of the vector which do not correspond to reactions are unused. A row contains a reaction if the corresponding row of the system status vector contains the value true.

The displacement coordinates of a bar are collected in the element displacement vector \mathbf{v}_e defined in (4.25). In order to relate the element displacement vector \mathbf{v}_e to the system displacement vector \mathbf{d}_s , vector \mathbf{v}_e is related to the element displacement vector \mathbf{d}_e with (4.29). Element vector \mathbf{d}_e in turn is related to the system displacement vector \mathbf{d}_s with the element topology matrix.

$$\mathbf{v}_e = \mathbf{R}_{esy}^T \mathbf{d}_e = \mathbf{R}_{esy}^T \mathbf{T}_e \mathbf{d}_s \quad (4.37)$$

- \mathbf{v}_e displacement vector of bar e referred to reference space
- \mathbf{d}_s system displacement vector referred to support space
- \mathbf{R}_{esy} element rotation matrix, see (4.29)
- \mathbf{T}_e topology matrix of bar e

If the generalised displacement with index k in the element vector \mathbf{v}_e corresponds to the generalised displacement with index m in the system vector \mathbf{d}_s , the coefficient $\mathbf{T}_e(k,m)$ equals 1. All other coefficients in row k of \mathbf{T}_e are null.

4.2.2 Governing Equations for a State of the Truss

The weak form (3.108) of the governing equations for the nonlinear behaviour of elastic bodies is now specialised for plane trusses. Each integral over the volume of the body is replaced by the sum of the integrals over the volumes of the bars. Since only the axial strain ε and the axial stress s are different from null, the scalar product $\delta \boldsymbol{\varepsilon}^T \boldsymbol{\sigma}$ is replaced by the product $\delta \varepsilon s$. It is assumed that the loads are applied only at the nodes, so that the volume load term is null. Equation (3.145) is used to replace the surface loads and reactions by loads and reactions at the nodes of the truss. With these modifications, equation (3.108) takes the following form:

$$\sum_e \int_{C_e} \delta \varepsilon s dv = \delta \mathbf{d}_s^T (\mathbf{q}_s + \mathbf{r}_s) \quad (4.38)$$

- C_e volume of bar e in the reference configuration
- ε axial strain in the bar (coordinate e_{11} of the Green tensor)
- s axial stress in the bar (coordinate s_{11} of the 2. Piola-Kirchhoff tensor)
- \mathbf{d}_s system displacement vector (coordinates in support space)
- \mathbf{q}_s system load vector (coordinates in support space)
- \mathbf{r}_s system reaction vector (coordinates in support space)

Equation (4.38) is not solved directly. A state of the truss is determined in the step-wise solution procedure by solving the incremental governing equations for the step and incrementing the previous state. This solution is approximate due to linearisation. Equation (4.38) is therefore not satisfied exactly. The unbalanced forces in (4.38) are computed as an error vector and compensated by iteration. The solution procedure is formulated in the remaining sections of this chapter.

4.2.3 Increments of the State Variables

Displacement increment: The incremental governing equations (3.126) for the nonlinear behaviour of elastic bodies are now specialised for plane trusses. The displacement vector of a material point in an unknown instant configuration \bar{C} of the truss is written as the sum of the displacement vector in a known instant configuration \hat{C} and the displacement increment from \hat{C} to \bar{C} . The following equations describe the approach if the displacement coordinates of a node are referred to the global space:

$$\hat{\mathbf{x}}_s = \mathbf{x} + \mathbf{u} \quad (4.39)$$

$$\hat{\mathbf{x}}_{s+1} = \hat{\mathbf{x}}_s + \Delta \mathbf{u} \quad (4.40)$$

- \mathbf{x} location vector of the node in the reference configuration
- $\hat{\mathbf{x}}_s$ known location vector of the node in the instant configuration \hat{C}_s
- $\hat{\mathbf{x}}_{s+1}$ unknown location vector of the node in the instant configuration \hat{C}_{s+1}
- \mathbf{u} known node displacement vector in the instant configuration \hat{C}_s
- $\Delta \mathbf{u}$ unknown node displacement increment from configuration \hat{C}_s to \hat{C}_{s+1}

If the displacement coordinates of a point P on the axis of the bar are referred to the reference space, the incremental displacement is defined as follows:

$$\hat{\mathbf{y}}_s = \mathbf{y} + \mathbf{v} \quad (4.41)$$

$$\hat{\mathbf{y}}_{s+1} = \hat{\mathbf{y}}_s + \Delta \mathbf{v} \quad (4.42)$$

- \mathbf{y} location vector of point P in the reference configuration
- $\hat{\mathbf{y}}_s$ known location vector of point \hat{P}_s in the instant configuration \hat{C}_s
- $\hat{\mathbf{y}}_{s+1}$ unknown location vector of point \hat{P}_{s+1} in the instant configuration \hat{C}_{s+1}
- \mathbf{v} known vector point displacement vector in configuration \hat{C}_s
- $\Delta \mathbf{v}$ unknown point displacement increment from location \hat{P}_s to \hat{P}_{s+1}

Strain increment: The strain in the known instant configuration \hat{C}_s is given by (4.32):

$$\varepsilon = v_{1,1} + \frac{1}{2}(v_{1,1}^2 + v_{2,1}^2)$$

The strain in the unknown instant configuration \hat{C}_{s+1} is similarly given by:

$$\varepsilon + \Delta \varepsilon = (v_{1,1} + \Delta v_{1,1}) + \frac{1}{2}(v_{1,1} + \Delta v_{1,1})^2 + \frac{1}{2}(v_{2,1} + \Delta v_{2,1})^2$$

The first equation is subtracted from the second equation to obtain the increment:

$$\Delta \varepsilon = \Delta \varepsilon^0 + \Delta \varepsilon^n$$

$$\Delta \varepsilon^0 = (1 + v_{1,1})\Delta v_{1,1} + v_{2,1}\Delta v_{2,1} \quad (4.43)$$

$$\Delta \varepsilon^n = \frac{1}{2}(\Delta v_{1,1}^2 + \Delta v_{2,1}^2) \quad (4.44)$$

- $\Delta \varepsilon$ strain increment from configuration \hat{C}_s to configuration \hat{C}_{s+1}
- $\Delta \varepsilon^0$ linear component of the strain increment
- $\Delta \varepsilon^n$ nonlinear component of the strain increment

4.2.4 Variation of the Incremental Variables

Assume that the instant configuration \hat{C}_{s+1} of the truss is to be determined. The residuals of the differential governing equations are therefore weighted with the variation of the total displacement in configuration \hat{C}_{s+1} . If the coordinates of the displacements of the nodes are referred to the support space, the variation equals

$$\delta(\mathbf{d} + \Delta\mathbf{d}) = \delta\mathbf{d} + \delta(\Delta\mathbf{d})$$

Since the state of the truss in configuration \hat{C}_s is known, the variation $\delta\mathbf{d}$ of the displacement is not required and the variation of the displacement vector is therefore:

$$\delta(\mathbf{d} + \Delta\mathbf{d}) = \delta(\Delta\mathbf{d}) \quad (4.45)$$

The variation of the bar displacement vector referred to the reference or the instant space is determined by analogous reasoning:

$$\delta(\mathbf{v}_e + \Delta\mathbf{v}_e) = \delta(\Delta\mathbf{v}_e) \quad (4.46)$$

$$\delta(\mathbf{w}_e + \Delta\mathbf{w}_e) = \delta(\Delta\mathbf{w}_e) \quad (4.47)$$

The variation of the strain coordinates follows from (4.43) and (4.44):

$$\delta(\Delta\varepsilon^0) = (1 + v_{1,1})\delta(\Delta v_{1,1}) + v_{2,1}\delta(\Delta v_{2,1}) \quad (4.48)$$

$$\delta(\Delta\varepsilon^n) = \Delta v_{1,1}\delta(\Delta v_{1,1}) + \Delta v_{2,1}\delta(\Delta v_{2,1}) \quad (4.49)$$

4.2.5 Governing Equations for the Displacement Increments

The governing equation (4.38) for plane truss behaviour is applied to configuration \hat{C}_{s+1} of the truss. The left-hand side of (4.38) yields:

$$\sum_e \int_{C_e} \delta(\varepsilon + \Delta\varepsilon)(s + \Delta s) dv = \sum_e \int_{C_e} \delta(\Delta\varepsilon^0 + \Delta\varepsilon^n)(s + \Delta s) dv \quad (4.50)$$

$$\sum_e \int_{C_e} \delta(\varepsilon + \Delta\varepsilon)(s + \Delta s) dv = \sum_e \int_{C_e} \delta(\Delta\varepsilon^0) s dv + \sum_e \int_{C_e} \delta(\Delta\varepsilon^n) s dv + \sum_e \int_{C_e} \delta(\Delta\varepsilon) \Delta s dv$$

The right-hand side of (4.38) yields:

$$\delta(\mathbf{d}_s + \Delta\mathbf{d}_s)^T (\mathbf{q}_s + \Delta\mathbf{q}_s) + \delta(\mathbf{d}_s + \Delta\mathbf{d}_s)^T (\mathbf{r}_s + \Delta\mathbf{r}_s) = \delta(\Delta\mathbf{d}_s)^T \mathbf{q}_s + \delta(\Delta\mathbf{d}_s)^T \Delta\mathbf{q}_s + \delta(\Delta\mathbf{d}_s)^T \mathbf{r}_s + \delta(\Delta\mathbf{d}_s)^T \Delta\mathbf{r}_s \quad (4.51)$$

Consider the terms in expressions (4.50) and (4.51) which are associated with the state of displacement, load and reaction in configuration \hat{C}_s . If the solution for \hat{C}_s were exact, these terms would cancel each other due to (4.38). Because of the approximations made in the preceding steps, the solution for \hat{C}_s is not exact and these terms lead to an error δe whose value is given by:

$$\delta e = - \sum_e \int_{C_e} \delta(\Delta\varepsilon^0) s dv + \delta(\Delta\mathbf{d}_s)^T \mathbf{q}_s + \delta(\Delta\mathbf{d}_s)^T \mathbf{r}_s \quad (4.52)$$

The variation of the strain increment is substituted from (4.48) into (4.52):

$$\delta e = - \sum_e \int_{C_e} ((1 + v_{1,1})\delta(\Delta v_{1,1}) + v_{2,1}\delta(\Delta v_{2,1})) s dv + \delta(\Delta\mathbf{d}_s)^T \mathbf{q}_s + \delta(\Delta\mathbf{d}_s)^T \mathbf{r}_s \quad (4.53)$$

Since the displacements are interpolated linearly over the length of the bar, the stress \mathbf{s} and the derivatives in (4.53) are constant over the length of the bar. The integration over the length of the bar is performed analytically:

$$\delta e = - \sum_{\mathbf{e}} \mathbf{A} \mathbf{a} \mathbf{s} \left((1 + \mathbf{v}_{1,1}) \Delta \mathbf{v}_{1,1} + \mathbf{v}_{2,1} \Delta \mathbf{v}_{2,1} \right) + \delta(\Delta \mathbf{d}_s)^T \mathbf{q}_s + \delta(\Delta \mathbf{d}_s)^T \mathbf{r}_s \quad (4.54)$$

a length of the bar in the reference configuration

The variation of the nonlinear component of the strain variation is substituted from (4.49) into the second term on the right-hand side of (4.50):

$$\sum_{\mathbf{e}} \int_{C_{\mathbf{e}}} \delta(\Delta \varepsilon^n) \mathbf{s} d\mathbf{v} = \sum_{\mathbf{e}} \int_{C_{\mathbf{e}}} \left(\Delta \mathbf{v}_{1,1} \delta(\Delta \mathbf{v}_{1,1}) + \Delta \mathbf{v}_{2,1} \delta(\Delta \mathbf{v}_{2,1}) \right) \mathbf{s} d\mathbf{v} \quad (4.55)$$

The third term on the right-hand side of equation (4.50) is nonlinear in the displacement increments $\Delta \mathbf{v}_k$. The nonlinear terms are approximated iteratively in each load step. In iteration 0, the nonlinear terms are set to null. In the subsequent iterations, the approximate values of the displacement derivatives from the preceding iteration are used to approximate the nonlinear terms. Approximate values of the variables are marked with a bar superscript. The strain increment $\Delta \varepsilon$ and its variation $\delta(\Delta \varepsilon)$ are expressed with (4.43), (4.44), (4.48) and (4.49):

$$\Delta \varepsilon = (1 + \mathbf{v}_{1,1} + \frac{1}{2} \Delta \mathbf{v}_{1,1}) \Delta \mathbf{v}_{1,1} + (\mathbf{v}_{2,1} + \frac{1}{2} \Delta \mathbf{v}_{2,1}) \Delta \mathbf{v}_{2,1} \quad (4.56)$$

$$\delta(\Delta \varepsilon) = (1 + \mathbf{v}_{1,1} + \Delta \mathbf{v}_{1,1}) \delta(\Delta \mathbf{v}_{1,1}) + (\mathbf{v}_{2,1} + \Delta \mathbf{v}_{2,1}) \delta(\Delta \mathbf{v}_{2,1})$$

$\mathbf{v}_{m,1}$ derivatives of the displacement coordinates in $C^{(s)}$ with respect to y_1

$\Delta \mathbf{v}_{m,1}$ derivatives of the displacement increments from $C^{(s)}$ to $C^{(s+1)}$

The strain increment and its variation are substituted from (4.56) into the third term on the right-hand side of equation (4.50). Terms approximated with values from the preceding cycle of iteration, or with null in the first cycle of iteration, are marked with a bar.

$$\sum_{\mathbf{e}} \int_{C_{\mathbf{e}}} \mathbf{E} \delta(\Delta \varepsilon) \Delta \varepsilon d\mathbf{v} = \quad (4.57)$$

$$\sum_{\mathbf{e}} \int_{C_{\mathbf{e}}} \mathbf{E} \left((1 + \mathbf{v}_{1,1}) \delta(\Delta \mathbf{v}_{1,1}) + \mathbf{v}_{2,1} \delta(\Delta \mathbf{v}_{2,1}) \right) \left((1 + \mathbf{v}_{1,1}) \Delta \mathbf{v}_{1,1} + \mathbf{v}_{2,1} \Delta \mathbf{v}_{2,1} \right) d\mathbf{v} +$$

$$\sum_{\mathbf{e}} \int_{C_{\mathbf{e}}} \frac{1}{2} \mathbf{E} \left((1 + \mathbf{v}_{1,1}) \delta(\Delta \mathbf{v}_{1,1}) + \mathbf{v}_{2,1} \delta(\Delta \mathbf{v}_{2,1}) \right) \left(\Delta \bar{\mathbf{v}}_{1,1} \Delta \mathbf{v}_{1,1} + \Delta \bar{\mathbf{v}}_{2,1} \Delta \mathbf{v}_{2,1} \right) d\mathbf{v} +$$

$$\sum_{\mathbf{e}} \int_{C_{\mathbf{e}}} \frac{1}{2} \mathbf{E} \left(\Delta \bar{\mathbf{v}}_{1,1} \delta(\Delta \mathbf{v}_{1,1}) + \Delta \bar{\mathbf{v}}_{2,1} \delta(\Delta \mathbf{v}_{2,1}) \right) \left((1 + \mathbf{v}_{1,1}) \Delta \mathbf{v}_{1,1} + \mathbf{v}_{2,1} \Delta \mathbf{v}_{2,1} \right) d\mathbf{v} +$$

$$\sum_{\mathbf{e}} \int_{C_{\mathbf{e}}} \frac{1}{2} \mathbf{E} \left(\Delta \bar{\mathbf{v}}_{1,1} \delta(\Delta \mathbf{v}_{1,1}) + \Delta \bar{\mathbf{v}}_{2,1} \delta(\Delta \mathbf{v}_{2,1}) \right) \left(\Delta \bar{\mathbf{v}}_{1,1} \Delta \mathbf{v}_{1,1} + \Delta \bar{\mathbf{v}}_{2,1} \Delta \mathbf{v}_{2,1} \right) d\mathbf{v} +$$

$$\sum_{\mathbf{e}} \int_{C_{\mathbf{e}}} \mathbf{E} \bar{\mathbf{h}} \left(\Delta \mathbf{v}_{1,1} \delta(\Delta \mathbf{v}_{1,1}) + \Delta \mathbf{v}_{2,1} \delta(\Delta \mathbf{v}_{2,1}) \right) d\mathbf{v}$$

$$\bar{\mathbf{h}} = \frac{1}{2} \left((1 + \mathbf{v}_{1,1}) \Delta \bar{\mathbf{v}}_{1,1} + \mathbf{v}_{2,1} \Delta \bar{\mathbf{v}}_{2,1} \right) \quad (4.58)$$

The terms of equation (4.57) are integrated analytically as in equation (4.54):

$$\begin{aligned}
 \sum_{\text{e}} \int_{C_{\text{e}}} \mathbf{E} \delta(\Delta \boldsymbol{\varepsilon}) \Delta \boldsymbol{\varepsilon} \, \text{d}\mathbf{v} &= \quad (4.59) \\
 \sum_{\text{e}} \mathbf{A} \mathbf{E} \mathbf{a} \left((1 + \mathbf{v}_{1,1}) \delta(\Delta \mathbf{v}_{1,1}) + \mathbf{v}_{2,1} \delta(\Delta \mathbf{v}_{2,1}) \right) \left((1 + \mathbf{v}_{1,1}) \Delta \mathbf{v}_{1,1} + \mathbf{v}_{2,1} \Delta \mathbf{v}_{2,1} \right) &+ \\
 \sum_{\text{e}} \frac{1}{2} \mathbf{A} \mathbf{E} \mathbf{a} \left((1 + \mathbf{v}_{1,1}) \delta(\Delta \mathbf{v}_{1,1}) + \mathbf{v}_{2,1} \delta(\Delta \mathbf{v}_{2,1}) \right) \left(\Delta \bar{\mathbf{v}}_{1,1} \Delta \mathbf{v}_{1,1} + \Delta \bar{\mathbf{v}}_{2,1} \Delta \mathbf{v}_{2,1} \right) &+ \\
 \sum_{\text{e}} \frac{1}{2} \mathbf{A} \mathbf{E} \mathbf{a} \left(\Delta \bar{\mathbf{v}}_{1,1} \delta(\Delta \mathbf{v}_{1,1}) + \Delta \bar{\mathbf{v}}_{2,1} \delta(\Delta \mathbf{v}_{2,1}) \right) \left((1 + \mathbf{v}_{1,1}) \Delta \mathbf{v}_{1,1} + \mathbf{v}_{2,1} \Delta \mathbf{v}_{2,1} \right) &+ \\
 \sum_{\text{e}} \frac{1}{2} \mathbf{A} \mathbf{E} \mathbf{a} \left(\Delta \bar{\mathbf{v}}_{1,1} \delta(\Delta \mathbf{v}_{1,1}) + \Delta \bar{\mathbf{v}}_{2,1} \delta(\Delta \mathbf{v}_{2,1}) \right) \left(\Delta \bar{\mathbf{v}}_{1,1} \Delta \mathbf{v}_{1,1} + \Delta \bar{\mathbf{v}}_{2,1} \Delta \mathbf{v}_{2,1} \right) &+ \\
 \sum_{\text{e}} \mathbf{A} \mathbf{E} \mathbf{a} \bar{\mathbf{h}} \left(\Delta \mathbf{v}_{1,1} \delta(\Delta \mathbf{v}_{1,1}) + \Delta \mathbf{v}_{2,1} \delta(\Delta \mathbf{v}_{2,1}) \right) &
 \end{aligned}$$

Substitution of expression (4.54) to (4.59) into equation (4.50) yields the incremental governing equations for plane trusses.

$$\begin{aligned}
 \sum_{\text{e}} \mathbf{A} \mathbf{E} \mathbf{a} \boldsymbol{\varepsilon} \left(\Delta \mathbf{v}_{1,1} \delta(\Delta \mathbf{v}_{1,1}) + \Delta \mathbf{v}_{2,1} \delta(\Delta \mathbf{v}_{2,1}) \right) &+ \quad (4.60) \\
 \sum_{\text{e}} \mathbf{A} \mathbf{E} \mathbf{a} \left((1 + \mathbf{v}_{1,1}) \delta(\Delta \mathbf{v}_{1,1}) + \mathbf{v}_{2,1} \delta(\Delta \mathbf{v}_{2,1}) \right) \left((1 + \mathbf{v}_{1,1}) \Delta \mathbf{v}_{1,1} + \mathbf{v}_{2,1} \Delta \mathbf{v}_{2,1} \right) &+ \\
 \sum_{\text{e}} \frac{1}{2} \mathbf{A} \mathbf{E} \mathbf{a} \left((1 + \mathbf{v}_{1,1}) \delta(\Delta \mathbf{v}_{1,1}) + \mathbf{v}_{2,1} \delta(\Delta \mathbf{v}_{2,1}) \right) \left(\Delta \bar{\mathbf{v}}_{1,1} \Delta \mathbf{v}_{1,1} + \Delta \bar{\mathbf{v}}_{2,1} \Delta \mathbf{v}_{2,1} \right) &+ \\
 \sum_{\text{e}} \frac{1}{2} \mathbf{A} \mathbf{E} \mathbf{a} \left(\Delta \bar{\mathbf{v}}_{1,1} \delta(\Delta \mathbf{v}_{1,1}) + \Delta \bar{\mathbf{v}}_{2,1} \delta(\Delta \mathbf{v}_{2,1}) \right) \left((1 + \mathbf{v}_{1,1}) \Delta \mathbf{v}_{1,1} + \mathbf{v}_{2,1} \Delta \mathbf{v}_{2,1} \right) &+ \\
 \sum_{\text{e}} \frac{1}{2} \mathbf{A} \mathbf{E} \mathbf{a} \left(\Delta \bar{\mathbf{v}}_{1,1} \delta(\Delta \mathbf{v}_{1,1}) + \Delta \bar{\mathbf{v}}_{2,1} \delta(\Delta \mathbf{v}_{2,1}) \right) \left(\Delta \bar{\mathbf{v}}_{1,1} \Delta \mathbf{v}_{1,1} + \Delta \bar{\mathbf{v}}_{2,1} \Delta \mathbf{v}_{2,1} \right) &+ \\
 \sum_{\text{e}} \mathbf{A} \mathbf{E} \mathbf{a} \bar{\mathbf{h}} \left(\Delta \mathbf{v}_{1,1} \delta(\Delta \mathbf{v}_{1,1}) + \Delta \mathbf{v}_{2,1} \delta(\Delta \mathbf{v}_{2,1}) \right) &= \delta \mathbf{e} + \delta(\Delta \mathbf{d}_{\text{s}})^{\text{T}} \Delta \mathbf{q}_{\text{s}} + \delta(\Delta \mathbf{d}_{\text{s}})^{\text{T}} \Delta \mathbf{r}_{\text{s}}
 \end{aligned}$$

The equations (4.60) are linear in the unknown displacement increments $\Delta \mathbf{v}_1$ and $\Delta \mathbf{v}_2$. They are approximate because the derivatives $\Delta \bar{\mathbf{v}}_{1,1}$ and $\Delta \bar{\mathbf{v}}_{2,1}$ are approximated with values from the previous cycle of iteration in the load step. None of the terms of the exact governing equation (4.50) has been neglected.

This completes the analytic formulation of the incremental governing equations for the nonlinear behaviour of plane trusses.

4.3 ALGEBRAIC GOVERNING EQUATIONS FOR PLANE TRUSSES

4.3.1 Interpolation of the State Variables for the Error Vector

Displacement: The interpolation of the displacement of points on the axis of the bar is stipulated in part (a) of the truss hypothesis in section 4.1.3. Since the axis is a straight line in the instant configuration, the displacement coordinates vary linearly between their values at the nodes of the bar.

$$\mathbf{v} = \mathbf{S} \mathbf{v}_e \quad (4.61)$$

$$\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$$

$$\mathbf{S} = \begin{bmatrix} 0.5(1-z) & 0 & 0.5(1+z) & 0 \\ 0 & 0.5(1-z) & 0 & 0.5(1+z) \end{bmatrix}$$

$$\mathbf{v}_e^T = \begin{bmatrix} v_{1A} & v_{2A} & v_{1B} & v_{2B} \end{bmatrix}$$

- \mathbf{v} displacement vector of point z from reference to instant configuration
- \mathbf{S} interpolation matrix for the displacement coordinates in instant space
- \mathbf{v}_e element displacement vector from reference to instant configuration
- z normalised axial coordinate, see (4.15)

The variation of displacement coordinate v_m depends on row m of the interpolation matrix \mathbf{S} , which is denoted by \mathbf{s}_m^T :

$$v_m = \mathbf{s}_m^T \mathbf{v}_e \quad (4.62)$$

$$\mathbf{S} = \begin{bmatrix} \mathbf{s}_1^T \\ \mathbf{s}_2^T \end{bmatrix}$$

\mathbf{s}_m interpolation vector for displacement coordinate v_m

Derivatives of the displacement: The first derivatives of the displacement coordinates with respect to y_1 follow from (4.61) with the chain rule:

$$\frac{d\mathbf{v}}{dy_1} = \frac{d\mathbf{S}}{dy_1} \mathbf{v}_e := \mathbf{S}_{,1} \mathbf{v}_e \quad (4.63)$$

$$\mathbf{S}_{,1} = \frac{d\mathbf{S}}{dz} \frac{dz}{dy_1} = \frac{2}{a} \frac{d\mathbf{S}}{dz}$$

$$\mathbf{S}_{,1} = \frac{1}{a} \begin{bmatrix} -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix}$$

a length of the member in the reference configuration

The interpolation of the derivative of the displacement coordinate v_m on the axis of the bar is formulated with the terms of row m of the interpolation matrix $\mathbf{S}_{,1}$:

$$v_{m,1} = \frac{1}{a} \mathbf{g}_m^T \mathbf{v}_e \quad (4.64)$$

$$\mathbf{g}_1 = \begin{bmatrix} -1 \\ 0 \\ 1 \\ 0 \end{bmatrix} \quad \mathbf{g}_2 = \begin{bmatrix} 0 \\ -1 \\ 0 \\ 1 \end{bmatrix}$$

\mathbf{g}_m displacement derivative interpolation vector for coordinate v_m

Strain: The interpolation of the axial strain on the axis of a bar is determined by substitution of (4.64) into (4.32):

$$\varepsilon = \mathbf{g}_1^T \mathbf{v}_e + \frac{1}{2} ((\mathbf{g}_1^T \mathbf{v}_e)^2 + (\mathbf{g}_2^T \mathbf{v}_e)^2) \quad (4.65)$$

\mathbf{g}_m displacement derivative interpolation vector, see (4.64)

4.3.2 Interpolation of the Displacement Increments

Displacement increment: Let the bar in figure 4.4 be displaced from location AB in the reference configuration C to location $\hat{A}_s \hat{B}_s$ in the instant configuration \hat{C}_s . Let the bar be displaced further to $\hat{A}_{s+1} \hat{B}_{s+1}$ in the following instant configuration \hat{C}_{s+1} .

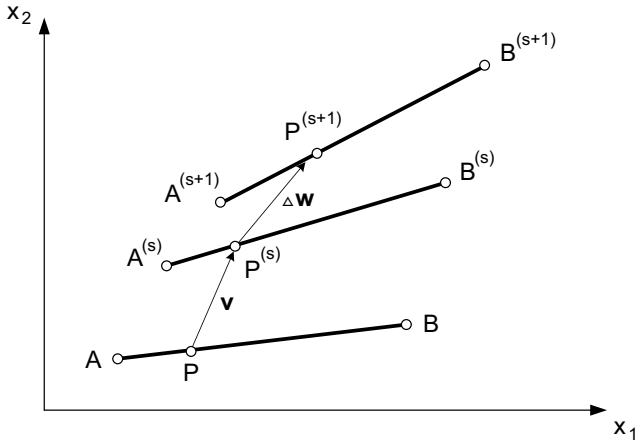


Figure 4.4 Incremental displacement of a bar

The increments Δw_1 and Δw_2 of the displacement coordinates of a point P on the axis of the bar in instant space are interpolated linearly with the normalised coordinate z :

$$\Delta \mathbf{w} = \mathbf{S} \Delta \mathbf{w}_e \quad (4.66)$$

$$\Delta \mathbf{w} = \begin{bmatrix} \Delta W_1 \\ \Delta W_2 \end{bmatrix}$$

$$\Delta \mathbf{W}_e^T = \begin{bmatrix} \Delta W_{1A} & \Delta W_{2A} & \Delta W_{1B} & \Delta W_{2B} \end{bmatrix}$$

$$\Delta \mathbf{w}_m = \mathbf{s}_m^T \Delta \mathbf{v}_e \quad (4.67)$$

$\Delta \mathbf{w}$ increment of the displacement vector of point z from configuration $\hat{\mathbf{C}}_s$ to $\hat{\mathbf{C}}_{s+1}$

$\Delta \mathbf{w}_e$ increment of the element displacement vector from configuration $\hat{\mathbf{C}}_s$ to $\hat{\mathbf{C}}_{s+1}$

The displacement coordinates are transformed with (4.23) so that they are referred to the reference space:

$$\Delta \mathbf{v} = \mathbf{R}_{yz} \Delta \mathbf{w}$$

$$\Delta \mathbf{v}_A = \mathbf{R}_{yz} \Delta \mathbf{w}_A \quad \Delta \mathbf{w}_A = \mathbf{R}_{yz}^T \Delta \mathbf{v}_A$$

$$\Delta \mathbf{v}_B = \mathbf{R}_{yz} \Delta \mathbf{w}_B \quad \Delta \mathbf{w}_B = \mathbf{R}_{yz}^T \Delta \mathbf{v}_B$$

The interpolation (4.66) is multiplied from the left with the rotation matrix \mathbf{R}_{yz} . The transformations of $\Delta \mathbf{w}_A$ and $\Delta \mathbf{w}_B$ are substituted on the right-hand side:

$$\Delta \mathbf{v} = \mathbf{R}_{yz} \begin{bmatrix} 0.5(1-z) \mathbf{I} & 0.5(1+z) \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{R}_{yz}^T \Delta \mathbf{v}_A \\ \mathbf{R}_{yz}^T \Delta \mathbf{v}_B \end{bmatrix} \quad (4.68)$$

Since the rotation matrix \mathbf{R}_{yz} is orthonormal, its products in (4.68) reduce to the unit matrix, so that the displacement increments vary linearly between nodes A and B in the reference space:

$$\Delta \mathbf{v} = \mathbf{S} \Delta \mathbf{v}_e \quad (4.69)$$

$$\Delta \mathbf{v} = \begin{bmatrix} \Delta v_1 \\ \Delta v_2 \end{bmatrix}$$

$$\Delta \mathbf{v}_e^T = \begin{bmatrix} \Delta v_{1A} & \Delta v_{2A} & \Delta v_{1B} & \Delta v_{2B} \end{bmatrix}$$

$$\Delta v_i = \mathbf{s}_i^T \Delta \mathbf{v}_e \quad (4.70)$$

$\Delta \mathbf{v}$ increment of the displacement vector of point z from configuration $\hat{\mathbf{C}}_s$ to $\hat{\mathbf{C}}_{s+1}$

$\Delta \mathbf{v}_e$ increment of the element displacement vector from configuration $\hat{\mathbf{C}}_s$ to $\hat{\mathbf{C}}_{s+1}$

\mathbf{s}_i displacement interpolation vector, see (4.62)

Displacement derivatives: The derivatives of the displacement increments with respect to y_1 follow from (4.66) with the chain rule:

$$\frac{d(\Delta \mathbf{v})}{dy_1} = \mathbf{S}_{,1} \Delta \mathbf{v}_e \quad (4.71)$$

$$\frac{d(\Delta \mathbf{v})}{dy_1} = \begin{bmatrix} \Delta v_{1,1} \\ \Delta v_{2,1} \end{bmatrix}$$

$$\Delta v_{m,1} = \frac{1}{a} \mathbf{g}_m^T \Delta \mathbf{v}_e \quad (4.72)$$

\mathbf{g}_m interpolation vector for the displacement derivative, see (4.64)

Variation of the incremental variables: The variation of the displacement increments follows from (4.70):

$$\delta(\Delta \mathbf{v}_m) = \mathbf{s}_m^T \delta(\Delta \mathbf{v}_e) \quad m \in \{1,2,3\} \quad (4.73)$$

The variation of the derivatives of the displacement increments follows from (4.72):

$$\delta(\Delta \mathbf{v}_{m,1}) = \mathbf{g}_m^T \delta(\Delta \mathbf{v}_e) \quad m \in \{1,2,3\} \quad (4.74)$$

The variation of the strain increments follows from (4.48), (4.49) and (4.56):

$$\delta(\Delta \varepsilon^0) = (1 + v_{1,1}) \mathbf{g}_1^T \delta(\Delta \mathbf{v}_e) + v_{2,1} \mathbf{g}_2^T \delta(\Delta \mathbf{v}_e) \quad (4.75)$$

$$\delta(\Delta \varepsilon^n) = \delta(\Delta \mathbf{v}_e)^T \mathbf{g}_1 \mathbf{g}_1^T \Delta \mathbf{v}_e + \delta(\Delta \mathbf{v}_e)^T \mathbf{g}_2 \mathbf{g}_2^T \Delta \mathbf{v}_e \quad (4.76)$$

$$\delta(\Delta \varepsilon) = \frac{1}{a} \left((1 + v_{1,1} + \frac{1}{2} \Delta \bar{v}_{1,1}) \mathbf{g}_1^T \delta(\Delta \mathbf{v}_e) + (v_{2,1} + \frac{1}{2} \Delta \bar{v}_{2,1}) \mathbf{g}_2^T \delta(\Delta \mathbf{v}_e) \right) + \frac{1}{2a} \delta(\Delta \mathbf{v}_e^T) (\Delta \bar{v}_{1,1} \mathbf{g}_1 + \Delta \bar{v}_{2,1} \mathbf{g}_2) \quad (4.77)$$

4.3.3 Error Vector in the Governing Equations

The increments of the displacement derivatives (4.72) are substituted into the error term (4.54) of the governing equations. The variation of the element displacement vector $\delta(\Delta \mathbf{v}_e)$ is replaced by the variation of the increment of the system displacement vector $\delta(\Delta \mathbf{d}_s)$ in (4.37):

$$\delta \mathbf{e} = \delta(\Delta \mathbf{d}_s)^T \mathbf{e}_s$$

$$\mathbf{e}_s = \sum_e \mathbf{T}_e^T \mathbf{R}_{esy} \mathbf{e}_e + \mathbf{q}_s + \mathbf{r}_s \quad (4.78)$$

$$\mathbf{e}_e = -AE\varepsilon \left((1 + v_{1,1}) \mathbf{g}_1 + v_{2,1} \mathbf{g}_2 \right) \quad (4.79)$$

$$\mathbf{e}_e = AE \left(v_{1,1} + \frac{1}{2} (v_{1,1}^2 + v_{2,1}^2) \right) \begin{array}{|c|} \hline -(1 + v_{1,1}) \\ \hline -v_{2,1} \\ \hline 1 + v_{1,1} \\ \hline v_{2,1} \\ \hline \end{array} \quad (4.80)$$

4.3.4 Algebraic Incremental Equations

The algebraic incremental equations are derived from the incremental governing equations (4.60). The derivatives of the displacement increments and their variations are substituted from (4.72) and (4.74):

$$\sum_e AE a \left((1 + v_{1,1}) \delta(\Delta v_{1,1}) + v_{2,1} \delta(\Delta v_{2,1}) \right) \left((1 + v_{1,1}) \Delta v_{1,1} + v_{2,1} \Delta v_{2,1} \right) = \sum_e \delta(\Delta \mathbf{v}_e)^T \left(AE a \left((1 + v_{1,1}) \mathbf{g}_1 + v_{2,1} \mathbf{g}_2 \right) \left((1 + v_{1,1}) \mathbf{g}_1^T + v_{2,1} \mathbf{g}_2^T \right) \right) \Delta \mathbf{v}_e \quad (4.81)$$

$$\sum_e AE a \varepsilon \left(\Delta v_{1,1} \delta(\Delta v_{1,1}) + \Delta v_{2,1} \delta(\Delta v_{2,1}) \right) = \sum_e \delta(\Delta \mathbf{v}_e)^T \left(\frac{AE}{a} \varepsilon \left(\mathbf{g}_1 \mathbf{g}_1^T + \mathbf{g}_2 \mathbf{g}_2^T \right) \right) \Delta \mathbf{v}_e \quad (4.82)$$

$$\begin{aligned} & \sum_e \frac{1}{2} A E a ((1+v_{1,1})\delta(\Delta v_{1,1}) + v_{2,1}\delta(\Delta v_{2,1})) (\Delta \bar{v}_{1,1} \Delta v_{1,1} + \Delta \bar{v}_{2,1} \Delta v_{2,1}) = \\ & = \sum_e \delta(\Delta \mathbf{v}_e)^T \left(\frac{A E}{2a} ((1+v_{1,1})\mathbf{g}_1 + v_{2,1}\mathbf{g}_2) (\Delta \bar{v}_{1,1} \mathbf{g}_1^T + \Delta \bar{v}_{2,1} \mathbf{g}_2^T) \right) \Delta \mathbf{v}_e \end{aligned} \quad (4.83)$$

$$\begin{aligned} & \sum_e \frac{1}{2} A E a (\Delta \bar{v}_{1,1} \delta(\Delta v_{1,1}) + \Delta \bar{v}_{2,1} \delta(\Delta v_{2,1})) ((1+v_{1,1})\Delta v_{1,1} + v_{2,1}\Delta v_{2,1}) = \\ & = \sum_e \delta(\Delta \mathbf{v}_e)^T \left(\frac{A E}{2a} (\Delta \bar{v}_{1,1} \mathbf{g}_1 + \Delta \bar{v}_{2,1} \mathbf{g}_2) ((1+v_{1,1})\mathbf{g}_1^T + v_{2,1}\mathbf{g}_2^T) \right) \Delta \mathbf{v}_e \end{aligned} \quad (4.84)$$

$$\begin{aligned} & \sum_e \frac{1}{2} A E a (\Delta \bar{v}_{1,1} \delta(\Delta v_{1,1}) + \Delta \bar{v}_{2,1} \delta(\Delta v_{2,1})) (\Delta \bar{v}_{1,1} \Delta v_{1,1} + \Delta \bar{v}_{2,1} \Delta v_{2,1}) = \\ & = \sum_e \delta(\Delta \mathbf{v}_e)^T \left(\frac{A E}{2a} (\Delta \bar{v}_{1,1} \mathbf{g}_1 + \Delta \bar{v}_{2,1} \mathbf{g}_2) (\Delta \bar{v}_{1,1} \mathbf{g}_1^T + \Delta \bar{v}_{2,1} \mathbf{g}_2^T) \right) \Delta \mathbf{v}_e \end{aligned} \quad (4.85)$$

$$\begin{aligned} & \sum_e A E a \bar{h} (\Delta v_{1,1} \delta(\Delta v_{1,1}) + \Delta v_{2,1} \delta(\Delta v_{2,1})) = \\ & = \sum_e \delta(\Delta \mathbf{v}_e)^T \left(\frac{A E}{a} \bar{h} (\mathbf{g}_1^T \mathbf{g}_1 + \mathbf{g}_2^T \mathbf{g}_2) \right) \Delta \mathbf{v}_e \end{aligned} \quad (4.86)$$

The variation of the element displacement vector is replaced by the variation of the system displacement vector, using the topology matrix defined in (4.37). Expressions (4.81) to (4.86) are substituted into the governing equations (4.60):

$$\delta(\Delta \mathbf{d}_s)^T \mathbf{K}_s \Delta \mathbf{d}_s = \delta(\Delta \mathbf{d}_s)^T (\mathbf{e}_s + \Delta \mathbf{q}_s + \Delta \mathbf{r}_s) \quad (4.87)$$

- \mathbf{K}_s system stiffness matrix
- $\Delta \mathbf{d}_s$ system displacement increment
- \mathbf{e}_s system error vector, see (4.79)
- $\Delta \mathbf{q}_s$ system load increment
- $\Delta \mathbf{r}_s$ system reaction increment

The system stiffness matrix is assembled from the element contributions:

$$\mathbf{K}_s = \sum_e \mathbf{T}_e^T \mathbf{R}_{\text{esy}} \mathbf{K}_e \mathbf{R}_{\text{esy}}^T \mathbf{T}_e \quad (4.88)$$

$$\mathbf{K}_e = \mathbf{K}_c + \mathbf{K}_d + \mathbf{K}_g + \mathbf{K}_m + \mathbf{K}_n + \mathbf{K}_r \quad (4.89)$$

The vector products of the derivative vectors \mathbf{g}_m are denoted as follows:

$$\mathbf{G}_{kk} := \mathbf{g}_k \mathbf{g}_k^T \quad k, m \in \{1, 2\} \quad (4.90)$$

$$\mathbf{G}_{km} := \mathbf{g}_k \mathbf{g}_m^T + \mathbf{g}_m \mathbf{g}_k^T \quad k \neq m$$

The constant component \mathbf{K}_c of the bar stiffness follows from (4.81):

$$\mathbf{K}_c = \frac{A E}{a} \mathbf{G}_{11} \quad (4.91)$$

The deviatoric component \mathbf{K}_d of the bar stiffness also follows from (4.81):

$$\mathbf{K}_d = \frac{AE}{a}(c_1 \mathbf{G}_{11} + c_2 \mathbf{G}_{22} + c_3 \mathbf{G}_{12}) \quad (4.92)$$

$$c_1 = v_{1,1}(2 + v_{1,1}) \quad c_2 = v_{2,1}^2 \quad c_3 = v_{2,1}(1 + v_{1,1})$$

The geometric component \mathbf{K}_g of the bar stiffness follows from (4.82):

$$\mathbf{K}_g = \frac{AE}{a} c_4 (\mathbf{G}_{11} + \mathbf{G}_{22}) \quad (4.93)$$

$$c_4 = v_{1,1} + \frac{1}{2}(v_{1,1}^2 + v_{2,1}^2)$$

The corrective component \mathbf{K}_m follows from (4.83) and (4.84):

$$\mathbf{K}_m = \frac{AE}{a}(c_5 \mathbf{G}_{11} + c_6 \mathbf{G}_{22} + c_7 \mathbf{G}_{12}) \quad (4.94)$$

$$c_5 = \Delta \bar{v}_{1,1}(1 + v_{1,1}) \quad c_6 = \Delta \bar{v}_{2,1} v_{2,1} \quad c_7 = \frac{1}{2}(\Delta \bar{v}_{2,1}(1 + v_{1,1}) + \Delta \bar{v}_{1,1} v_{2,1})$$

The corrective component \mathbf{K}_n follows from (4.85):

$$\mathbf{K}_n = \frac{AE}{a}(c_8 \mathbf{G}_{11} + c_9 \mathbf{G}_{22} + c_{10} \mathbf{G}_{12}) \quad (4.95)$$

$$c_8 = \frac{1}{2} \Delta \bar{v}_{1,1}^2 \quad c_9 = \frac{1}{2} \Delta \bar{v}_{2,1}^2 \quad c_{10} = \frac{1}{2} \Delta \bar{v}_{1,1} \Delta \bar{v}_{2,1}$$

The corrective component \mathbf{K}_r follows from (4.86):

$$\mathbf{K}_r = \frac{AE}{a}(c_{11} \mathbf{G}_{11} + c_{11} \mathbf{G}_{22}) \quad (4.96)$$

$$c_{11} = \bar{h}$$

The substitution of expressions (4.91) to (4.96) into (4.89) leads to the following coefficients of the element stiffness matrix \mathbf{K}_e :

$$\mathbf{K}_e = \frac{EA}{a} \begin{array}{|c|c|c|c|} \hline a_1 & a_3 & -a_1 & -a_3 \\ \hline a_3 & a_2 & -a_3 & -a_2 \\ \hline -a_1 & -a_3 & a_1 & a_3 \\ \hline -a_3 & -a_2 & a_3 & a_2 \\ \hline \end{array} \quad (4.97)$$

$$a_1 = 1 + c_1 + c_4 + c_5 + c_8 + c_{11} \quad a_2 = c_2 + c_4 + c_6 + c_9 + c_{11} \quad a_3 = c_3 + c_7 + c_{10}$$

Unit values of the displacement variables at the nodes of the truss are chosen as variations of the system displacement vector in (4.87). Each variation contributes one equation to the following system of algebraic governing equations for the increments of the displacements and the reactions of a plane truss in a load step:

$$\mathbf{K}_s \Delta \mathbf{d}_s = \mathbf{e}_s + \Delta \mathbf{q}_s + \Delta \mathbf{r}_s \quad (4.98)$$

The governing equations (4.98) are linear. The assembly and solution of these equations are described in chapter 6.

5 PLANE FRAMES

5.1 CONFIGURATIONS OF PLANE FRAMES

5.1.1 Properties of a Plane Frame

An elastic plane frame is an elastic body whose shape, external influences and internal behaviour have specific properties which are defined in this section.

Members and nodes: Consider a body composed of members with straight axes, as shown in figure 5.1. The end points of the axis of a member are called nodes. It is assumed that a member does not have internal nodes. Let the cross-section of the member normal to its axis be constant over the length of the member, and assume that the centroid of the cross-section lies on the axis of the member. Two or more members can be connected at a common node.

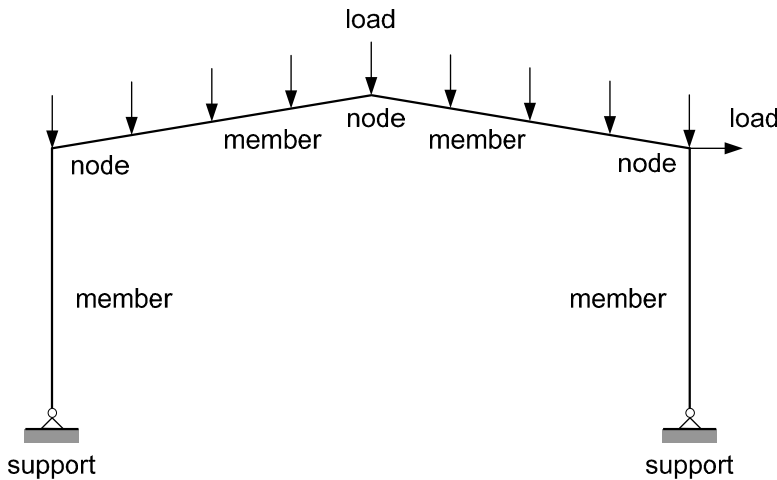


Figure 5.1 Components of a plane frame

Loads and supports: The external influences acting on the body are prescribed forces and prescribed displacements. The term force is generalised to include the concentrated forces and moments that act at the nodes and the distributed forces that act on the members. The prescribed generalised forces are called loads. The term displacement is generalised to include rotations. The prescribed displacements are called supports. The supports act only at nodes.

Frame: A body consisting of members and nodes is called a frame if the dimensions of the cross-section of each member are much smaller than the length of its axis, and if the connections can transfer both forces and moments between the members. The frame is called plane if the axes of all of its members are located in a common plane, if this plane is a plane of symmetry of the cross-sections of the members and if the external influences acting on the frame satisfy the following conditions:

- The loads are prescribed forces that act in the plane of the frame and moments that act about axes normal to the plane of the frame.

– The supports are prescribed displacements of nodes in the plane of the frame and prescribed rotations at nodes about axes normal to the plane of the frame.

Let the nodes and bars of the plane frame in figure 5.1 be named. The topology of the frame is described by specifying the names of the nodes at the beginning and at the end of each member. The geometry of the frame is described by specifying the coordinates of the nodes and the area, as well as the moment of inertia of the cross-sections of the members. The only specified material property is the modulus of elasticity of the members.

For each concentrated prescribed force, the name of the node where it is applied, the direction in which it acts and the magnitude are specified. For each concentrated prescribed moment, the name of the node where it is applied and the magnitude are specified. For each prescribed distributed load, the member on which it acts and the coordinates of the force per unit length are specified.

For each prescribed displacement, the name of the node where it is prescribed, the magnitude of the prescribed coordinate and its direction are specified. For each prescribed rotation, the name of the node where it is prescribed and its magnitude are specified.

The displacements and reactions at the nodes, as well as the forces and bending moments in the members due to the external influences, are to be determined.

5.1.2 Coordinate Systems

The nonlinear behaviour of frames is described with the four Cartesian coordinate systems shown in figure 5.2. The dimension of the location vectors is reduced from three for the general theory of elasticity to two for plane frames. The transformation of the coordinates of tensors between the coordinate spaces in figure 5.2 plays an important role in the nonlinear theory of plane frames.

Global space: The global coordinate system is used to describe the geometry of the frame and to specify the location of the origin, as well as the coordinates of the base vectors of the other coordinate systems. Some loads are referred to the global space. The location coordinates of a point P in the global space are denoted by x_1 , x_2 :

$$\mathbf{x} = x_1 \mathbf{i}_1 + x_2 \mathbf{i}_2 \quad (5.1)$$

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = x_1 \begin{bmatrix} 1 \\ 0 \end{bmatrix} + x_2 \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Reference space: Consider the reference configuration of a plane frame. A local reference coordinate system is defined for each member of the frame. The start node of the member is chosen as the origin, and the axis of the member is chosen as the local axis y_1 of the local coordinate system. The reference coordinate system is used to identify the points on the axis of the member in all phases of the nonlinear analysis (total Lagrangian approach).

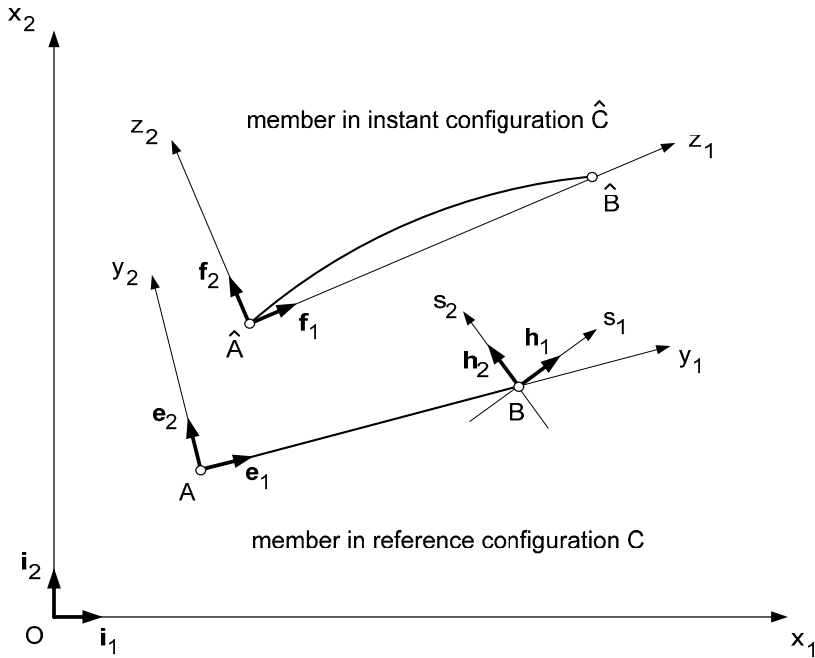


Figure 5.2 Coordinate systems for nonlinear frame analysis

- x_1, x_2 global space with base i_1, i_2
- y_1, y_2 reference space with base e_1, e_2
- z_1, z_2 instant space with base f_1, f_2
- s_1, s_2 support space with base h_1, h_2

Consider member AB in the reference configuration of the frame in figure 5.2. Let the global coordinates of points A and B be specified:

$$\mathbf{x}_A = \begin{bmatrix} x_{1A} \\ x_{2A} \end{bmatrix} \quad \mathbf{x}_B = \begin{bmatrix} x_{1B} \\ x_{2B} \end{bmatrix} \quad (5.2)$$

- \mathbf{x}_A location vector for node A in the reference configuration
- \mathbf{x}_B location vector for node B in the reference configuration
- x_{1A} global coordinates of node A in the reference configuration
- x_{1B} global coordinates of node B in the reference configuration

Node A is chosen as the origin of the reference coordinate system of the member. The unit base vector e_1 for axis y_1 points from the start node A to the end node B of the member. The global coordinates of e_1 are computed as follows:

$$L = \sqrt{(x_{1B} - x_{1A})^2 + (x_{2B} - x_{2A})^2} \quad (5.3)$$

$$\mathbf{e}_1 := \begin{bmatrix} e_{11} \\ e_{21} \end{bmatrix} = \frac{1}{L} \begin{bmatrix} (x_{1B} - x_{1A}) \\ (x_{2B} - x_{2A}) \end{bmatrix}$$

The unit base vector \mathbf{e}_2 is chosen so that it is orthogonal to \mathbf{e}_1 :

$$\mathbf{e}_2 = \begin{bmatrix} \mathbf{e}_{12} \\ \mathbf{e}_{22} \end{bmatrix} = \begin{bmatrix} -\mathbf{e}_{21} \\ \mathbf{e}_{11} \end{bmatrix} \quad (5.4)$$

The columns of the rotation matrix \mathbf{R}_{xy} for the transformation of the coordinates of a point between the global and reference spaces contain the base vectors:

$$\mathbf{x} = \mathbf{x}_A + \mathbf{R}_{xy} \mathbf{y} \quad (5.5)$$

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_{1A} \\ x_{2A} \end{bmatrix} + \begin{bmatrix} \mathbf{e}_{11} & \mathbf{e}_{12} \\ \mathbf{e}_{21} & \mathbf{e}_{22} \end{bmatrix} * \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$$

$$\mathbf{R}_{xy} = \begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 \end{bmatrix}$$

\mathbf{R}_{xy} rotation matrix global - reference space

Instant space: Each member of the frame has a specific instant coordinate system for each instant configuration of the truss. The instant coordinate system is used to interpolate the displacements between the nodes of the member. The axial force in the member is referred to the instant space.

The member of the frame in figure 5.2 has displaced from its reference location AB to an instant location $\hat{A}\hat{B}$. Let the global coordinates of points \hat{A} and \hat{B} be known:

$$\hat{\mathbf{x}}_A = \begin{bmatrix} \hat{x}_{1A} \\ \hat{x}_{2A} \end{bmatrix} \quad \hat{\mathbf{x}}_B = \begin{bmatrix} \hat{x}_{1B} \\ \hat{x}_{2B} \end{bmatrix} \quad (5.6)$$

$\hat{\mathbf{x}}_A, \hat{\mathbf{x}}_B$ location vectors for nodes A and B in the instant configuration
 $\hat{x}_{1A}, \hat{x}_{1B}$ global coordinates of nodes A and B in the instant configuration

The displaced node \hat{A} is chosen as the origin of the instant coordinate system of the member in the instant configuration \hat{C} . The unit base vector \mathbf{f}_1 for axis z_1 points from the start node \hat{A} to the end node \hat{B} of the member. The global coordinates of \mathbf{f}_1 are computed as follows:

$$\hat{L} = \sqrt{(\hat{x}_{1B} - \hat{x}_{1A})^2 + (\hat{x}_{2B} - \hat{x}_{2A})^2} \quad (5.7)$$

$$\mathbf{f}_1 := \begin{bmatrix} f_{11} \\ f_{21} \end{bmatrix} = \frac{1}{\hat{L}} \begin{bmatrix} (\hat{x}_{1B} - \hat{x}_{1A}) \\ (\hat{x}_{2B} - \hat{x}_{2A}) \end{bmatrix}$$

The unit base vector \mathbf{f}_2 is chosen so that it is orthogonal to \mathbf{f}_1 :

$$\mathbf{f}_2 = \begin{bmatrix} f_{12} \\ f_{22} \end{bmatrix} = \begin{bmatrix} -f_{21} \\ f_{11} \end{bmatrix} \quad (5.8)$$

The columns of the rotation matrix \mathbf{R}_{xz} for the transformation of the coordinates of a point between the global and reference spaces contain the base vectors:

$$\mathbf{x} = \hat{\mathbf{x}}_A + \mathbf{R}_{xz} \mathbf{z} \quad (5.9)$$

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \hat{x}_{1A} \\ \hat{x}_{2A} \end{bmatrix} + \begin{bmatrix} f_{11} & f_{12} \\ f_{21} & f_{22} \end{bmatrix} * \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}$$

$$\mathbf{R}_{xz} = \begin{bmatrix} | & | \\ \mathbf{f}_1 & \mathbf{f}_2 \\ | & | \end{bmatrix}$$

\mathbf{R}_{xz} rotation matrix global - instant space

Support space: A Cartesian node coordinate system is defined at each node of the frame. The node itself is the origin of the coordinate system. If the directions of the node coordinate axes are not specified explicitly, they coincide with the directions of the global coordinate axes. The coordinates of the support space are denoted by s_1, s_2 , and the orthonormal base vectors by $\mathbf{h}_1, \mathbf{h}_2$.

$$\mathbf{x} = \mathbf{x}_N + \mathbf{R}_{xs} \mathbf{s} \quad (5.10)$$

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_{1N} \\ x_{2N} \end{bmatrix} + \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} * \begin{bmatrix} s_1 \\ s_2 \end{bmatrix}$$

\mathbf{R}_{xs} rotation matrix global - support space

Transformations between location coordinates: The transformations between the coordinate systems of a member in (5.5), (5.9) and (5.10) are inverted using (3.9):

$$\text{reference - global: } \mathbf{y} = \mathbf{R}_{xy}^T (\mathbf{x} - \mathbf{x}_A) \quad (5.11)$$

$$\text{instant - global: } \mathbf{z} = \mathbf{R}_{xz}^T (\mathbf{x} - \hat{\mathbf{x}}_A)$$

$$\text{support - global: } \mathbf{s} = \mathbf{R}_{xs}^T (\mathbf{x} - \mathbf{x}_N)$$

The transformation between reference and instant space follows from (5.5) and (5.9):

$$\mathbf{x}_A + \mathbf{R}_{xy} \mathbf{y} = \hat{\mathbf{x}}_A + \mathbf{R}_{xz} \mathbf{z}$$

$$\mathbf{y} = \mathbf{y}_A + \mathbf{R}_{yz} \mathbf{z} \quad (5.12)$$

$$\mathbf{z} = \mathbf{R}_{yz}^T (\mathbf{y} - \mathbf{y}_A)$$

$$\mathbf{y}_A = \mathbf{R}_{xy}^T (\hat{\mathbf{x}}_A - \mathbf{x}_A)$$

$$\mathbf{R}_{yz} = \mathbf{R}_{xy}^T \mathbf{R}_{xz}$$

$$\mathbf{R}_{yz} = \begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix}$$

\mathbf{R}_{yz} rotation matrix reference - instant space

The transformation between the support and the reference space follows from (5.5) and (5.10):

$$\begin{aligned} \mathbf{x}_A + \mathbf{R}_{xy} \mathbf{y} &= \mathbf{x}_N + \mathbf{R}_{xs} \mathbf{s} \\ \mathbf{s} &= \mathbf{s}_A + \mathbf{R}_{sy} \mathbf{y} \end{aligned} \quad (5.13)$$

$$\mathbf{y} = \mathbf{R}_{sy}^T (\mathbf{s} - \mathbf{s}_A)$$

$$\mathbf{s}_A = \mathbf{R}_{xs}^T (\mathbf{x}_A - \mathbf{x}_N)$$

$$\mathbf{R}_{sy} = \mathbf{R}_{xs}^T \mathbf{R}_{xy}$$

\mathbf{R}_{sy} rotation matrix support - reference space

The transformation between support and instant space follows from (5.9) and (5.10):

$$\begin{aligned} \hat{\mathbf{x}}_A + \mathbf{R}_{xz} \mathbf{z} &= \mathbf{x}_N + \mathbf{R}_{xs} \mathbf{s} \\ \mathbf{s} &= \hat{\mathbf{s}}_A + \mathbf{R}_{sz} \mathbf{z} \end{aligned} \quad (5.14)$$

$$\mathbf{z} = \mathbf{R}_{sz}^T (\mathbf{s} - \mathbf{s}_A)$$

$$\hat{\mathbf{s}}_A = \mathbf{R}_{xs}^T (\hat{\mathbf{x}}_A - \mathbf{x}_N)$$

$$\mathbf{R}_{sz} = \mathbf{R}_{xs}^T \mathbf{R}_{xz}$$

\mathbf{R}_{sz} rotation matrix support - instant space

Transformation of tensor coordinates: The rules for the transformation of tensor coordinates with changes of the coordinate space were derived in section 3.1.3. Results (3.12) and (3.14) are now specialised for two-dimensional space.

Consider a vector denoted by \mathbf{u} if it is expressed as a linear combination of the global base vectors \mathbf{i}_k , and by \mathbf{v} if it is expressed as a linear combination of the local base vectors \mathbf{j}_k . The coordinates of the vector transform according to (3.12):

$$\mathbf{u} = \mathbf{R} \mathbf{v} \quad (5.15)$$

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} e_{11} & e_{12} \\ e_{21} & e_{22} \end{bmatrix} * \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$$

Consider a matrix denoted by \mathbf{U} if it is expressed as a linear combination of the vector products of the global base vectors \mathbf{i}_k , and by \mathbf{V} if it is expressed as a linear combination of the vector products of the local base vectors \mathbf{j}_k . The coordinates of the matrix transform according to (3.14):

$$\mathbf{U} = \mathbf{R} \mathbf{V} \mathbf{R}^T \quad (5.16)$$

$$\begin{bmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{bmatrix} = \begin{bmatrix} e_{11} & e_{12} \\ e_{21} & e_{22} \end{bmatrix} * \begin{bmatrix} v_{11} & v_{12} \\ v_{21} & v_{22} \end{bmatrix} * \begin{bmatrix} e_{11} & e_{21} \\ e_{12} & e_{22} \end{bmatrix}$$

The rotation and moment variables acting about the axis normal to the plane of the frame are not affected by the changes in the coordinate systems considered in this chapter.

5.1.3 Plane Frame Hypothesis

The plane frame hypothesis is formulated to reduce the three-dimensional theory of elasticity to a one-dimensional theory for a member of a plane frame. Consider a point P on the axis AB of the reference configuration C of the member in figure 5.3. Let SPT be a plane cross-section of the member whose normal \mathbf{n} is parallel to the axial base vector \mathbf{e}_1 . The plane frame hypothesis postulates the following properties of the material points of the member in the instant configuration \hat{C} :

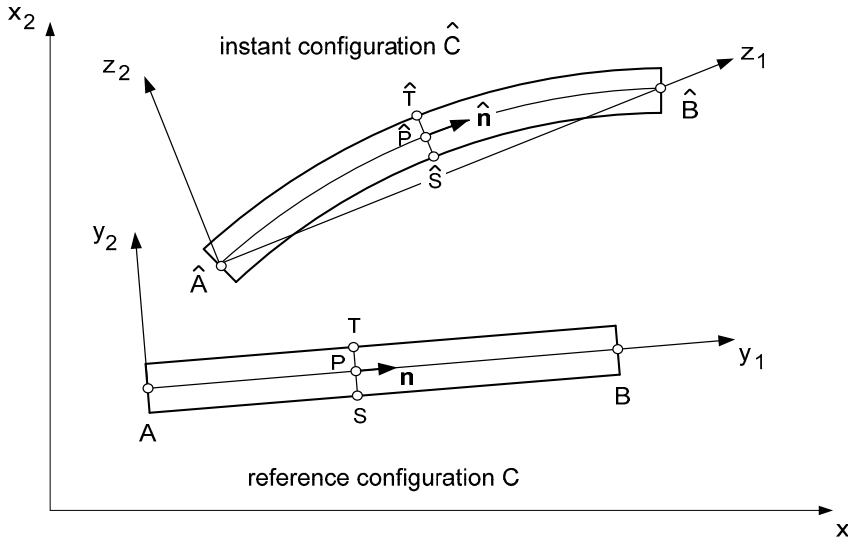


Figure 5.3 Displacement of a member

(a) **Plane:** The material points of the plane cross-section SPT of the member in the reference configuration displace to the plane cross-section $\hat{S}\hat{P}\hat{T}$ of the member in the instant configuration.

(b) **Rigid:** The material points of cross-section SPT in the reference configuration displace as a rigid body to location $\hat{S}\hat{P}\hat{T}$ in the instant configuration.

(c) **Normal:** The material points that lie on the normal to cross-section SPT in the reference configuration, lie on the axis of the member in the instant configuration. The tangent to the axis in the instant configuration at \hat{P} is parallel to the normal of cross-section $\hat{S}\hat{P}\hat{T}$ at \hat{P} .

As a consequence of stipulations (a) to (c), the member has the following properties:

(d) **Congruence:** The shape and dimensions of a cross-section do not change when the member displaces. Cross-sections SPT and $\hat{S}\hat{P}\hat{T}$ are congruent.

(e) **Displacement:** The displacement of a point Q on cross-section SPT in the reference configuration of the member to its location \hat{Q} on cross-section $\hat{S}\hat{P}\hat{T}$ of the instant configuration is the vector sum of the displacement of point P to point \hat{P} and the displacement due to a rotation of plane SPT about the normal to the plane of the frame at P .

5.1.4 State of Displacement

Node displacement vectors: A node displacement vector contains the coordinates of the displacement of the node and the angle of rotation β of the normal \mathbf{n} about the axis normal to plane of the frame. The displacement vector is denoted by $\mathbf{u}_n, \mathbf{v}_n, \mathbf{w}_n$ or \mathbf{d}_n if the displacement coordinates are referred to the global, reference, instant or support space. The rotation β is independent of the coordinate space.

$$\mathbf{u}_n = \begin{bmatrix} u_1 \\ u_2 \\ \beta \end{bmatrix} \quad \mathbf{v}_n = \begin{bmatrix} v_1 \\ v_2 \\ \beta \end{bmatrix} \quad \mathbf{w}_n = \begin{bmatrix} w_1 \\ w_2 \\ \beta \end{bmatrix} \quad \mathbf{d}_n = \begin{bmatrix} d_1 \\ d_2 \\ \beta \end{bmatrix} \quad (5.17)$$

u_1, u_2 displacement coordinates in global space x_1, x_2
 v_1, v_2 displacement coordinates in reference space y_1, y_2
 w_1, w_2 displacement coordinates in instant space z_1, z_2
 d_1, d_2 displacement coordinates in support space s_1, s_2
 β angle of rotation, positive anti-clockwise

The coordinates of the node displacement vectors are transformed by means of the rotation matrices defined in section 5.1.2:

$$\mathbf{u}_n = \mathbf{R}_{nxy} \mathbf{v}_n \quad (5.18)$$

$$\begin{bmatrix} u_1 \\ u_2 \\ \beta \end{bmatrix} = \begin{bmatrix} e_{11} & e_{12} & 0 \\ e_{21} & e_{22} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ \beta \end{bmatrix}$$

$$\mathbf{u}_n = \mathbf{R}_{nxz} \mathbf{w}_n \quad (5.19)$$

$$\begin{bmatrix} u_1 \\ u_2 \\ \beta \end{bmatrix} = \begin{bmatrix} f_{11} & f_{12} & 0 \\ f_{21} & f_{22} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \beta \end{bmatrix}$$

$$\mathbf{v}_n = \mathbf{R}_{nyz} \mathbf{w}_n \quad (5.20)$$

$$\begin{bmatrix} v_1 \\ v_2 \\ \beta \end{bmatrix} = \begin{bmatrix} g_{11} & g_{12} & 0 \\ g_{21} & g_{22} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \beta \end{bmatrix}$$

$$\mathbf{u}_n = \mathbf{R}_{nxs} \mathbf{d}_n \quad (5.21)$$

$$\begin{bmatrix} u_1 \\ u_2 \\ \beta \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & 0 \\ h_{21} & h_{22} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ \beta \end{bmatrix}$$

\mathbf{R}_{nxy} node rotation matrix global - reference space
 \mathbf{R}_{nxz} node rotation matrix global - instant space
 \mathbf{R}_{nyz} node rotation matrix reference - instant space
 \mathbf{R}_{nxs} node rotation matrix global - support space

Since the rotation matrices for the transformation of the location coordinates in section 5.1.2 are orthonormal, the node rotation matrices in (5.18) to (5.21) are also orthonormal:

$$\mathbf{R}_{nxy}^T \mathbf{R}_{nxy} = \mathbf{R}_{nxz}^T \mathbf{R}_{nxz} = \mathbf{R}_{nyz}^T \mathbf{R}_{nyz} = \mathbf{R}_{nxs}^T \mathbf{R}_{nxs} = \mathbf{I} \quad (5.22)$$

Member displacement vectors: The member displacement vector contains the node displacement vectors for the start node and the end node of the member. It is denoted by $\mathbf{u}_e, \mathbf{v}_e, \mathbf{w}_e$ or \mathbf{d}_e if the displacement coordinates are referred to the global, reference, instant or support space:

$$\mathbf{u}_e = \begin{bmatrix} u_{1A} \\ u_{2A} \\ \beta_A \\ u_{1B} \\ u_{2B} \\ \beta_B \end{bmatrix} \quad \mathbf{v}_e = \begin{bmatrix} v_{1A} \\ v_{2A} \\ \beta_A \\ v_{1B} \\ v_{2B} \\ \beta_B \end{bmatrix} \quad \mathbf{w}_e = \begin{bmatrix} w_{1A} \\ w_{2A} \\ \beta_A \\ w_{1B} \\ w_{2B} \\ \beta_B \end{bmatrix} \quad \mathbf{d}_e = \begin{bmatrix} d_{1A} \\ d_{2A} \\ \beta_A \\ d_{1B} \\ d_{2B} \\ \beta_B \end{bmatrix} \quad (5.23)$$

The coordinate transformations for the member displacement vectors follow from the coordinate transformations (5.18) to (5.21) for the node displacement vectors:

$$\mathbf{u}_e = \mathbf{R}_{exy} \mathbf{v}_e \quad \mathbf{R}_{exy} = \begin{bmatrix} \mathbf{R}_{nxy} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_{nxy} \end{bmatrix} \quad (5.24)$$

$$\mathbf{u}_e = \mathbf{R}_{exz} \mathbf{w}_e \quad \mathbf{R}_{exz} = \begin{bmatrix} \mathbf{R}_{nxz} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_{nxz} \end{bmatrix} \quad (5.25)$$

$$\mathbf{v}_e = \mathbf{R}_{eyz} \mathbf{w}_e \quad \mathbf{R}_{eyz} = \begin{bmatrix} \mathbf{R}_{nyz} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_{nyz} \end{bmatrix} \quad (5.26)$$

$$\mathbf{u}_e = \mathbf{R}_{exs} \mathbf{d}_e \quad \mathbf{R}_{exs} = \begin{bmatrix} \mathbf{R}_{nxs} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_{nxs} \end{bmatrix} \quad (5.27)$$

- \mathbf{R}_{exy} element rotation matrix global - reference space
- \mathbf{R}_{exz} element rotation matrix global - instant space
- \mathbf{R}_{eyz} element rotation matrix reference - instant space
- \mathbf{R}_{exs} element rotation matrix global - support space

Since the node rotation matrices are orthonormal, the element rotation matrices are also orthonormal:

$$\mathbf{R}_{exy}^T \mathbf{R}_{exy} = \mathbf{R}_{exz}^T \mathbf{R}_{exz} = \mathbf{R}_{eyz}^T \mathbf{R}_{eyz} = \mathbf{R}_{eny}^T \mathbf{R}_{eny} = \mathbf{I} \quad (5.28)$$

Displacement of points on a cross-section: Consider a point P on the axis of the reference configuration of the member in figure 5.3. Its location vector is:

$$\mathbf{y}_P = y_1 \mathbf{j}_1 \quad (5.29)$$

y_1 coordinate of point P in the reference configuration of member AB

\mathbf{j}_1 local base vector in the reference configuration

Let the displacement vector of point P in the instant configuration of the member be referred to the reference space:

$$\mathbf{v}_P = v_1 \mathbf{j}_1 + v_2 \mathbf{j}_2 \quad (5.30)$$

v_1, v_2 displacement coordinates of point P in the reference space of member AB

Consider the cross-section SPT of member AB which passes through point P. Let Q be a point on this cross-section. Its location vector is:

$$\mathbf{y}_Q = y_1 \mathbf{j}_1 + y_2 \mathbf{j}_2 \quad (5.31)$$

y_1, y_2 coordinates of point Q in the reference configuration of member AB

$\mathbf{j}_1, \mathbf{j}_2$ local base vectors in the reference configuration

Assume that cross-section SPT rotates as a rigid body through an angle β in the anti-clockwise direction about the axis through P. The displacement \mathbf{v}_β of Q due to this rotation is given by (3.25) and (3.27):

$$\mathbf{v}_\beta = (\mathbf{R} - \mathbf{I})\mathbf{p} = (\mathbf{A} \sin \beta + \mathbf{A}^2(1 - \cos \beta))\mathbf{p} \quad (5.32)$$

$$\mathbf{a} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad \mathbf{p} = \begin{bmatrix} 0 \\ y_2 \\ 0 \end{bmatrix} \quad \mathbf{A} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{v}_\beta = \begin{bmatrix} -(1 - \cos \beta) & -\sin \beta & 0 \\ \sin \beta & -(1 - \cos \beta) & 0 \\ 0 & 0 & 0 \end{bmatrix} \circ \begin{bmatrix} 0 \\ y_2 \\ 0 \end{bmatrix} = \begin{bmatrix} -y_2 \sin \beta \\ -y_2(1 - \cos \beta) \\ 0 \end{bmatrix}$$

The plane frame hypothesis stipulates that the total displacement of point Q is the sum of the displacement of point P on the axis of the member and the displacement due to the rigid body rotation of the cross-section:

$$\mathbf{v}_Q = \mathbf{v}_P + y_2 \bar{\mathbf{v}}_P \quad (5.33)$$

$$\begin{bmatrix} v_{1Q} \\ v_{2Q} \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} + y_2 \begin{bmatrix} \bar{v}_1 \\ \bar{v}_2 \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} + y_2 \begin{bmatrix} -\sin \beta \\ -(1 - \cos \beta) \end{bmatrix}$$

β angle of rotation of the cross-section

\bar{v}_1, \bar{v}_2 displacement derivatives with respect to y_2 at point P

The instant base vectors at the displaced point \hat{Q} in the instant configuration are denoted by $\mathbf{c}_1, \mathbf{c}_2$ and determined as the derivatives of the location vector $\mathbf{y}_Q + \mathbf{v}_Q$:

$$\mathbf{c}_1 = \frac{\partial(\mathbf{y}_Q + \mathbf{v}_Q)}{\partial y_1} = \frac{\partial(\mathbf{y}_Q + \mathbf{v}_P)}{\partial y_1} + y_2 \frac{\partial \bar{\mathbf{v}}_P}{\partial y_1}$$

$$\mathbf{c}_1 = \begin{bmatrix} 1 + v_{1,1} \\ v_{2,1} \end{bmatrix} + y_2 \frac{\partial}{\partial y_1} \begin{bmatrix} -\sin\beta \\ -(1 - \cos\beta) \end{bmatrix} = \begin{bmatrix} 1 + v_{1,1} \\ v_{2,1} \end{bmatrix} + y_2 \begin{bmatrix} -\cos\beta \\ -\sin\beta \end{bmatrix} \frac{d\beta}{dy_1} \quad (5.34)$$

$$\mathbf{c}_2 = \frac{\partial(\mathbf{y}_Q + \mathbf{v}_Q)}{\partial y_2} = \frac{\partial \mathbf{y}_Q}{\partial y_2} + \bar{\mathbf{v}}_P$$

$$\mathbf{c}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} + \begin{bmatrix} -\sin\beta \\ -(1 - \cos\beta) \end{bmatrix} = \begin{bmatrix} -\sin\beta \\ \cos\beta \end{bmatrix} \quad (5.35)$$

The plane frame hypothesis stipulates that a fibre that was normal to the cross-section in the reference configuration is normal to the rotated cross-section in the instant configuration. Vectors \mathbf{c}_1 and \mathbf{c}_2 must therefore be orthogonal to each other. Expressions (5.34) and (5.35) show that this is the case if the ratio of $1 + v_{1,1}$ to $\cos\beta$ equals the ratio of $v_{2,1}$ to $\sin\beta$:

$$\tan\beta = \frac{v_{2,1}}{1 + v_{1,1}} \quad \sin\beta = \frac{v_{2,1}}{h} \quad \cos\beta = \frac{1 + v_{1,1}}{h} \quad (5.36)$$

$$h = \sqrt{(1 + v_{1,1})^2 + (v_{2,1})^2}$$

The tangent to the axis of the member at point \hat{P} in the instant configuration is determined by differentiation of the displacement on the axis. It follows from (5.34) and (5.36) that base vector \mathbf{c}_1 is tangential to the displaced axis of the member:

$$\mathbf{c}_1 = \begin{bmatrix} \cos\beta \\ \sin\beta \end{bmatrix} (h - y_2 \frac{d\beta}{dy_1}) \quad (5.37)$$

$$\mathbf{c}_2 = \begin{bmatrix} -\sin\beta \\ \cos\beta \end{bmatrix}$$

The derivative of β is determined by differentiating $\sin\beta$ in (5.36):

$$\frac{d}{dy_1}(\sin\beta) = \cos\beta \frac{d\beta}{dy_1} = \frac{d}{dy_1} \left(\frac{v_{2,1}}{h} \right) = \frac{1}{h^2} \left(h v_{2,11} - \frac{v_{2,1}}{2h} (2(1 + v_{1,1})v_{1,11} + 2v_{2,1}v_{2,11}) \right)$$

$$\cos\beta \frac{d\beta}{dy_1} = \frac{v_{2,11}}{h} - \frac{\sin\beta}{h} (v_{1,11} \cos\beta + v_{2,11} \sin\beta) = \frac{\cos\beta}{h} (v_{2,11} \cos\beta - v_{1,11} \sin\beta)$$

$$\frac{d\beta}{dy_1} = \frac{1}{h} (v_{2,11} \cos\beta - v_{1,11} \sin\beta) \quad (5.38)$$

The derivative of β from (5.38) and the trigonometric ratios for β from (5.36) are substituted into (5.37):

$$\mathbf{c}_1 = \left(h + \frac{y_2}{h} (v_{1,11} \sin \beta - v_{2,11} \cos \beta) \right) \frac{\cos \beta}{\sin \beta} \quad (5.39)$$

$$= \left(1 + \frac{y_2}{h^2} (v_{1,11} \sin \beta - v_{2,11} \cos \beta) \right) \frac{1 + v_{1,1}}{v_{2,1}}$$

$$\mathbf{c}_2 = \frac{-\sin \beta}{\cos \beta} = \frac{1}{h} \frac{-v_{2,1}}{1 + v_{1,1}}$$

The derivatives of the displacement of point Q are required for the determination of the state of strain at point Q. They are obtained by differentiation of (5.33):

$$\frac{\partial \mathbf{v}_Q}{\partial y_1} = \frac{v_{1,1}}{v_{2,1}} + y_2 \frac{-\cos \beta}{-\sin \beta} \frac{d\beta}{dy_1} \quad (5.40)$$

$$\frac{\partial \mathbf{v}_Q}{\partial y_2} = \frac{-\sin \beta}{-(1 - \cos \beta)}$$

The derivative of β is substituted from (5.38):

$$\frac{\partial \mathbf{v}_Q}{\partial y_1} = \frac{v_{1,1}}{v_{2,1}} + \frac{y_2}{h} (v_{1,11} \sin \beta - v_{2,11} \cos \beta) \frac{\cos \beta}{\sin \beta} \quad (5.41)$$

$$\frac{\partial \mathbf{v}_Q}{\partial y_2} = \frac{-\sin \beta}{\cos \beta - 1} \quad (5.42)$$

5.1.5 State of Strain

Plane frame hypothesis: The state of strain for a general three-dimensional body is given by the strain tensor (3.38) of Green. The material point for which the strain is computed is identified by its location y_1, y_2 in the reference configuration, and the coordinates of the strain tensor are referred to the reference space. The strain will now be specialised for plane frames.

Due to the symmetry of the plane frame about the plane $y_3 = 0$, displacement v_3 of the points in the plane $y_3 = 0$ is null. Since the plane frame hypothesis stipulates that the cross-sections of the members displace as rigid bodies, the displacements v_3 of all the points of the members are null and the displacements v_1, v_2 do not vary with y_3 . The plane frame hypothesis therefore leads to the following values of the displacement derivatives at an arbitrary point of a member:

$$v_{1,3} = v_{2,3} = v_{3,3} = v_{3,1} = v_{3,2} = 0 \quad (5.43)$$

Substitution of the displacement derivatives (5.43) into expressions (3.38) shows that the following strain coordinates are null:

$$\varepsilon_{13} = \varepsilon_{23} = \varepsilon_{33} = \varepsilon_{32} = \varepsilon_{31} = 0 \quad (5.44)$$

The remaining strains reduce to:

$$e_{11Q} = \{v_{1,1} + \frac{1}{2}(v_{1,1}^2 + v_{2,1}^2)\}_Q \quad (5.45)$$

$$e_{22Q} = \{v_{2,2} + \frac{1}{2}(v_{1,2}^2 + v_{2,2}^2)\}_Q$$

$$e_{12Q} = \frac{1}{2}\{v_{1,2} + v_{2,1} + v_{1,1}v_{1,2} + v_{2,1}v_{2,2}\}_Q$$

The partial derivatives of the displacement coordinates follow from (5.41):

$$(v_{1,1})_Q = v_{1,1} + \frac{y_2 t}{h} \cos \beta \quad (5.46)$$

$$(v_{2,1})_Q = v_{2,1} + \frac{y_2 t}{h} \sin \beta$$

$$(v_{1,2})_Q = -\sin \beta$$

$$(v_{2,2})_Q = \cos \beta - 1$$

$$t := v_{1,1} \sin \beta - v_{2,1} \cos \beta \quad (5.47)$$

Substitution of the partial derivatives from (5.46) and the trigonometric ratios from (5.36) into expressions (5.45) shows that e_{22} and e_{12} are null. The remaining strain e_{11Q} is given by:

$$e_{11Q} = v_{1,1} + \frac{y_2 t}{h} \cos \beta + \frac{1}{2} \left(v_{1,1}^2 + 2 \left(\frac{y_2 t}{h} \right) v_{1,1} \cos \beta + \left(\frac{y_2 t}{h} \right)^2 \cos^2 \beta \right) +$$

$$\frac{1}{2} \left(v_{2,1}^2 + 2 \left(\frac{y_2 t}{t} \right) v_{2,1} \sin \beta + \left(\frac{y_2 t}{h} \right)^2 \sin^2 \beta \right)$$

$$e_{11Q} = v_{1,1} + \frac{1}{2}(v_{1,1}^2 + v_{2,1}^2) + \frac{y_2 t}{h}(\cos \beta + v_{1,1} \cos \beta + v_{2,1} \sin \beta) + \frac{1}{2} \left(\frac{y_2 t}{h} \right)^2$$

Expressions (5.36) are used to replace the partial derivatives $v_{1,1}$ and $v_{2,1}$ in the third term on the right-hand side by trigonometric ratios. This leads to:

$$e_{11Q} = v_{1,1} + \frac{1}{2}(v_{1,1}^2 + v_{2,1}^2) + y_2 t + \frac{1}{2} \left(\frac{y_2 t}{h} \right)^2 \quad (5.48)$$

$v_{k,1}$ first partial derivative of displacement v_k of point P with respect to y_1
 t function (5.47) of the second partial derivatives of v_k with respect to y_1

Expression (5.48) is valid for large displacements, rotations and deformations of the member. Angle β in function t depends on the displacement of the axis of the member, as shown in (5.36), and generally varies along the axis.

Large deformations: Figure 5.4 shows the axis $\hat{A}\hat{S}\hat{T}\hat{B}$ of an instant configuration of a member with reference configuration ASTB, which has undergone large displacements, rotations and deformations. Axis z_1 of the instant coordinate system z_1, z_2 with origin \hat{A} points in the direction of chord $\hat{A}\hat{B}$ of the deformed member.

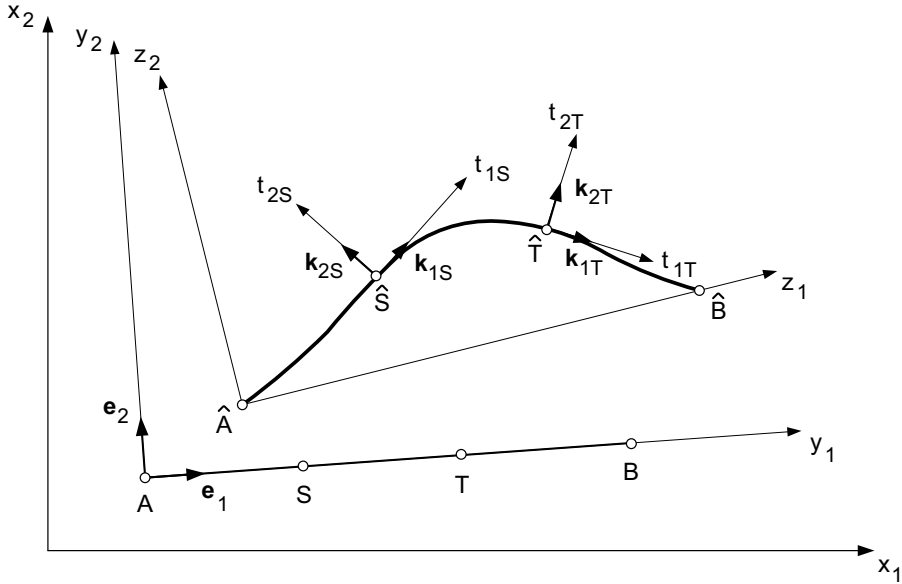


Figure 5.4 Coordinate systems for members with large deformations

At the instant points \hat{S} and \hat{T} , local coordinate systems with axes t_1, t_2 and with orthogonal base vectors $\mathbf{k}_1, \mathbf{k}_2$ are constructed so that \mathbf{k}_{1S} is tangent to the instant axis of the member at \hat{S} , while \mathbf{k}_{1T} is tangent to the instant axis at \hat{T} . The state of displacement of points \hat{S} and \hat{T} in the instant configuration can be described with the following variables whose coordinates are referred to the reference space:

- v_{1S}, v_{2S} displacement coordinates of point S
- β_S angle of rotation from \mathbf{e}_1 to \mathbf{k}_{1S} , positive anti-clockwise
- v_{1T}, v_{2T} displacement coordinates of point T
- β_T angle of rotation from \mathbf{e}_1 to \mathbf{k}_{1T} , positive anti-clockwise

The displacement coordinates v_1 and v_2 in the strain expressions (5.47) and (5.48) refer to the base vectors \mathbf{e}_1 and \mathbf{e}_2 of the reference space. The displacements $v_1\mathbf{e}_1$ and $v_2\mathbf{e}_2$ are not in the direction of the tangent \mathbf{k}_1 and the normal \mathbf{k}_2 of the deformed axis of the member. It is therefore difficult to find suitable interpolation functions for v_1 and v_2 when the finite element theory for the member is developed.

It seems attractive to transform (5.47) and (5.48) into the local coordinate spaces at points \hat{S} and \hat{T} , since the strain will then be expressed in terms of the tangential and the normal components of the displacement. Consider the displacement coordinates r_{1S}, r_{2S} of point S in the local space $\mathbf{k}_{1S}, \mathbf{k}_{2S}$ and r_{1T}, r_{2T} of point T in the local space $\mathbf{k}_{1T}, \mathbf{k}_{2T}$. These coordinates are related to the displacement coordinates of the points in the reference space in analogy to transformation (5.20):

$$\begin{aligned} r_{1S} &= v_{1S} \cos \beta_S + v_{2S} \sin \beta_S & r_{1T} &= v_{1T} \cos \beta_T + v_{2T} \sin \beta_T \\ r_{2S} &= -v_{1S} \sin \beta_S + v_{2S} \cos \beta_S & r_{2T} &= -v_{1T} \sin \beta_T + v_{2T} \cos \beta_T \end{aligned} \quad (5.49)$$

r_1, r_2 displacement coordinates in the local space with base $\mathbf{k}_1, \mathbf{k}_2$

The interpolation is simplified if the derivatives of v_1, v_2 and β with respect to y_1 are replaced by derivatives of r_1, r_2 and β with respect to t_1 . In the case of rotations β_S and β_T this is possible, since the rotations refer to the base vectors \mathbf{k}_{3S} and \mathbf{k}_{3T} , which are parallel. The displacement coordinates r_{1S} and r_{1T} , however, refer to base vectors \mathbf{k}_{1S} and \mathbf{k}_{1T} , which are not parallel. Similarly, the displacement coordinates r_{2S} and r_{2T} refer to base vectors that are not parallel. Therefore, the derivatives of the tangential displacement r_1 and the normal displacement r_2 with respect to t_1 cannot be determined by partial differentiation. Instead, the covariant partial derivatives of tensor calculus must be used, which contain the Christoffel symbols to account for the change in the base vectors [44].

The tensor calculus approach is suitable if large deformations are taken into account in the formulation of the governing equations for the individual member. In this book, however, large deformations are not taken into account in the formulation for the individual member. Instead, members subject to large deformations are subdivided into finite elements that are sufficiently small to permit a stepwise solution with a small deformation theory.

It should be noted that equations (5.49) cannot be used to replace the expression $t = v_{1,11} \sin\beta - v_{2,11} \cos\beta$ by the term $-r_{2,11}$ because β is a function of y_1 .

Displacement in instant space: In order to prepare for the derivation of a theory of small deformations and the subdivision of members of the frame into finite elements, derivatives $v_{1,11}$ and $v_{2,11}$ of the displacement coordinates in the reference space are replaced by derivatives $w_{1,11}$ and $w_{2,11}$ of the displacement coordinates in the instant space. The material points on the axis are identified by their reference coordinate y_1 , so that the differentiation in $w_{1,11}$ and $w_{2,11}$ is performed with respect to y_1 .

Consider the member AB in figure 5.5 that is displaced to $\hat{A}\hat{B}$ in an instant configuration. The line connecting the displaced nodes \hat{A} and \hat{B} is called the chord of the member in the instant configuration. Let the angle θ from axis y_1 to axis z_1 be

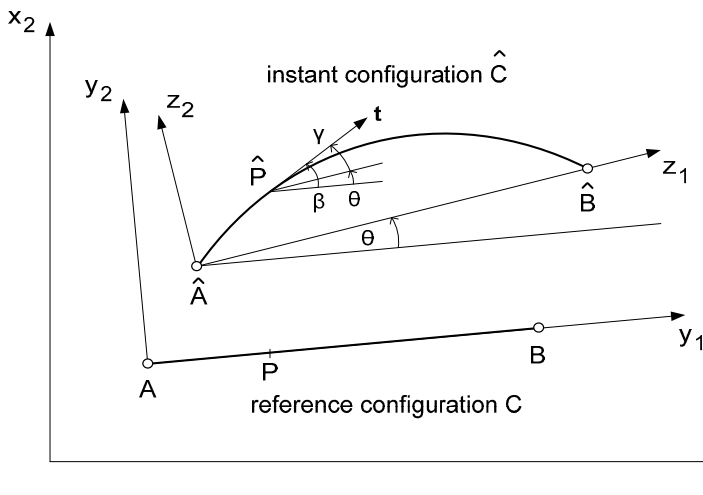


Figure 5.5 Relative angle $\gamma = \beta - \theta$ between chord and tangent of a member

positive if it is measured in the anti-clockwise direction. The displacement transformation (5.12) is used to replace coordinates v_1, v_2 in the reference space by coordinates w_1, w_2 in the instant space:

$$\mathbf{v} = \mathbf{R}_{yz} \mathbf{w} \quad (5.50)$$

$$\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$$

θ angle from axis y_1 to axis z_1

Angle θ is independent of coordinate y_1 . Denote the angle from the instant chord $\hat{A}\hat{B}$ to the tangent \mathbf{t} at point \hat{P} of the axis in the instant configuration by γ . Function t in (5.47) is expressed in terms of the angle γ :

$$\begin{aligned} t &= \frac{d^2}{dy_1^2} (w_1 \cos \theta - w_2 \sin \theta) \sin \beta - \frac{d^2}{dy_1^2} (w_1 \sin \theta + w_2 \cos \theta) \cos \beta \\ &= \frac{d^2 w_1}{dy_1^2} (\sin \beta \cos \theta - \cos \beta \sin \theta) - \frac{d^2 w_2}{dy_1^2} (\sin \beta \sin \theta + \cos \beta \cos \theta) \\ t &= w_{1,11} \sin \gamma - w_{2,11} \cos \gamma \end{aligned} \quad (5.51)$$

$$\gamma := \beta - \theta$$

γ angle of rotation of the axis relative to the chord in the instant configuration

Since factor h in (5.36) is of order 1.0, the last term in (5.48) is of order $O[y_2 t]^2$. Term t is of the order of magnitude of the second derivative $w_{2,11}$. The absolute value of $w_{2,11}$ in linear beam theory equals the applied bending moment M divided by the bending stiffness EJ . Thus $O[y_2 t]$ equals the ratio of the bending stress to the modulus of elasticity, which for steel is of the order of 10^{-3} .

$$O[y_2 h] = O[y_2 w_{2,11}] = O\left[\frac{y_2 M}{EJ}\right] = O\left[\frac{\sigma_{\text{bending}}}{E}\right] = O[10^{-3}] \quad (5.52)$$

The last term in (5.48), which is of the order $O[y_2 h]^2$, can therefore be neglected relative to the second last term, which is of the order $O[y_2 h]$. Strain ε follows from (5.48) and (5.51):

$$\varepsilon = v_{1,1} + \frac{1}{2}(v_{1,1}^2 + v_{2,1}^2) + y_2 (w_{1,11} \sin \gamma - w_{2,11} \cos \gamma) \quad (5.53)$$

ε longitudinal strain in the member
 v_1, v_2 displacement coordinates in reference space
 $v_{k,1}$ 1. derivative of v_k with respect to y_1
 w_1, w_2 displacement coordinates in instant space
 $w_{k,11}$ 2. derivative of w_k with respect to y_1
 γ rotation of the axis relative to the chord

The longitudinal strain ε is decomposed into an axial strain ε_a and a bending strain derivative ε_b :

$$\varepsilon = \varepsilon_a + y_2 \varepsilon_b \quad (5.54)$$

$$\varepsilon_a = v_{1,1} + \frac{1}{2}(v_{1,1}^2 + v_{2,1}^2) \quad (5.55)$$

$$\varepsilon_b = w_{1,11} \sin \gamma - w_{2,11} \cos \gamma \quad (5.56)$$

ε_a axial strain
 ε_b bending strain derivative with respect to y_2

5.1.6 Small Deformation Theory

Small deformations: The displacement of the axis of a member of the frame will now be formulated in such a way that it becomes possible to define small deformations of the member. Consider member APB of the reference configuration of the frame in figure 5.5, which is displaced to $\hat{A}\hat{P}\hat{B}$ in an instant configuration of the frame. The displacement of the axis of the member is decomposed into two components:

(a) **Chord displacement:** Chord displacement w_c is a linear function of the distance z_1 on the chord of the member in the instant configuration. It moves the physical points at nodes A and B of the reference configuration to nodes \hat{A} and \hat{B} in the instant configuration. Intermediate points on the axis of the reference configuration, such as P, are moved to the chord of the member in the instant configuration.

$$w_c = w_A + \frac{z_1}{\hat{a}}(w_B - w_A) \quad (5.57)$$

$$\begin{bmatrix} w_{c1} \\ w_{c2} \end{bmatrix} = \begin{bmatrix} w_{1A} \\ w_{2A} \end{bmatrix} + \frac{z_1}{\hat{a}} \begin{bmatrix} w_{1B} - w_{1A} \\ w_{2B} - w_{2A} \end{bmatrix} = \begin{bmatrix} w_{1A} \\ w_{2A} \end{bmatrix} + \frac{z_1}{\hat{a}} \begin{bmatrix} \Delta w_1 \\ \Delta w_2 \end{bmatrix}$$

$w_{ci}(z_1)$ chord displacement coordinate of point z_1 in instant space
 w_{iA} displacement coordinate of node A in instant space
 w_{iB} displacement coordinate of node B in instant space
 \hat{a} length of the member in the instant configuration

(b) **Relative displacement:** The relative displacement w_r is a nonlinear function of the distance z_1 on the chord of the member. The relative translation of the physical points at nodes \hat{A} and \hat{B} is null, but the relative angle of rotation from the chord to the tangent of the axis at \hat{A} and \hat{B} generally differs from null. The relative displacement moves the physical points from chord $\hat{A}\hat{B}$ to the deformed axis of the member in the instant configuration.

$$w_r = w_r(z_1) \quad (5.58)$$

$$\begin{bmatrix} w_{r1} \\ w_{r2} \\ \gamma \end{bmatrix} = \begin{bmatrix} w_{r1}(z_1) \\ w_{r2}(z_1) \\ \gamma(z_1) \end{bmatrix}$$

$w_{ri}(z_1)$ relative displacement of point z_1 in instant space
 γ rotation of the tangent to the axis at point z_1 relative to the chord

The deformation of a member of a frame is considered to be small if the following approximations can be made:

(a) The chord displacements (5.57) in figure 5.6 are a linear function of the distance z_1 on the chord:

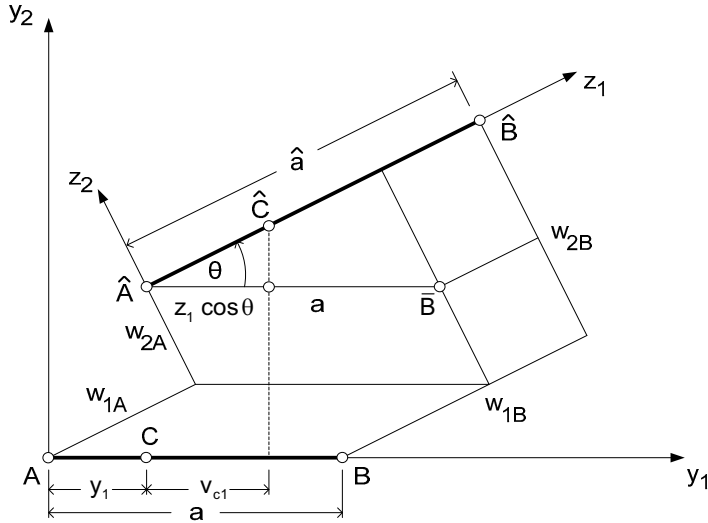


Figure 5.6 Chord displacement of a member

Consider a point $C = (y_1, 0)$ on the axis AB of the reference configuration of the member and refer all coordinates to the reference space. The y_1 coordinate of the displaced point \hat{C} , which is determined by adding chord displacement v_{c1} to the location y_1 of point C , equals the y_1 coordinate that is determined by adding the y_1 coordinate of the displaced node \hat{A} and the y_1 component of vector $\hat{A}\hat{C}$:

$$y_1 + v_{c1} = v_{1A} + z_1 \cos \theta$$

The coordinates of the chord displacement are referred to instant space with (5.50):

$$y_1 + w_{c1} \cos \theta - w_{c2} \sin \theta = v_{1A} + z_1 \cos \theta$$

The equation is differentiated with respect to z_1 and the coordinates of the chord displacement are substituted from (5.57):

$$\frac{dy_1}{dz_1} + \frac{\Delta w_1}{\hat{a}} \cos \theta - \frac{\Delta w_2}{\hat{a}} \sin \theta = \cos \theta \quad (5.59)$$

$$\frac{dw_{c1}}{dz_1} = \frac{w_{1B} - w_{1A}}{\hat{a}} := \frac{\Delta w_1}{\hat{a}}$$

$$\frac{dw_{c2}}{dz_1} = \frac{w_{2B} - w_{2A}}{\hat{a}} := \frac{\Delta w_2}{\hat{a}}$$

Figure 5.6 shows that the trigonometric ratios are given by:

$$\cos \theta = \frac{\hat{a} - \Delta w_1}{a} \quad \sin \theta = \frac{\Delta w_2}{a} \quad (5.60)$$

Substitution of (5.60) into (5.59) yields the derivative of y_1 with respect to z_1 :

$$\frac{dy_1}{dz_1} = \frac{a}{\hat{a}} \quad (5.61)$$

Approximation (5.61) makes it possible to express the coordinates y_1 and z_1 in terms of a common normalised coordinate z :

$$\begin{aligned} y_1 &= \frac{1}{2}(1+z) a & dy_1 &= \frac{a}{2} dz & -1 \leq z \leq 1 & \quad (5.62) \\ z_1 &= \frac{1}{2}(1+z) \hat{a} & dz_1 &= \frac{\hat{a}}{2} dz \end{aligned}$$

z normalised axial coordinate of the bar
 a length of the member in the reference configuration
 \hat{a} length of the member in the instant configuration

(b) The influence of the relative displacement on the axial strain is neglected, so that the axial strain is due only to the chord displacement. Consider a point $C = (y_1, 0)$ of the reference configuration of a member undergoing a chord displacement. Let the location of the displaced point in the instant space be $(z_1, 0)$. The displacement coordinates of the point in the reference space are:

$$\begin{aligned} v_{c1} &= v_{1A} + z_1 \cos \theta - y_1 \\ v_{c2} &= v_{2A} + z_1 \sin \theta \end{aligned}$$

The partial derivatives of the chord displacement with respect to y_1 follow from these equations by differentiation and substitution of (5.61):

$$\begin{aligned} \frac{dv_{c1}}{dy_1} &:= v_{c1,1} = \frac{\hat{a}}{a} \cos \theta - 1 \\ \frac{dv_{c2}}{dy_1} &:= v_{c2,1} = \frac{\hat{a}}{a} \sin \theta \end{aligned}$$

The axial strain for the strain tensor of Green is given by (5.55):

$$\varepsilon_a = v_{c1,1} + \frac{1}{2}v_{c1,1}^2 + \frac{1}{2}v_{c2,1}^2 \quad (5.63)$$

Consider the coordinates of the chord displacement referred to the instant coordinate system of the member. Their derivatives with respect to y_1 and the instant length of the member are given by:

$$\begin{aligned} \frac{dw_{c1}}{dy_1} &:= w_{c1,1} = \frac{w_{1B} - w_{1A}}{a} \\ \frac{dw_{c2}}{dy_1} &:= w_{c2,1} = \frac{w_{2B} - w_{2A}}{a} \end{aligned} \quad (5.64)$$

w_{c1} coordinate of the chord displacement on axis z_1
 w_{c2} coordinate of the chord displacement on axis z_2

The displacement coordinates v_{1c}, v_{2c} and w_{1c}, w_{2c} are related by (5.15):

$$\begin{aligned} v_{c1} &= w_{c1} \cos \theta - w_{c2} \sin \theta \\ v_{c2} &= w_{c1} \sin \theta + w_{c2} \cos \theta \end{aligned}$$

For small deformations, the ratio of the instant to the reference length of the member is approximately 1.0, so that (5.60) and (5.64) yield the following trigonometric ratios:

$$\cos \theta = 1 - w_{c1,1} \quad \sin \theta = w_{c2,1}$$

The derivatives of v_{c1} and v_{c2} with respect to y_1 are expressed as functions of the derivatives of w_{c1} and w_{c2} with respect to y_1 and substituted into (5.63) to obtain the axial strain in terms of the displacement coordinates in instant space:

$$\begin{aligned} v_{c1,1} &= w_{c1,1} \cos \theta - w_{c2,1} \sin \theta = w_{c1,1} - w_{c1,1}^2 - w_{c2,1}^2 \\ v_{c1,1}^2 + v_{c2,1}^2 &= w_{c1,1}^2 + w_{c2,1}^2 \\ \varepsilon_a &= w_{c1,1} - \frac{1}{2} w_{c1,1}^2 - \frac{1}{2} w_{c2,1}^2 \end{aligned} \quad (5.65)$$

(c) Strain derivative ε_b in (5.56) does not depend on the chord displacement since the chord displacement is linear in y_1 . The second derivative $w_{r1,11}$ of the relative displacement coordinate w_{r1} is assumed to be negligible compared to $w_{r2,11}$. It is also assumed that displacement w_{2r} is so small compared to the length of the member that the relative rotation $\gamma \ll 1.0$. The trigonometric ratio $\cos \gamma$ and $\sin \gamma$ are therefore approximated by 1.0 and γ . The bending strain derivative ε_b in expression (5.56) reduces to:

$$\varepsilon_b = -w_{r2,11} \quad (5.66)$$

For a member that satisfies assumptions (a) to (c) for small deformations, the strain expressions (5.54) to (5.56) are replaced by:

$$\varepsilon = \varepsilon_a + y_2 \varepsilon_b \quad (5.67)$$

$$\varepsilon_a = w_{c1,1} - \frac{1}{2} w_{c1,1}^2 - \frac{1}{2} w_{c2,1}^2 \quad (5.68)$$

$$\varepsilon_b = -w_{r2,11} \quad (5.69)$$

- $w_{c1,1}$ 1. derivative of the chord displacement w_{c1} with respect to y_1
- $w_{c2,1}$ 1. derivative of the chord displacement w_{c2} with respect to y_1
- $w_{r2,11}$ 2. derivative of the relative displacement w_{r2} with respect to y_1

Finite elements: For a given loading condition, the members of a plane frame may not satisfy conditions (a) to (c) of the small deformation theory. Figure 5.4 shows an example of such a member.

If a frame contains members that do not satisfy the conditions of the small deformation theory for a given loading condition, these members are subdivided into a suitable number of finite elements by defining additional nodes S_1, S_2, \dots , as shown in figure 5.7. Each finite element is a member of the frame and has its own instant coordinate space z_1, z_2 .

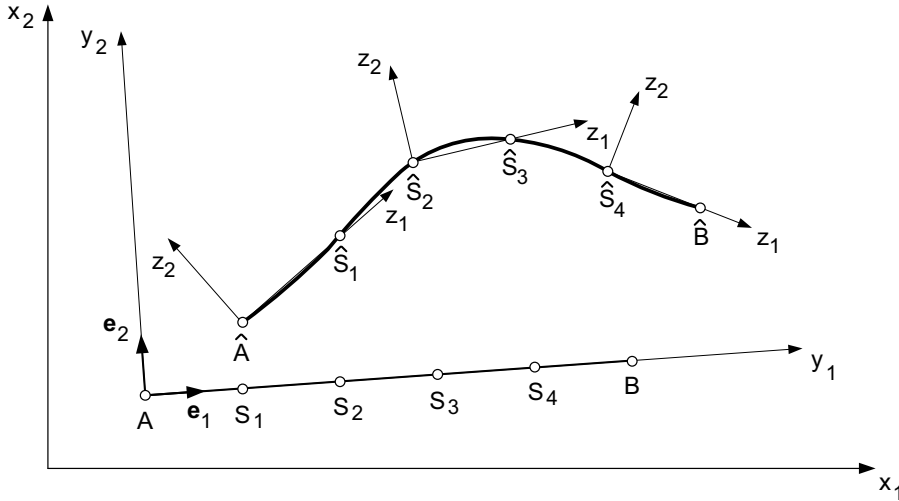


Figure 5.7 Subdivision of a frame member into finite elements

Figure 5.7 shows that the subdivision of member AB into finite elements reduces the magnitude of the relative displacements and rotations significantly. The number of finite elements can be chosen so that the small deformation theory can be applied to each member. This subdivision can be automated or left to the discretion of the user.

The subdivision of a member into finite elements does not affect the order of magnitude of the total displacements and rotations to which the member is subjected. The elements in Figure 5.7 are subject to small deformations, but large displacements and rotations.

The subdivision of a member into finite elements permits the modelling of tension stiffening even though the elements undergo small deformations. The chord displacement of the nodes S_1, S_2, \dots leads to stretching of the elements.

5.1.7 Stress Resultants

The strain ε in the member is given by expression (5.67). The product of strain ε and the modulus of elasticity E yields the axial 2. Piola-Kirchhoff stress s , which is the coefficient of base vector \mathbf{c}_1 of the instant configuration:

$$\mathbf{s} = s\mathbf{c}_1 = E\varepsilon\mathbf{c}_1 = E\mathbf{c}_1(\varepsilon_a + y_2\varepsilon_b) \quad (5.70)$$

- \mathbf{s} axial stress vector per unit area of the cross-section
- \mathbf{c}_1 instant base vector, see (5.39)
- s longitudinal 2. Piola-Kirchhoff stress
- ε_a axial strain, see (5.65)
- ε_b longitudinal strain derivative, see (5.66)
- E modulus of elasticity

The instant base vector \mathbf{c}_1 is tangential to the chord of the member in the instant configuration. The term containing $t = v_{1,11} \sin \beta - v_{2,11} \cos \beta = w_{1,11} \sin \gamma - w_{2,11} \cos \gamma$ in (5.39) is neglected due to (5.52), so that the expression for the base vector is:

$$\mathbf{c}_1 = \begin{bmatrix} 1 + v_{c1,1} \\ v_{c2,1} \end{bmatrix}$$

$$|\mathbf{c}_1| = \sqrt{(1 + v_{c1,1})^2 + (v_{c2,1})^2} \quad (5.71)$$

The physical stress σ acting on the cross-section of the member is referred to a vector of unit length in the direction of \mathbf{c}_1 . It is decomposed into an axial stress σ_a and a bending stress derivative σ_b :

$$\sigma = |\mathbf{c}_1|s = \sigma_a + y_2 \sigma_b \quad (5.72)$$

$$\sigma_a = E \varepsilon_a |\mathbf{c}_1|$$

$$\sigma_b = E \varepsilon_b |\mathbf{c}_1|$$

σ_a physical axial stress

σ_b physical longitudinal stress derivative

Stress σ_a leads to an axial force n that acts at the centroid of the cross-section in the direction of the tangent to the axis in the instant configuration:

$$n = \int_A (\sigma_a + y_2 \sigma_b) da = A \sigma_a \quad (5.73)$$

n axial force acting on the cross-section (tension positive)

A area of the cross-section

Stress $y_2 \sigma_b$ leads to a bending moment m that acts about an axis through the centroid of the cross-section normal to the plane of the frame:

$$m = \int_A (\sigma_a + y_2 \sigma_b) y_2 da = J \sigma_b \quad (5.74)$$

m bending moment acting on the cross-section

J moment of inertia of the cross-section

A positive bending moment causes tension in the fibres with positive coordinate y_2 .

5.2 ANALYTICAL GOVERNING EQUATIONS FOR PLANE FRAMES

5.2.1 System Vectors

The two coordinates of the displacement and the rotation about the axis normal to the plane of the frame are called the generalised displacements at a node. The two coordinates of the force acting on the node and the moment acting about the axis normal to the plane of the frame are called the generalised forces at the node. The generalised variables are collected in system vectors. The analytical governing equations for a plane frame are formulated in terms of these system vectors.

$$\mathbf{d}_s = \begin{bmatrix} \dots \\ x \\ x \\ \dots \\ x \\ \dots \end{bmatrix} \quad \mathbf{q}_s = \begin{bmatrix} x \\ x \\ x \\ x \\ x \\ x \end{bmatrix} \quad \mathbf{r}_s = \begin{bmatrix} \circ \\ \dots \\ \dots \\ \circ \\ \dots \\ \circ \end{bmatrix} \quad \mathbf{s}_s = \begin{bmatrix} f \\ t \\ t \\ f \\ t \\ f \end{bmatrix} \quad (5.75)$$

... computed value
 x prescribed value
 ◦ unused coefficient

t : prescribed displacement
 f : free displacement

\mathbf{d}_s system displacement vector
 \mathbf{q}_s system load vector
 \mathbf{r}_s system reaction vector
 \mathbf{s}_s system status vector

The nodes of the frame are numbered consecutively. The node vectors are entered into the system vectors in the order of the node numbers. The coordinates of the displacement and force vectors at a node are referred to the support space. In each coordinate direction at each node of the truss, either the coordinate of the generalised displacement of the node, or the coordinate of the generalised force acting on the node is prescribed. The values that are not prescribed are determined in the analysis of the frame.

A generalised displacement coordinate at a node can be prescribed or free. The prescribed and the free displacement coordinates are stored in a common system vector \mathbf{d}_s . The distinction between prescribed and free displacements is made by means of a system status vector \mathbf{s}_s , whose dimension equals that of the system displacement vector. The value of the status coefficient is “true” if the displacement or rotation is prescribed.

A generalised force coordinate at a node can be prescribed or free. A prescribed force coordinate is called a load. The loads are collected in a system load vector \mathbf{q}_s . The dimension of the system load vector equals that of the system displacement vector.

A free generalised force coordinate at a node is called a reaction. The reactions are collected in a system reaction vector \mathbf{r}_s . The dimension of the system reaction vector equals that of the system displacement vector. Rows of the vector that do not correspond to reactions are unused and assumed to contain null. A row contains a reaction if the corresponding row of the status vector contains the value "true".

The generalised displacement coordinates of a member are collected in the element displacement vector \mathbf{w}_e defined in (5.23). In order to relate the element displacement vector \mathbf{w}_e to the system displacement vector \mathbf{d}_s , vector \mathbf{w}_e is related to the element displacement vector \mathbf{d}_e . Element vector \mathbf{d}_e in turn is related to the system displacement vector \mathbf{d}_s with the element topology matrix \mathbf{T}_e .

$$\mathbf{w}_e = \mathbf{R}_{esz}^T \mathbf{d}_e = \mathbf{R}_{esz}^T \mathbf{T}_e \mathbf{d}_s \quad (5.76)$$

$$\mathbf{R}_{esz} = \mathbf{R}_{exs}^T \mathbf{R}_{exz}$$

- \mathbf{w}_e displacement vector of member e referred to instant space
- \mathbf{d}_s system displacement vector referred to node support spaces
- \mathbf{R}_{esz} element rotation matrix support - instant space
- \mathbf{T}_e topology matrix of member e

If the generalised displacement with index k in the element vector \mathbf{w}_e corresponds to the generalised displacement with index m in the system vector \mathbf{d}_s , the coefficient $\mathbf{T}_e(k,m)$ equals 1. All other coefficients in row k of \mathbf{T}_e are null.

5.2.2 Governing Equations for a State of the Frame

The weak form (3.108) of the governing equations for the nonlinear behaviour of elastic bodies is now specialised for plane frames. Each integral over the volume of the body is replaced by the sum of the integrals over the volumes of the members. Since only the longitudinal strain ε and the longitudinal stress s are different from null, the scalar product $\delta \boldsymbol{\varepsilon}^T \boldsymbol{\sigma}$ is replaced by the product of the variation $\delta \varepsilon$ of the longitudinal strain with the longitudinal stress s . The volume load $\rho \mathbf{q}$ is replaced by a load \mathbf{p}_a per unit length of the member axis in the reference configuration. The surface loads and reactions are replaced by loads and reactions at the joints of the frame. Equation (3.145) is used to replace the surface loads and reactions by loads and reactions at the nodes of the frame. With these modifications, equation (3.108) takes the following form:

$$\sum_e \int_{C_e} \delta \varepsilon s dv = \sum_e \int_{L_e} \delta \mathbf{v}^T \mathbf{p}_a dy_1 + \delta \mathbf{q}_s^T (\mathbf{q}_s + \mathbf{r}_s) \quad (5.77)$$

- C_e volume of member e in the reference configuration
- ε longitudinal strain in the member (strain tensor of Green)
- s longitudinal stress in the member (2. Piola-Kirchhoff stress tensor)
- \mathbf{d}_s system displacement vector (coordinates in support space)
- \mathbf{q}_s system load vector (coordinates in support space)
- \mathbf{r}_s system reaction vector (coordinates in support space)

Equation (5.77) is not solved directly. The states of the frame are determined by solving an initial value problem, starting with the reference configuration and an

increment of the applied load. In each step of the solution procedure, incremental governing equations for the change in the displacements and forces are solved and the values of the variables at the end of the previous step are incremented with the increments for the current step. This solution procedure is approximate because of linearisation. Equation (5.77) is therefore not satisfied exactly. The unbalanced forces in (5.77) are computed as an error vector and compensated by means of iteration in the load step. The incremental governing equations are formulated in the remaining sections of this chapter. The solution procedure is treated in chapter 6.

5.2.3 Increments of the State Variables

Displacement increment: Figure 5.8 shows a member AB of a frame in the reference configuration, the instant configuration s and the instant configuration $s+1$. The coordinate systems in the different configurations are denoted as follows:

- y_1, y_2 reference coordinate system of member AB
- z_1, z_2 instant coordinate system for configuration s with nodes \hat{A}, \hat{B}
- t_1, t_2 instant coordinate system for configuration $s+1$ with nodes \bar{A}, \bar{B}

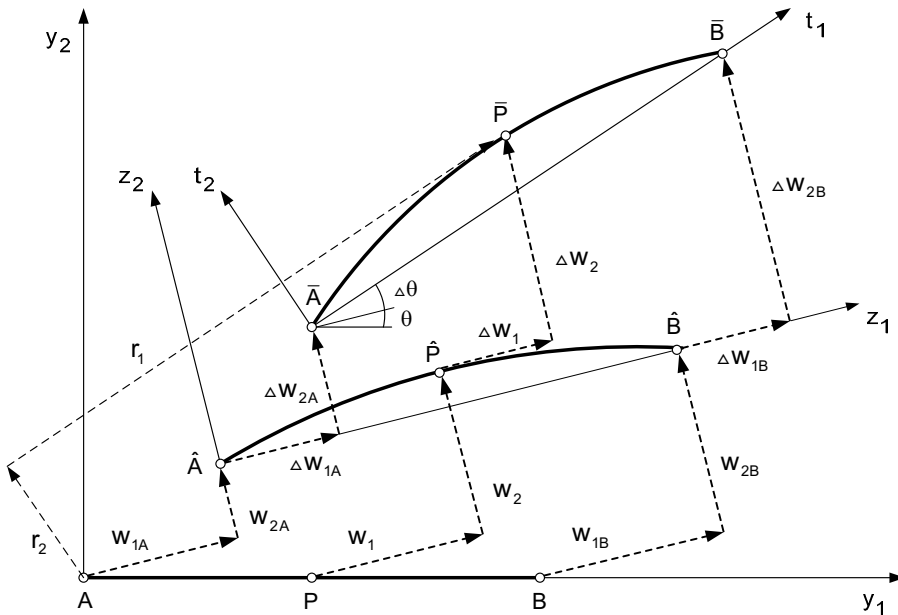


Figure 5.8 Displacement increments of points on the axis of a member

If the coordinates of the displacement of a point P on the axis of member AB are referred to the instant coordinate system y_1, y_2 of that configuration, they are denoted as follows:

- w_1, w_2 displacement in configuration s
- $\Delta w_1, \Delta w_2$ displacement increment in step s
- $w_1 + \Delta w_1, w_2 + \Delta w_2$ displacement in configuration $s+1$

For the computation of the stretching and bending strain in configuration $s + 1$, the coordinates of the displacement must be referred to the coordinate system t_1, t_2 of that configuration. The angle from axis z_1 to axis t_1 is given by:

$$\Delta\theta \cong \frac{\Delta W_{2B} - \Delta W_{2A}}{a} \quad (5.78)$$

$$\sin \Delta\theta \cong \Delta\theta$$

$$\cos \Delta\theta \cong 1 - \frac{1}{2} \Delta\theta^2$$

The coordinates of the displacement of point P referred to the coordinate system t_1, t_2 follow from (5.15):

$$r_1 = (w_1 + \Delta w_1) \cos \Delta\theta + (w_2 + \Delta w_2) \sin \Delta\theta \quad (5.79)$$

$$r_2 = -(w_1 + \Delta w_1) \sin \Delta\theta + (w_2 + \Delta w_2) \cos \Delta\theta$$

r_1, r_2 displacement of P in configuration $s + 1$, referred to coordinate space t_1, t_2

Approximation (5.78) for the trigonometric ratios is substituted into (5.79):

$$r_1 = (w_1 + \Delta w_1) \left(1 - \frac{1}{2} \Delta\theta^2\right) + (w_2 + \Delta w_2) \Delta\theta \quad (5.80)$$

$$r_2 = -(w_1 + \Delta w_1) \Delta\theta + (w_2 + \Delta w_2) \left(1 - \frac{1}{2} \Delta\theta^2\right)$$

Strain increment: The longitudinal strain ε at point (y_1, y_2) of a member in configuration s is given by (5.67) to (5.69):

$$\varepsilon = \varepsilon_a + y_2 \varepsilon_b \quad (5.81)$$

$$\varepsilon_a = w_{c1,1} - \frac{1}{2} w_{c1,1}^2 - \frac{1}{2} w_{c2,1}^2 \quad (5.82)$$

$$\varepsilon_b = -w_{r2,1} \quad (5.83)$$

The axial strain at point y_1 in configuration $s + 1$ is determined with (5.82), using the coordinates of the displacement referred to the instant space of that configuration:

$$\varepsilon_a + \Delta\varepsilon_a = r_{c1,1} - \frac{1}{2} r_{c1,1}^2 - \frac{1}{2} r_{c2,1}^2 \quad (5.84)$$

$\Delta\varepsilon_a$ increment of the axial strain in step s

The derivatives $r_{c1,1}$ and $r_{c2,1}$ correspond to (5.64) and are derived with (5.80):

$$r_{c1,1} = \frac{r_{1B} - r_{1A}}{a} = \left(\frac{w_{1B} - w_{1A}}{a} + \frac{\Delta w_{1B} - \Delta w_{1A}}{a} \right) \left(1 - \frac{1}{2} \Delta\theta^2 \right) + \left(\frac{w_{2B} - w_{2A}}{a} + \frac{\Delta w_{2B} - \Delta w_{2A}}{a} \right) \Delta\theta \quad (5.85)$$

$$r_{c2,1} = \frac{r_{2B} - r_{2A}}{a} = \left(\frac{w_{2B} - w_{2A}}{a} + \frac{\Delta w_{2B} - \Delta w_{2A}}{a} \right) \left(1 - \frac{1}{2} \Delta\theta^2 \right) - \left(\frac{w_{1B} - w_{1A}}{a} + \frac{\Delta w_{1B} - \Delta w_{1A}}{a} \right) \Delta\theta \quad (5.86)$$

Equation (5.64) shows that the increment $\Delta\theta$ of the chord angle in equation (5.78) equals the increment $\Delta w_{c2,1}$ of the chord displacement derivative:

$$\Delta\theta = \Delta w_{c2,1} \quad (5.87)$$

Expression (5.87) is substituted into (5.85) and (5.86):

$$r_{c1,1} = (w_{c1,1} + \Delta w_{c1,1})\left(1 - \frac{1}{2}\Delta w_{c2,1}^2\right) + w_{c2,1}\Delta w_{c2,1} + \Delta w_{c2,1}^2 \quad (5.88)$$

$$r_{c2,1} = (w_{c2,1} + \Delta w_{c2,1})\left(1 - \frac{1}{2}\Delta w_{c2,1}^2\right) - w_{c1,1}\Delta w_{c2,1} - \Delta w_{c1,1}\Delta w_{c2,1} \quad (5.89)$$

The increment of the axial strain follows from (5.82), (5.84), (5.88) and (5.89):

$$\begin{aligned} \Delta\varepsilon_a = & (w_{c1,1} + \Delta w_{c1,1})\left(1 - \frac{1}{2}\Delta w_{c2,1}^2\right) + w_{c2,1}\Delta w_{c2,1} + \Delta w_{c2,1}^2 - w_{c1,1} - \\ & \frac{1}{2}\left((w_{c1,1} + \Delta w_{c1,1})\left(1 - \frac{1}{2}\Delta w_{c2,1}^2\right) + w_{c2,1}\Delta w_{c2,1} + \Delta w_{c2,1}^2\right)^2 + \frac{1}{2}w_{c1,1}^2 - \\ & \frac{1}{2}\left((w_{c2,1} + \Delta w_{c2,1})\left(1 - \frac{1}{2}\Delta w_{c2,1}^2\right) - w_{c1,1}\Delta w_{c2,1} - \Delta w_{c1,1}\Delta w_{c2,1}\right)^2 + \frac{1}{2}w_{c2,1}^2 \end{aligned} \quad (5.90)$$

The axial strain increment is split into a term $\Delta\varepsilon_a^0$ that is linear in the displacement increments, and a term $\Delta\varepsilon_a^n$ that is quadratic in the displacement increments. Higher powers of the displacement increments are neglected.

$$\Delta\varepsilon_a = \Delta\varepsilon_a^0 + \Delta\varepsilon_a^n \quad (5.91)$$

$$\Delta\varepsilon_a^0 = (1 - w_{c1,1})\Delta w_{c1,1} \quad (5.92)$$

$$\Delta\varepsilon_a^n = -\frac{1}{2}\Delta w_{c1,1}^2 + \frac{1}{2}(1 - w_{c1,1})\Delta w_{c2,1}^2 \quad (5.93)$$

The bending strain derivative in configuration $s+1$ is determined with (5.80) and (5.83) for the coordinates of the displacement referred to the instant space of that configuration:

$$\varepsilon_b + \Delta\varepsilon_b = -\frac{d^2}{dy_1^2}\left(- (w_1 + \Delta w_1)\Delta\theta + (w_2 + \Delta w_2)\left(1 - \frac{1}{2}\Delta\theta^2\right)\right) \quad (5.94)$$

$\Delta\varepsilon_b$ increment of the bending strain derivative in step s

The increment $\Delta\varepsilon_b$ is split into a term that is linear and a term that is quadratic in the displacement increments. Higher powers of the displacement increments and derivatives of order higher than 2 are neglected.

$$\Delta\varepsilon_b = \Delta\varepsilon_b^0 + \Delta\varepsilon_b^n \quad (5.95)$$

$\Delta\varepsilon_b^0$ linear component of the bending strain derivative

$\Delta\varepsilon_b^n$ quadratic component of the bending strain derivative

The increment $\Delta\theta$ is substituted from (5.87) into (5.94):

$$\Delta\varepsilon_b = -\frac{d^2}{dy_1^2} \left(-(w_1 + \Delta w_1) \Delta w_{c2,1} + (w_2 + \Delta w_2) \left(1 - \frac{1}{2} \Delta w_{c2,1}^2\right) \right) + w_{2,11}$$

$$\Delta\varepsilon_b^0 = -\Delta w_{2,11} + w_{1,11} \Delta w_{c2,1} \quad (5.96)$$

$$\Delta\varepsilon_b^n = \frac{1}{2} w_{2,11} \Delta w_{c2,1}^2 + \Delta w_{1,11} \Delta w_{c2,1} \quad (5.97)$$

For small deformations, the second derivative $w_{1,11}$ of the axial displacement is negligible compared to the second derivative $w_{2,11}$ of the transverse displacement. In section 2.2, for example, $w_{1,11}$ for the cantilever beam is null for all values of the applied end moment. Expressions (5.96) and (5.97) simplify as follows:

$$\Delta\varepsilon_b^0 = -\Delta w_{2,11} \quad (5.98)$$

$$\Delta\varepsilon_b^n = \frac{1}{2} w_{2,11} \Delta w_{c2,1}^2 \quad (5.99)$$

Stress increment: The stress in the known instant configuration s is given by:

$$s = s_a + y_2 s_b \quad (5.100)$$

$s_a = E \varepsilon_a$ axial 2. Piola-Kirchhoff stress

$s_b = E \varepsilon_b$ bending stress derivative

The stress in the unknown instant configuration $s+1$ is similarly given by:

$$s + \Delta s = s_a + \Delta s_a + y_2 (s_b + \Delta s_b) \quad (5.101)$$

The increment in the stress in step s is thus:

$$\Delta s = \Delta s_a + y_2 \Delta s_b \quad (5.102)$$

$\Delta s_a = E \Delta \varepsilon_a$ increment of the axial 2. Piola-Kirchhoff stress

$\Delta s_b = E \Delta \varepsilon_b$ increment of the bending stress derivative

5.2.4 Variation of the Incremental Variables

Assume that the instant configuration $C^{(s+1)}$ of the frame is to be determined. The residuals of the differential governing equations are therefore weighted with the variation of the system displacement vector in configuration $C^{(s+1)}$:

$$\delta(\mathbf{d} + \Delta \mathbf{d}) = \delta \mathbf{d} + \delta(\Delta \mathbf{d})$$

Since the state of the frame in configuration $C^{(s)}$ is known, the variation $\delta \mathbf{d}$ of the displacement is not required and the variation of the nodal displacement is therefore:

$$\delta(\mathbf{d} + \Delta \mathbf{d}) = \delta(\Delta \mathbf{d}) \quad (5.103)$$

The variation of the member displacement vector referred to reference and to instant space is determined by analogous reasoning:

$$\delta(\mathbf{v}_e + \Delta \mathbf{v}_e) = \delta(\Delta \mathbf{v}_e) \quad (5.104)$$

$$\delta(\mathbf{w}_e + \Delta \mathbf{w}_e) = \delta(\Delta \mathbf{w}_e) \quad (5.105)$$

The variation of the axial strain increment follows from (5.92) and (5.93):

$$\delta(\Delta \varepsilon_a) = \delta(\Delta \varepsilon_a^0) + \delta(\Delta \varepsilon_a^n) \quad (5.106)$$

$$\delta(\Delta \varepsilon_a^0) = (1 - w_{c1,1}) \delta(\Delta w_{c1,1}) \quad (5.107)$$

$$\delta(\Delta \varepsilon_a^n) = -\Delta w_{c1,1} \delta(\Delta w_{c1,1}) + (1 - w_{c1,1}) \Delta w_{c2,1} \delta(\Delta w_{c2,1}) \quad (5.108)$$

The variation of the increment of the bending strain derivative follows from (5.98) and (5.99):

$$\delta(\Delta \varepsilon_b) = \delta(\Delta \varepsilon_b^0) + \delta(\Delta \varepsilon_b^n) \quad (5.109)$$

$$\delta(\Delta \varepsilon_b^0) = -\delta(\Delta w_{2,11}) \quad (5.110)$$

$$\delta(\Delta \varepsilon_b^n) = w_{2,11} \Delta w_{c2,1} \delta(\Delta w_{c2,1}) \quad (5.111)$$

5.2.5 Governing Equations for the Displacement Increments

The governing equation (5.77) for plane frame behaviour is applied to configuration C_{s+1} of the frame. The left-hand side of (5.77) yields:

$$\begin{aligned} \sum_{\mathbf{e}} \int_{C_e} \delta(\varepsilon + \Delta \varepsilon)(s + \Delta s) dv &= \quad (5.112) \\ \sum_{\mathbf{e}} \int_{C_e} \delta(\Delta \varepsilon_a^0 + \Delta \varepsilon_a^n)(s_a + \Delta s_a) dv &+ \sum_{\mathbf{e}} \int_{C_e} y_2^2 \delta(\Delta \varepsilon_b^0 + \Delta \varepsilon_b^n)(s_b + \Delta s_b) dv = \\ \sum_{\mathbf{e}} \int_{C_e} \delta(\Delta \varepsilon_a^0) s_a dv &+ \sum_{\mathbf{e}} \int_{C_e} \delta(\Delta \varepsilon_a^n) s_a dv + \sum_{\mathbf{e}} \int_{C_e} \delta(\Delta \varepsilon_a) \Delta s_a dv + \\ \sum_{\mathbf{e}} \int_{C_e} y_2^2 \delta(\Delta \varepsilon_b^0) s_b dv &+ \sum_{\mathbf{e}} \int_{C_e} y_2^2 \delta(\Delta \varepsilon_b^n) s_b dv + \sum_{\mathbf{e}} \int_{C_e} y_2^2 \delta(\Delta \varepsilon_b) \Delta s_b dv \end{aligned}$$

The right-hand side of equation (5.77) yields:

$$\begin{aligned} \sum_{\mathbf{e}} \int_{L_e} \delta(\mathbf{v} + \Delta \mathbf{v})^T (\mathbf{p}_a + \Delta \mathbf{p}_a) dy_1 &+ \delta(\Delta \mathbf{d}_s)^T (\mathbf{q}_s + \Delta \mathbf{q}_s + \mathbf{r}_s + \Delta \mathbf{r}_s) = \quad (5.113) \\ \sum_{\mathbf{e}} \int_{L_e} \delta(\Delta \mathbf{v})^T \mathbf{p}_a dy_1 &+ \sum_{\mathbf{e}} \int_{L_e} \delta(\Delta \mathbf{v})^T \Delta \mathbf{p}_a dy_1 + \delta(\Delta \mathbf{d}_s)^T \mathbf{q}_s + \delta(\Delta \mathbf{d}_s)^T \Delta \mathbf{q}_s + \delta(\Delta \mathbf{d}_s)^T \mathbf{r}_s + \delta(\Delta \mathbf{d}_s)^T \Delta \mathbf{r}_s \end{aligned}$$

Consider the terms in (5.112) and (5.113) which are associated with the state of displacement, load and reaction in configuration $C^{(s)}$. If the solution for $C^{(s)}$ were exact, these terms would cancel each other. Due to the approximations which were made in the preceding steps, the solution for $C^{(s)}$ is not exact and these terms lead to an error δe whose value is given by:

$$\begin{aligned} \delta e = - \sum_{\mathbf{e}} \int_{C_e} \delta(\Delta \varepsilon_a^0) s_a dv &- \sum_{\mathbf{e}} \int_{C_e} y_2^2 \delta(\Delta \varepsilon_b^0) s_b dv + \sum_{\mathbf{e}} \int_{a_e} \delta(\Delta \mathbf{v})^T \mathbf{p}_a dy_1 + \\ &\delta(\Delta \mathbf{d}_s)^T \mathbf{q}_s + \delta(\Delta \mathbf{d}_s)^T \mathbf{r}_s \quad (5.114) \end{aligned}$$

Variations (5.107) and (5.110) of the strain increments are substituted into (5.114):

$$\begin{aligned} \delta e = - \sum_{\mathbf{e}} \int_{C_e} (1 - w_{c1,1}) \delta(\Delta w_{c1,1}) s_a dv &+ \sum_{\mathbf{e}} \int_{C_e} y_2^2 \delta(\Delta w_{2,11}) s_b dv + \\ &\sum_{\mathbf{e}} \int_{a_e} \delta(\Delta \mathbf{v})^T \mathbf{p}_a dy_1 + \delta(\Delta \mathbf{d}_s)^T \mathbf{q}_s + \delta(\Delta \mathbf{d}_s)^T \mathbf{r}_s \quad (5.115) \end{aligned}$$

The variation $\delta(\Delta \varepsilon_a^n)$ of the nonlinear component of the axial strain is substituted from (5.108) into the second term on the right-hand side of equation (5.112):

$$\sum_e \int_{C_e} \delta(\Delta \varepsilon_a^n) s_a dv = \sum_e \int_{C_e} \left(-\Delta w_{c1,1} \delta(\Delta w_{c1,1}) + (1 - w_{c1,1}) \Delta w_{c2,1} \delta(\Delta w_{c2,1}) \right) s_a dv \quad (5.116)$$

The nonlinear third term on the right-hand side of (5.112) is approximated iteratively in each load step. In iteration 0, the nonlinear part is set to null. In the subsequent iterations, the approximate values of the displacement derivatives from the preceding iteration are used to approximate the nonlinear term. The strain increment $\Delta \varepsilon_a$ and its variation $\delta(\Delta \varepsilon_a)$ are taken from (5.92), (5.93), (5.107) and (5.108):

$$\Delta \varepsilon_a = q \Delta w_{c1,1} - \frac{1}{2} \Delta w_{c1,1}^2 + \frac{1}{2} q \Delta w_{c2,1}^2 \quad (5.117)$$

$$\delta(\Delta \varepsilon_a) = q \delta(\Delta w_{c1,1}) - \Delta w_{c1,1} \delta(\Delta w_{c1,1}) + q \Delta w_{c2,1} \delta(\Delta w_{c2,1}) \quad (5.118)$$

$$q = 1 - w_{c1,1} \quad (5.119)$$

$w_{cm,1}$ derivatives of the chord displacement coordinates in $C^{(s)}$ with respect to y_1
 $\Delta w_{cm,1}$ derivatives of the chord displacement increments from $C^{(s)}$ to $C^{(s+1)}$

The product of the strain increment with its variation is expressed as follows:

$$\begin{aligned} \Delta \varepsilon_a \delta(\Delta \varepsilon_a) &= q^2 \Delta w_{c1,1} \delta(\Delta w_{c1,1}) - q \Delta w_{c1,1}^2 \delta(\Delta w_{c1,1}) + q^2 \Delta w_{c1,1} \Delta w_{c2,1} \delta(\Delta w_{c2,1}) \\ &\quad - \frac{1}{2} q \Delta w_{c1,1}^2 \delta(\Delta w_{c1,1}) + \frac{1}{2} \Delta w_{c1,1}^3 \delta(\Delta w_{c1,1}) - \frac{1}{2} q \Delta w_{c1,1} \Delta w_{c2,1} \delta(\Delta w_{c2,1}) \\ &\quad + \frac{1}{2} q^2 \Delta w_{c2,1}^2 \delta(\Delta w_{c2,1}) - \frac{1}{2} q \Delta w_{c1,1} \Delta w_{c2,1}^2 \delta(\Delta w_{c1,1}) + \frac{1}{2} q^2 \Delta w_{c2,1}^3 \delta(\Delta w_{c2,1}) \end{aligned}$$

The expressions are rearranged in such a fashion that they become symmetric in the linearised terms. Approximate displacement increments are marked with a bar:

$$\begin{aligned} \Delta \varepsilon_a \delta(\Delta \varepsilon_a) &= \left(q - \frac{1}{2} \Delta \bar{w}_{c1,1} \right) \left(q - \Delta \bar{w}_{c1,1} \right) \Delta w_{c1,1} \delta(\Delta w_{c1,1}) + \quad (5.120) \\ &\quad \frac{1}{2} q \Delta \bar{w}_{c2,1} \left(q - \Delta \bar{w}_{c1,1} \right) \left(\Delta w_{c1,1} \delta(\Delta w_{c2,1}) + \Delta w_{c2,1} \delta(\Delta w_{c1,1}) \right) + \\ &\quad \frac{1}{2} q^2 \left(\Delta \bar{w}_{c1,1} + \Delta \bar{w}_{c2,1}^2 \right) \Delta w_{c2,1} \delta(\Delta w_{c2,1}) \end{aligned}$$

The expression $\Delta \varepsilon_a \delta(\Delta \varepsilon_a)$ is substituted from equation (5.120) into the third term on the right-hand side of equation (5.112):

$$\begin{aligned} \sum_e \int_{C_e} \delta(\Delta \varepsilon_a) \Delta s_a dv &= \sum_e \int_{C_e} E \left(q - \frac{1}{2} \Delta \bar{w}_{c1,1} \right) \left(q - \Delta \bar{w}_{c1,1} \right) \delta(\Delta w_{c1,1}) \Delta w_{c1,1} dv + \quad (5.121) \\ &\quad \sum_e \int_{C_e} \frac{1}{2} E q \left(q - \Delta \bar{w}_{c1,1} \right) \Delta \bar{w}_{c2,1} \left(\Delta w_{c1,1} \delta(\Delta w_{c2,1}) + \Delta w_{c2,1} \delta(\Delta w_{c1,1}) \right) dv + \\ &\quad \sum_e \int_{C_e} \frac{1}{2} E q^2 \left(\Delta \bar{w}_{c1,1} + \Delta \bar{w}_{c2,1}^2 \right) \Delta w_{c2,1} \delta(\Delta w_{c2,1}) dv \end{aligned}$$

The variation $\delta(\Delta \varepsilon_b^n)$ of the nonlinear component of the bending strain derivative is substituted from (5.111) into the fifth term on the right-hand side of equation (5.112):

$$\sum_e \int_{C_e} y_2^2 \delta(\Delta \varepsilon_b^n) s_b dv = - \sum_e \int_{C_e} y_2^2 E (w_{2,11})^2 \Delta w_{c2,1} \delta(\Delta w_{c2,1}) dv \quad (5.122)$$

Expression (5.122) represents the influence of the change in direction of the shear force in a member on the equilibrium of the frame and is neglected. The nonlinear sixth term on the right-hand side of (5.112) is approximated iteratively in the same manner as the third term. The strain derivative increment $\Delta \varepsilon_b$ and its variation $\delta(\Delta \varepsilon_b)$ are taken from (5.98), (5.99), (5.110) and (5.111):

$$\Delta \varepsilon_b = -\Delta w_{2,11} + \frac{1}{2} w_{2,11} \Delta w_{c2,1}^2 \quad (5.123)$$

$$\delta(\Delta \varepsilon_b) = -\delta(\Delta w_{2,11}) + w_{2,11} \Delta w_{c2,1} \delta(\Delta w_{c2,1}) \quad (5.124)$$

The terms containing third powers of the displacement increments are neglected in the product of the strain increment with its variation:

$$\Delta \varepsilon_b \delta(\Delta \varepsilon_b) = \Delta w_{2,11} \delta(\Delta w_{2,11}) \quad (5.125)$$

Expression $\Delta \varepsilon_b \delta(\Delta \varepsilon_b)$ is substituted from equation (5.125) into the sixth term on the right-hand side of equation (5.112):

$$\sum_e \int_{C_e} y_2^2 \delta(\Delta \varepsilon_b) \Delta s_b dv = \sum_e \int_{C_e} y_2^2 E \delta(\Delta w_{2,11}) \Delta w_{2,11} dv \quad (5.126)$$

Expressions (5.115), (5.116), (5.121) and (5.126) are substituted into expressions (5.112) and (5.113) for the left- and right-hand sides of the governing equations for the displacement increments of plane frames:

$$\begin{aligned} & \sum_e \int_{C_e} (-\Delta w_{c1,1} \delta(\Delta w_{c1,1}) + q \Delta w_{c2,1} \delta(\Delta w_{c2,1})) s_a dv + \quad (5.127) \\ & \sum_e \int_{C_e} E (q - \frac{1}{2} \Delta \bar{w}_{c1,1}) (q - \Delta \bar{w}_{c1,1}) \delta(\Delta w_{c1,1}) \Delta w_{c1,1} dv + \\ & \sum_e \int_{C_e} \frac{1}{2} E q (q - \Delta \bar{w}_{c1,1}) \Delta \bar{w}_{c2,1} (\Delta w_{c1,1} \delta(\Delta w_{c2,1}) + \Delta w_{c2,1} \delta(\Delta w_{c1,1})) dv + \\ & \sum_e \int_{C_e} \frac{1}{2} E q^2 (\Delta \bar{w}_{c1,1} + \Delta \bar{w}_{c2,1}^2) \Delta w_{c2,1} \delta(\Delta w_{c2,1}) dv + \\ & \sum_e \int_{C_e} y_2^2 E \delta(\Delta w_{2,11}) \Delta w_{2,11} dv = \\ & \delta e + \sum_e \int_{L_e} \delta(\Delta \mathbf{v})^T \Delta \mathbf{p}_a dy_1 + \delta(\Delta \mathbf{d}_s)^T \Delta \mathbf{q}_s + \delta(\Delta \mathbf{d}_s)^T \Delta \mathbf{r}_s \end{aligned}$$

Expression (5.127) is the incremental governing equation for plane frames. Because of the assumptions made in the derivation, the equation is approximate.

5.2.6 Governing Equations for a Member

The governing equations for the chord displacements and the relative displacements of a frame member are formulated for the treatment of member loads. It is assumed that the change in geometry in step s can be neglected in formulating the governing equations for configuration s . Equation (5.127) is reduced as follows:

$$\sum_e \int_0^a EA \delta(\Delta w_{c1,1}) \Delta w_{c1,1} dy_1 + \sum_e \int_0^a EJ \delta(\Delta w_{2,11}) \Delta w_{2,11} dy_1 = \delta e + \sum_e \int_0^a \delta(\Delta \mathbf{w})^T \Delta \mathbf{p}_c dy_1 + \delta(\Delta \mathbf{d}_s)^T \Delta \mathbf{q}_s + \delta(\Delta \mathbf{d}_s)^T \Delta \mathbf{r}_s \quad (5.128)$$

Chord displacement: Consider a frame member with prescribed node displacement increments Δw_{1A} and Δw_{1B} that carries an incremental load Δp_{c1} per unit length in the reference configuration. Equation (5.128) reduces to the following incremental governing equation for the member:

$$\int_0^a EA \delta(\Delta w_{c1,1}) \Delta w_{c1,1} dy_1 = \int_0^a \delta(\Delta w_{c1}) \Delta p_{c1} dy_1 + \delta(\Delta w_{1A}) \Delta r_{1A} + \delta(\Delta w_{1B}) \Delta r_{1B} \quad (5.129)$$

The left-hand side of the equation is integrated by parts:

$$-\int_0^a EA \delta(\Delta w_{c1}) \Delta w_{c1,1} dy_1 + EA \delta(\Delta w_{c1}) \Delta w_{c1,1} \Big|_B - EA \delta(\Delta w_{c1}) \Delta w_{c1,1} \Big|_A = \int_0^a \delta(\Delta w_{c1}) \Delta p_{c1} dy_1 + \delta(\Delta w_{1A}) \Delta r_{1A} + \delta(\Delta w_{1B}) \Delta r_{1B}$$

This equation is satisfied for arbitrary permissible variations of the displacement if the displacement and reaction increments satisfy the following equations:

$$EA \Delta w_{c1,11} + \Delta p_{c1} = 0 \quad (5.130)$$

$$\begin{aligned} \Delta r_{1A} &= -EA \Delta w_{c1,1} \Big|_A & \Delta w_{c1} \Big|_A &= \text{prescribed value at node A} \\ \Delta r_{1B} &= EA \Delta w_{c1,1} \Big|_B & \Delta w_{c1} \Big|_B &= \text{prescribed value at node B} \end{aligned}$$

Relative displacement: Consider a frame member with prescribed node displacement increments $\Delta w_{2A}, \Delta w_{2B}$ and prescribed node rotation increments $\Delta \beta_A, \Delta \beta_B$ that carries an incremental load Δp_{c2} per unit length in the reference configuration. Equation (5.127) reduces to the following incremental governing equation for the member:

$$\int_0^a EJ \delta(\Delta w_{r2,11}) \Delta w_{r2,11} dy_1 = \int_0^a \delta(\Delta w_{r2}) \Delta p_{c2} dy_1 + \delta(\Delta w_{2A}) \Delta r_{2A} + \delta(\Delta w_{2B}) \Delta r_{2B} + \delta(\Delta \beta_A) \Delta m_A + \delta(\Delta \beta_B) \Delta m_B \quad (5.131)$$

The left-hand side of the equation is integrated by parts:

$$-\int_0^a EJ \delta(\Delta w_{r2,1}) \Delta w_{r2,1,1} dy_1 + EJ \delta(\Delta w_{r2,1}) \Delta w_{r2,1} \Big|_B - EJ \delta(\Delta w_{r2,1}) \Delta w_{r2,1} \Big|_A = \int_0^a \delta(\Delta w_{r2}) \Delta p_{c2} dy_1 + \delta(\Delta w_{2A}) \Delta r_{2A} + \delta(\Delta w_{2B}) \Delta r_{2B} + \delta(\Delta \beta_A) \Delta m_A + \delta(\Delta \beta_B) \Delta m_B$$

The second and third terms on the left-hand side and the last two terms on the right-hand side must be equal for arbitrary variations $\delta(\Delta w_{2B})$. The terms are equal if the following equations are satisfied:

$$\begin{aligned} \Delta m_A &= -EJ \Delta w_{r2,11}|_A & \Delta \beta|_A &= \text{prescribed value at node A} \\ \Delta m_B &= EJ \Delta w_{r2,11}|_B & \Delta \beta|_B &= \text{prescribed value at node B} \end{aligned} \quad (5.132)$$

The left-hand side of the equation is integrated by parts once more:

$$\begin{aligned} &\int_0^a EJ \delta(\Delta w_{r2}) \Delta w_{r2,1111} dy_1 + EJ \delta(\Delta w_{r2}) \Delta w_{r2,111}|_B - EJ \delta(\Delta w_{r2}) \Delta w_{r2,111}|_A = \\ &\int_0^a \delta(\Delta w_{r2}) \Delta p_{c2} dy_1 + \delta(\Delta w_{2A}) \Delta r_{2A} + \delta(\Delta w_{2B}) \Delta r_{2B} \end{aligned}$$

This equation is satisfied for arbitrary permissible variations of the displacement if the displacement and reaction increments satisfy the following equations:

$$EJ \Delta w_{r2,1111} - \Delta p_{c2} = 0 \quad (5.133)$$

$$\begin{aligned} \Delta r_{2A} &= -EJ \Delta w_{r1,111}|_A & \Delta w_{r2}|_A &= \text{prescribed value at node A} \\ \Delta r_{2B} &= EJ \Delta w_{r1,111}|_B & \Delta w_{r2}|_B &= \text{prescribed value at node B} \end{aligned}$$

5.2.7 Member Loads

Interpolation functions: Governing equations (5.128) for the displacement increments are linear. The displacement increment due to the increment of the member load can therefore be superimposed with the displacement increments due to the increments of the generalised node displacements to yield the total increments of the member displacement. This raises the following question: which of the displacement functions must be used as interpolation functions in the finite element method to compute the element stiffness matrix and the element load vector?

Chord displacement: Let the total chord displacement of a member in (5.130) be expressed as the sum of the particular solution of the governing equation (5.130) for the load Δp_{c1} and fixed supports with the homogeneous solutions for unit node displacements, multiplied by the node displacement increments:

$$\Delta w_{c1} = s_0 + \mathbf{s}^T \Delta \mathbf{w}_1 \quad (5.134)$$

$$\Delta w_{c1} = s_0 + \begin{bmatrix} s_1 & s_2 \end{bmatrix} \begin{bmatrix} \Delta w_{1A} \\ \Delta w_{1B} \end{bmatrix}$$

Displacement increment s_0 satisfies equation (5.130) for the fixed-end member:

$$EA \frac{d^2 s_0}{dz_1^2} + p_{c1} = 0 \quad (5.135)$$

$$y_1 = 0: \quad s_0 = 0$$

$$y_1 = a: \quad s_0 = 0$$

The displacement increments s_1, s_2 satisfy the homogeneous equation (5.130) for unit values of the node displacement increments:

$$EA \frac{d^2 s_k}{dy_1^2} = 0 \quad k \in \{1,2\} \quad (5.136)$$

$$y_1 = 0: \quad s_1 = 1 \quad s_2 = 0$$

$$y_1 = a: \quad s_1 = 0 \quad s_2 = 1$$

The contribution of the member to equation (5.128) is found by substituting (5.134) into (5.128):

$$\begin{aligned} & \int_0^a EA \delta \left(\frac{ds_0}{dy_1} + \Delta \mathbf{w}_1^T \frac{d\mathbf{s}}{dy_1} \right) \left(\frac{ds_0}{dy_1} + \Delta \mathbf{w}_1^T \frac{d\mathbf{s}}{dy_1} \right) dy_1 - \int_0^a \delta (s_0 + \Delta \mathbf{w}_1^T \mathbf{s}) p_{c1} dy_1 = \\ & \int_0^a EA \delta \left(\frac{ds_0}{dy_1} \right) \frac{ds_0}{dy_1} dy_1 + \int_0^a EA \delta \left(\frac{ds_0}{dy_1} \right) (\Delta \mathbf{w}_1^T \frac{d\mathbf{s}}{dy_1}) dy_1 + \int_0^a EA \delta (\Delta \mathbf{w}_1^T \frac{d\mathbf{s}}{dy_1}) \frac{ds_0}{dy_1} dy_1 + \\ & \int_0^a EA \delta (\Delta \mathbf{w}_1^T \frac{d\mathbf{s}}{dy_1}) (\Delta \mathbf{w}_1^T \frac{d\mathbf{s}}{dy_1}) dy_1 - \int_0^a \delta (s_0) \Delta p_{c1} dy_1 - \int_0^a \delta (\Delta \mathbf{w}_1^T \mathbf{s}) \Delta p_{c1} dy_1 \end{aligned} \quad (5.137)$$

The particular solution s_0 with the boundary conditions (5.135) satisfies (5.129):

$$\int_0^a EA \delta \left(\frac{ds_0}{dy_1} \right) \frac{ds_0}{dy_1} dy_1 - \int_0^a \delta (s_0) \Delta p_{c1} dy_1 = 0 \quad (5.138)$$

The second and third terms on the right-hand side of (5.137) are integrated by parts:

$$\begin{aligned} & \int_0^a EA \delta \left(\frac{ds_0}{dy_1} \right) (\Delta \mathbf{w}_1^T \frac{d\mathbf{s}}{dy_1}) dy_1 = - \int_0^a \delta s_0 \Delta \mathbf{w}_1^T (EA \frac{d^2 \mathbf{s}}{dy_1^2}) dy_1 + \delta s_0 EA \Delta \mathbf{w}_1^T \frac{d\mathbf{s}}{dy_1} \Big|_A^B \\ & \int_0^a EA \delta (\Delta \mathbf{w}_1^T) \frac{d\mathbf{s}}{dy_1} \frac{ds_0}{dy_1} dy_1 = - \int_0^a \delta (\Delta \mathbf{w}_1^T)^T (EA \frac{d^2 \mathbf{s}}{dy_1^2}) s_0 dy_1 + \delta (\Delta \mathbf{w}_1^T)^T (EA \frac{d\mathbf{s}}{dy_1}) s_0 \Big|_A^B \end{aligned}$$

Equation (5.136) shows that the first term on the right-hand side of both equations is null. The second terms are also null because $\delta s_0 = s_0 = 0$ at nodes A and B of the fixed-end member. The contribution of the member to (5.129) is thus:

$$\int_0^a EA \delta (\Delta \mathbf{w}_1^T \frac{d\mathbf{s}}{dy_1}) (\Delta \mathbf{w}_1^T \frac{d\mathbf{s}}{dy_1}) dy_1 - \int_0^a \delta (\Delta \mathbf{w}_1^T \mathbf{s}) \Delta p_{c1} dy_1 \quad (5.139)$$

Relative displacement: Let the total relative displacement of a member in (5.131) be expressed as the sum of the particular solution t_0 of the governing equation (5.133) for the load Δp_{c2} and fixed supports with the homogeneous solutions for unit node displacements, multiplied by the node displacement increments:

$$\Delta \mathbf{w}_{r2} = t_0 + \mathbf{t}^T \Delta \mathbf{w}_2 \quad (5.140)$$

$$\Delta \mathbf{w}_{r2} = t_0 + \begin{array}{|c|c|c|c|} \hline t_1 & t_2 & t_3 & t_4 \\ \hline \end{array} \begin{array}{|c|} \hline \Delta \mathbf{w}_{2A} \\ \hline \Delta \beta_A \\ \hline \Delta \mathbf{w}_{2B} \\ \hline \Delta \beta_B \\ \hline \end{array}$$

Displacement increment t_0 satisfies equation (5.133) for the fixed-end member:

$$EJ \frac{d^4 t_0}{dy_1^4} - \Delta p_{c2} = 0 \quad (5.141)$$

$$y_1 = 0: t_0 = 0 \quad \frac{dt_0}{dy_1} = 0 \quad y_1 = a: t_0 = 0 \quad \frac{dt_0}{dy_1} = 0$$

Displacement increments t_1, \dots, t_4 satisfy the homogeneous equation (5.133) for unit values of the node displacement increments:

$$EJ \frac{d^4 t_k}{dy_1^4} = 0 \quad k \in \{1, 2, 3, 4\} \quad (5.142)$$

$$\begin{aligned} y_1 = 0: t_1 = 1 \quad \frac{dt_1}{dy_1} = 0 \quad y_1 = a: t_1 = 0 \quad \frac{dt_1}{dy_1} = 0 \\ t_2 = 0 \quad \frac{dt_2}{dy_1} = 1 \quad t_2 = 0 \quad \frac{dt_2}{dy_1} = 0 \\ t_3 = 0 \quad \frac{dt_3}{dy_1} = 0 \quad t_3 = 1 \quad \frac{dt_3}{dy_1} = 0 \\ t_4 = 0 \quad \frac{dt_4}{dy_1} = 0 \quad t_4 = 0 \quad \frac{dt_4}{dy_1} = 1 \end{aligned}$$

The contribution of the member to (5.128) is found by substituting (5.140) in (5.128):

$$\begin{aligned} \int_0^a EJ \delta \left(\frac{d^2 t_0}{dy_1^2} + \Delta \mathbf{w}_{r2}^T \frac{d^2 \mathbf{t}}{dy_1^2} \right) \left(\frac{d^2 t_0}{dz_1^2} + \Delta \mathbf{w}_{r2}^T \frac{d^2 \mathbf{t}}{dy_1^2} \right) dy_1 - \int_0^a \delta (t_0 + \Delta \mathbf{w}_{r2}^T \mathbf{t}) p_{c2} dy_1 = \\ \int_0^a EJ \delta \left(\frac{d^2 t_0}{dy_1^2} \right) \frac{d^2 t_0}{dy_1^2} dy_1 + \int_0^a EJ \delta \left(\frac{d^2 t_0}{dy_1^2} \right) (\Delta \mathbf{w}_{r2}^T \frac{d^2 \mathbf{t}}{dy_1^2}) dy_1 + \int_0^a EJ \delta (\Delta \mathbf{w}_{r2})^T \frac{d^2 \mathbf{t}}{dy_1^2} \frac{d^2 t_0}{dy_1^2} dy_1 + \\ \int_0^a EJ \delta (\Delta \mathbf{w}_{r2})^T \frac{d^2 \mathbf{t}}{dy_1^2} \left(\frac{d^2 \mathbf{t}}{dy_1^2} \right)^T \Delta \mathbf{w}_{r2} dy_1 - \int_0^a \delta t_0 p_2 dy_1 - \int_0^a \delta (\Delta \mathbf{w}_{r2})^T \mathbf{t} \Delta p_{c2} dy_1 \end{aligned}$$

The particular solution t_0 satisfies equation (5.141) so that:

$$\int_0^a EJ \delta \left(\frac{d^2 t_0}{dy_1^2} \right) \frac{d^2 t_0}{dy_1^2} dy_1 - \int_0^a \delta (t_0) \Delta p_{c2} dy_1 = 0 \quad (5.143)$$

The second and third terms on the right-hand side are integrated by parts:

$$\begin{aligned} \int_0^a EJ \delta \left(\frac{d^2 t_0}{dy_1^2} \right) (\Delta \mathbf{w}_{r2}^T \frac{d^2 \mathbf{t}}{dy_1^2}) dy_1 = - \int_0^a EJ \delta \left(\frac{dt_0}{dy_1} \right) (\Delta \mathbf{w}_{r2}^T \frac{d^3 \mathbf{t}}{dy_1^3}) dy_1 + \delta \left(\frac{dt_0}{dy_1} \right) EJ \Delta \mathbf{w}_{r2}^T \frac{d^2 \mathbf{t}}{dy_1^2} \Big|_A^B \\ \int_0^a EJ \delta (\Delta \mathbf{w}_{r2})^T \frac{d^2 \mathbf{t}}{dy_1^2} \frac{d^2 t_0}{dy_1^2} dy_1 = - \int_0^a EJ \delta (\Delta \mathbf{w}_{r2})^T \frac{d^3 \mathbf{t}}{dy_1^3} \frac{dt_0}{dy_1} dy_1 + EJ \delta (\Delta \mathbf{w}_{r2})^T \frac{d^2 \mathbf{t}}{dy_1^2} \frac{dt_0}{dy_1} \Big|_A^B \end{aligned}$$

The second term on the right-hand side of both equations is null due to (5.141). Both equations are integrated once more by parts:

$$\begin{aligned} \int_0^a EJ \delta \left(\frac{d^2 t_0}{dy_1^2} \right) (\Delta \mathbf{w}_{r2}^T \frac{d^2 \mathbf{t}}{dy_1^2}) dy_1 = \int_0^a \delta t_0 \Delta \mathbf{w}_{r2}^T (EJ \frac{d^4 \mathbf{t}}{dy_1^4}) dy_1 - \delta t_0 EJ \Delta \mathbf{w}_{r2}^T \frac{d^3 \mathbf{t}}{dy_1^3} \Big|_A^B \\ \int_0^a EJ \delta (\Delta \mathbf{w}_{r2})^T \frac{d^2 \mathbf{t}}{dy_1^2} \frac{d^2 t_0}{dy_1^2} dy_1 = \int_0^a \delta (\Delta \mathbf{w}_{r2})^T (EJ \frac{d^4 \mathbf{t}}{dy_1^4}) t_0 dy_1 - EJ \delta (\Delta \mathbf{w}_{r2})^T \frac{d^3 \mathbf{t}}{dy_1^3} t_0 \Big|_A^B \end{aligned}$$

Equation (5.142) shows that the first term on the right-hand side of both equations is null. The second term is also null because $\delta t_0 = t_0 = 0$ at nodes A and B of the fixed-end member in (5.141). The contribution of the member to (5.131) is thus:

$$\int_0^a E J \delta(\Delta \mathbf{w}_{r2})^T \frac{d^2 \mathbf{t}}{dy_1^2} \left(\frac{d^2 \mathbf{t}}{dy_1^2} \right)^T \Delta \mathbf{w}_{r2} dy_1 - \int_0^a \delta(\Delta \mathbf{w}_{r2})^T \mathbf{t} \Delta p_{e2} dy_1 \quad (5.144)$$

Since expressions (5.139) and (5.144) contain the homogeneous solutions \mathbf{s} and \mathbf{t} of the governing equations, the element stiffness matrix and the element load vector must be computed with the trial functions \mathbf{s} and \mathbf{t} . The member loads are taken into account, as follows:

- (a) The member load increments are applied to the fixed-end members. The reactions they cause are computed and applied with the inverse sign as loads at the nodes of the frame.
- (b) The displacement increments of the nodes of the frame are computed by solving the governing equations for the frame as a whole.
- (c) The total displacements and stress resultants in the members are computed by adding the displacements and stress resultants caused by the node displacements to the displacements and stress resultants caused by the member loads.

5.3 ALGEBRAIC GOVERNING EQUATIONS FOR PLANE FRAMES

5.3.1 Interpolation Functions for the Finite Element Method

The nonlinear analysis of plane frames contains two operations that are performed with finite elements: the computation of the error vector for a given displacement and reaction state of the frame, and the computation of the displacement and reaction increments for a load step. In preparation of the computer implementation of the method of analysis, expressions (5.115) and (5.127) for these operations are now transformed from the analytical form to an algebraic form by means of the finite element method.

For the computation of the error vector, the total displacement of the member in its instant configuration is interpolated. For the computation of a load step, however, the increments of the displacements are interpolated. Although both interpolations are linear in z for the axial direction and cubic in z for the transverse direction, the two interpolations differ significantly.

In the interpolation for the error vector, the instant location of the nodes is known and the total displacement of the axis of a member is the sum of a linear chord displacement and a cubic relative displacement. These interpolation functions are not the exact solution of the differential governing equations for the members. The error vector is therefore not exact. The accuracy of an analysis can be enhanced by subdividing the given members of a frame into finite elements, so that the approximation of the exact member displacement is improved.

The interpolation of the displacement increments is made for a load step in which the node displacements change. The angle of rotation of the chord changes as a result of the node displacements, which, in turn, changes the relative angle of rotation independent of the change in the total angle of rotation. The results of section 5.2.7 show that the interpolation functions for the displacement increments must be exact solutions of the homogeneous incremental governing equations of the members, if the total displacement of points on the axis of a member is to be treated as the sum of the displacement due to member load and the displacement due to displacement of the nodes of the member.

The interpolation functions for the total displacement and for the displacement increment must be consistent. This is achieved by choosing the consistent linearisations of the displacement interpolation functions for the error vector as interpolation functions for the displacement increments.

The theory on which the computer implementation is based was developed for large displacements and rotations but small deformations of the members. If large deformations occur in a member of the frame, this member cannot be modelled by a single finite element, but must be subdivided into several elements, as shown in section 5.1.6.

5.3.2 Interpolation of the Displacements and Strains

The total displacement in the instant configuration, for which the error vector is to be determined, is interpolated in instant space with the normalised coordinate z defined in (5.62). The total displacement is the sum of the chord displacement and the relative displacement. The coordinates of the chord displacement vary linearly between their node values. The relative displacement is normal to the chord and varies cubically. Its value is null at both nodes.

It is convenient for the interpolation to replace the element displacement vectors \mathbf{v}_e and \mathbf{w}_e , which are defined in (5.23), by modified element displacement vectors that contain the tangent of the relative angle of rotation $\gamma = \beta - \theta$ instead of the total angle of rotation β as node variable. The modified member displacement vector is denoted by $\tilde{\mathbf{w}}_e$ if the displacement coordinates are referred to instant space, and by $\tilde{\mathbf{v}}_e$ if they are referred to reference space.

Chord displacement: The chord displacement of the axis of the member is interpolated linearly in the instant space for which the error vector is determined:

$$\mathbf{w}_c = \mathbf{S}_c \tilde{\mathbf{w}}_e \quad (5.145)$$

$$\mathbf{w}_c^T = \begin{bmatrix} w_{c1} & w_{c2} \end{bmatrix}$$

$$\mathbf{S}_c = \frac{1}{2} \begin{bmatrix} 1-z & 0 & 0 & 1+z & 0 & 0 \\ 0 & 1-z & 0 & 0 & 1+z & 0 \end{bmatrix} \quad -1 \leq z \leq 1$$

$$\tilde{\mathbf{w}}_e^T = \begin{bmatrix} w_{1A} & w_{2A} & * & w_{1B} & w_{1B} & * \end{bmatrix}$$

w_{ck} chord displacement coordinates of point P in instant space, $k \in \{1,2\}$

z normalised coordinate on axis z_1 , see (5.62)

w_{kA}, w_{kB} displacement coordinates of nodes A, B of the member in instant space

* any value

Transformations (5.20) and (5.26) are applied to (5.145) to obtain the interpolation of the chord displacement coordinates in reference space as a function of the modified element displacement vector $\tilde{\mathbf{v}}_e$ for the reference space:

$$\mathbf{v}_c = \mathbf{R}_{nyz} \mathbf{S}_c \mathbf{R}_{eyz}^T \tilde{\mathbf{v}}_e = \mathbf{S}_c \tilde{\mathbf{v}}_e \quad (5.146)$$

$$\mathbf{v}_c^T = \begin{bmatrix} v_{c1} & v_{c2} \end{bmatrix}$$

$$\tilde{\mathbf{v}}_e^T = \begin{bmatrix} v_{1A} & v_{2A} & * & v_{1B} & v_{2B} & * \end{bmatrix}$$

v_{ck} chord displacement coordinates of point P in reference space

v_{kA}, v_{kB} displacement coordinates of nodes A and B in reference space

The first derivative of chord displacement w_c with respect to z follows from (5.145):

$$\frac{d\mathbf{w}_c}{dz} = \frac{d\mathbf{S}_c}{dz} \tilde{\mathbf{w}}_e := \mathbf{S}_{c,z} \tilde{\mathbf{w}}_e \quad (5.147)$$

$$\mathbf{S}_{c,z} = \frac{1}{2} \begin{array}{|c|c|c|c|c|c|c|} \hline -1 & 0 & 0 & 1 & 0 & 0 & \\ \hline 0 & -1 & 0 & 0 & 1 & 0 & \\ \hline \end{array}$$

The first derivative of the chord displacement with respect to coordinate y_1 in the reference space is determined with the chain rule using expression (5.62):

$$\frac{d\mathbf{w}_c}{dy_1} = \frac{d\mathbf{w}_c}{dz} \frac{dz}{dy_1} = \frac{2}{a} \frac{d\mathbf{w}_c}{dz} \quad (5.148)$$

Define the differentiation vectors \mathbf{g}_1 and \mathbf{g}_2 :

$$\mathbf{g}_1 = \begin{array}{|c|} \hline -1 \\ \hline 0 \\ \hline 0 \\ \hline 1 \\ \hline 0 \\ \hline 0 \\ \hline \end{array} \quad \mathbf{g}_2 = \begin{array}{|c|} \hline 0 \\ \hline -1 \\ \hline 0 \\ \hline 0 \\ \hline 1 \\ \hline 0 \\ \hline \end{array} \quad (5.149)$$

The first derivatives of the chord displacements follow from (5.148):

$$\mathbf{w}_{ck,1} = \frac{1}{a} \mathbf{g}_k^T \tilde{\mathbf{w}}_e \quad k \in \{1,2\} \quad (5.150)$$

The second derivatives of \mathbf{w}_c with respect to y_1 are null.

Relative displacement: The relative displacement of the axis of the member is interpolated cubically in the instant space for which the error vector is determined:

$$\mathbf{w}_r = \mathbf{S}_r \bar{\mathbf{w}}_e \quad (5.151)$$

$$\begin{array}{|c|c|} \hline w_{r1} \\ \hline w_{r2} \\ \hline \end{array} = \frac{\hat{a}}{8} \begin{array}{|c|c|c|c|c|c|c|} \hline 0 & 0 & 0 & 0 & 0 & 0 & \\ \hline 0 & 0 & 1-z-z^2+z^3 & 0 & 0 & -1-z+z^2+z^3 & \\ \hline \end{array} \bar{\mathbf{w}}_e$$

$$\bar{\mathbf{w}}_e^T = \begin{array}{|c|c|c|c|c|c|} \hline w_{1A} & w_{2A} & \tan(\beta_A - \theta) & w_{1B} & w_{2B} & \tan(\beta_B - \theta) \\ \hline \end{array}$$

$\bar{\mathbf{w}}_e$ element displacement vector containing the relative rotations $\lambda = \beta - \theta$

w_{rk} relative displacement coordinates of point P in instant space

\hat{a} instant length of the member

The first derivative of \mathbf{w}_r with respect to z is derived from (5.151):

$$\frac{d\mathbf{w}_r}{dz} = \frac{d\mathbf{S}_r}{dz} \bar{\mathbf{w}}_e := \mathbf{S}_{r,z} \bar{\mathbf{w}}_e \quad (5.152)$$

$$\mathbf{S}_{r,z} = \frac{\hat{a}}{8} \begin{array}{|c|c|c|c|c|c|} \hline 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & -1-2z+3z^2 & 0 & 0 & -1+2z+3z^2 \\ \hline \end{array}$$

The second derivative of \mathbf{w}_r with respect to y_1 is derived from (5.152):

$$\frac{d^2 \mathbf{w}_r}{dz^2} = \frac{d^2 \mathbf{S}_r}{dz^2} \bar{\mathbf{w}}_e := \mathbf{S}_{r,zz} \bar{\mathbf{w}}_e \quad (5.153)$$

$$\mathbf{S}_{r,zz} = \frac{\hat{a}}{4} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1+3z & 0 & 0 & 1+3z \end{bmatrix}$$

The derivatives of the relative displacement with respect to y_1 are determined from expressions (5.151) to (5.153) using the chain rule and (5.62). The relative angle of rotation γ is assumed to be small for the small deformation theory and can be controlled by subdivision of the members. The trigonometric ratio $\tan \gamma$ can therefore be approximated by γ :

$$\frac{dw_r}{dy_1} = \frac{dw_r}{dz} \frac{dz}{dy_1} = \frac{2}{a} \mathbf{S}_{r,z} \bar{\mathbf{w}}_e \quad (5.154)$$

$$\frac{d^2 w_r}{dy_1^2} = \frac{d^2 w_r}{dz^2} \left(\frac{dz}{dy_1} \right)^2 = \frac{4}{a^2} \mathbf{S}_{r,zz} \bar{\mathbf{w}}_e$$

$$w_{r2,11} = \frac{\hat{a}}{a^2} ((-1+3z)\gamma_A + (1+3z)\gamma_B) \quad (5.155)$$

$$\gamma_A = \beta_A - \theta$$

$$\gamma_B = \beta_B - \theta$$

Strain: The strain in the member is determined with expressions (5.67) to (5.69). Expressions (5.150) for the first derivatives $w_{c1,1}$ and $w_{c2,1}$ of the chord displacement are substituted into expression (5.68) for the axial strain ε_a :

$$\varepsilon_a = w_{c1,1} - \frac{1}{2} w_{c1,1}^2 - \frac{1}{2} w_{c2,1}^2 \quad (5.156)$$

$$w_{c1,1} = \frac{1}{a} (w_{1B} - w_{1A})$$

$$w_{c2,1} = \frac{1}{a} (w_{2B} - w_{2A})$$

Expression (5.155) for the second derivative $w_{r2,11}$ of the relative displacement are substituted into expression (5.69) for the bending strain derivative ε_b :

$$\varepsilon_b = -w_{r2,11} = -\frac{\hat{a}}{a^2} ((-1+3z)\gamma_A + (1+3z)\gamma_B) \quad (5.157)$$

Expressions (5.156) and (5.157) are used to compute the strains for a given state of displacement of the member.

5.3.3 Interpolation of the Displacement and Strain Increments

Chord displacement increment: Consider equation (5.145) and let the element displacement vector in configuration $C^{(s+1)}$ be $\tilde{\mathbf{w}}_e + \Delta\tilde{\mathbf{w}}_e$:

$$\Delta\tilde{\mathbf{w}}_e = \begin{array}{|c|c|c|c|c|c|} \hline \Delta W_{1A} & \Delta W_{2A} & * & \Delta W_{1B} & \Delta W_{2B} & * \\ \hline \end{array} \quad (5.158)$$

The chord displacement in configuration $C^{(s+1)}$ follows from (5.145):

$$\begin{aligned} \mathbf{w}_c + \Delta\mathbf{w}_c &= \mathbf{S}_c(\tilde{\mathbf{w}}_e + \Delta\tilde{\mathbf{w}}_e) \\ \Delta\mathbf{w}_c &= \mathbf{S}_c\Delta\tilde{\mathbf{w}}_e \end{aligned} \quad (5.159)$$

$\Delta\mathbf{w}_c$ increment of the chord displacement

The first derivative of the increment of the chord displacement with respect to the axial coordinate y_1 in the reference space follows from (5.150):

$$\Delta W_{ck,1} = \frac{1}{a} \mathbf{g}_k^T \Delta\tilde{\mathbf{w}}_e \quad k \in \{1,2\} \quad (5.160)$$

The variation of the first derivative of the increment of the chord displacement with respect to the axial coordinate y_1 in the reference space follows directly from (5.160):

$$\delta(\Delta W_{ck,1}) = \frac{1}{a} \mathbf{g}_k^T \delta(\Delta\tilde{\mathbf{w}}_e) \quad k \in \{1,2\} \quad (5.161)$$

Relative displacement increment: The relative displacement w_{r2} in expression (5.151) is a function of a set of independent variables $\varphi := \{w_{2A}, \beta_A, w_{2B}, \beta_B\}$ and the dependent chord rotation $\theta(\varphi)$ of the member. The increment Δw_{r2} in the relative displacement due to an increment $\Delta\varphi$ in the independent variables is determined by consistent linearisation:

$$\Delta w_{r2} = \frac{d}{d\eta} \langle w_{r2}(\varphi + \eta\Delta\varphi) \rangle_{\eta=0} \quad (5.162)$$

The directional derivatives of the total translation coordinates w_i in the element displacement vector $\tilde{\mathbf{w}}_e$ in (5.151) are:

$$\frac{d}{d\eta} \langle w_i + \eta\Delta w_i \rangle_{\eta=0} = \Delta w_i \quad (5.163)$$

The directional derivative of the coefficient $\tan(\beta - \theta)$ in (5.151) is:

$$\frac{d}{d\eta} \langle \tan(\beta + \eta\Delta\beta - \theta - \eta\Delta\theta) \rangle_{\eta=0} = \frac{1}{\cos^2(\beta - \theta)} (\Delta\beta - \Delta\theta) \quad (5.164)$$

The term $\cos^2(\beta - \theta)$ is approximated by $\cos^2 \gamma = 1$. The increment $\Delta\theta$ of the chord rotation is expressed in terms of the total displacements in (5.78):

$$\Delta\theta = \frac{\Delta W_{2B} - \Delta W_{2A}}{a} \quad (5.165)$$

The increment $\Delta\tilde{\mathbf{w}}_e$ of the element displacement vector follows from expressions (5.163) to (5.165):

$$\Delta\tilde{\mathbf{w}}_e = \begin{array}{|c|c|c|c|c|c|} \hline \Delta W_{1A} & \Delta W_{2A} & \Delta\beta_A - \frac{\Delta W_{2B} - \Delta W_{2A}}{a} & \Delta W_{1B} & \Delta W_{2B} & \Delta\beta_B - \frac{\Delta W_{2B} - \Delta W_{2A}}{a} \\ \hline \end{array} \quad (5.166)$$

The increment in the relative displacement is expressed in terms of the element displacement vector \mathbf{w}_e containing the total rotation β by substituting (5.166) into (5.151) and rearranging the resulting product:

$$\Delta \mathbf{w}_{2r} = \frac{\hat{a}}{8} \begin{bmatrix} 0 & \frac{2}{a}(-z+z^3) & 1-z-z^2+z^3 & 0 & -\frac{2}{a}(-z+z^3) & -1-z+z^2+z^3 \end{bmatrix} \quad (5.167)$$

Total displacement increment: The total displacement increment $\Delta \mathbf{w}_p$ of a point on the axis of a member is the sum of the chord displacement increment $\Delta \mathbf{w}_c$ and the relative displacement increment $\Delta \mathbf{w}_r$ of the point:

$$\Delta \mathbf{w}_p = \Delta \mathbf{w}_c + \Delta \mathbf{w}_r \quad (5.168)$$

$\Delta \mathbf{w}_p$ increment of the total displacement of a point on the axis

Substitution of (5.160) and (5.167) into (5.168) leads to the interpolation formulas of linear frame theory, modified by the effect of the change in length of the member:

$$\Delta \mathbf{w}_p = \mathbf{S}_p \Delta \mathbf{w}_e \quad (5.169)$$

$$\Delta \mathbf{w}_m = \mathbf{s}_m^T \Delta \mathbf{w}_e \quad (5.170)$$

$$\mathbf{S}_p = \begin{bmatrix} \mathbf{s}_1^T \\ \mathbf{s}_2^T \end{bmatrix}$$

$$\mathbf{s}_1 = \begin{bmatrix} 0.50(1-z) \\ 0 \\ 0 \\ 0.50(1+z) \\ 0 \\ 0 \end{bmatrix}$$

$$\mathbf{s}_2 = \begin{bmatrix} 0 \\ 0.50(1-z) + 0.25q(-z+z^3) \\ 0.125\hat{a}(1-z-z^2+z^3) \\ 0 \\ 0.50(1+z) - 0.25q(-z+z^3) \\ 0.125\hat{a}(-1-z+z^2+z^3) \end{bmatrix}$$

$$q := \frac{\hat{a}}{a}$$

The first derivative of $\Delta \mathbf{w}_p$ with respect to z is obtained from (5.169):

$$\frac{d(\Delta \mathbf{w}_p)}{dz} = \frac{d\mathbf{S}_p}{dz} \Delta \mathbf{w}_e := \mathbf{S}_{p,z} \Delta \mathbf{w}_e \quad (5.171)$$

$$\mathbf{S}_{p,z} = \begin{bmatrix} \mathbf{t}_1^T \\ \mathbf{t}_2^T \end{bmatrix} \quad (5.172)$$

$$\mathbf{t}_1 = \begin{bmatrix} -0.50 \\ 0 \\ 0 \\ 0.50 \\ 0 \\ 0 \end{bmatrix}$$

$$\mathbf{t}_2 = \begin{bmatrix} 0 \\ -0.50 + 0.25q(-1+3z^2) \\ 0.125\hat{a}(-1-2z+3z^2) \\ 0 \\ 0.50 - 0.25q(-1+3z^2) \\ 0.125\hat{a}(-1+2z+3z^2) \end{bmatrix}$$

The second derivative of the displacement increment $\Delta \mathbf{w}_p$ with respect to z follows by differentiation of (5.171):

$$\frac{d^2(\Delta \mathbf{w}_P)}{dz^2} = \frac{d^2 \mathbf{S}_P}{dz^2} \Delta \mathbf{w}_e := \mathbf{S}_{P,zz} \Delta \mathbf{w}_e \quad (5.173)$$

$$\mathbf{S}_{P,zz} = \begin{array}{|c|} \hline \mathbf{0} \\ \hline 0.25 \mathbf{h}_2^\top \\ \hline \end{array}$$

$$\mathbf{h}_2 = \begin{array}{|c|} \hline 0 \\ \hline 6qz \\ \hline \hat{a}(-1+3z) \\ \hline 0 \\ \hline -6qz \\ \hline \hat{a}(1+3z) \\ \hline \end{array}$$

The second derivative of Δw_2 with respect to the axial coordinate y_1 is determined with the chain rule from (5.173) and (5.62):

$$\Delta w_{2,11} = \frac{1}{a^2} \mathbf{h}_2^\top \Delta \mathbf{w}_e \quad (5.174)$$

The variation of the second derivative of Δw_2 follows directly from (5.174):

$$\delta(\Delta w_{2,11}) = \frac{1}{a^2} \mathbf{h}_2^\top \delta(\Delta \mathbf{w}_e) \quad (5.175)$$

Since the chord displacement is linear, the second derivative of the relative displacement equals the second derivative of the total displacement.

Strain increment: Expression (5.160) for the first derivatives $\Delta w_{c1,1}$ and $w_{c2,1}$ of the chord displacement increment is substituted into expressions (5.91) to (5.93) for the axial strain increment:

$$\Delta \varepsilon_a = \Delta \varepsilon_a^0 + \Delta \varepsilon_a^n$$

$$\Delta \varepsilon_a^0 = (1 - w_{c1,1}) \Delta w_{c1,1} \quad (5.176)$$

$$\Delta \varepsilon_a^n = -\frac{1}{2} \Delta w_{c1,1}^2 + \frac{1}{2} (1 - w_{c1,1}) \Delta w_{c2,1}^2 \quad (5.177)$$

$$\Delta w_{c1,1} = \frac{1}{a} \mathbf{g}_1^\top \Delta \tilde{\mathbf{w}}_e$$

$$\Delta w_{c2,1} = \frac{1}{a} \mathbf{g}_2^\top \Delta \tilde{\mathbf{w}}_e$$

The variation of the axial strain increment follows from equations (5.106) to (5.108):

$$\delta(\Delta \varepsilon_a) = \delta(\Delta \varepsilon_a^0) + \delta(\Delta \varepsilon_a^n)$$

$$\delta(\Delta \varepsilon_a^0) = (1 - w_{c1,1}) \delta(\Delta w_{c1,1}) \quad (5.178)$$

$$\delta(\Delta \varepsilon_a^n) = -\Delta w_{c1,1} \delta(\Delta w_{c1,1}) + (1 - w_{c1,1}) \Delta w_{c2,1} \delta(\Delta w_{c2,1}) \quad (5.179)$$

$$\delta(\Delta w_{c1,1}) = \frac{1}{a} \mathbf{g}_1^\top \delta(\Delta \tilde{\mathbf{w}}_e)$$

$$\delta(\Delta w_{c2,1}) = \frac{1}{a} \mathbf{g}_2^\top \delta(\Delta \tilde{\mathbf{w}}_e)$$

Expression (5.174) for the second derivative $\Delta w_{2,11}$ of the displacement increment is substituted into expressions (5.98) and (5.99) for the increment of the bending strain derivative:

$$\begin{aligned}\Delta \varepsilon_b &= \Delta \varepsilon_b^0 + \Delta \varepsilon_b^n \\ \Delta \varepsilon_b^0 &= -\Delta w_{2,11}\end{aligned}\quad (5.180)$$

$$\Delta \varepsilon_b^n = -\frac{1}{2} w_{2,11} \Delta w_{c2,1}^2 \quad (5.181)$$

$$\Delta w_{2,11} = \frac{1}{a^2} \mathbf{h}_2^T \Delta \mathbf{w}_e$$

The variation of the increment of the bending strain derivative $\delta(\Delta \varepsilon_b)$ follows from expressions (5.109) to (5.111):

$$\begin{aligned}\delta(\Delta \varepsilon_b) &= \delta(\Delta \varepsilon_b^0) + \delta(\Delta \varepsilon_b^n) \\ \delta(\Delta \varepsilon_b^0) &= -\delta(\Delta w_{2,11})\end{aligned}\quad (5.182)$$

$$\delta(\Delta \varepsilon_b^n) = -w_{2,11} \Delta w_{c2,1} \delta(\Delta w_{c2,1}) \quad (5.183)$$

$$\delta(\Delta \varepsilon_b^0) = -\frac{1}{a^2} \mathbf{h}_2^T \delta(\Delta \mathbf{w}_e)$$

$$\delta(\Delta \varepsilon_b^n) = -\frac{1}{a} w_{2,11} \Delta w_{c2,1} \mathbf{g}_2^T \Delta \tilde{\mathbf{w}}_e$$

The interpolations of the strain increments and their variations are used to derive the algebraic form of the incremental governing equations.

5.3.4 Error Vector in the Governing Equations

The error vector in the governing equations is determined from the error term δe in (5.115). Analytic integration is applied over the cross-section of the member and the variation of the variables in the axial direction of the member is expressed with the normalised coordinate z in (5.62):

$$\int_{c_e} f(y_1) dv = \int_0^a f(y_1) \left(\int_0^h b(y_2) dy_2 \right) dy_1 = \frac{1}{2} A a \int_{-1}^1 f(z) dz \quad (5.184)$$

$$\int_{c_e} y_2^2 f(y_1) dv = \int_0^a f(y_1) \left(\int_0^h y_2^2 b(y_2) dy_2 \right) dy_1 = \frac{1}{2} J a \int_{-1}^1 f(z) dz \quad (5.185)$$

A area of the cross-section of the member

J moment of inertia of the cross-section of the member

The variation $\delta(\Delta w_{c1,1})$ of the first derivative of the chord displacement increment is substituted from (5.161) into the first term of (5.115). The strain ε_a is computed with expression (5.156).

$$\int_{-1}^1 \frac{1}{2} A E a \varepsilon_a (1 - w_{c1,1}) \delta(\Delta w_{c1,1}) dz = \delta(\Delta \mathbf{w}_e)^T \int_{-1}^1 \frac{1}{2} A E \varepsilon_a (1 - w_{c1,1}) \mathbf{g}_1 dz \quad (5.186)$$

The variation $\delta(\Delta w_{2,11})$ of the second derivative of the displacement increment is substituted from (5.175) into the second term on the right-hand side of (5.115). The bending strain derivative ε_b is computed with expression (5.157).

$$\int_{-1}^1 \frac{1}{2} J E a \varepsilon_b \delta(\Delta w_{r2,11}) dz = \delta(\Delta \mathbf{w}_e)^T \int_{-1}^1 \frac{1}{2a} E J \varepsilon_b \mathbf{h}_2 dz \quad (5.187)$$

The scalar product $\delta(\Delta \mathbf{v})^T \mathbf{p}_a$ in the third term on the right-hand side of (5.115) equals the scale product $\delta(\Delta \mathbf{w})^T \mathbf{p}_c$:

$$\delta(\Delta \mathbf{v})^T \mathbf{p}_a = \delta(\Delta \mathbf{w})^T \mathbf{p}_c \quad (5.188)$$

v total displacement increment referred to reference space

w total displacement increment referred to instant space

p_a load per unit length of the member, referred to reference space

p_a load per unit length of the chord, referred to instant space

The variation of the displacement increment $\Delta \mathbf{w}$ is substituted from (5.170) into the third term on the right-hand side of (5.115):

$$\int_{-1}^1 \frac{1}{2} \mathbf{a} \delta(\Delta \mathbf{w})^T \mathbf{p}_c dz = \delta(\Delta \mathbf{w}_e)^T \int_{-1}^1 \frac{\mathbf{a}}{2} (p_{c1} \mathbf{s}_1 + p_{c2} \mathbf{s}_2) dz \quad (5.189)$$

p_{ck} coordinates of the load **p_c** per unit chord length in instant space

s_k interpolation vectors for displacement increments in (5.170)

In order to assemble the contributions of the elements to the error term, the element displacement vector is transformed to support space with (5.25) and (5.27). The element vector is then related to the system displacement vector **d_s**, as shown in (5.76), with the topology matrix **T_e** of the member:

$$\delta(\Delta \mathbf{w}_e)^T = \delta(\Delta \mathbf{d}_e)^T \mathbf{R}_{esz} = \delta(\Delta \mathbf{d}_s)^T \mathbf{T}_e^T \mathbf{R}_{esz} \quad (5.190)$$

$$\mathbf{R}_{esz} = \mathbf{R}_{exs}^T \mathbf{R}_{exz} \quad (5.191)$$

\mathbf{R}_{esz} element rotation matrix support - instant space

Expressions (5.184) to (5.190) are substituted into the error term (5.115) to obtain the system error vector \mathbf{e}_s :

$$\delta e = \delta(\Delta \mathbf{d}_s)^T \mathbf{e}_s \quad (5.192)$$

$$\mathbf{e}_s = \sum_e \mathbf{T}_e^T \mathbf{R}_{esz} (\mathbf{f}_{pe} - \mathbf{f}_{ae} - \mathbf{f}_{be}) + \mathbf{q}_s + \mathbf{r}_s \quad (5.193)$$

$$\mathbf{f}_{ae} = \int_{-1}^1 \frac{1}{2} AE \varepsilon_a (1 - w_{c1,1}) \mathbf{g}_1 dz \quad (5.194)$$

$$\mathbf{f}_{be} = - \int_{-1}^1 \frac{1}{2a} JE \varepsilon_b \mathbf{h}_2 dz \quad (5.195)$$

$$\mathbf{f}_{pe} = \int_{-1}^1 \frac{a}{2} (p_{c1} \mathbf{s}_1 + p_{c2} \mathbf{s}_2) dz \quad (5.196)$$

\mathbf{f}_{ae} member end forces due to stretching

\mathbf{f}_{be} member end forces due to bending

\mathbf{f}_{pe} member end forces due to member loads

Expressions (5.150) show that the derivatives $w_{c1,1}$ and $w_{c2,1}$ are independent of z . It follows from (5.156) that the axial strain ε_a is also independent of z . The vector \mathbf{f}_{ae} in (5.194) is integrated analytically:

$$\mathbf{f}_{ae} = AE \varepsilon_a (1 - w_{c1,1}) \begin{bmatrix} -1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \quad (5.197)$$

The strain derivative ε_b is given by (5.157), and the vector \mathbf{h}_2 by (5.173). The vector \mathbf{f}_{be} in (5.195) is integrated analytically:

$$\mathbf{f}_{be} = \frac{2JEq}{a^2} \begin{bmatrix} 0 & 0 \\ 3q & 3q \\ 2\hat{a} & \hat{a} \\ 0 & 0 \\ -3q & -3q \\ \hat{a} & 2\hat{a} \end{bmatrix} \begin{bmatrix} \gamma_A \\ \gamma_B \end{bmatrix} \quad (5.198)$$

$$\gamma_A = \beta_A - \theta \quad \gamma_B = \beta_B - \theta \quad q = \frac{\hat{a}}{a}$$

Let the member load vary linearly over the length of the element:

$$\mathbf{p}_a = \mathbf{S}_L \mathbf{p}_{ae} \quad (5.199)$$

$$\begin{bmatrix} p_{a1} \\ p_{a2} \end{bmatrix} = \begin{bmatrix} 0.5(1-z) & 0 & 0.5(1+z) & 0 \\ 0 & 0.5(1-z) & 0 & 0.5(1+z) \end{bmatrix} * \begin{bmatrix} p_{a1A} \\ p_{a2A} \\ p_{a1B} \\ p_{a2B} \end{bmatrix}$$

- \mathbf{p}_a load per unit length at point z referred to reference space
- \mathbf{p}_{ae} member load vector per unit length referred to reference space
- p_{akA} member load per unit length at node A
- p_{akB} member load per unit length at node B

The load coordinates are transformed to instant space with expression (5.15) using the assumptions of the small deformation theory:

$$\mathbf{p}_c = \mathbf{S}_L \mathbf{p}_{ce} \quad (5.200)$$

$$\begin{bmatrix} p_{c1} \\ p_{c2} \end{bmatrix} = \begin{bmatrix} 0.5(1-z) & 0 & 0.5(1+z) & 0 \\ 0 & 0.5(1-z) & 0 & 0.5(1+z) \end{bmatrix} * \begin{bmatrix} p_{c1A} \\ p_{c2A} \\ p_{c1B} \\ p_{c2B} \end{bmatrix}$$

- \mathbf{p}_c load per unit length at point z referred to instant space
- \mathbf{p}_{ce} member load vector per unit length referred to instant space
- p_{ckA} member load per unit length at node A
- p_{ckB} member load per unit length at node B

The coordinates p_{c1} and p_{c2} in (5.196) are given by (5.200) and the vectors \mathbf{s}_1 and \mathbf{s}_2 by (5.170). The vector \mathbf{f}_{pe} in (5.196) is integrated analytically:

$$\mathbf{f}_{pe} = \mathbf{Q}_e \mathbf{p}_{ce} \quad (5.201)$$

$$\mathbf{Q}_e = \frac{a}{60} \begin{bmatrix} 20 & 0 & 10 & 0 \\ 0 & 20+q & 0 & 10-q \\ 0 & 3\hat{a} & 0 & 2\hat{a} \\ 10 & 0 & 20 & 0 \\ 0 & 10-q & 0 & 20+q \\ 0 & -2\hat{a} & 0 & -3\hat{a} \end{bmatrix}$$

$$q = \frac{\hat{a}}{a}$$

Results (5.197), (5.198) and (5.201) are substituted into expression (5.193) for the system error vector \mathbf{e}_s .

5.3.5 Algebraic Incremental Equations

The algebraic incremental governing equations are derived from the incremental governing equation (5.127). The axial coordinate increment is expressed in terms of the normalised coordinate z in (5.62) and the integration over the cross-section of the member is performed analytically. The resulting equation is of the following form:

$$\sum_e \delta(\Delta \mathbf{w}_e)^T \mathbf{K}_e \Delta \mathbf{w}_e = \sum_e \delta(\Delta \mathbf{w}_e)^T \mathbf{q}_e + \delta(\Delta \mathbf{d}_s)^T \Delta \mathbf{q}_s + \delta(\Delta \mathbf{d}_s)^T \Delta \mathbf{r}_s \quad (5.202)$$

The element stiffness matrix \mathbf{K}_e in (5.202) contains components of axial stiffness, which are marked with the subscript a in (5.203), and components of bending stiffness, which are marked with the subscript b. The constant component is denoted by subscript c, the geometric component is denoted by subscript g and nonlinear components are denoted by subscripts m, n and r:

$$\mathbf{K}_e = \mathbf{K}_{ac} + \mathbf{K}_{ag} + \mathbf{K}_{am} + \mathbf{K}_{an} + \mathbf{K}_{ar} + \mathbf{K}_{bc} \quad (5.203)$$

The terms in (5.203) are determined by substituting the following variables into the incremental governing equation (5.127): the derivatives of the chord displacement increments $\Delta w_{c1,1}$ and $\Delta w_{c2,1}$ from (5.160), their variations from (5.161), the second derivative $\Delta w_{2,11}$ of the displacement increment Δw_2 from (5.174) and its variation from (5.175). The products of the differentiation vectors are denoted as follows:

$$\mathbf{G}_{kk} = \mathbf{g}_k \mathbf{g}_k^T \quad (5.204)$$

$$\mathbf{G}_{km} = \mathbf{g}_k \mathbf{g}_m^T + \mathbf{g}_m \mathbf{g}_k^T$$

The component matrices of \mathbf{K}_e are determined by analytic integration, as follows:

– Component \mathbf{K}_{ac} of \mathbf{K}_e :

$$\delta(\Delta \mathbf{w}_e)^T \mathbf{K}_{ac} \Delta \mathbf{w}_e = \int_{C_e} E q^2 \delta(\Delta w_{c1,1}) \Delta w_{c1,1} dv$$

$$\mathbf{K}_{ac} = \int_{-1}^1 \frac{AE}{2a} t^2 \mathbf{G}_{11} dz = \frac{AE}{a} c_0 \mathbf{G}_{11} \quad (5.205)$$

$$c_0 = t^2$$

$$t = 1 - w_{c1,1}$$

– Component \mathbf{K}_{ag} of \mathbf{K}_e :

$$\delta(\Delta \mathbf{w}_e)^T \mathbf{K}_{ag} \Delta \mathbf{w}_e = \int_{C_e} (-\Delta w_{c1,1} \delta(\Delta w_{c1,1}) + t \Delta w_{c2,1} \delta(\Delta w_{c2,1})) s_a dv$$

$$\mathbf{K}_{ag} = \int_{-1}^1 \frac{AE}{2a} \varepsilon_a (-\mathbf{G}_{11} + t \mathbf{G}_{22}) dz = \frac{AE}{a} (c_1 \mathbf{G}_{11} + c_2 \mathbf{G}_{22}) \quad (5.206)$$

$$c_1 = -(w_{c1,1} - 0.5 w_{c1,1}^2 - 0.5 w_{c2,1}^2) \quad c_2 = -t c_1$$

– Component \mathbf{K}_{am} of \mathbf{K}_e :

$$\delta(\Delta \mathbf{w}_e)^T \mathbf{K}_{am} \Delta \mathbf{w}_e = \int_{C_e} \frac{1}{2} E \Delta \bar{w}_{c1,1} (-3t + \Delta \bar{w}_{c1,1}) \Delta w_{c1,1} \delta(\Delta w_{c1,1}) dv$$

$$\mathbf{K}_{am} = \int_{-1}^1 \frac{AE}{4a} \Delta \bar{w}_{c1,1} (-3t + \Delta \bar{w}_{c1,1}) \mathbf{G}_{11} dz = \frac{AE}{a} c_3 \mathbf{G}_{11} \quad (5.207)$$

$$c_3 = \frac{1}{2} \Delta \bar{w}_{c1,1} (-3t + \Delta \bar{w}_{c1,1})$$

– Component \mathbf{K}_{an} of \mathbf{K}_e :

$$\delta(\Delta \mathbf{w}_e)^T \mathbf{K}_{an} \Delta \mathbf{w}_e = \int_{C_e} \frac{1}{2} E t (t - \Delta \bar{w}_{c1,1}) \Delta \bar{w}_{c2,1} (\Delta w_{c1,1} \delta(\Delta w_{c2,1}) + \Delta w_{c2,1} \delta(\Delta w_{c1,1})) dv$$

$$\mathbf{K}_{an} = \int_{-1}^1 \frac{AE}{4a} t (t - \Delta \bar{w}_{c1,1}) \Delta \bar{w}_{c2,1} \mathbf{G}_{12} dz = \frac{AE}{a} c_4 \mathbf{G}_{12} \quad (5.208)$$

$$c_4 = \frac{1}{2} t (t - \Delta \bar{w}_{c1,1}) \Delta \bar{w}_{c2,1}$$

– Component \mathbf{K}_{ar} of \mathbf{K}_e :

$$\delta(\Delta \mathbf{w}_e)^T \mathbf{K}_{ar} \Delta \mathbf{w}_e = \int_{C_e} \frac{1}{2} E t^2 (\Delta \bar{w}_{c1,1} + \Delta \bar{w}_{c2,1}^2) \Delta w_{c2,1} \delta(\Delta w_{c2,1}) dv$$

$$\mathbf{K}_{ar} = \int_{-1}^1 \frac{AE}{4a} t^2 (\Delta \bar{w}_{c1,1} + \Delta \bar{w}_{c2,1}^2) \mathbf{G}_{22} dz = \frac{AE}{a} c_5 \mathbf{G}_{22} \quad (5.209)$$

$$c_5 = \frac{1}{2} t^2 (\Delta \bar{w}_{c1,1} + \Delta \bar{w}_{c2,1}^2)$$

– Component \mathbf{K}_{bc} of \mathbf{K}_e :

$$\delta(\Delta \mathbf{w}_e)^T \mathbf{K}_{bc} \Delta \mathbf{w}_e = \int_{C_e} y^2 E \delta(\Delta w_{2,11}) \Delta w_{2,11} dv$$

$$\mathbf{K}_{bc} = \int_{-1}^1 \frac{JE}{2a^3} \mathbf{h}_2 \mathbf{h}_2^T dz = \frac{JE}{a^3} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & c_6 & c_7 & 0 & -c_6 & c_7 \\ 0 & c_7 & c_8 & 0 & -c_7 & c_9 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -c_6 & -c_7 & 0 & c_6 & -c_7 \\ 0 & c_7 & c_9 & 0 & -c_7 & c_8 \end{bmatrix} \quad (5.210)$$

$$c_6 = 12q^2 \quad c_7 = 6\hat{a}q \quad c_8 = 4\hat{a}^2 \quad c_9 = 2\hat{a}^2 \quad q = \frac{\hat{a}}{a}$$

The variations $\delta(\Delta w_m)$ of the displacement increments are substituted from (5.160) into the second term on the right-hand side of (5.127) to determine the contribution of an element to the load vector of the system:

$$\delta(\Delta \mathbf{w}_e)^T \Delta \mathbf{q}_e = \int_{L_e} \delta(\Delta \mathbf{w})^T \Delta \mathbf{p}_c dy_1$$

$$\Delta \mathbf{q}_e = \int_{-1}^1 \frac{a}{2} (\Delta p_{c1} \mathbf{s}_1 + \Delta p_{c2} \mathbf{s}_2) dz \quad (5.211)$$

The increment of the member load, like the total member load in expression (5.200), varies linearly on the chord of the element:

$$\Delta \mathbf{p}_c = \mathbf{S}_L \Delta \mathbf{p}_{ce} \quad (5.212)$$

$$\begin{bmatrix} \Delta p_{c1} \\ \Delta p_{c2} \end{bmatrix} = \begin{bmatrix} 0.5(1-z) & 0 & 0.5(1+z) & 0 \\ 0 & 0.5(1-z) & 0 & 0.5(1+z) \end{bmatrix} * \begin{bmatrix} \Delta p_{c1A} \\ \Delta p_{c2A} \\ \Delta p_{c1B} \\ \Delta p_{c2B} \end{bmatrix}$$

$\Delta \mathbf{p}_c$ increment of the member load per unit length referred to instant space

Δp_{kA} increment of the member load per unit length at node A

Δp_{kB} increment of the member load per unit length at node B

The element load vector increment (5.211) is obtained by analytic integration:

$$\Delta \mathbf{q}_e = \mathbf{Q}_e \Delta \mathbf{p}_e \quad (5.213)$$

$$\mathbf{Q}_e = \frac{a}{60} \begin{bmatrix} 20 & 0 & 10 & 0 \\ 0 & 20+q & 0 & 10-q \\ 0 & 3\hat{a} & 0 & 2\hat{a} \\ 10 & 0 & 20 & 0 \\ 0 & 10-q & 0 & 20+q \\ 0 & -2\hat{a} & 0 & -3\hat{a} \end{bmatrix} \quad q = \frac{\hat{a}}{a}$$

Substitution of (5.205) to (5.213) into (5.203) leads to the following incremental element stiffness matrix:

$$\mathbf{K}_e = \frac{AE}{a} \mathbf{M}_{e1} + \frac{JE}{a^3} \mathbf{M}_{e2} \quad (5.214)$$

A area of the cross-section of the member

J moment of inertia of the cross-section of the member

a length of the member in the reference configuration

The coefficients of the component matrices \mathbf{M}_{e1} and \mathbf{M}_{e2} are collected from the component matrices (5.205) to (5.215) of the element stiffness matrix \mathbf{K}_e :

$$\mathbf{M}_{e1} = \begin{bmatrix} a_1 & a_2 & 0 & -a_1 & -a_2 & 0 \\ a_2 & a_3 & 0 & -a_2 & -a_3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -a_1 & -a_2 & 0 & a_1 & a_2 & 0 \\ -a_2 & -a_3 & 0 & a_2 & a_3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$a_1 = c_0 + c_1 + c_3$$

$$a_2 = c_4$$

$$a_3 = c_2 + c_5$$

$$\mathbf{M}_{e2} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & a_4 & a_5 & 0 & -a_4 & a_6 \\ 0 & a_5 & a_7 & 0 & -a_5 & a_8 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -a_4 & -a_5 & 0 & a_4 & -a_6 \\ 0 & a_6 & a_8 & 0 & -a_6 & a_7 \end{bmatrix}$$

$$a_4 = c_6$$

$$a_7 = c_8$$

$$a_5 = c_7$$

$$a_8 = c_9$$

$$a_6 = c_7$$

$$(5.215)$$

The variation $\delta(\Delta \mathbf{w}_e)$ of the increment of the element displacement vector in the incremental governing equations (5.202) is replaced by the variation $\delta(\Delta \mathbf{d}_s)$ of the increment of the system displacement vector, using the topology matrix \mathbf{T}_e defined in equation (5.76):

$$\delta(\Delta \mathbf{d}_s)^T \mathbf{K}_s \Delta \mathbf{d}_s = \delta(\Delta \mathbf{d}_s)^T (\mathbf{e}_s + \Delta \mathbf{q}_s + \Delta \mathbf{r}_s) \quad (5.216)$$

$$\mathbf{K}_s = \sum_e \mathbf{T}_e^T \mathbf{R}_{esz} \mathbf{K}_e \mathbf{R}_{esz}^T \mathbf{T}_e \quad (5.217)$$

$$\Delta \mathbf{q}_s = \Delta \mathbf{q}_n + \sum_e \mathbf{T}_e^T \mathbf{R}_{esz} \Delta \mathbf{q}_e \quad (5.218)$$

\mathbf{K}_s system stiffness matrix

\mathbf{K}_e element stiffness matrix, see (5.214)

\mathbf{R}_{esz} element rotation matrix support - instant space

\mathbf{T}_e element topology matrix, see (5.76)

$\Delta \mathbf{d}_s$ increment of the system displacement vector

$\Delta \mathbf{q}_n$ increment of the system vector of the node loads

$\Delta \mathbf{q}_e$ increment of the element load vector, see (5.213)

$\Delta \mathbf{q}_s$ increment of the system load vector

\mathbf{e}_s system error vector, see (5.193)

Unit values of the displacement variables at the nodes of the frame are chosen as variations of the system displacement vector in (5.216). Each variation contributes one equation to the following system of algebraic governing equations for the increments of the displacements and the reactions of a plane frame in a load step:

$$\mathbf{K}_s \Delta \mathbf{d}_s = \mathbf{e}_s + \Delta \mathbf{q}_s + \Delta \mathbf{r}_s \quad (5.219)$$

The governing equations (5.219) are linear. The assembly and solution of these equations are treated in chapter 6.

5.4 STRESS RESULTANTS IN A MEMBER

5.4.1 Governing Equations

The solution of the system governing equations for a plane frame yields the values of the generalised displacements and reactions at the nodes of the frame. A complete solution includes the determination of the displacements and the stress resultants in the members of the frame.

Assume that the deformations in the frame are small and that the members of the frame are subdivided into a suitable number of finite elements, as discussed in section 5.1.5. Assume further that the particular solution for the chord displacements, the relative displacements and the stress resultants in a frame can be determined with the governing equations of linear beam theory.

It has been shown in section 5.2.7 that the total displacements in the members of a frame with small deformations are approximately equal to the sum of the displacements due to the node displacements and the displacements in the fixed-end member subjected to the member load. The stress resultants are computed in a similar manner.

The stress resultants due to the node displacements are determined in section 5.4.2. The particular solutions for a fixed-end member subjected to linearly distributed axial and transverse loads are determined in sections 5.5.3 and 5.4.4. The analysis is performed in the instant configuration of the member. It is assumed that the axis of the member coincides with its chord before the member load is applied. This is an approximation.

5.4.2 Stress Resultants due to Node Displacements

The normal force n in a member is given by expression (5.73), where the axial stress σ_A is given by expressions (5.71), (5.72) and (5.156):

$$n = AE|c_1|\varepsilon_a \quad (5.220)$$

$$|c_1| = \sqrt{(1 + v_{c1,1})^2 + v_{2,1}^2} = \text{instant length / reference length}$$

$$\varepsilon_a = w_{c1,1} - \frac{1}{2}w_{c1,1}^2 - \frac{1}{2}w_{c2,1}^2$$

The bending moment m in the member is given by expression (5.74), where the bending stress derivative is given by (5.71), (5.72) and (5.157):

$$m = JE|c_1|\varepsilon_b \quad (5.221)$$

$$\varepsilon_b = -\frac{d^2w_{r2}}{dy_1^2} = -\frac{\hat{a}}{a^2}((-1+3z)\gamma_A + (1+3z)\gamma_B) \quad (5.222)$$

The shear force q is the derivative of the bending moment with respect to the distance y_1 on the reference axis:

$$q = \frac{dm}{dy_1} \quad (5.223)$$

Expression (5.221) for the moment m is substituted into expression (5.223):

$$q = JE |c_1| \frac{d\varepsilon_b}{dy_1} \quad (5.224)$$

Expression (5.222) is differentiated with respect to y_1 :

$$\frac{d\varepsilon_b}{dy_1} = -\frac{1}{a} \frac{d^3 w_{r2}}{dy_1^3} = -\frac{6\hat{a}}{a^3} (\gamma_A + \gamma_B) \quad (5.225)$$

Expressions (5.224) and (5.225) yield the shear force in the member.

5.4.3 Displacements and Stress Resultants due to Axial Load

Consider a member AB in an instant configuration and let the location of a point on the chord be specified with the normalised coordinate z defined in expression (5.62):

$$z_1 = \frac{1}{2}(1-z)\hat{a} \quad dz_1 = \frac{1}{2}\hat{a}dz \quad -1 \leq z \leq 1 \quad (5.226)$$

\hat{a} length of the member in the instant configuration

Let the axial load on the member be p_1 per unit length, and consider a linear distribution of the axial load:

$$p_1 = \frac{1}{2}(1-z)p_{1A} + \frac{1}{2}(1+z)p_{1B} \quad (5.227)$$

p_1 axial load per unit length of the chord

p_{1A} axial load intensity at node A

p_{1B} axial load intensity at node B

The equilibrium condition for an infinitesimal element of the member is:

$$\frac{dn}{dz_1} = -p_1 \quad (5.228)$$

n axial force in the member at point z

Coordinate z_1 is replaced by the normalised coordinate z in (5.226):

$$\frac{dn}{dz} = -\frac{1}{2}\hat{a}p_1 = -\frac{1}{4}\hat{a}(1-z)p_{1A} - \frac{1}{4}\hat{a}(1+z)p_{1B} \quad (5.229)$$

The axial force is determined by integration:

$$n = \frac{1}{8}\hat{a}(1-z)^2 p_{1A} - \frac{1}{8}\hat{a}(1+z)^2 p_{1B} + c_1 \quad (5.230)$$

The axial force n and the axial displacement w_{c1} are related as follows:

$$n = EA \frac{dw_{c1}}{dz_1} = \frac{2EA}{\hat{a}} \frac{dw_{c1}}{dz} \quad (5.231)$$

E modulus of elasticity of the member

A area of the cross-section of the member

w_{c1} axial displacement

The axial force is substituted from (5.230) into (5.231), which is then solved for the axial displacement:

$$\begin{aligned}\frac{dw_{c1}}{dz} &= \frac{\hat{a}}{2EA} \left(\frac{1}{8} \hat{a} (1-z)^2 p_{1A} - \frac{1}{8} \hat{a} (1+z)^2 p_{1B} + c_1 \right) \\ w_{c1} &= \frac{\hat{a}}{2EA} \left(-\frac{1}{24} \hat{a} (1-z)^3 p_{1A} - \frac{1}{24} \hat{a} (1+z)^3 p_{1B} + c_1 z + c_2 \right)\end{aligned}\quad (5.232)$$

The displacement at both ends of the member is null:

$$\begin{aligned}z = -1: \quad w_{1c} &= -\frac{1}{3} \hat{a} p_{1A} - c_1 + c_2 = 0 \\ z = 1: \quad w_{1c} &= -\frac{1}{3} \hat{a} p_{1B} + c_1 + c_2 = 0\end{aligned}$$

These equations are solved for the integration constants:

$$c_1 = \frac{1}{6} \hat{a} (p_{1B} - p_{1A}) \quad (5.233)$$

$$c_2 = \frac{1}{6} \hat{a} (p_{1A} + p_{1B}) \quad (5.234)$$

The axial displacement follows from (5.232):

$$w_{c1} = \frac{\hat{a}^2}{48EA} \left(-(1-z)^3 p_{1A} - (1+z)^3 p_{1B} + 4(p_{1B} - p_{1A})z + 4(p_{1A} + p_{1B}) \right) \quad (5.235)$$

The axial force follows from (5.230):

$$n = \frac{\hat{a}}{8} (1-z)^2 p_{1A} - \frac{\hat{a}}{8} (1+z)^2 p_{1B} + \frac{\hat{a}}{6} (p_{1B} - p_{1A}) \quad (5.236)$$

5.4.4 Displacement and Stress Resultants due to Transverse Load

Consider a member AB in an instant configuration and let the location of a point on the chord be specified with the normalised coordinate z in expression (5.226). Let the transverse load on the member be p_2 per unit length and consider a linear distribution of the transverse load:

$$p_2 = \frac{1}{2} (1-z) p_{2A} + \frac{1}{2} (1+z) p_{2B} \quad -1 \leq z \leq 1 \quad (5.237)$$

p_{2A} transverse load intensity at node A

p_{2B} transverse load intensity at node B

The equilibrium equation for the shear force q in the member is:

$$\frac{dq}{dz_1} = -p_2 \quad (5.238)$$

q shear force acting on the cross-section of the member

Coordinate z_1 is replaced by the normalised coordinate z in (5.226):

$$\frac{dq}{dz} = -\frac{1}{2} \hat{a} p_2 = -\frac{1}{4} (1-z) \hat{a} p_{2A} - \frac{1}{4} (1+z) \hat{a} p_{2B} \quad (5.239)$$

The shear force is determined by integration:

$$q = \frac{1}{8}(1-z)^2 \hat{a} p_{2A} - \frac{1}{8}(1+z)^2 \hat{a} p_{2B} + c_1 \quad (5.240)$$

c_1 integration constant

The equilibrium equation for the bending moment m in the member is:

$$\frac{dm}{dz_1} = q \quad (5.241)$$

m bending moment acting on the cross-section of the member

Coordinate z_1 is replaced by the normalised coordinate z in (5.226) and the shear force q is substituted from (5.240) into (5.241):

$$\frac{dm}{dz} = \frac{1}{2} \hat{a} q = \frac{1}{16}(1-z)^2 \hat{a}^2 p_{2A} - \frac{1}{16}(1+z)^2 \hat{a}^2 p_{2B} + \frac{1}{2} \hat{a} c_1 \quad (5.242)$$

The bending moment is determined by integration:

$$m = -\frac{1}{48}(1-z)^3 \hat{a}^2 p_{2A} - \frac{1}{48}(1+z)^3 \hat{a}^2 p_{2B} + \frac{1}{2} \hat{a} c_1 z + c_2 \quad (5.243)$$

The bending moment m and the transverse displacement w_{r2} are related, as follows:

$$m = -EJ \frac{d^2 w_{r2}}{dz_1^2} = -\frac{4EJ}{\hat{a}^2} \frac{d^2 w_{r2}}{dz^2} \quad (5.244)$$

J moment of inertia of the cross-section

w_{2r} transverse relative displacement

The bending moment m is substituted from (5.243) into (5.244), which is then solved for the transverse displacement:

$$\begin{aligned} \frac{d^2 w_{r2}}{dz^2} &= \frac{\hat{a}^2}{4EJ} \left(\frac{1}{48}(1-z)^3 \hat{a}^2 p_{2A} + \frac{1}{48}(1+z)^3 \hat{a}^2 p_{2B} - \frac{1}{2} \hat{a} c_1 z - c_2 \right) \\ \frac{dw_{r2}}{dz} &= \frac{\hat{a}^2}{4EJ} \left(-\frac{1}{192}(1-z)^4 \hat{a}^2 p_{2A} + \frac{1}{192}(1+z)^4 \hat{a}^2 p_{2B} - \frac{1}{4} \hat{a} c_1 z^2 - c_2 z + c_3 \right) \\ \frac{dw_{r2}}{dz_1} &= \frac{\hat{a}}{2EJ} \left(-\frac{1}{192}(1-z)^4 \hat{a}^2 p_{2A} + \frac{1}{192}(1+z)^4 \hat{a}^2 p_{2B} - \frac{1}{4} \hat{a} c_1 z^2 - c_2 z + c_3 \right) \\ w_{2r} &= \frac{\hat{a}^2}{4EJ} \left(\frac{1}{960}(1-z)^5 \hat{a}^2 p_{2A} + \frac{1}{960}(1+z)^5 \hat{a}^2 p_{2B} - \frac{1}{12} \hat{a} c_1 z^3 - \frac{1}{2} c_2 z^2 + c_3 z + c_4 \right) \end{aligned} \quad (5.245)$$

The boundary conditions are used to determine the integration constants. Both the displacements and the derivatives of the displacements are null at both ends of the member:

$$z = -1: \quad w_{r2} = \frac{\hat{a}^2}{4EJ} \left(\frac{1}{30} \hat{a}^2 p_{2A} + \frac{1}{12} \hat{a} c_1 - \frac{1}{2} c_2 - c_3 + c_4 \right) = 0 \quad (5.246)$$

$$z = 1: \quad w_{r2} = \frac{\hat{a}^2}{4EJ} \left(\frac{1}{30} \hat{a}^2 p_{2B} - \frac{1}{12} \hat{a} c_1 - \frac{1}{2} c_2 + c_3 + c_4 \right) = 0 \quad (5.247)$$

$$z = -1: \quad \frac{dw_{r2}}{dz} = \frac{\hat{a}^2}{4EJ} \left(-\frac{1}{12} \hat{a}^2 p_{2A} - \frac{1}{4} \hat{a} c_1 + c_2 + c_3 \right) = 0 \quad (5.248)$$

$$z = 1: \quad \frac{dw_{r2}}{dz} = \frac{\hat{a}^2}{4EJ} \left(\frac{1}{12} \hat{a}^2 p_{2B} - \frac{1}{4} \hat{a} c_1 - c_2 + c_3 \right) = 0 \quad (5.249)$$

Equations (5.246) to (5.249) are solved for the constants c_1 to c_4 :

$$c_1 = \frac{3}{20} \hat{a} (p_{2B} - p_{2A})$$

$$c_2 = \frac{1}{24} \hat{a}^2 (p_{2A} + p_{2B})$$

$$c_3 = -\frac{1}{240} \hat{a}^2 (p_{2B} - p_{2A})$$

$$c_4 = \frac{1}{240} \hat{a}^2 (p_{2A} + p_{2B})$$

The relative bending displacement w_{r2} of points on the axis of the member follows by substitution of the integration constants into (5.245):

$$w_{r2} = \frac{\hat{a}^4}{960EJ} \left(\frac{1}{4} (1-z)^5 p_{2A} + \frac{1}{4} (1+z)^5 p_{2B} - z(1+3z^2)(p_{2B} - p_{2A}) + (1-5z^2)(p_{2A} + p_{2B}) \right) \quad (5.250)$$

The bending moment acting on the cross-sections of the member follows by substitution of the integration constants into (5.243):

$$m = \frac{\hat{a}^2}{240} \left(-5(1-z)^3 p_{1A} - 5(1+z)^3 p_{1B} + 18z(p_{2B} - p_{2A}) + 10(p_{2A} + p_{2B}) \right) \quad (5.251)$$

The shear forces acting on the cross-sections of the member follow by substitution of the integration constants into (5.240):

$$q = \frac{\hat{a}}{40} \left(5(1-z)^2 p_{2A} - 5(1+z)^2 p_{2B} + 6(p_{2B} - p_{2A}) \right) \quad (5.252)$$

This completes the particular solution for the fixed-end beam subjected to linearly distributed axial and transverse loads.

6 SOLUTION OF THE GOVERNING EQUATIONS

6.1 STEPWISE SOLUTION PROCEDURE

6.1.1 Concept of the Numerical Algorithm

The solution of the nonlinear governing equations for plane trusses and frames is treated as an initial value problem. It is assumed that the geometry and topology of the reference configuration are known, as well as the properties of the cross-sections of the members and their modulus of elasticity. The displacements, strains and stresses in the reference configuration are null.

The loading of the structure is specified with a prescribed force pattern and a prescribed displacement pattern. It is assumed that the loading in a particular instant configuration of the structure consists of the prescribed force and displacement patterns, scaled with the load factor for that configuration. The displacements and reactions of the structure in the instant configurations are to be determined.

The curve relating a norm of the generalised displacements of the nodes of the structure with a norm of the generalised forces acting on the structure is called a load path. Each point of the load path corresponds to an instant configuration of the structure. The displacement and force norms in the reference configuration are null, so that the load curve starts at the origin of the displacement-load diagram. Figure 6.1 shows a typical load path.

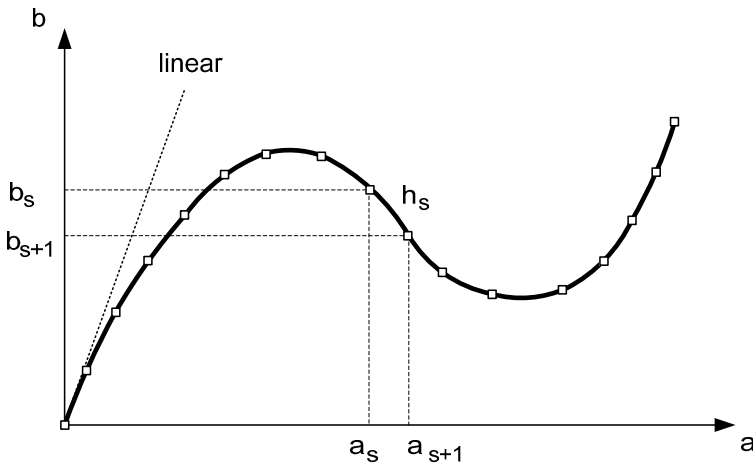


Figure 6.1 Normalised displacement - force diagram (load path)

The load path of a structure is subdivided into arc increments. Each arc increment corresponds to a step in the step-wise solution procedure for the initial value problem. The load steps are computed in sequence. At the beginning of the first load step, the displacements and forces of the reference configuration are known. The first trial solution for the first load step is computed with a prescribed load factor increment. This trial solution for the displacement and force increments is improved by iteration which accounts for the nonlinearity in the behaviour of the structure. After convergence of the iteration, the displacements and forces in the structure at the

beginning of the second step are set equal to the displacement and forces at the end of the first step. The procedure is repeated for the second and all succeeding steps. The procedure terminates when the load factor reaches or exceeds 1.0.

In the derivation of the numerical solution procedure, it is assumed that the displacements and forces in the instant configuration at the beginning of the current load step are known. The displacement and force increments in the load step are computed by solving the linearised incremental governing equations. The increments are added to the displacements and forces at the beginning of the load step to yield the displacements and forces in the trial state. If the norm of the error vector for the trial state is less than a prescribed limit, the displacements and forces of the trial state are considered to be the solution for the state of the structure at the end of the load step.

If the norm of the error vector exceeds the limit, the trial solution is improved by iteration. The displacements and forces of the trial solution are used to approximate the nonlinear terms of the incremental governing equations with linear terms, as described in chapters 4 and 5. The revised governing equations are then used to compute an improved trial solution for the step. The error norm of the revised error vector is used to determine whether the iteration in the load step must be continued or can be terminated.

The load factor increment for a load step is not known a priori. Only the load factor increment for the first load step is prescribed. The load factor increments for the load steps are determined as part of the solution for the load step. If the structure is not near a limit configuration, the value of the load step increment is determined by the constant arc increment method: the length of the chord of the current arc increment is set equal to the length of the chord of the arc increment in the first load step. If the structure is near a limit point, the limit configuration is determined by means of the limit analysis described in chapter 7.

6.1.2 Normalised Load Paths

Norms: The displacement and force norms used to construct a load path can be defined in different ways. If only the free displacements and the applied loads in the system vectors are included in the norms, the norms and the load path are called special. If all displacements and all forces in the system vectors are included in the norms, the norms and the load path are called general.

The special load path is used to show the relationship between the applied loads and the system displacements. The general load path is used to control the flow of the computations in the load steps. Since the general force norm includes the reaction forces, the error in the reactions is taken into account when the decision is made to continue or to terminate the iteration in a load step. It will be shown by means of an example in section 6.1.3 that the tangent to the load path at a limit point is not necessarily parallel to the displacement axis.

In the mathematical definition, the sign of a norm is always positive. A vector \mathbf{x} and the corresponding negative vector $-\mathbf{x}$ therefore have the same norm. It is shown in section 6.1.3 that this property can lead to discontinuities of the first derivative of the load curve if unmodified norms are used in the load diagram. In order to retain the

continuity of the first derivative, which is essential for the numerical algorithms, a sign \odot is associated with the norms. In the first step of the load path, the sign of both norms is by definition positive. In the succeeding steps, the sign of a norm changes if the norm assumes the value null at a point in this step.

For the definition of the special and general norms, the system displacement vector \mathbf{d}_s is reordered so that it contains first the free node displacement coordinates and then the prescribed node displacement coordinates. The system load vector \mathbf{q}_s and the system reaction vector \mathbf{r}_s are arranged in the same order as \mathbf{d}_s . The sum of the system load and reaction vectors is called the system force vector \mathbf{p}_s :

$$\mathbf{p}_s = \mathbf{q}_s + \mathbf{r}_s$$

$$\mathbf{d}_s = \begin{bmatrix} \mathbf{d}_{s1} \\ \mathbf{d}_{s2} \end{bmatrix} \quad \mathbf{q}_s = \begin{bmatrix} \mathbf{q}_{s1} \\ \mathbf{q}_{s2} \end{bmatrix} \quad \mathbf{r}_s = \begin{bmatrix} \mathbf{0} \\ \mathbf{r}_{s2} \end{bmatrix} \quad \mathbf{p}_s = \begin{bmatrix} \mathbf{p}_{s1} \\ \mathbf{p}_{s2} \end{bmatrix} \quad (6.1)$$

\mathbf{p}_s system force vector
 \mathbf{q}_s system load vector
 \mathbf{r}_s system reaction vector

General displacement norm: The general displacement norm a is computed for the system displacement vector \mathbf{d}_s . If the structure is a truss, the system displacement vector contains only displacements and no rotations. The scaled length of the system displacement vector is used as the norm. The displacement calibration factor μ is necessary because the displacements and forces have different dimensions. The calibration factor is used to make the load path independent of the chosen units of measurement.

$$a = \odot \mu \sqrt{\mathbf{d}_s^T \mathbf{d}_s} \quad (6.2)$$

a general displacement norm
 \odot sign (+ or - as determined by the preceding load path)
 μ displacement calibration factor
 \mathbf{d}_s system displacement vector

If the structure is a frame, the system displacement vector contains rotations whose dimension differs from that of the displacements. The rotations are therefore scaled with a factor that is a fraction of the average length of the members of the frame. The displacement norm is then computed with (6.3) instead of (6.2):

$$a = \odot \mu \sqrt{\sum_i d_i^2 + g^2 \sum_k \beta_k^2} \quad (6.3)$$

d_i node displacement coordinate
 β_k node rotation
 g rotation scale factor

General force norm: The general force norm is computed for the system force vector \mathbf{p}_s . If the structure is a truss, the system force vector contains only forces and no moments. The length of the system force vector is used as the norm:

$$b = \ominus \sqrt{\mathbf{p}_s^T \mathbf{p}_s} \quad (6.4)$$

b force norm

\ominus sign (+ or – as determined by the preceding load path)

If the structure is a frame, the system force vector contains moments whose dimension differs from that of the forces. The moments are therefore scaled with a factor that is the inverse of a fraction of the average length of the members of the frame. The general force norm is then computed with (6.5) instead of (6.4):

$$b = \ominus \sqrt{\sum_i p_i^2 + \frac{1}{g^2} \sum_k m_k^2} \quad (6.5)$$

The calibration factor μ is chosen so that the slope of the curve for the first step of the analysis has a specified value m :

$$\mu = \frac{1}{m} \sqrt{\frac{\mathbf{p}_t^T \mathbf{p}_t}{\mathbf{d}_t^T \mathbf{d}_t}} \quad (6.6)$$

\mathbf{p}_t normalised force computed by linear analysis for the load pattern

\mathbf{d}_t normalised displacement computed by linear analysis for the load pattern

Special norms: The special displacement norm for trusses is defined with the free system displacement vector \mathbf{d}_{s1} , as follows:

$$\bar{a} = \ominus \mu \sqrt{\mathbf{d}_{s1}^T \mathbf{d}_{s1}} \quad (6.7)$$

\bar{a} special displacement norm

\mathbf{d}_{s1} free system displacement vector

\ominus sign (+ or – as determined by the preceding load path)

The special displacement norm for frames is given by:

$$\bar{a} = \ominus \mu \sqrt{\sum_i d_i^2 + g^2 \sum_k \beta_k^2} \quad (6.8)$$

d_i node displacement coordinate in system vector \mathbf{d}_{s1}

β_k node rotation in system vector \mathbf{d}_{s1}

The special force norm is defined with the system load vector \mathbf{q}_s . The special force norm for trusses is given by:

$$\bar{b} = \ominus \sqrt{\mathbf{q}_s^T \mathbf{q}_s} \quad (6.9)$$

\bar{b} special force norm

The special displacement norm for frames is given by:

$$\bar{b} = \ominus \sqrt{\sum_i p_i^2 + \frac{1}{g^2} \sum_k m_k^2} \quad (6.10)$$

p_i node load coordinate in system vector \mathbf{q}_s

m_k node rotation coordinate in system vector \mathbf{q}_s

Chord length: Each configuration of the structure is mapped to a point on the load path by computing its displacement and force norms. If the start or end configuration of a load step is mapped to the load path, that point is called a station. The segment of the curve between two successive stations is called an arc increment of the load path. The length of an arc increment is approximated by the length of the chord connecting its stations:

$$h_s = \sqrt{(a_{s+1} - a_s)^2 + (b_{s+1} - b_s)^2} \quad (6.11)$$

h_s chord length between stations s and $s + 1$

Bifurcation: Each step of the solution of the initial value problem is mapped to an arc of the load path. At a bifurcation point, the displacement and force increments in the step following the limit point are not unique. The load path following a bifurcation point is therefore not unique. Each of the possible load paths is represented by its own curve in the normalised displacement - force diagram. The following example shows load paths branching after a bifurcation.

6.1.3 Load Paths of a 2-Bar Truss

General path: Consider the 2-bar truss treated in section 2.1. Let the load P_1 on the truss be null. The displacements and the reactions due to the load P_2 are determined by linear elastic analysis:

$$u_2 = \frac{h_0}{2s} P_2 \quad s := AE \left(\frac{h_0}{L_0} \right)^3$$

$$R_{1A} = -R_{1B} = \frac{m}{2} P_2$$

$$R_{2A} = R_{2B} = \frac{1}{2} P_2$$

The displacement calibration factor is computed with (6.6) for a slope of 5:

$$\mu = \frac{1}{5} \sqrt{\frac{P_2^2 + R_{1A}^2 + R_{1B}^2 + R_{2A}^2 + R_{2B}^2}{u_2^2}} = \frac{s}{5 h_0} \sqrt{2(3 + m^2)}$$

The load P_2 and the reactions corresponding to a displacement u_2 in solution 1 of the nonlinear theory are given by (2.13), (2.14), (2.20), (2.24) and (2.25):

$$P_2 = s w_2 (1 + w_2)(2 + w_2)$$

$$R_{1B} = -R_{1A} = \frac{s}{2} m w_2 (2 + w_2)$$

$$R_{2A} = R_{2B} = \frac{s}{2} w_2 (1 + w_2)(2 + w_2)$$

The general displacement norm for solution 1 is computed with (6.2):

$$a_1 = \odot \mu \sqrt{\mathbf{d}_s^T \mathbf{d}_s} = \odot \mu h_0 w_2 = \odot \frac{s}{5} w_2 \sqrt{2(3 + m^2)}$$

The general force norm for solution 1 is computed with (6.4):

$$b_1 = \odot \sqrt{P_2^2 + R_{1A}^2 + R_{1B}^2 + R_{2A}^2 + R_{2B}^2} = \odot s w_2 (2 + w_2) \sqrt{\frac{3}{2}(1 + w_2)^2 + \frac{1}{2} m^2}$$

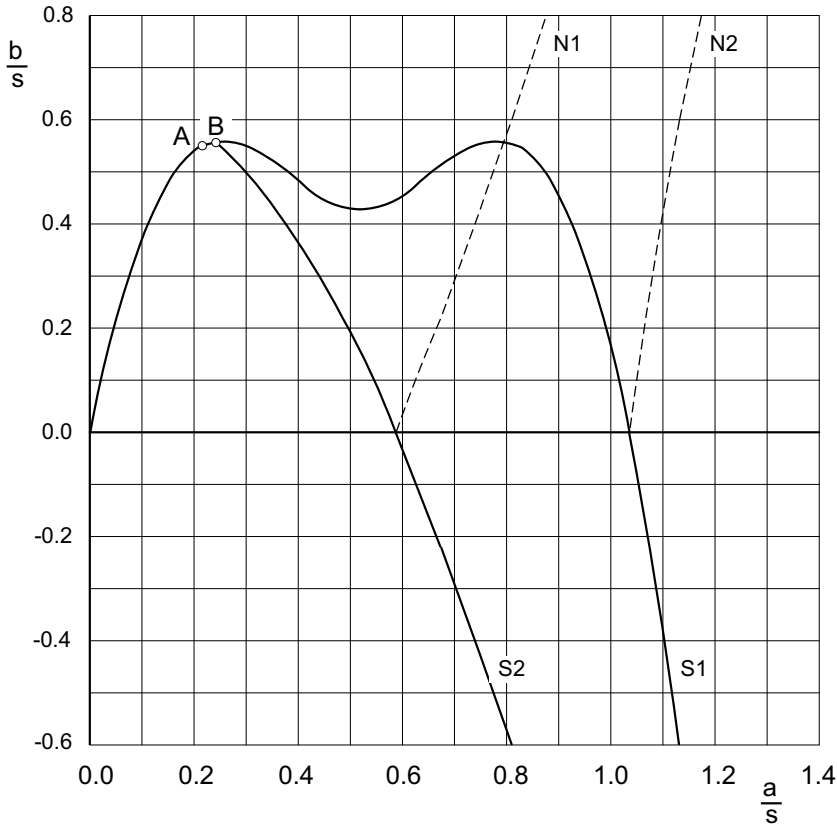


Figure 6.2 General load path of a 2-bar-truss with aspect ratio $m = 0.6$

A	snap-through point	B	bifurcation point
S1	solution 1	S2	solution 2
N1	positive norm for S1	N2	positive norm for S2

Figure 6.2 shows the general load path of solution 1 for a truss with aspect ratio $m = 0.60$. The sign of the force norm changes for the displacement $w_2 = -2.00$ which corresponds to point $(a_1, b_1) = (1.037s, 0.0)$. The snap-through displacement $w_2 = -0.422650$ in (2.21) corresponds to point A with coordinates $(a_1, b_1) = (0.2191s, 0.5497s)$ on the load path.

Since the shape factor m of the truss satisfies condition (2.38), the load path has a bifurcation point. The bifurcation displacement $w_2 = -0.470850$ is computed with (2.40) setting $m = 0.60$. The coordinates of the corresponding point B on the load path are $(a_1, b_1) = (a_2, b_2) = (0.2441s, 0.5577s)$.

After the bifurcation point, the load path can follow either solution 1 or solution 2. The displacement u_1 for solution 2 is given by (2.41):

$$w_1^2 = 1 - 2m^2 - (1 + w_2)^2 \quad \text{for } w_2 \leq -1 + \sqrt{1 - 2m^2} = -0.470850$$

The general displacement norm for solution 2 is computed with (6.2):

$$a_2 = \ominus \mu h_0 \sqrt{w_1^2 + w_2^2} = \ominus \mu h_0 \sqrt{-2(m^2 + w_2)} = \ominus \frac{2s}{5} \sqrt{-(m^2 + w_2)(m^2 + 3)}$$

The load P_2 corresponding to displacement u_2 in solution 2 is computed with (2.29). The reactions are computed with (2.32) to (2.35):

$$\begin{aligned} P_2 &= -2sm^2(1+w_2) \\ R_{1A} &= -R_{1B} = sm(m+w_1)(m-w_1) \\ R_{2A} &= sm(1+w_2)(m-w_1) \\ R_{2B} &= sm(1+w_2)(m+w_1) \end{aligned}$$

The general force norm for solution 2 is computed with (6.4) using w_1^2 from (2.41):

$$b_2 = \ominus \sqrt{P_2^2 + R_{1A}^2 + R_{1B}^2 + R_{2A}^2 + R_{2B}^2} = \ominus sm \sqrt{2(1+w_2)^2(3m^2 + w_1^2) + 2(m^2 - w_1^2)^2}$$

The load path for solution 2 is shown in figure 6.2. It cuts the load path for solution 1 in the bifurcation point B.

Special path: The displacement calibration factor is computed for the load and the displacement at the apex. Contrary to the procedure for general paths, the reactions are not included in the determination of the calibration factor for special paths.

$$\mu = \frac{1}{5} \sqrt{\frac{P_2^2}{u_2^2}} = \frac{2s}{5h_0}$$

The special displacement norm for solution 1 is computed with (6.7):

$$\bar{a}_1 = \ominus \mu \sqrt{\mathbf{d}_{s1}^T \mathbf{d}_{s1}} = \ominus \mu h_0 w_2 = \ominus \frac{2}{5} s w_2$$

The special force displacement norm is computed with (6.9):

$$\bar{b}_1 = \ominus \sqrt{\mathbf{q}_s^T \mathbf{q}_s} = \ominus P_2 = \ominus s w_2 (1+w_2)(2+w_2)$$

The special load path for solution 1 is shown in figure 6.3 by curve S1. The two snap-through points on the curve are marked B and E. Their normalised coordinates are given by (2.21) and (2.22). The corresponding coordinates on the load path are (a,b) = (0.1691s, 0.3849s) and (0.6309, -0.3849s).

Since the shape factor m of the truss satisfies condition (2.38), the load path also has two bifurcation points. These are marked C and D in figure 6.3. Their normalised coordinates $(w_2, p_2) = (-0.4709, 0.3810)$ and $(-1.5292, -0.3810)$ are computed with (2.39) and (2.40). Their coordinates on the load path are (a,b) = (0.1883s, 0.3810s) and (0.6117s, -0.3810s).

After bifurcation point C, the load path can follow either solution 1 or solution 2. The displacement u_1 for solution 2 is given by (2.41). The special displacement norm for solution 2 is computed with (6.7):

$$\bar{a}_2 = \ominus \mu h_0 \sqrt{w_1^2 + w_2^2} = \ominus \mu h_0 \sqrt{-2(m^2 + w_2)} = \ominus \frac{2s}{5} \sqrt{-2(m^2 + w_2)}$$

The special force norm for solution 2 is computed with (6.9):

$$\bar{b}_2 = \ominus \sqrt{P_2^2} = \ominus 2sm^2(1+w_2)$$

The load path for solution 2 is shown in figure 6.3 by curve S2. It cuts the load path for solution 1 in the bifurcation points C and D. Curves N1 and N2 in figure 6.3 show the load path if the positive sign is used for all norms.

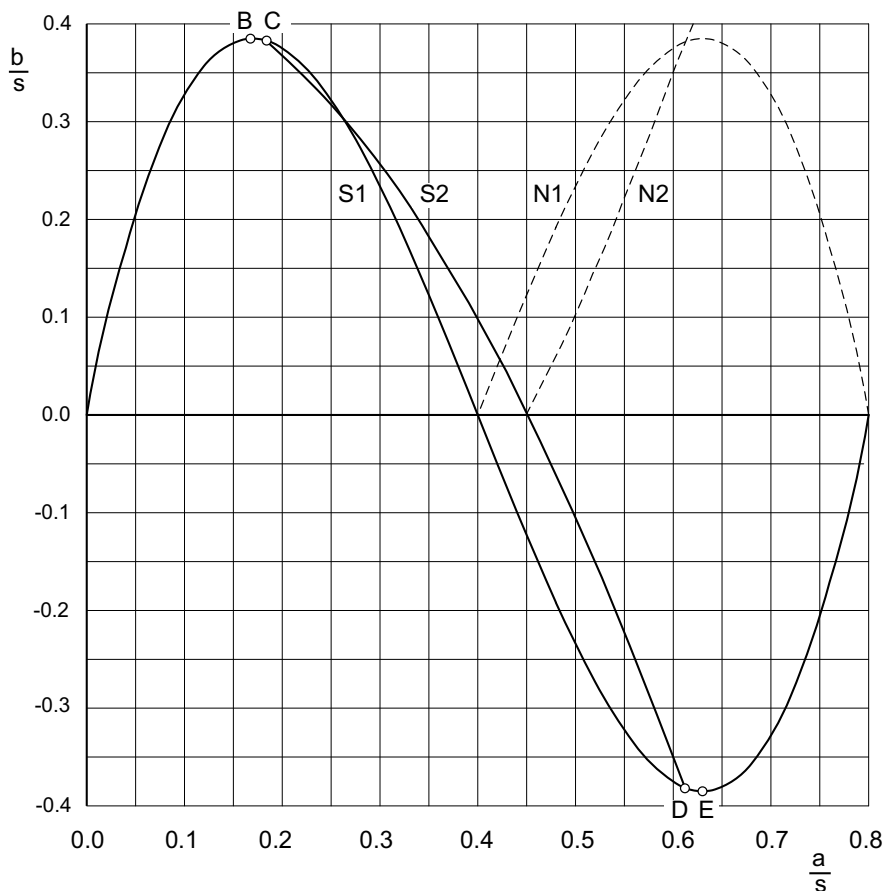


Figure 6.3 Special load paths of a 2-bar truss with aspect ratio $m = 0.60$

B,E	snap-through points	C,D	bifurcation points
S1	solution 1	S2	solution 2
N1	positive norm for S1	N2	positive norm for S2

The tangents to load path S1 at the snap-through points B and E are parallel to the axis of the displacement norm. Due to the singularity, an infinitesimal displacement Δw_2 can take place while w_1 , p_1 and p_2 remain constant. The change in w_2 leads to a change in a_1 , but not in b_1 .

At the bifurcation point C, the tangent to the load path of solution 2 is not parallel to the axis of the displacement norm. The singularity at point C is due to an infinitesimal displacement increment Δw_1 while w_2 , p_1 and p_2 remain constant. The change in displacement w_1 does not change the norms \bar{a}_2 or \bar{b}_2 .

The example demonstrates that the tangent to a load path at a singular point need not be parallel to the axis of the displacement norm. The two-dimensional load path is not capable of reflecting all properties of the multi-dimensional stiffness matrix, which is singular at C.

6.2 ASSEMBLY OF THE INCREMENTAL GOVERNING EQUATIONS

System stiffness matrix: In order to compute the step from station s to station $s+1$ on the load path, the linear incremental governing equations for the structure are assembled and solved. It is assumed that the structure contains both bars of trusses and members of frames. The common designation for bars and members is element. The governing equations (4.98) for trusses and (5.219) for frames are combined:

$$\mathbf{K}_s \Delta \mathbf{d}_s = \mathbf{e}_s + \Delta \mathbf{q}_s + \Delta \mathbf{r}_s \quad (6.12)$$

- \mathbf{K}_s incremental system stiffness matrix
- \mathbf{e}_s system error vector
- $\Delta \mathbf{d}_s$ increment of the system displacement vector
- $\Delta \mathbf{q}_s$ increment of the system load vector
- $\Delta \mathbf{r}_s$ increment of the system reaction vector

The contribution of a bar to matrix \mathbf{K}_s follows from (4.88):

$$d_e \mathbf{K}_s = \mathbf{T}_e^T \mathbf{R}_{esy} \mathbf{K}_e \mathbf{R}_{esy}^T \mathbf{T}_e \quad (6.13)$$

- $d_e \mathbf{K}_s$ contribution of bar e to the incremental system stiffness matrix
- \mathbf{T}_e topology matrix for bar e
- \mathbf{R}_{esy} bar rotation matrix from support space to reference space, see (4.29)
- \mathbf{K}_e incremental bar stiffness matrix, see (4.89)

The contribution of a frame member to matrix \mathbf{K}_s follows from (5.217):

$$d_e \mathbf{K}_s = \mathbf{T}_e^T \mathbf{R}_{esz} \mathbf{K}_e \mathbf{R}_{esz}^T \mathbf{T}_e \quad (6.14)$$

- $d_e \mathbf{K}_s$ contribution of member e to the incremental system stiffness matrix
- \mathbf{T}_e topology matrix for member e
- \mathbf{R}_{esz} member rotation matrix from support space to instant space, see (5.76)
- \mathbf{K}_e incremental element stiffness matrix, see (5.203)

Rotation to support space: The incremental stiffness matrix for a given configuration of the structure is assembled from the contributions of the individual elements. The elements are traversed in arbitrary sequence. For each element, the incremental element stiffness matrix is computed. In the case of bars, the coefficients of \mathbf{K}_e are referred to reference space. In the case of members, the coefficients of \mathbf{K}_e are referred to instant space. In both cases, the matrix containing the element stiffness coefficients in the support spaces of the end nodes of the element is computed by means of the element rotation matrices as shown in (6.13) and (6.14):

$$\mathbf{K}_{es} = \mathbf{R}_e^T \mathbf{K}_e \mathbf{R}_e \quad (6.15)$$

- \mathbf{K}_{es} incremental element stiffness matrix referred to the support spaces

The product (6.15) is computed efficiently by means of the block operations in (6.16) which avoid the multiplication with the null submatrices of the element rotation matrix. The block size is 2 by 2 for the bars due to the dimension of the node rotation matrix (4.12), and 3 by 3 for the frame members due to the dimension of the node rotation matrix (5.21), which accounts for the rotation variable at the node.

$$\begin{array}{cc|cc|cc}
 & & \mathbf{K}_e & & & \mathbf{R}_e^T \\
 & & \mathbf{K}_{11} & \mathbf{K}_{12} & \mathbf{R}_{na}^T & \mathbf{0}^T \\
 & & \mathbf{K}_{21} & \mathbf{K}_{22} & \mathbf{0}^T & \mathbf{R}_{nb}^T \\
 \mathbf{R}_{na} & \mathbf{0} & \mathbf{R}_{na} \mathbf{K}_{11} & \mathbf{R}_{na} \mathbf{K}_{12} & \mathbf{R}_{na} \mathbf{K}_{11} \mathbf{R}_{na}^T & \mathbf{R}_{na} \mathbf{K}_{12} \mathbf{R}_{nb}^T \\
 \mathbf{0} & \mathbf{R}_{nb} & \mathbf{R}_{nb} \mathbf{K}_{21} & \mathbf{R}_{nb} \mathbf{K}_{22} & \mathbf{R}_{nb} \mathbf{K}_{21} \mathbf{R}_{na}^T & \mathbf{R}_{nb} \mathbf{K}_{22} \mathbf{R}_{nb}^T \\
 \mathbf{R}_e & & & & & \mathbf{K}_{es}
 \end{array} \tag{6.16}$$

Addition to the system matrix: The rotated incremental stiffness matrix \mathbf{K}_{es} is used to compute the contribution of the element to the incremental system stiffness matrix by determining the product $\mathbf{T}_e^T \mathbf{K}_{es} \mathbf{T}_e$ in (6.13) and (6.14). Figure 6.4 shows a bar e whose end nodes have the system displacement coordinates d_i, d_k, d_m and d_s . Variable d_i is associated with row i and column i of the system matrix. The product $\mathbf{T}_e^T \mathbf{K}_{es} \mathbf{T}_e$ is evaluated explicitly in the figure. It can be seen that this multiplication is inefficient because most of the coefficients of \mathbf{T}_e are null and the remaining coefficients are 1.

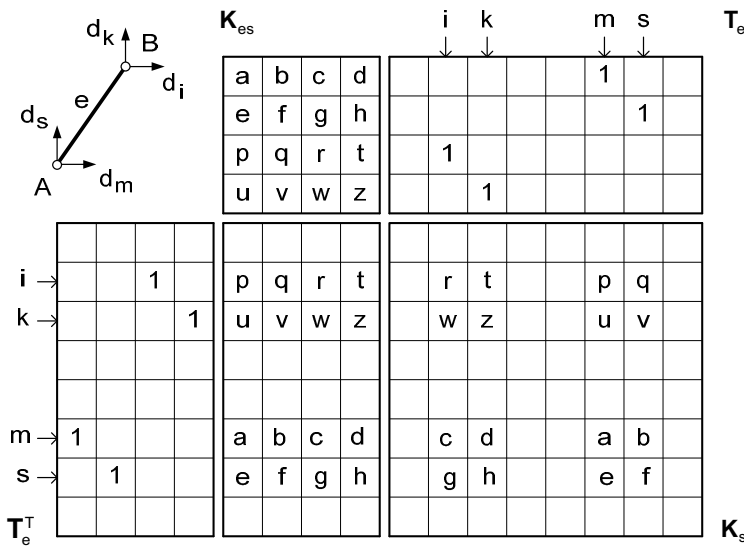


Figure 6.4 Contribution of bar e to the system stiffness matrix

The multiplication in figure 6.4 is avoided by mapping of the local index j of the element displacement coordinate to the system index n_j of the same variable. The mapping is defined in an index table, as follows:

$$\begin{array}{|c|c|c|c|}
 \hline
 n_0 & n_1 & n_2 & n_3 \\
 \hline
 \end{array} \tag{6.17}$$

$$j = 0 \quad 1 \quad 2 \quad 3$$

n_j system index of the displacement coordinate with index j in the element vector

The index table for the element in figure 6.4 is

m	s	i	k
0	1	2	3

(6.18)

Consider the mapping $p: \{0,1,2,3\} \rightarrow \{n_0, n_1, n_2, n_3\}$ of the index table (6.18). Let the coefficients of the rotated matrix \mathbf{K}_{es} be renamed k_{im} with $i, m \in \{0,1,2,3\}$ and let the coefficients of the system matrix be called s_{jn} . The following rule assures that the coefficients of \mathbf{K}_{es} are added at the same location in the system matrix \mathbf{K}_s as in the multiplication in figure 6.4:

$$\begin{aligned}
 p: \{0,1,2,3\} &\rightarrow \{n_0, n_1, n_2, n_3\} \\
 j &= p(i) \quad n = p(m) \\
 s_{jn} &\leftarrow s_{jn} + k_{im}
 \end{aligned}
 \tag{6.19}$$

The mapping (6.19) for the example in figure 6.4 is illustrated in figure 6.5.

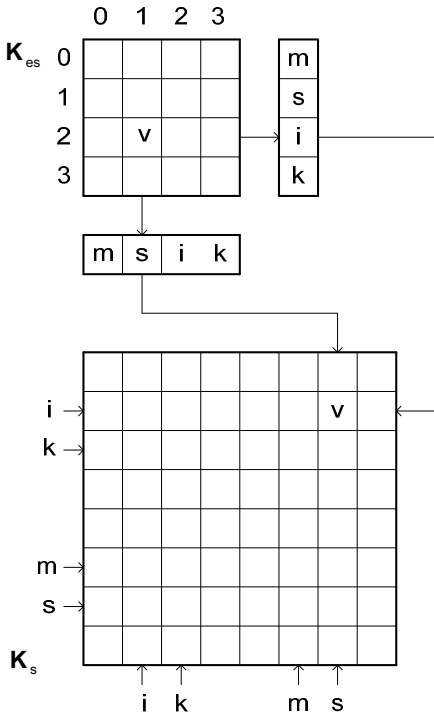


Figure 6.5 Contribution of element stiffness matrix \mathbf{K}_{es} to system matrix \mathbf{K}_s

System load vector: The system load vector is computed by assembling the loads applied at nodes of the structure and the loads applied to members of a frame. The coordinates of loads applied at a node are prescribed in the support space. Moment loads can only be prescribed at nodes where at least one frame member is incident. The system index of a node load coordinate is known once the system displacement

variables have been numbered. The prescribed load coordinate is added directly to the system load vector:

$$q_j \leftarrow q_j + \bar{q}_j \quad (6.20)$$

q_j coefficient of the system load vector

\bar{q}_j prescribed load at a node

Bars of trusses do not carry an element load. The contribution of a frame member to the system load vector follows from (5.218):

$$d_e \mathbf{q}_s = \mathbf{T}_e^T \mathbf{R}_{esz} \mathbf{q}_e \quad (6.21)$$

$d_e \mathbf{q}_s$ contribution of member e to the system load vector

\mathbf{T}_e topology matrix for bar e

\mathbf{R}_{esz} element rotation matrix from support space to instant space, see (5.76)

\mathbf{q}_e member load, see (5.201) and (5.213)

Rotation to support space: The coordinates of the member load vector are referred to instant space. The vector that contains the element load coefficients in the support spaces of the nodes of the member is computed by means of the element rotation matrix, as shown in (6.21):

$$\mathbf{q}_{es} = \mathbf{R}_{esz} \mathbf{q}_e \quad (6.22)$$

\mathbf{q}_{es} element load vector referred to the support spaces

The product (6.22) is computed efficiently by means of the block operations in (6.23) which avoid multiplication with the null submatrices of the element rotation matrix. The block size is 3 by 3 in the matrix and 3 by 1 in the vector.

$$\begin{array}{c}
 \mathbf{q}_e \\
 \begin{array}{|c|} \hline \mathbf{q}_1 \\ \hline \mathbf{q}_2 \\ \hline \end{array} \\
 \begin{array}{|c|c|c|} \hline \mathbf{R}_a & \mathbf{0} & \mathbf{R}_a \mathbf{q}_1 \\ \hline \mathbf{0} & \mathbf{R}_b & \mathbf{R}_b \mathbf{q}_2 \\ \hline \end{array} \\
 \mathbf{R}_{esz} \qquad \qquad \mathbf{q}_{es}
 \end{array} \quad (6.23)$$

Addition to the system load vector: The element load vector is added to the system load vector by means of the index table (6.17). Let the coefficient in the system load vector be denoted by s_n and the coefficient in the element load vector by q_k . The coefficient q_k is added to the system load vector with the following rule:

$$p: \{0,1,2,3\} \rightarrow \{n_0, n_1, n_2, n_3\}$$

$$n = p(k)$$

$$s_n \leftarrow s_n + q_k \quad (6.24)$$

6.3 TREATMENT OF THE ERROR VECTOR

The incremental governing equation (6.12) contains the error vector \mathbf{e}_s due to the unbalanced forces in state s of the frame. The forces and moments in vector \mathbf{e}_s are in equilibrium if they act on the instant configuration of the structure for which the error vector is determined. These forces and moments cause displacements, reactions and stress resultants in the frame. It will be shown that the unbalanced forces do not contribute to the load on the frame.

Consider the exact governing equations for the total displacements and reactions of the frame in state s :

$$\mathbf{K}_e^{(s)} \mathbf{d}_e^{(s)} = \mathbf{q}_e^{(s)} + \mathbf{r}_e^{(s)} \quad (6.25)$$

$\mathbf{K}_e^{(s)}$ exact total stiffness matrix for state s

$\mathbf{d}_e^{(s)}$ exact total displacement in state s

$\mathbf{q}_e^{(s)}$ exact total load in state s

$\mathbf{r}_e^{(s)}$ exact total reaction in state s

In order to derive the exact secant stiffness matrix for step s , the exact governing equation for the total displacements and reactions in state $s-1$ are also considered:

$$\mathbf{K}_e^{(s-1)} \mathbf{d}_e^{(s-1)} = \mathbf{q}_e^{(s-1)} + \mathbf{r}_e^{(s-1)} \quad (6.26)$$

Equation (6.26) is subtracted from equation (6.25):

$$\mathbf{K}_e^{(s)} \mathbf{d}_e^{(s)} - \mathbf{K}_e^{(s-1)} \mathbf{d}_e^{(s-1)} = \mathbf{q}_e^{(s)} - \mathbf{q}_e^{(s-1)} + \mathbf{r}_e^{(s)} - \mathbf{r}_e^{(s-1)} \quad (6.27)$$

The exact displacement, load and reaction increments for load step s are defined as follows:

$$\Delta \mathbf{d}_e^{(s)} = \mathbf{d}_e^{(s)} - \mathbf{d}_e^{(s-1)} \quad (6.28)$$

$$\Delta \mathbf{q}_e^{(s)} = \mathbf{q}_e^{(s)} - \mathbf{q}_e^{(s-1)} \quad (6.29)$$

$$\Delta \mathbf{r}_e^{(s)} = \mathbf{r}_e^{(s)} - \mathbf{r}_e^{(s-1)} \quad (6.30)$$

$\Delta \mathbf{d}_e^{(s)}$ exact displacement increment in step s

$\Delta \mathbf{q}_e^{(s)}$ exact load increment in step s

$\Delta \mathbf{r}_e^{(s)}$ exact reaction increment in step s

Expressions (6.28) to (6.30) are substituted into equation (6.27):

$$\left(\mathbf{K}_e^{(s)} - \mathbf{K}_e^{(s-1)} \right) \mathbf{d}_e^{(s-1)} + \mathbf{K}_e^{(s)} \Delta \mathbf{d}_e^{(s)} = \Delta \mathbf{q}_e^{(s)} + \Delta \mathbf{r}_e^{(s)} \quad (6.31)$$

Since the difference of the total stiffness matrices in states s and $s-1$ is a function of the displacement increment in step s , the first term in equation (6.31) can be written in the following form:

$$\left(\mathbf{K}_e^{(s)} - \mathbf{K}_e^{(s-1)} \right) \mathbf{d}_e^{(s-1)} = \Delta \mathbf{K}_e^{(s)} \Delta \mathbf{d}_e^{(s)} \quad (6.32)$$

$\Delta \mathbf{K}_e^{(s)}$ auxiliary matrix

Substitution of expression (6.32) into equation (6.31) leads to the exact governing equation for the displacement increments in step s , which contain the exact secant stiffness matrix:

$$\mathbf{K}_{se}^{(s)} \Delta \mathbf{d}_e^{(s)} = \Delta \mathbf{q}_e^{(s)} + \Delta \mathbf{r}_e^{(s)} \quad (6.33)$$

$$\mathbf{K}_{se}^{(s)} = \mathbf{K}_e^{(s)} + \Delta \mathbf{K}_e^{(s)} \quad (6.34)$$

$\mathbf{K}_{se}^{(s)}$ exact secant stiffness matrix

The exact governing equation (6.33) for the displacement and reaction increments is approximated by a governing equation that contains the exact load increment for the load step, but approximations for the secant matrix as well as the displacement and reaction increments:

$$\mathbf{K}_s^{(s)} \Delta \mathbf{d}^{(s)} = \Delta \mathbf{q}_e^{(s)} + \Delta \mathbf{r}^{(s)} \quad (6.35)$$

$\mathbf{K}_s^{(s)}$ approximate secant stiffness matrix for step s

$\Delta \mathbf{d}^{(s)}$ approximate displacement increment for step s

$\Delta \mathbf{r}^{(s)}$ approximate reaction increment for step s

Equation (6.35) is solved for the approximate displacement increment $\Delta \mathbf{d}^{(s)}$ and the approximate reaction increment $\Delta \mathbf{r}^{(s)}$ in load step s. Due to approximation (6.35), the frame is not in exact equilibrium in state s. The unbalanced forces are equal to the difference between the external and the internal forces acting at the nodes of the frame:

$$\mathbf{e}^{(s)} = \mathbf{q}_e^{(s)} + \mathbf{r}^{(s)} - \mathbf{K}_e^{(s)} \mathbf{d}^{(s)} \quad (6.36)$$

$\mathbf{e}^{(s)}$ error vector containing the unbalanced forces in the approximate state s

$\mathbf{d}^{(s)}$ approximate total displacement in state s

$\mathbf{r}^{(s)}$ approximate total reaction in state s

Assume that the unbalanced forces are due to the use of the approximate secant matrix $\mathbf{K}_s^{(s)}$ instead of the exact secant matrix in equation (6.33):

$$\mathbf{K}_s^{(s)} \Delta \mathbf{d}_e^{(s)} = \Delta \mathbf{q}_e^{(s)} + \Delta \mathbf{r}_e^{(s)} + \mathbf{e}^{(s)} \quad (6.37)$$

The exact load increment $\Delta \mathbf{q}_e^{(s)}$ is substituted from expression (6.35) into (6.37):

$$\mathbf{K}_s^{(s)} (\Delta \mathbf{d}_e^{(s)} - \Delta \mathbf{d}^{(s)}) = \mathbf{e}^{(s)} + (\Delta \mathbf{r}_e^{(s)} - \Delta \mathbf{r}^{(s)}) \quad (6.38)$$

Equation (6.38) is solved for the correction $\Delta \mathbf{d}_e^{(s)} - \Delta \mathbf{d}^{(s)}$ in the displacement and the correction $\Delta \mathbf{r}_e^{(s)} - \Delta \mathbf{r}^{(s)}$ in the reaction in state s. The load in state s is not corrected.

6.4 SOLUTION OF THE INCREMENTAL GOVERNING EQUATIONS

Assume that the incremental governing equations (6.12) for a load step have been assembled and that the structure is not near a singular point. Equations (6.12) contain free and prescribed values for the generalised displacements and for the generalised forces. The equations therefore have unknowns on both their left- and their right-hand side. In addition, the incremental stiffness matrix usually has a profile structure. These characteristics are taken into account in the development of the solution procedure for the incremental governing equations.

The equations and variables of the system equations are ordered in such a way that the system displacement vector contains first the free and then the prescribed values. In the computer implementation, this reordering is avoided. The reordering leads to the following block structure of the system equations if the value of any variable x in iteration cycle m of step s is denoted by $x^{(s,m)}$:

$$\begin{bmatrix} \mathbf{K}_{11}^{(s,m)} & \mathbf{K}_{12}^{(s,m)} & \Delta \mathbf{d}_1^{(s,m)} \\ \mathbf{K}_{21}^{(s,m)} & \mathbf{K}_{22}^{(s,m)} & \Delta \mathbf{d}_2^{(s,m)} \end{bmatrix} = \begin{bmatrix} \Delta \mathbf{q}_1^{(s,m)} \\ \Delta \mathbf{q}_2^{(s,m)} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \Delta \mathbf{r}_2^{(s,m)} \end{bmatrix} + \begin{bmatrix} \mathbf{e}_1^{(s)} \\ \mathbf{e}_2^{(s)} \end{bmatrix} \quad (6.39)$$

$\mathbf{K}_{ik}^{(s,m)}$ secant stiffness matrix for iteration cycle m in step s

$\mathbf{e}_k^{(s)}$ error in state s

$\Delta \mathbf{d}_1^{(s,m)}$ free displacement increment in cycle m of step s

$\Delta \mathbf{d}_2^{(s,m)}$ prescribed displacement increment in cycle m of step s

$\Delta \mathbf{q}_k^{(s,m)}$ load increment in cycle m of step s

$\Delta \mathbf{r}_2^{(s,m)}$ reaction increment in cycle m of step s

Let the load pattern \mathbf{q}_t and the displacement pattern \mathbf{d}_{2t} that act on the structure be specified. The fraction λ of the load and displacement patterns, which acts on the structure in a given configuration, is called the load factor of that configuration. The load factor increment specifies the change in the loading during a load step:

$$\Delta \mathbf{q}_1^{(s,m)} = \Delta \lambda^{(s,m)} \mathbf{q}_{1t} \quad (6.40)$$

$$\Delta \mathbf{q}_2^{(s,m)} = \Delta \lambda^{(s,m)} \mathbf{q}_{2t} \quad (6.41)$$

$$\Delta \mathbf{d}_2^{(s,m)} = \Delta \lambda^{(s,m)} \mathbf{d}_{2t} \quad (6.42)$$

$\Delta \lambda^{(s,m)}$ load factor increment in cycle m of step s

The load factor increment for a load step is not known a priori. It is determined by the following iterative procedure, which also yields the displacement and reaction increments for the load step.

(a) **Pattern load solution for iteration m in step s :** The incremental stiffness matrix and the load vector of the structure are computed in each iteration m of step s . The current geometry of the frame members is taken into account in computing the member load vectors. The nonlinear terms are neglected in iteration 0 and are approximated with the results of iteration $m-1$ in the subsequent iterations. The displacements $\mathbf{d}_{1t}^{(s,m)}$ and the reactions $\mathbf{r}_{2t}^{(s,m)}$ for the load pattern $\mathbf{q}_t^{(s,m)}$ and the displacement pattern \mathbf{d}_{2t} are computed with the secant stiffness matrix:

$$\mathbf{K}_s^{(s,m)} \mathbf{d}_t^{(s,m)} = \mathbf{q}_t^{(s,m)} + \mathbf{r}_t^{(s,m)} \quad (6.43)$$

$$\mathbf{d}_{1t}^{(s,m)} = (\mathbf{K}_{11}^{(s,m)})^{-1} (\mathbf{q}_{1t}^{(s,m)} - \mathbf{K}_{12} \mathbf{d}_{2t})$$

$$\mathbf{r}_{2t}^{(s,m)} = \mathbf{K}_{21} \mathbf{d}_{1t}^{(s,m)} + \mathbf{K}_{22} \mathbf{d}_{2t} - \mathbf{q}_{2t}^{(s,m)}$$

$$\begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} \\ \mathbf{K}_{21} & \mathbf{K}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{d}_{1t}^{(s,m)} \\ \mathbf{d}_{2t} \end{bmatrix} = \begin{bmatrix} \mathbf{q}_{1t}^{(s,m)} \\ \mathbf{q}_{2t}^{(s,m)} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{r}_{2t}^{(s,m)} \end{bmatrix}$$

$\mathbf{K}^{(s,m)}$ incremental stiffness matrix for iteration m in step s

$\mathbf{d}_{1t}^{(s,m)}$ free displacements due to the pattern load in iteration m of step s

$\mathbf{r}_{2t}^{(s,m)}$ reactions due to the pattern load in iteration m of step s

The force vector $\mathbf{p}_t^{(s,m)}$ for the total load is the sum of the load and reaction vectors:

$$\mathbf{p}_t^{(s,m)} = \mathbf{q}_t^{(s,m)} + \mathbf{r}_t^{(s,m)} \quad (6.44)$$

$$\begin{bmatrix} \mathbf{p}_{1t}^{(s,m)} \\ \mathbf{p}_{2t}^{(s,m)} \end{bmatrix} = \begin{bmatrix} \mathbf{q}_{1t}^{(s,m)} \\ \mathbf{q}_{2t}^{(s,m)} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{r}_{2t}^{(s,m)} \end{bmatrix}$$

(b) **Incremental solution for iteration m in step s:** The increments (6.40) to (6.42) of the load and of the prescribed displacement are substituted into (6.39). Equations (6.39) are solved for the displacement increment and the reaction increment as functions of the unknown load factor increment $\Delta\lambda$:

$$\mathbf{K}^{(s,m)} \Delta \mathbf{d}^{(s,m)} = \Delta\lambda^{(s,m)} \mathbf{q}_t^{(s)} + \Delta \mathbf{r}^{(s,m)} + \mathbf{e}^{(s,m)} \quad (6.45)$$

$$\begin{bmatrix} \mathbf{K}_{11}^{(s,m)} & \mathbf{K}_{12}^{(s,m)} \\ \mathbf{K}_{21}^{(s,m)} & \mathbf{K}_{22}^{(s,m)} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{d}_1^{(s,m)} \\ \Delta \lambda^{(s,m)} \mathbf{d}_{2t} \end{bmatrix} = \Delta \lambda^{(s,m)} \begin{bmatrix} \mathbf{q}_{1t}^{(s)} \\ \mathbf{q}_{2t}^{(s)} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \Delta \mathbf{r}_2^{(s,0)} \end{bmatrix} + \begin{bmatrix} \mathbf{e}_1^{(s)} \\ \mathbf{e}_2^{(s)} \end{bmatrix}$$

$\Delta\lambda^{(s,m)}$ load factor increment for iteration m in step s

$\Delta \mathbf{d}_1^{(s,m)}$ increment of the free displacements for iteration m in step s

$\Delta \mathbf{r}_2^{(s,m)}$ increment of the reactions for iteration m in step s

$\mathbf{e}_k^{(s)}$ error vector in state s

The first row of equation (6.45) is solved for the free displacement increment; the second row of the equation is solved for the reaction increment in iteration 0:

$$\Delta \mathbf{d}_1^{(s,m)} = \Delta\lambda^{(s,m)} \mathbf{d}_{1t}^{(s)} + \mathbf{f}_1^{(s,m)} \quad (6.46)$$

$$\mathbf{f}_1^{(s,m)} = (\mathbf{K}_{11}^{(s,m)})^{-1} \mathbf{e}_1^{(s,m)} \quad (6.47)$$

$$\Delta \mathbf{r}_2^{(s,m)} = \Delta\lambda^{(s,m)} \mathbf{r}_{2t}^{(s)} + \mathbf{t}_2^{(s,m)} \quad (6.48)$$

$$\mathbf{t}_2^{(s,m)} = \mathbf{K}_{21} \mathbf{f}_1^{(s,m)} - \mathbf{e}_2^{(s)} \quad (6.49)$$

The solution for the displacement increments in equations (6.45) is:

$$\Delta \mathbf{d}^{(s,m)} = \Delta\lambda^{(s,m)} \mathbf{d}_t^{(s)} + \mathbf{f}^{(s,m)} \quad (6.50)$$

$$\begin{bmatrix} \Delta \mathbf{d}_1^{(s,m)} \\ \Delta \mathbf{d}_2^{(s,m)} \end{bmatrix} = \Delta \lambda^{(s,m)} \begin{bmatrix} \mathbf{d}_{1t}^{(s)} \\ \mathbf{d}_{2t} \end{bmatrix} + \begin{bmatrix} \mathbf{f}_1^{(s,m)} \\ \mathbf{0} \end{bmatrix}$$

The error vector is treated as shown in equation (6.38) to find the solution for the load and reaction increments in equations (6.45):

$$\Delta \mathbf{q}^{(s,m)} = \Delta \lambda^{(s,m)} \mathbf{q}_t^{(m)} \quad (6.51)$$

$$\begin{bmatrix} \Delta \mathbf{q}_1^{(s,m)} \\ \Delta \mathbf{q}_2^{(s,m)} \end{bmatrix} = \Delta \lambda^{(s,m)} \begin{bmatrix} \mathbf{q}_{1t}^{(s,m)} \\ \mathbf{q}_{2t}^{(s,m)} \end{bmatrix}$$

$$\Delta \mathbf{r}^{(s,m)} = \Delta \lambda^{(s,m)} \mathbf{r}_t^{(s)} + \mathbf{t}^{(s,m)} \quad (6.52)$$

$$\begin{bmatrix} \mathbf{0} \\ \Delta \mathbf{r}_2^{(s,m)} \end{bmatrix} = \Delta \lambda^{(s,m)} \begin{bmatrix} \mathbf{0} \\ \mathbf{r}_{2t}^{(s,m)} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{t}_2^{(s,m)} \end{bmatrix}$$

The increment of the force vector follows from (6.51) and (6.52):

$$\Delta \mathbf{p}^{(s,m)} = \Delta \lambda^{(s,m)} \mathbf{p}_t^{(s,m)} + \mathbf{t}^{(s,m)} \quad (6.53)$$

$$\begin{bmatrix} \Delta \mathbf{p}_1^{(s,m)} \\ \Delta \mathbf{p}_2^{(s,m)} \end{bmatrix} = \Delta \lambda^{(s,m)} \begin{bmatrix} \mathbf{p}_{1t}^{(s,m)} \\ \mathbf{p}_{2t}^{(s,m)} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{t}_2^{(s,m)} \end{bmatrix}$$

The values of the displacement, load and reaction vectors in the trial state at the end of iteration m in step s can now be determined. The value $\Delta \lambda$ of the load factor increment is computed as shown in section 6.5 and substituted into (6.50) to (6.52) to obtain the incremental vectors $\Delta \mathbf{d}^{(s,m)}$, $\Delta \mathbf{q}^{(s,m)}$ and $\Delta \mathbf{r}^{(s,m)}$. The incremental vectors are added to the state of the structure at the beginning of the step to obtain the trial values for state $s+1$ in iteration cycle m of step s :

$$\mathbf{d}^{(s+1,m)} = \mathbf{d}^{(s)} + \Delta \mathbf{d}^{(s,m)} \quad (6.54)$$

$$\mathbf{q}^{(s+1,m)} = \mathbf{q}^{(s)} + \Delta \mathbf{q}^{(s,m)} \quad (6.55)$$

$$\mathbf{r}^{(s+1,m)} = \mathbf{r}^{(s)} + \Delta \mathbf{r}^{(s,m)} \quad (6.56)$$

$\mathbf{d}^{(s+1,m)}$ trial system displacement vector at the start of step $s+1$ in iteration m

$\mathbf{q}^{(s+1,m)}$ trial system load vector at the start of step $s+1$ in iteration m

$\mathbf{r}^{(s+1,m)}$ trial system reaction vector at the start of step $s+1$ in iteration m

(c) **Termination of the iteration:** The system error vector \mathbf{e}_s is computed for the trial state (6.54) to (6.56) of the structure, using expressions (4.78) and (5.193). The iteration in step s is terminated if the norm of the system error vector is less than a preset limit. Otherwise, the computations in step s are continued with iteration $m+1$. The error limit can, for example, be a prescribed fraction of the force norm in the first load step.

If the iteration in step s has converged, the algorithm proceeds to step $s+1$ and the steps that follow. The process is terminated when the load factor 1.0 is reached. Since the load factor is calculated and not prescribed, the value 1.0 will usually be exceeded in the last step of the analysis. The values of the displacements, reactions and stress resultants corresponding to $\lambda = 1.0$ are determined by interpolation in the last load step.

6.5 LOAD FACTOR INCREMENT

The load factor increment is chosen so that the length of the chord of the arc for step s in the load path is equal to the length h_0 of the chord for the first load step, which follows from the prescribed value $\Delta\lambda_0$ of the load factor in the first iteration of the first load step:

$$h_0^2 = \Delta\lambda_0^2 \mathbf{p}_0^T \mathbf{p}_0 + \mu^2 \Delta\lambda_0^2 \mathbf{d}_0^T \mathbf{d}_0 \quad (6.57)$$

h_0 chord length for the first load step

$\Delta\lambda_0$ prescribed load factor increment for the first step

\mathbf{p}_0 system force vector (6.44) in the first step due to the pattern load

\mathbf{d}_0 system displacement vector due to \mathbf{p}_0

The arc length h_0 , which is computed with (6.57) in the first cycle of iteration of the first load step, accounts for the loads, reactions and displacements of the linear elastic structure, but not for the corrective displacements \mathbf{f}_1 in (6.47) or the corrective reactions \mathbf{t}_2 in (6.49). In order to account for these corrections, the initial arc length h_0 is recomputed at the end of each cycle of iteration in the first load step.

The displacement norms a_1 and a_2 and the force norms b_1 and b_2 at the beginning and the end of the iteration in the load step are:

$$a_1 = \odot \mu \sqrt{\mathbf{d}^T \mathbf{d}} \quad a_2 = \odot \mu \sqrt{(\mathbf{d} + \Delta\mathbf{d})^T (\mathbf{d} + \Delta\mathbf{d})} \quad (6.58)$$

$$b_1 = \odot \sqrt{\mathbf{p}^T \mathbf{p}} \quad b_2 = \odot \sqrt{(\mathbf{p} + \Delta\mathbf{p})^T (\mathbf{p} + \Delta\mathbf{p})} \quad (6.59)$$

\odot path-dependent sign, see (6.2) and (6.5)

Separate procedures are required to determine the load factor increment in step s for both equal and different path-dependent signs of the displacement and force norms. These procedures are developed in the following sections in such a way that they can be used for both the standard arc increment method and the continuation beyond a singular point (see chapter 7).

6.5.1 Path-Dependent Norms with Equal Signs

In order to simplify the notation for the expressions derived in this section, the notation for the variables is redefined. The system displacement vector at the end of the step is denoted by:

$$\bar{\mathbf{d}} = \mathbf{d} + \mathbf{f} + z \mathbf{a} \quad (6.60)$$

$\bar{\mathbf{d}}$ system displacement vector at the end of the step

\mathbf{d} system displacement vector at the beginning of the step

\mathbf{f} displacement correction vector

\mathbf{a} displacement reference vector (pattern displacement or eigenvector)

z load factor increment $\Delta\lambda$ for the step

The system force vector at the end of the step is denoted by:

$$\bar{\mathbf{p}} = \mathbf{p} + \mathbf{t} + z\mathbf{b} \quad (6.61)$$

- $\bar{\mathbf{p}}$ system force vector at the end of the step
 \mathbf{p} system force vector at the beginning of the step
 \mathbf{t} force correction vector
 \mathbf{b} force reference vector (pattern force or reaction due to eigenvector)
 z load factor increment $\Delta\lambda$ for the step

The displacement norms at the beginning and the end of the step are:

$$a_1 = \mu \sqrt{(\mathbf{d} + \mathbf{f})^T (\mathbf{d} + \mathbf{f})} \quad (6.62)$$

$$a_2 = \mu \sqrt{(\mathbf{d} + \mathbf{f} + z\mathbf{a})^T (\mathbf{d} + \mathbf{f} + z\mathbf{a})}$$

μ displacement calibration factor, see (6.2)

The force norms at the beginning and the end of the step are:

$$b_1 = \sqrt{(\mathbf{p} + \mathbf{t})^T (\mathbf{p} + \mathbf{t})} \quad (6.63)$$

$$b_2 = \sqrt{(\mathbf{p} + \mathbf{t} + z\mathbf{b})^T (\mathbf{p} + \mathbf{t} + z\mathbf{b})}$$

The chord length in step s is set equal to the chord length h_0 in the first load step:

$$(a_2 - a_1)^2 + (b_2 - b_1)^2 = h_0^2 \quad (6.64)$$

h_0 chord length of the first step

The following dimensionless parameters are defined:

$$n_0 = (\mathbf{d} + \mathbf{f})^T (\mathbf{d} + \mathbf{f}) \quad n_1 = \frac{\mathbf{a}^T (\mathbf{d} + \mathbf{f})}{n_0} \quad n_2 = \frac{\mathbf{a}^T \mathbf{a}}{n_0} \quad (6.65)$$

$$n_3 = (\mathbf{p} + \mathbf{t})^T (\mathbf{p} + \mathbf{t}) \quad n_4 = \frac{\mathbf{b}^T (\mathbf{p} + \mathbf{t})}{n_3} \quad n_5 = \frac{\mathbf{b}^T \mathbf{b}}{n_3} \quad (6.66)$$

The terms on the left-hand side of equation (6.64) are expressed in terms of the parameters (6.65) and (6.66):

$$(a_2 - a_1)^2 = \mu^2 n_0 (2 + 2n_1 z + n_2 z^2 - 2\sqrt{1 + 2n_1 z + n_2 z^2}) \quad (6.67)$$

$$(b_2 - b_1)^2 = n_3 (2 + 2n_4 z + n_5 z^2 - 2\sqrt{1 + 2n_4 z + n_5 z^2}) \quad (6.68)$$

Substitution of expressions (6.67) and (6.68) into equation (6.64) shows that the equation for the determination of the load factor increment z is highly nonlinear. It is solved iteratively by the Newton-Raphson method. The residual function $f(z)$ is defined as follows:

$$f(z) = (a_2 - a_1)^2 + (b_2 - b_1)^2 - h_0^2 \quad (6.69)$$

$f(z)$ residual function for Newton-Raphson iteration

Since the derivative of the residual function is null at the origin $z=0$, the iteration cannot be started at the origin. In order to obtain a suitable starting value z_0 for the iteration, the residual function (6.69) is approximated by a Taylor series at $z=0$:

$$(a_2 - a_1)^2 = \mu^2 n_0 (2 + 2n_1 z + n_2 z^2 - 2r) \quad (6.70)$$

$$(b_2 - b_1)^2 = n_3 (2 + 2n_4 z + n_5 z^2 - 2s)$$

$$r = \sqrt{1 + 2n_1 z + n_2 z^2} \quad (6.71)$$

$$s = \sqrt{1 + 2n_4 z + n_5 z^2}$$

The derivatives of the residual function are given by:

$$\frac{df}{dz} = 2\mu^2 n_0 (n_1 + n_2 z) \left(1 - \frac{1}{r}\right) + 2n_3 (n_4 + n_5 z) \left(1 - \frac{1}{s}\right) \quad (6.72)$$

$$\frac{d^2 f}{dz^2} = 2\mu^2 n_0 \left(n_2 \left(1 - \frac{1}{r}\right) + \frac{(n_1 + n_2 z)^2}{r^3} \right) + 2n_3 \left(n_5 \left(1 - \frac{1}{s}\right) + \frac{(n_4 + n_5 z)^2}{s^3} \right)$$

$$\frac{d^3 f}{dz^3} = 6\mu^2 n_0 \left(\frac{n_2 (n_1 + n_2 z)}{r^3} - \frac{(n_1 + n_2 z)^3}{r^5} \right) + 6n_3 \left(\frac{n_5 (n_4 + n_5 z)}{s^3} - \frac{(n_4 + n_5 z)^3}{s^5} \right)$$

$$\frac{d^4 f}{dz^4} = 6\mu^2 n_0 \left(\frac{n_2^2}{r^3} - \frac{6n_2 (n_1 + n_2 z)^2}{r^5} + \frac{5(n_1 + n_2 z)^4}{r^7} \right) + 6n_3 \left(\frac{n_5^2}{s^3} - \frac{6n_5 (n_4 + n_5 z)^2}{s^5} + \frac{5(n_4 + n_5 z)^4}{s^7} \right)$$

At the origin $z = 0$, the roots are $r = s = 1$, so that the function and its derivatives are given by the following expressions:

$$f_0 = -h_0^2 \quad (6.73)$$

$$\frac{df_0}{dz} = 0$$

$$\frac{d^2 f_0}{dz^2} = 2\mu^2 n_0 n_1^2 + 2n_3 n_4^2$$

$$\frac{d^3 f_0}{dz^3} = 6\mu^2 n_0 n_1 (n_2 - n_1^2) + 6n_3 n_4 (n_5 - n_4^2)$$

$$\frac{d^4 f_0}{dz^4} = 6\mu^2 n_0 (n_2 - n_1^2)(n_2 - 5n_1^2) + 6n_3 (n_5 - n_4^2)(n_5 - 5n_4^2)$$

The Taylor series expansion of the residual function at the origin $z = 0$ is truncated after the fifth term in order to obtain a quartic equation in the load factor increment z :

$$f(z) = f_0 + \frac{df_0}{dz} z + \frac{1}{2} \frac{d^2 f_0}{dz^2} z^2 + \frac{1}{6} \frac{d^3 f_0}{dz^3} z^3 + \frac{1}{24} \frac{d^4 f_0}{dz^4} z^4 \quad (6.74)$$

In special cases, for example at a bifurcation point of the 2-bar truss in section 2.1, the second and third derivatives of the residual function are null at the origin $z = 0$, so that a quadratic approximation of residual function $f(z)$ would not be adequate. Equation (6.74) is solved for the load factor increment by one of the available standard algorithms. The equation can have 4 real roots, or 2 real and 2 conjugate complex roots, or 2 pairs of conjugate complex roots. If it has 4 real roots, the two roots with the smallest absolute value are considered. If it has at least two real roots,

the root that leads to the largest distance between the pivot state $s - 1$ and new state $s + 1$ on the load path is chosen as starting point for the Newton-Raphson iteration for the load factor increment and is denoted by z_0 .

The approximate solution z_0 of equation (6.69), which is determined with the Taylor series expansion (6.74), is improved by Newton-Raphson iteration. At the beginning of cycle m of the iteration, the residual function f_m and its first derivative are computed with equations (6.69) and (6.72). The iterated value z_{m+1} of the load factor increment is obtained by solving the linearisation of equation (6.64):

$$f_m + \frac{df_m}{dz}(z_{m+1} - z_m) = 0$$

$$z_{m+1} = z_m - \left(\frac{df_m}{dz} \right)^{-1} f_m \quad (6.75)$$

The iteration is continued until the change in the load factor increment in a cycle of the iteration is less than a prescribed value.

6.5.2 Change in Sign of the Path-Dependent Force Norm

Let the path-dependent signs of the force norms at the beginning and at the end of a load step be opposite. The force norm then equals null at a point of the load path lying inside the load step. If the sign of the norm is not taken into account, the slope of the path between the end points of the load step is discontinuous on the axis where the load factor is null. The distance between the two end points therefore does not equal the length of the arc. The methods for determining the load factor increment and the distance of the end point to the pivot point, which depend on a smooth curve, can therefore no longer be used.

Points in the vicinity of the axis, where the load factor is null, are avoided in the procedure used to determine the load factor increment if the path-dependent sign of the norm changes. The procedure is therefore not affected by the discontinuity in the slope. It consists of the following steps:

- (1) Let the absolute value of the load factor at the start of the step be f . If f exceeds a prescribed limit g , the absolute value of the load factor at the end of the step is set to $0.5f$ and the sign of the load factor is reversed.
- (2) If the absolute value of the load factor at the start of the step is less than or equal to g , the absolute value of the load factor at the end of the step is set to g and the sign of the load factor is reversed.
- (3) The sign of the load factor increment in the load step in which the sign of the load factor changes and in the next load step are set equal.

6.5.3 Load Step for a 2-Bar Truss

This example illustrates the iteration in a load step of the numerical solution of the governing equations (without error vector \mathbf{e}). Consider the load curve for solution 1 of the 2-bar truss shown in figure 2.4. The pattern load on the truss is $p_{2t} = -0.3849$. Let the displacement and loading of the truss in configuration C be $(w_1, w_2) = (0.0, -0.250000)$ and $(p_1, p_2) = (0.0, -0.328125)$. The load step s following configuration C is to be computed. Let the desired arc length of the load step be $h = 0.1$ and assume that configuration C is an exact equilibrium configuration.

Denote w_2 with u and p_2 with q . The part of the load curve for the displacement range $(-0.24, -0.35)$ that contains the load step is enlarged in figure 6.6. The equilibrium equation (2.20) of the truss in configuration C is written in terms of the total stiffness k of the truss:

$$k u = q \quad \text{with} \quad k = (1+u)(2+u) \quad (6.76)$$

Let the displacement increment in iteration 0 of step s be Δu . The equilibrium equation of the truss in state $s+1$ with displacement $u + \Delta u$ follows from (2.20):

$$k(u + \Delta u) = q + \Delta q \quad \text{with} \quad k = (1+u + \Delta u)(2+u + \Delta u) \quad (6.77)$$

The secant incremental stiffness is determined by subtracting (6.76) from (6.77):

$$\begin{aligned} k_t \Delta u &= \Delta q - q_c \\ k_t &= (2 + 6u + 3u^2) + 3(1+u)\Delta u + \Delta u^2 \end{aligned} \quad (6.78)$$

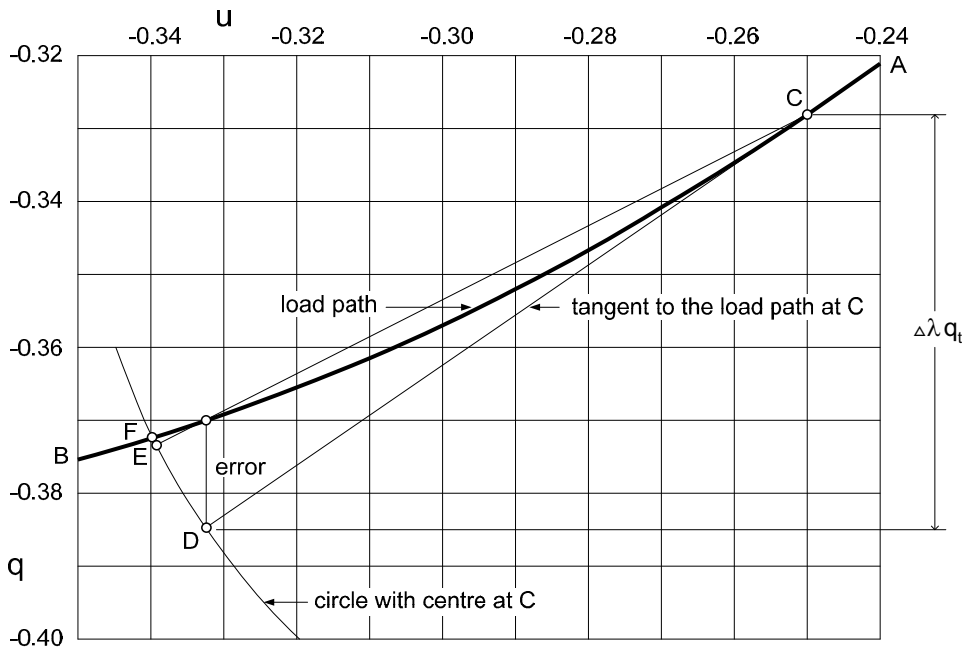


Figure 6.6 Load step for the 2-bar truss in figure 2.1

- | | | | |
|---|---------------------|---|-----------------------|
| C | start configuration | D | result of iteration 0 |
| F | end configuration | E | result of iteration 1 |

The secant stiffness k_t is computed by iteration. In cycle 0, the stiffness is computed for a displacement increment 0. In cycle m , the stiffness is approximated with the displacement increment computed in cycle $m-1$.

Cycle 0: The incremental stiffness is computed by substitution of $\Delta u^{(s,0)}=0.0$ and $u^{(s)}=-0.2500$ into (6.78). The pattern displacement u_t is computed for the pattern load $q_t=-0.3849$:

$$k_t^{(s,0)} = (1+u^{(s)})(2+u^{(s)}) = 0.687500$$

$$k_t^{(s,0)} u_t^{(s,0)} = q_t : u_t^{(s,0)} = \frac{-0.384900}{0.687500} = -0.559855$$

The load factor increment $\Delta\lambda^{(s,0)}$ is computed so that the length of the arc increment is $h=0.1$:

$$(\Delta\lambda^{(s,0)})^2 ((u_t^{(s,0)})^2 + q_t^2) = (0.10)^2 : \quad \Delta\lambda^{(s,0)} = 0.147189$$

$$\Delta u^{(s,0)} = \Delta\lambda^{(s,0)} u_t^{(s,0)} = -0.082404 \quad u^{(s+1,0)} = u^{(s)} + \Delta u^{(s,0)} = -0.332404$$

$$\Delta q^{(s,0)} = \Delta\lambda^{(s,0)} q_t = -0.056653 \quad q^{(s+1,0)} = q^{(s)} + \Delta q^{(s,0)} = -0.384778$$

Cycle 1: The substitution of $\Delta u^{(s,0)}$ from cycle 0 into expression (6.78) leads to an improved value of the secant incremental stiffness:

$$k_t^{(s,1)} = 0.6875 + 3(1+u)\Delta u^{(s,0)} + (\Delta u^{(s,0)})^2 = 0.508881$$

The pattern displacement is computed for $q_t=-0.384900$ as in cycle 0:

$$k_t^{(s,1)} u_t^{(s,1)} = q_t : u_t^{(s,1)} = \frac{-0.384900}{0.508881} = -0.756365$$

The load factor increment $\Delta\lambda^{(s,1)}$ is computed as in cycle 0 so that the length of the arc increment in step s is $h=0.1$:

$$(\Delta\lambda^{(s,1)})^2 ((u_t^{(s,1)})^2 + q_t^2) = (0.10)^2 : \quad \Delta\lambda^{(s,1)} = 0.117832$$

$$\Delta u^{(s,0)} = \Delta\lambda^{(s,0)} u_t^{(s,0)} = -0.089664 \quad u^{(s+1,0)} = u^{(s)} + \Delta u^{(s,0)} = -0.339124$$

$$\Delta q^{(s,0)} = \Delta\lambda^{(s,0)} q_t = -0.044276 \quad q^{(s+1,0)} = q^{(s)} + \Delta q^{(s,0)} = -0.373478$$

Convergence: The iteration is continued, starting with the displacement increment $\Delta u^{(s,1)}=-0.089664$. The exact load corresponding to the computed displacement u is computed with (2.20). The results converge in cycle 4 to values that are correct to six digits. The rate of convergence in the example is high.

m	k_t	u_t	$\Delta\lambda$	u	q	q_{exact}
2	0.494914	-0.777710	0.115241	-0.339624	-0.372481	-0.372388
3	0.493878	-0.779343	0.115047	-0.339661	-0.372407	-0.372400
4	0.493801	-0.779463	0.115033	-0.339664	-0.372401	-0.372401

The results of the iteration are shown in figure 6.6. The trial solution at the end of cycle 0 is D, the trial solution at the end of cycle 1 is E and the exact solution for the load step is F. The distances \overline{CD} , \overline{CE} and \overline{CF} equal h , so that the points D, E and F lie on a circle with centre C and radius h .

6.6 LINEAR EQUATIONS WITH PROFILE STRUCTURE AND STATUS

The incremental stiffness matrix for an iteration in a load step is assembled at the beginning of the load step and used to compute the pattern displacement in (6.43), as well as the increment (6.46) of the free displacements and of the increment (6.48) of the reaction. The coefficients of the stiffness matrix are real and the matrix is symmetric. Because the near-limit configurations are treated in a separate algorithm, it can be assumed when the system of linear equations is solved that the secant stiffness matrix is regular. The secant matrix is not necessarily positive definite.

The linear incremental governing equations (6.12) solved in the nonlinear structural analysis have special features that influence the solution algorithm. The stiffness matrix is generally sparsely populated and has a profile structure. The system displacement vector on the left-hand side and the reaction vector on the right-hand side of the equation both contain unknowns.

The displacements in the system vector were reordered in the solution concept of section 6.4. This reordering destroys the profile structure of the stiffness matrix and would lead to a very significant increase of the computational effort if it were part of the implementation of the algorithm. A method will be developed that avoids reordering by distinguishing between free and prescribed variables in the algorithm, and thus avoids the need to destroy the profile.

The algorithm for the solution of equation (6.12) is developed in stages. In the first stage, the elementary linear system $\mathbf{A}\mathbf{u} = \mathbf{q}$ with fully populated coefficient matrix \mathbf{A} and fully prescribed absolute term \mathbf{q} is considered. The unknown vector \mathbf{u} is determined. In the second stage, the profile structure of the matrix is considered. In the third stage, the method that avoids reordering is developed.

Full matrix: Consider the following system of linear equations with a real symmetric matrix \mathbf{A} and an absolute term \mathbf{q} that is prescribed. The term \mathbf{u} is to be determined.

$$\mathbf{A}\mathbf{u} = \mathbf{q} \quad (6.79)$$

\mathbf{A} coefficient matrix
 \mathbf{u} unknown vector
 \mathbf{q} absolute vector

The square matrix \mathbf{A} is decomposed into the product of a left triangular matrix \mathbf{S} with unit diagonal coefficients, a diagonal matrix \mathbf{D} whose diagonal coefficients can be positive or negative but not null, and the transpose of \mathbf{S} .

$$\mathbf{A} := \mathbf{S}\mathbf{D}\mathbf{S}^T \quad (6.80)$$

\mathbf{S} left triangular matrix with diagonal coefficients 1.0
 \mathbf{D} diagonal matrix with diagonal coefficients which are not null

Decomposition (6.80) is substituted into (6.79). A temporary vector \mathbf{y} is defined and substituted into the equation:

$$\mathbf{S}\mathbf{D}\mathbf{S}^T\mathbf{u} = \mathbf{q} \quad (6.81)$$

$$\mathbf{y} := \mathbf{S}^T\mathbf{u} \quad (6.81)$$

$$\mathbf{S}\mathbf{D}\mathbf{y} = \mathbf{q} \quad (6.82)$$

Due to the triangular population of \mathbf{S} , equation (6.82) is conveniently solved for the temporary vector \mathbf{y} :

$$\mathbf{D}\mathbf{y} = \mathbf{S}^{-1}\mathbf{q} \quad (6.83)$$

$$\mathbf{y} = \mathbf{D}^{-1}\mathbf{S}^{-1}\mathbf{q} \quad (6.84)$$

The temporary vector \mathbf{y} is substituted into (6.81), which is then solved for \mathbf{u} :

$$\begin{aligned} \mathbf{S}^T\mathbf{u} &= \mathbf{y} \\ \mathbf{u} &= \mathbf{S}^{-T}\mathbf{y} \end{aligned} \quad (6.85)$$

The operations performed on the coefficients of the matrix and the vectors in the described phases of the solution will now be considered.

Decomposition: The matrix product for decomposition (6.80) of \mathbf{A} is shown in figure 6.7. The known matrix \mathbf{A} with dimension n is decomposed by columns, starting with column 0. Index p of the current column is called the pivot of the decomposition. After the pivot column has been decomposed, columns 0 to p of \mathbf{S} and the diagonal coefficients d_0, \dots, d_n of \mathbf{D} are known.

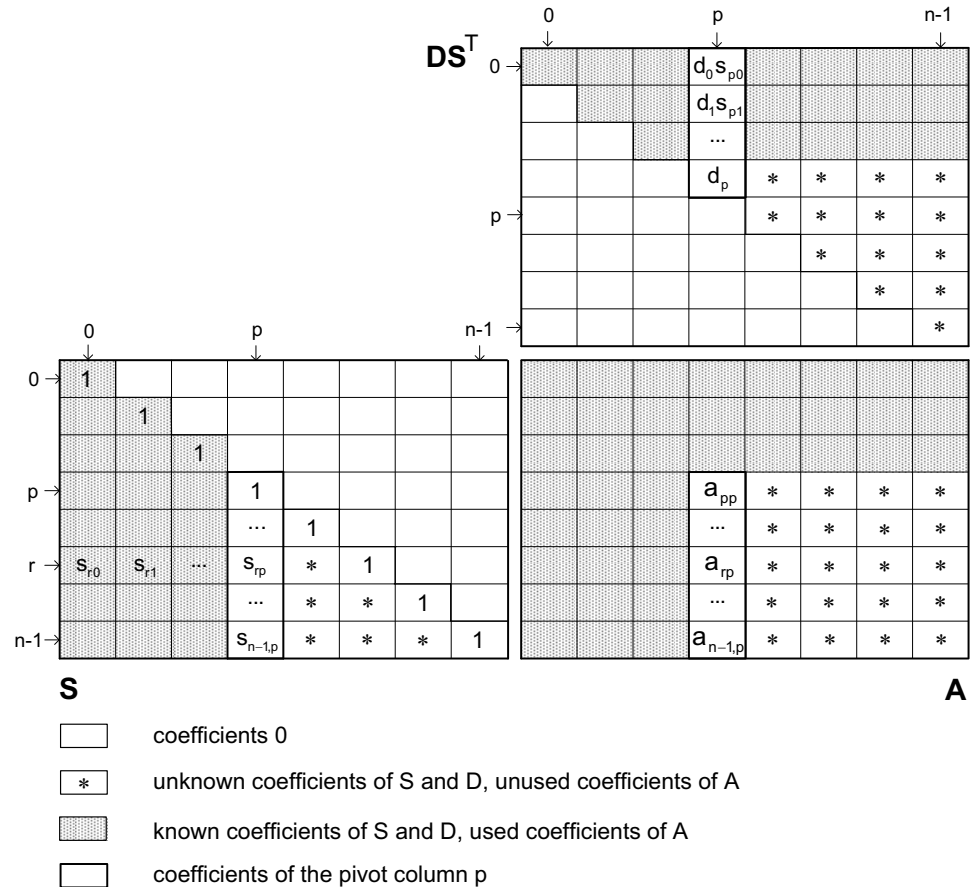


Figure 6.7 Decomposition of a symmetric real matrix

The decomposition is started by computing d_0 and column 0 of \mathbf{S} , using the known value $s_{00} = 1$:

$$d_0 s_{00} = a_{00} : \quad d_0 = a_{00} \quad (6.86)$$

$$d_0 s_{r0} = a_{r0} : \quad s_{r0} = \frac{a_{r0}}{d_0} \quad r = 1, \dots, n-1 \quad (6.87)$$

Assume that the diagonals d_0, \dots, d_{p-1} of \mathbf{D} and the coefficients of columns 0 to $p-1$ of \mathbf{S} have been computed. The decomposition is continued by computing the diagonal d_p and the coefficients of column p of \mathbf{S} using the known value $s_{dd} = 1$:

$$\sum_{k=0}^{p-1} s_{pk} (d_k s_{pk}) + d_p = a_{pp}$$

$$d_p = a_{pp} - \sum_{k=0}^{p-1} s_{pk} (d_k s_{pk}) \quad (6.88)$$

$$\sum_{k=0}^{p-1} s_{rk} (d_k s_{pk}) + s_{rp} d_p = a_{rp} \quad r = p+1, \dots, n-1$$

$$s_{rp} = \frac{1}{d_p} \left(a_{rp} - \sum_{k=0}^{p-1} s_{rk} (d_k s_{pk}) \right) \quad (6.89)$$

Profile matrix: Let matrix \mathbf{A} be sparsely populated and assume that the coefficients that are not null cluster near the diagonal. The column index of the first coefficient that is not null in row k of matrix \mathbf{A} is called the row profile in row k . The row index of the last coefficient that is not null in column m of matrix \mathbf{A} is called the column profile in column m . The row and column profiles of a coefficient matrix are shown in figure 6.8. The matrix is called a profile matrix.

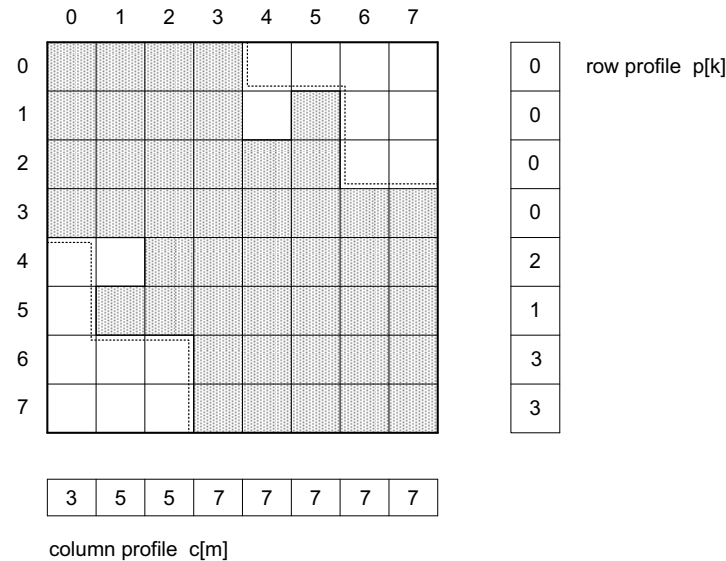


Figure 6.8 Row and column profiles of a coefficient matrix

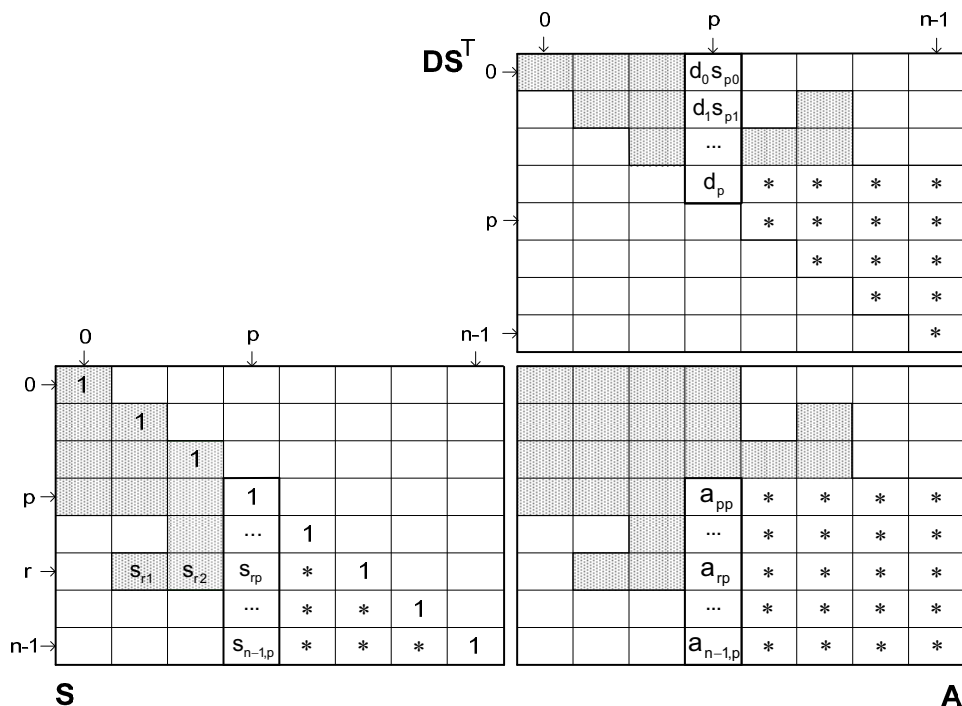


Figure 6.9 Decomposition of a profile matrix

The profile structure influences the decomposition of matrix **A** in several ways:

- (a) If the row profile $p[r]$ of row r is larger than the pivot p , row r is not considered in the decomposition of column p of matrix **A**.
- (b) The index of the last row of **A** that must be considered in the decomposition of column p is the column profile $c[p]$ of the pivot column.
- (c) The sum in (6.88) starts with the row profile $p[p]$ of the pivot row. The sum in (6.89) starts with the maximum of $p[r]$ for row r or $p[p]$ for column p .

Coefficients of **A** that lie outside the row profile remain null during the decomposition of the matrix. In order to prove this property, consider a coefficient s_{ij} that lies outside the profile of the matrix so that $j < p[r]$. If this coefficient were included in the decomposition, the corresponding coefficient in **S** would be given by (6.90):

$$s_{ij} = \frac{1}{d_j} \left(a_{ij} - \sum_{k=m}^{j-1} s_{rk} d_k s_{jk} \right) \quad (6.90)$$

The start index m for the loop in (6.90) is the maximum of the row profiles $p[j]$ and $p[r]$. Since $p[j] \leq j < p[r]$, the start index is $m = p[r] > j$. The sum in (6.90) is therefore empty and has the value null. Since the coefficient a_{ij} is also null, the coefficient s_{ij} is null. The profile of the matrix is not affected by the decomposition.

Forward sweep: The computation of the temporary vector \mathbf{y} in (6.84) is called the forward sweep of the solution algorithm. The coefficients of \mathbf{y} are computed in sequence from y_0 to y_{n-1} :

$$d_r \sum_{k=p[r]}^r s_{rk} y_k = q_r \quad r = 0, \dots, n-1$$

$$y_r = \frac{1}{d_r} \left(q_r - \sum_{k=p[r]}^{r-1} s_{rk} y_k \right) \quad (6.91)$$

Backward sweep: The computation of the solution vector \mathbf{u} in (6.85) is called the backward sweep of the solution algorithm. The matrix \mathbf{S}^T is not stored in the solution procedure. It is derived from matrix \mathbf{S} , which is stored row-wise. It is advantageous to arrange the operations on \mathbf{S}^T column-wise, so that the corresponding operations on \mathbf{S} can be performed row-wise. The coefficients of \mathbf{u} are computed in sequence from u_{n-1} to u_0 . Once the value of u_r has been computed, it is immediately substituted into the rows with indices $p[r]$ of r to $r-1$:

$$y_k \leftarrow y_k - s_{rk} u_r \quad r = n-1, \dots, 1; \quad k = p[r], \dots, r-1 \quad (6.92)$$

Status of variables: The distinction between free and prescribed displacements is made by means of a status vector. If the variable d_k in the system displacement vector is prescribed, the status variable s_k has the value true.

In equation (6.39), the status vector is used to order the system equations in such a way that a solution procedure can be formulated in matrix notation. The chosen order of the displacements destroys the profile of the stiffness matrix and thus causes a significant increase in the storage requirements, as well as the number of operations for the solution. This disadvantage is avoided in the implementation by using the status vector direct in the algorithms of the solver. The original order of the variables is retained, and thus the profile of the stiffness matrix.

- (a) The decomposition of \mathbf{A} is restricted to the rows and columns with status false:
 - The substitution of the prescribed displacements is restricted to the rows with status false.
 - The pivot column p is only decomposed if the status s_p is false.
 - Row r of the pivot column is only computed if the status s_r is false.
 - The term with index k of the sum in (6.88) and (6.89) is only included in the sum if s_k is false.
- (b) The forward sweep is restricted to the rows with status false:
 - The substitution of the prescribed displacements is restricted to the rows with status false and the columns with status true.
 - The multiplication with \mathbf{S}^{-1} is restricted to rows and columns with status false.
 - The division with the diagonal d_r is restricted to rows with status false.
- (c) The backward sweep is restricted to rows of \mathbf{S} with status false.
- (d) The computation of the reactions is restricted to rows with status true.

This concludes the solution of linear equations with profile and status.

7 STABILITY ANALYSIS

7.1 STIFFNESS OF STRUCTURES

7.1.1 Concepts of Stiffness

Three concepts are used in this chapter to describe the stiffness of a structure: total stiffness, tangent stiffness and secant stiffness. The distinction between these three concepts of stiffness is essential for the mathematical treatment of the stability of structures. The concepts are defined in this section and illustrated with the stiffness of a two-bar-truss in the next sections.

Total stiffness: Consider the finite element formulation of the equilibrium equations of a structure. Let the equilibrium equations for the total displacements from the reference configuration to the instant configuration be written as the product of a square matrix \mathbf{K} , a total displacement vector \mathbf{d} and a total force vector \mathbf{p} :

$$\mathbf{Kd} = \mathbf{p} \quad (7.1)$$

\mathbf{K} total stiffness matrix
 \mathbf{d} total displacement vector
 \mathbf{p} total force vector

The coefficient k_{im} of matrix \mathbf{K} is the total force p_i required to cause a unit total displacement d_m . The matrix \mathbf{K} is called the total stiffness matrix of the structure. Since the equilibrium equations are highly nonlinear functions of the displacement vector, the matrix \mathbf{K} is not unique. The left-hand side of the equilibrium equations is decomposed into the product \mathbf{Kd} so that \mathbf{K} is symmetric.

In the solution procedure described in chapter 6, the total stiffness matrix is not used explicitly. It is, however, contained indirectly in the determination of the error vector \mathbf{e} in expressions (4.80) for trusses and expressions (5.193) for frames.

Tangent stiffness: The increment Δp_i of the load coordinate p_i due to an increment $\Delta \mathbf{d}$ of the displacement in a load step depends on the partial derivatives of p_i with respect to the displacement coordinates:

$$p_i = \sum_m k_{im} d_m$$
$$\Delta p_i = \sum_n \frac{\partial p_i}{\partial d_n} \Delta d_n = \sum_n \frac{\partial}{\partial d_n} \left(\sum_m k_{im} d_m \right) \Delta d_n \quad (7.2)$$

Equation (7.2) is written in matrix form. The resulting stiffness matrix is called the tangent stiffness matrix of the structure and denoted by \mathbf{K}^t . The coefficients of \mathbf{K}^t are not the partial derivatives of the coefficients of the stiffness matrix \mathbf{K} in (7.1), but rather the derivatives of the products $k_{im} d_m$.

$$\mathbf{K}^t \Delta \mathbf{d} = \Delta \mathbf{p} \quad (7.3)$$

\mathbf{K}^t tangent stiffness matrix
 $\Delta \mathbf{d}$ incremental displacement vector
 $\Delta \mathbf{p}$ incremental force vector

The tangent stiffness matrix is used in the constant arc increment method of chapter 6 to perform the first cycle of the iteration in a load step. It will be used in this chapter to define singular configurations of a structure.

Secant stiffness: The displacement increment $\Delta \mathbf{d}$, which is computed for a load increment $\Delta \mathbf{p}$ with equation (7.3), is not exact, since the variation of the tangential stiffness matrix \mathbf{K}^t with the displacement is not taken into account. The total displacement $\mathbf{d} + \Delta \mathbf{d}$ computed for the total load $\mathbf{p} + \Delta \mathbf{p}$ therefore does not fall exactly on the load path. In order to be able to compute at a given point A of the load path a neighbouring point B that also lies on the load path, the secant stiffness matrix \mathbf{K}^s at point A is defined so that the solution $\Delta \mathbf{d}^s$ of equation (7.4) leads to the total displacement of point B.

$$\mathbf{K}^s \Delta \mathbf{d} = \Delta \mathbf{p} \quad (7.4)$$

\mathbf{K}^s secant stiffness matrix from point A to point B on the load path
 $\Delta \mathbf{d}$ displacement vector increment from point A to point B on the load path
 $\Delta \mathbf{p}$ total force vector increment from point A to point B on the load path

In contrast with the coefficients of the tangent matrix \mathbf{K}^t , which are determined by differentiation, the coefficients of the secant matrix \mathbf{K}^s are determined by forming the finite difference between the equilibrium equations at points A and B using the total stiffness matrices at A and B:

$$\begin{aligned} \text{equilibrium at A: } & \mathbf{K}(\mathbf{d})\mathbf{d} = \mathbf{p} \\ \text{equilibrium at B: } & \mathbf{K}(\mathbf{d} + \Delta \mathbf{d})(\mathbf{d} + \Delta \mathbf{d}) = \mathbf{p} + \Delta \mathbf{p} \\ & (\mathbf{K}(\mathbf{d} + \Delta \mathbf{d}) - \mathbf{K}(\mathbf{d}))\mathbf{d} + \mathbf{K}(\mathbf{d} + \Delta \mathbf{d})\Delta \mathbf{d} = \Delta \mathbf{p} \end{aligned} \quad (7.5)$$

The first term in equation (7.5) is transformed into the product of an auxiliary matrix \mathbf{K}^a with the displacement increment $\Delta \mathbf{d}$:

$$\mathbf{K}^a \Delta \mathbf{d} = (\mathbf{K}(\mathbf{d} + \Delta \mathbf{d}) - \mathbf{K}(\mathbf{d}))\mathbf{d} \quad (7.6)$$

The substitution of equations (7.5) and (7.6) into equation (7.4) yields expression (7.7) for the secant matrix:

$$\mathbf{K}^s = \mathbf{K}(\mathbf{d} + \Delta \mathbf{d}) + \mathbf{K}^a \quad (7.7)$$

Since the equilibrium equations (7.5) and (7.6) are highly nonlinear functions of the total displacement coordinates, the secant matrix \mathbf{K}^s is not unique. The decomposition $\mathbf{K}^s \Delta \mathbf{d}$ in equation (7.4) is made so that matrix \mathbf{K}^s is symmetric.

In the algorithm of the constant arc increment method in chapter 6, the secant matrix is determined numerically by iteration in each load step. The procedure is illustrated in figure 6.6. The secant matrix at point C leads to point F on the load curve.

Consider the load steps from a point A on a load path to a singular point L, and from the singular point L to a point B on the load path. The secant matrices for the steps \overline{AL} and \overline{LB} are in general regular, even though the tangent matrix at the singular point L is by definition singular. This property of secant matrices is used later in this chapter to develop an algorithm for load steps near singular points.

7.1.2 Total Stiffness of a 2-Bar Truss

Total stiffness matrix: Consider the equilibrium equations (2.18) and (2.19) of the 2-bar truss:

$$p_1 = w_1(2m^2 + w_1^2 + w_2(2 + w_2)) \quad (7.8)$$

$$p_2 = (1 + w_2)(w_1^2 + w_2(2 + w_2)) \quad (7.9)$$

These equations are arranged in the matrix form (7.1):

$$\mathbf{K} \mathbf{w} = \mathbf{p} \quad (7.10)$$

$$\begin{array}{|c|c|c|} \hline 2m^2 + w_1^2 + w_2 & w_1(1 + w_2) & w_1 \\ \hline w_1(1 + w_2) & (1 + w_2)(2 + w_2) & w_2 \\ \hline \end{array} = \begin{array}{|c|} \hline p_1 \\ \hline p_2 \\ \hline \end{array}$$

\mathbf{K} total stiffness matrix of the truss

The truss with load $p_1 = 0$ must satisfy the equilibrium equations. For solution 1 of equations (7.8) and (7.9) in section 2.1, the displacement $w_1 = 0$. The total stiffness matrix \mathbf{K}_1 for solution 1 is obtained by substituting $w_1 = 0$ into expression (7.10):

$$\mathbf{K}_1 \mathbf{w} = \mathbf{p}_2 \quad (7.11)$$

$$\begin{array}{|c|c|c|} \hline 2m^2 + w_2 & 0 & w_1 \\ \hline 0 & (1 + w_2)(2 + w_2) & w_2 \\ \hline \end{array} = \begin{array}{|c|} \hline 0 \\ \hline p_2 \\ \hline \end{array}$$

\mathbf{K}_1 total stiffness matrix for solution 1

The total stiffness matrix \mathbf{K}_2 for solution 2 in section 2.1 is obtained by substituting the condition $2m^2 + w_1^2 + w_2(2 + w_2) = 0$ into expression (7.10):

$$\mathbf{K}_2 \mathbf{w} = \mathbf{p}_2 \quad (7.12)$$

$$\begin{array}{|c|c|c|} \hline -w_2(1 + w_2) & w_1(1 + w_2) & w_1 \\ \hline w_1(1 + w_2) & (1 + w_2)(2 + w_2) & w_2 \\ \hline \end{array} = \begin{array}{|c|} \hline 0 \\ \hline p_2 \\ \hline \end{array}$$

$$w_1 = \pm \sqrt{-w_2(2 + w_2) - 2m^2}$$

\mathbf{K}_2 total stiffness matrix for solution 2

7.1.3 Tangent Stiffness of a 2-Bar Truss

The load coordinates p_1 and p_2 in expressions (7.8) and (7.9) are differentiated according to rule (7.2). The resulting equations are written in the form (7.3):

$$\mathbf{K}^t \Delta \mathbf{w} = \Delta \mathbf{p} \quad (7.13)$$

$$\begin{array}{|c|c|c|} \hline 2m^2 + 3w_1^2 + w_2(2 + w_2) & 2w_1(1 + w_2) & \Delta w_1 \\ \hline 2w_1(1 + w_2) & w_1^2 + 3w_2^2 + 6w_2 + 2 & \Delta w_2 \\ \hline \end{array} = \begin{array}{|c|} \hline \Delta p_1 \\ \hline \Delta p_2 \\ \hline \end{array}$$

\mathbf{K}^t tangent stiffness matrix of the truss

The tangent stiffness matrix for solution 1 in section 2.1 is obtained by substituting the displacement $w_1 = 0$ into expression (7.13):

$$\mathbf{K}_1^t \Delta \mathbf{w} = \Delta \mathbf{p}_2 \quad (7.14)$$

$$\begin{array}{|c|c|c|} \hline 2m^2 + w_2(2 + w_2) & 0 & \Delta w_1 \\ \hline 0 & 3w_2^2 + 6w_2 + 2 & \Delta w_2 \\ \hline \end{array} = \begin{array}{|c|} \hline 0 \\ \hline \Delta p_2 \\ \hline \end{array}$$

\mathbf{K}_1^t tangent stiffness matrix for solution 1

The tangent stiffness matrix for solution 2 in section 2.1 is obtained by substituting the condition $2m^2 + w_1^2 + w_2(2 + w_2) = 0$ into expression (7.13):

$$\mathbf{K}_2^t \Delta \mathbf{w} = \Delta \mathbf{p}_2 \quad (7.15)$$

$$\begin{array}{|c|c|c|} \hline 2w_1^2 & 2w_1(1 + w_2) & \Delta w_1 \\ \hline 2w_1(1 + w_2) & 2(1 + w_2)^2 - 2m^2 & \Delta w_2 \\ \hline \end{array} = \begin{array}{|c|} \hline 0 \\ \hline \Delta p_2 \\ \hline \end{array}$$

$$w_1 = \pm \sqrt{-w_2(2 + w_2) - 2m^2}$$

\mathbf{K}_2^t tangent stiffness matrix for solution 2

7.1.4 Secant Stiffness of a 2-Bar Truss

The total stiffness matrix $\mathbf{K}^{(s)}$ of the truss for a displacement (w_1, w_2) of the apex in step s is given in expression (7.10):

$$\mathbf{K}^{(s)} \mathbf{w} = \mathbf{p} \quad (7.16)$$

If the displacement of the apex is incremented to $(w_1 + \Delta w_1, w_2 + \Delta w_2)$ and the load to $(p_1 + \Delta p_1, p_2 + \Delta p_2)$ in state $s + 1$, the total stiffness matrix $\mathbf{K}^{(s+1)}$ becomes:

$$\mathbf{K}^{(s+1)} (\mathbf{w} + \Delta \mathbf{w}) = \mathbf{p} + \Delta \mathbf{p} \quad (7.17)$$

$$\begin{array}{|c|c|c|} \hline 2m^2 + (w_1 + \Delta w_1)^2 + w_2 + \Delta w_2 & (w_1 + \Delta w_1)(1 + w_2 + \Delta w_2) & w_1 + \Delta w_1 \\ \hline (w_1 + \Delta w_1)(1 + w_2 + \Delta w_2) & (1 + w_2 + \Delta w_2)(2 + w_2 + \Delta w_2) & w_2 + \Delta w_2 \\ \hline \end{array} = \begin{array}{|c|} \hline p_1 + \Delta p_1 \\ \hline p_2 + \Delta p_2 \\ \hline \end{array}$$

The secant stiffness matrix for the step from state s to state $s + 1$ is obtained by subtracting (7.16) from (7.17):

$$\mathbf{K}^s \Delta \mathbf{w} = \Delta \mathbf{p} \quad (7.18)$$

$$\begin{array}{|c|c|c|} \hline k_{11}^s & k_{12}^s & \Delta w_1 \\ \hline k_{21}^s & k_{22}^s & \Delta w_2 \\ \hline \end{array} = \begin{array}{|c|} \hline \Delta p_1 \\ \hline \Delta p_2 \\ \hline \end{array}$$

The stiffness coefficients k_{im}^s are computed in the following steps:

$$p_{11} = k_{11}^{(s+1)}(w_1 + \Delta w_1) - k_{11}^{(s)} w_1 \quad (7.19)$$

$$\begin{aligned} &= (2m^2 + (w_1 + \Delta w_1)^2 + w_2 + \Delta w_2)(w_1 + \Delta w_1) - (2m^2 + w_1^2 + w_2) w_1 \\ &= (2m^2 + 3w_1^2 + w_2 + 3w_1 \Delta w_1 + \Delta w_1^2 + \Delta w_2) \Delta w_1 + w_1 \Delta w_2 \end{aligned}$$

$$p_{22} = k_{22}^{(s+1)}(w_2 + \Delta w_2) - k_{22}^{(s)} w_2 \quad (7.20)$$

$$\begin{aligned} &= (1 + w_2 + \Delta w_2)(2 + w_2 + \Delta w_2)(w_2 + \Delta w_2) - (1 + w_2)(2 + w_2) w_2 \\ &= (2 + 6w_2 + 3w_2^2 + 3(1 + w_2) \Delta w_2 + \Delta w_2^2) \Delta w_2 \end{aligned}$$

$$p_{12} = k_{12}^{(s+1)}(w_2 + \Delta w_2) - k_{12}^{(s)} w_2 \quad (7.21)$$

$$\begin{aligned} &= (w_1 + \Delta w_1)(1 + w_2 + \Delta w_2)(w_2 + \Delta w_2) - w_1(1 + w_2)w_2 \\ &= (w_1(1 + 2w_2) + (1 + w_2)\Delta w_1 + w_1\Delta w_2)\Delta w_2 + (w_2(1 + w_2) + (w_2 + \Delta w_2)\Delta w_2)\Delta w_1 \end{aligned}$$

$$p_{21} = k_{21}^{(s+1)}(w_1 + \Delta w_1) - k_{21}^{(s)} w_1 \quad (7.22)$$

$$\begin{aligned} &= (w_1 + \Delta w_1)(1 + w_2 + \Delta w_2)(w_1 + \Delta w_1) - w_1(1 + w_2)w_1 \\ &= (2w_1(1 + w_2) + (1 + w_2)\Delta w_1 + w_1\Delta w_2)\Delta w_1 + (w_1^2 + \Delta w_1(w_1 + \Delta w_1))\Delta w_2 \end{aligned}$$

Term k_{11}^s is formed by collecting the terms in p_{11} and p_{12} that are multiplied by Δw_1 . Term k_{12}^s is formed by collecting the terms multiplied by Δw_2 . Similarly, term k_{21}^s is formed by collecting the terms in p_{21} and p_{22} that are multiplied by Δw_1 . Term k_{22}^s is formed by collecting the terms multiplied by Δw_2 .

$$k_{11}^s = 2m^2 + 3w_1^2 + w_2(2 + w_2) + 3w_1\Delta w_1 + (1 + w_2)\Delta w_2 + \Delta w_1^2 + \Delta w_2^2 \quad (7.23)$$

$$k_{12}^s = k_{21}^s = 2w_1(1 + w_2) + (1 + w_2)\Delta w_1 + w_1\Delta w_2$$

$$k_{22}^s = 2 + w_1^2 + 6w_2 + 3w_2^2 + w_1\Delta w_1 + 3(1 + w_2)\Delta w_2 + \Delta w_1^2 + \Delta w_2^2$$

The secant stiffness matrix for solution 1 in section 2.1 is obtained by substituting the displacement $w_1 = \Delta w_1 = 0$ into expression (7.23):

$$k_{11}^s = 2m^2 + w_2(2 + w_2) + (1 + w_2)\Delta w_2 + \Delta w_2^2 \quad (7.24)$$

$$k_{12}^s = k_{21}^s = 0$$

$$k_{22}^s = 2 + 6w_2 + 3w_2^2 + 3(1 + w_2)\Delta w_2 + \Delta w_2^2$$

The secant stiffness matrix for solution 2 in section 2.1 is obtained by substituting equation (2.28) into expression (7.23):

$$k_{11}^s = 2w_1^2 + 3w_1\Delta w_1 + (1 + w_2)\Delta w_2 + \Delta w_1^2 + \Delta w_2^2 \quad (7.25)$$

$$k_{12}^s = k_{21}^s = 2w_1(1 + w_2) + (1 + w_2)\Delta w_1 + w_1\Delta w_2$$

$$k_{22}^s = 2 - 6m^2 - 2w_1^2 + w_1\Delta w_1 + 3(1 + w_2)\Delta w_2 + \Delta w_1^2 + \Delta w_2^2$$

Equation (2.28) is applied at the beginning and at the end of the load step and the resulting equations are subtracted:

$$w_1^2 + w_2(2 + w_2) + 2m^2 = 0$$

$$(w_1 + \Delta w_1)^2 + (w_2 + \Delta w_2)(2 + w_2 + \Delta w_2) + 2m^2 = 0$$

$$\Delta w_1^2 + \Delta w_2^2 = -2w_1\Delta w_1 - 2(1 + w_2)\Delta w_2 \quad (7.26)$$

Equation (7.26) is substituted into expressions (7.25):

$$k_{11}^s = 2w_1^2 + w_1\Delta w_1 - (1 + w_2)\Delta w_2 \quad (7.27)$$

$$k_{12}^s = k_{21}^s = 2w_1(1 + w_2) + (1 + w_2)\Delta w_1 + w_1\Delta w_2$$

$$k_{22}^s = 2 - 6m^2 - 2w_1^2 - w_1\Delta w_1 + (1 + w_2)\Delta w_2$$

The influence of the coefficients of the different stiffness matrices on the stability of a structure will be studied in later sections of this chapter.

7.2 SINGULAR CONFIGURATIONS OF STRUCTURES

7.2.1 Concept of a Singular Configuration

Structural stability: The analysis of the stability of a structure is restricted to the investigation of equilibrium states of the structure. If an equilibrium state is perturbed and the cause of the perturbation is then removed, the structure may either return to the equilibrium state, or it may remain in the perturbed state, or the perturbation may grow. If the structure returns to the equilibrium state, the structure is called stable. If the perturbations grow, the structure is called unstable.

Mathematical conditions: The necessary condition for the instability of an instant configuration of a structure is the singularity of its tangent stiffness matrix. The singularity implies that the structure can undergo an infinitesimal displacement \mathbf{du} without an infinitesimal change $d\mathbf{q}$ in the applied load. In general, the infinitesimal displacement will cause an increment $d\mathbf{r}$ in the reactions of the structure.

The singularity of the tangent stiffness matrix is not sufficient for the instability of a structure. Consider, for example, a load that is the product of a load pattern and a load factor. The first derivative of the load factor for displacements in the direction of the infinitesimal displacement \mathbf{du} is null if the tangent stiffness matrix is singular.

If the second derivative of the load factor in the direction of the displacement \mathbf{du} is positive, the load factor increases with the displacement, even though the tangent stiffness matrix of the limit configuration is singular. If the second derivative of the load factor in the direction of \mathbf{du} is negative, the load factor decreases with the displacement. The sign of the second derivative of the load factor determines the stability of the structure. This reasoning is extended to derivatives of higher order if the second derivative is null.

Singular configuration and singular points: A configuration of a structure is called singular if the determinant of the tangent stiffness matrix of the structure for that configuration is null. The point that represents the singular configuration on the load path is called a singular point.

$$\det \mathbf{K}^t = 0 \rightarrow \text{the structure is in a singular configuration} \quad (7.28)$$

\mathbf{K}^t tangent stiffness matrix of the structure

In order to determine whether a singular configuration is stable or unstable, the load path is continued beyond the singular point. If the load factor increases on the continuation, the structure is stable. Otherwise it is unstable.

Snap-through point: A singular point is called a snap-through point (turning point) if there is a single continuation (branch) of the load path beyond the singular point. Point B in figure 6.2 and points B and E in figure 6.3 are snap-through points.

Bifurcation point: A singular point is called a bifurcation point if there is more than one continuation of the load path beyond the singular point. Point B in figure 6.2 and points C and D in figure 6.3 are bifurcation points. The variation of the load factor beyond the singular point determines which of the branches the structure follows in nature.

7.2.2 Singular Configurations of a 2-Bar Truss

The 2-bar truss introduced in section 2.1 is used to illustrate the concept of singular configurations of structures. Governing equations (2.18) and (2.19) of the truss have two solutions: solution S1 in (2.20) and solution S2 in (2.28) and (2.29):

$$\text{S1: } w_1 = 0 \quad (7.29)$$

$$p_1 = 0$$

$$p_2 = w_2(1+w_2)(2+w_2)$$

$$\text{S2: } p_1 = 0 \quad \text{for } m_c \leq \frac{1}{\sqrt{2}} = 0.7071 \quad (7.30)$$

$$p_2 = -2m^2(1+w_2)$$

$$w_1^2 + w_2(2+w_2) + 2m^2 = 0$$

Singular configurations: The truss is in a singular configuration if the tangent stiffness matrix \mathbf{K}^t is singular so that a nontrivial solution $\Delta \mathbf{w}$ exists for the homogeneous equation (7.13). Consider solution S1. Matrix \mathbf{K}_1^t in (7.14) is singular if at least one of its diagonal coefficients is null. The displacement w_2 and the diagonal coefficients on the load path of solution S1 have the following values at the singular points:

$$\text{case 1: } k_{22}^t = 3w_2^2 + 6w_2 + 2 = 0$$

$$w_2 = -1 \pm \frac{1}{\sqrt{3}} \quad (7.31)$$

$$k_{11}^t = 2m^2 - \frac{2}{3}$$

$$\text{case 2: } k_1^t = w_2(2+w_2) + 2m^2 = 0$$

$$w_2 = -1 \pm \sqrt{1-2m^2} \quad (7.32)$$

$$k_{22}^t = 2 - 6m^2$$

The value of the determinant of \mathbf{K}_2^t in (7.15) is $-4w_1^2m^2$. Thus \mathbf{K}_2^t is singular if $w_1 = 0$. The displacements w_2 in (7.31) and (7.32) are equal for a special value m_s of the shape factor:

$$m_s^2 = \frac{1}{3}: \quad m_s = 0.5774 \quad (7.33)$$

$$w_{2s} = -0.4226 \quad (7.34)$$

Load-displacement curves: It is not obvious which of solutions S1 and S2 in (7.31) and (7.32) will control the behaviour of a specific 2-bar truss, or which limit configurations the truss will reach during its displacement. These questions are investigated with the displacement-load curves in figure 7.1, which show the vertical load p_2 as a function of the displacement w_2 .

The load required in solution S2 for equilibrium with displacement $w_2 = 0$ is $p_2 = -2m^2$. The curve for solution S2 therefore does not pass through the reference configuration at origin A of figure 7.1. The load path at the origin must follow curve "a" for solution S1 since it is the only curve that passes through origin A. Shape factor m determines the subsequent displacement as follows:

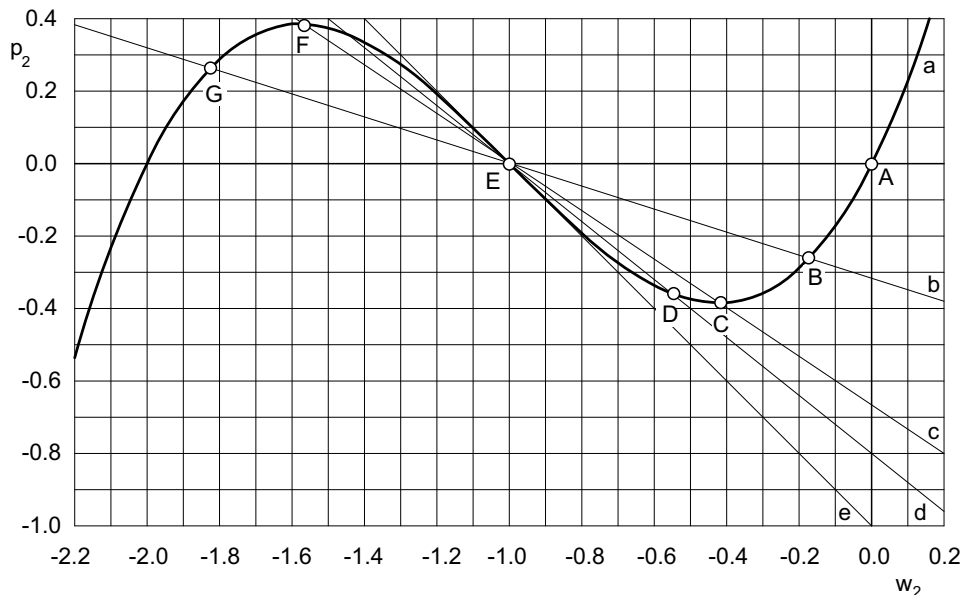


Figure 7.1 Load paths of a 2-bar truss

- | | | | |
|---|--------------------------|---|----------------------|
| a | S1 for all values of m | d | S2 with $m = 0.6325$ |
| b | S2 with $m = 0.4000$ | e | S2 with $m = 0.7071$ |
| c | S2 with $m_s = 0.5774$ | | |

(a) $m > m_c = 0.7071$: Equation (2.40) shows that the displacement w_{2c} for solution S2 is imaginary. The truss can only follow the curve "a" for solution S1 with the displacement $w_1 = 0$. The first singular configuration of the truss is reached at point C where $w_{2c} = -0.422650$ and $p_{2c} = -0.384900$ from (2.21).

Let the first displacement increment past the singular configuration C be $(\Delta w_1, \Delta w_2)$. Since $w_1 = 0$ on the entire curve "a", the displacement increment Δw_1 is null. The truss moves to a neighbouring equilibrium configuration by a displacement increment $(0, \Delta w_2)$. Because there is only one displacement direction in which the truss can move beyond the singular configuration to a neighbouring equilibrium configuration, the singular point C is a snap-through point.

(b) $m \leq 0.7071$: Solutions S1 and S2 are both possible. The two solutions intersect at a point whose coordinates follow from (7.29) and (7.30):

$$w_{2x}(1 + w_{2x})(2 + w_{2x}) = -2m^2(1 + w_{2x}) = p_{2x}$$

$$w_{2x} = -1 \quad \text{or} \quad w_{2x} = -1 + \sqrt{1 - 2m^2} \tag{7.35}$$

$$p_{2x} = 0 \quad \text{or} \quad p_{2x} = -2m^2\sqrt{1 - 2m^2} \tag{7.36}$$

(b1) $0 < m < m_s = 0.5774$: The truss reaches the singular point B for S2 before it reaches the singular point C for S1.

Let $(\Delta w_{1c}, \Delta w_{2c})$ be the first displacement increment that follows after the singular point B. If the load path follows curve "a" of solution S1, the displacement increment Δw_1 is null and the load increment is:

$$\Delta p_2 = (w_2 + \Delta w_2)(1 + w_2 + \Delta w_2)(2 + w_2 + \Delta w_2) - w_2(1 + w_2)(2 + w_2) \quad (7.37)$$

If the load path follows curve "b" of S2, the displacement increment Δw_1 is chosen freely and the load increment is:

$$\Delta p_2 = -2m^2(1 + w_2 + \Delta w_2) + 2m^2(1 + w_2) = -2m^2 \Delta w_2 \quad (7.38)$$

The truss can thus move to an equilibrium configuration that neighbours B either by a displacement increment $(0, \Delta w_{2c})$ on the curve for solution S1, or by a displacement increment $(\Delta w_{1c}, \Delta w_{2c})$ on the curve for solution S2. Figure 7.1 shows that the absolute value of load p_2 for a given displacement w_2 is less for S2 than for S1. Because there is more than one displacement direction in which the truss can move from the singular point B to a neighbouring equilibrium configuration, the singular configuration (w_{2x}, p_{2x}) at B is a bifurcation configuration. Once the load path starts to follow curve "b" towards E, all subsequent equilibrium states are solutions of type S2.

(b2) $m = m_s$: The curves "a" for solution S1 and "c" for solution S2 in figure 7.1 meet in point C which is a singular point for both S1 and S2. The coordinates of singular point C are given by (2.21):

$$(w_{2x}, p_{2x}) = (-0.4226, -0.3849) \quad (7.39)$$

(b3) $m_s < m < 0.7071$: The curves "a" for solution S1 and "d" for solution S2 meet in point D. The singular point C on curve "a" is reached before the point of intersection D. The truss thus snaps through at point C and continues to displace on curve "a" until it reaches a singular point D for S2, which is a bifurcation point.

(b4) If the shape factor of the truss has the limit value $m = 0.7071$, the curves "a" for S1 and "e" for S2 intersect at a single bifurcation point E.

In the following sections, a method of analysis for singular points is presented which accounts for the different cases illustrated by the behaviour of the 2-bar truss. This method has three components: the detection of nearly singular configurations, the exact computation of the singular configuration and the algorithm for the continuation of the load path on a new branch beyond the singular point.

7.3 NEARLY SINGULAR CONFIGURATIONS OF STRUCTURES

7.3.1 Concept of a Nearly Singular Configuration

The stepwise solution procedure in chapter 6 uses the inverse of the submatrix \mathbf{K}_{11} of the incremental governing equations (6.39). If this submatrix is singular, the pattern displacement \mathbf{d}_t in (6.43) and the corrective displacement \mathbf{f}_t in (6.47) cannot be determined. The submatrix \mathbf{K}_{11} is by definition singular at a singular point so that the stepwise solution procedure breaks down in the vicinity of singular points. An additional algorithm is therefore required to solve the three tasks below:

- (a) detection of configurations in which the structure is near a singular point
- (b) computation of the system displacement and reaction vectors at a singular point
- (c) continuation of the load path on a new branch beyond the singular point

In principle, the detection of configurations in which the tangent stiffness matrix of a structure is nearly singular is not difficult. The value of the determinant is computed and compared to a limit value. In numerical practice, however, significant difficulties arise.

Since the determinant of a matrix is equal to the product of its eigenvalues, which can be large, the value of the determinant of the stiffness matrix can lie outside the range of data type double for numbers in the computer. The determinant of the tangent stiffness matrix is therefore not a suitable indicator for singular points.

The determinant of the tangent stiffness matrix will usually change sign at a singular point. However, if the structure is symmetric, an even number of eigenvalues may change their sign at the same singular point. In this case, the sign of the determinant does not change at the singular point. The sign of the determinant is therefore not a suitable indicator for singular points.

If the instability of a structure is local, the stiffness change is restricted to a small portion of the structure. Since the determinant of the incremental stiffness matrix is a property of the structure as a whole, it is not suited as an indicator for such local effects.

Consider the decomposition (6.80) of matrix \mathbf{K}_{11} into the product of a left triangular matrix \mathbf{S} with unit diagonal coefficients, a diagonal matrix \mathbf{D} that can contain positive and negative coefficients d_k , and a right triangular matrix \mathbf{S}^T . The determinant of \mathbf{K}_{11} is equal to the product of the coefficients of \mathbf{D} :

$$\begin{aligned}\mathbf{K}_{11} &= \mathbf{S}\mathbf{D}\mathbf{S}^T \\ \det\mathbf{K}_{11} &= \det\mathbf{S} \det\mathbf{D} \det\mathbf{S}^T = d_0 \dots d_{n-1}\end{aligned}\tag{7.40}$$

In each step of the incremental solution procedure, the diagonal matrix \mathbf{D} is computed. Its diagonal coefficients contain significantly more information than the determinant of the incremental stiffness \mathbf{K}_{11} , which is a single number. \mathbf{D} is therefore used as follows to detect nearly singular configurations of a structure:

(a) Assume that the secant stiffness matrix for the last iteration max of step s has been decomposed in the form (7.41):

$$\mathbf{K}^{s(s,\max)} = \mathbf{S}^{s(s,\max)} \mathbf{D}^{s(s,\max)} \mathbf{S}^{s(s,\max)T} \quad (7.41)$$

Let the diagonal coefficients of matrix \mathbf{D} be $d_k^{s(s,\max)}$.

(b) Assume that the tangent stiffness matrix at the start of load step s + 1 is decomposed in the form (7.42):

$$\mathbf{K}^{t(s+1)} = \mathbf{S}^{t(s+1)} \mathbf{D}^{t(s+1)} \mathbf{S}^{t(s+1)T} \quad (7.42)$$

Let the sign of one or several of the diagonal coefficients in $\mathbf{D}^{s(s,\max)}$ and $\mathbf{D}^{t(s)}$ differ, and assume that the total number of negative coefficients in the two matrices is not equal. Then the structure has passed a singular state in load step s. The computation of the singular point is initiated, using the state of the frame at the beginning of step s as the nearly singular state.

(c) Assume that there is no change in the number of negative diagonal coefficients at the beginning of load step s + 1 for the tangent stiffness matrix described in (b) above. Let the secant matrix for iteration m in step s + 1 be decomposed similar to the secant matrix in step s in the form (7.43):

$$\mathbf{K}^{s(s+1,\max)} = \mathbf{S}^{s(s+1,\max)} \mathbf{D}^{s(s+1,\max)} \mathbf{S}^{s(s+1,\max)T} \quad (7.43)$$

Let the sign of one or several of the diagonal coefficients in $\mathbf{D}^{t(s+1)}$ and $\mathbf{D}^{s(s+1,m)}$ differ and assume that the total number of negative signs in the two matrices is not equal. Then the structure has passed a singular state in load step s + 1. The computation of the singular point is initiated, using the state of the frame at the beginning of step s + 1 as the nearly singular state.

The variation of the diagonal coefficients of the diagonal matrix \mathbf{D} for two-bar trusses is shown in the next section. The computation of the singular points for trusses and frames is treated in section 7.4.

7.3.2 Nearly Singular Configurations of a 2-Bar Truss

The nearly singular configurations of the 2-bar truss of section 2.1 are used as an example for the nearly singular configuration concept. In order to determine the nearly singular states of the truss, the tangent stiffness matrix \mathbf{K}_1^t in (7.14) is decomposed into a product $\mathbf{S}\mathbf{D}\mathbf{S}^T$. The diagonal coefficients of the left triangular matrix \mathbf{S} are 1; the diagonal coefficients of the diagonal matrix \mathbf{D} are denoted by d_m .

$$\mathbf{K}_1^t: \quad d_0 = 2m^2 + w_2(2 + w_2)$$

$$d_1 = 3w_2^2 + 6w_2 + 2$$

The variation of the coefficients d_m of the tangent stiffness matrix \mathbf{K}_1^t with displacement w_2 is shown in figure 7.2. Curve "a" shows the variation of d_1 that is independent of the shape factor m . Curves "b" to "e" show the variation of d_0 for different values of the shape factor m . It can be seen that most of the curves cut the axis $d_m = 0$, but that the curve "b" for $m = 0.7071$ touches the axis. Curves "a" and "c" intersect at $d_m = 0$, since the shape factor $m = 0.5774$ is the shape factor for the peak point.

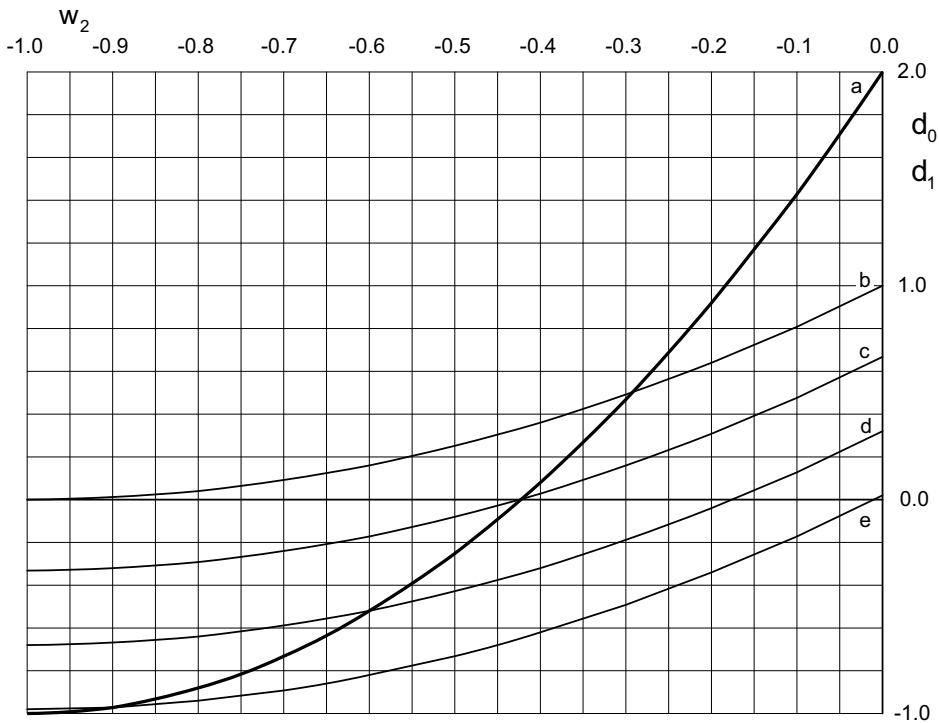


Figure 7.2 Variation of the diagonal coefficients of the diagonal matrix \mathbf{D} for a 2-bar truss with displacement w_2 and shape factor m

- | | | | |
|---|-----------------------------|---|------------------------|
| a | d_1 for all values of m | d | d_0 for $m = 0.4000$ |
| b | d_0 for $m = 0.7071$ | e | d_0 for $m = 0.1000$ |
| c | d_0 for $m_s = 0.5774$ | | |

7.4 DETERMINATION OF SINGULAR CONFIGURATIONS

7.4.1 Determination of the Limit Load Factor

If a structure is near a singular configuration, the solution procedure described in chapter 6 cannot be continued. Two strategies can be followed: either the singular point is avoided empirically or the system displacement vector and the system reaction vector at the singular point are determined. The first approach is simpler and can be used at snap-through points where there is only one direction in which the truss can displace beyond the singular point. The second approach must be used for controlled continuation of a load path beyond a bifurcation point.

A singular point is determined by computing the displacement and reaction vectors for which the tangent stiffness matrix in the governing equation (6.12) is singular. Let A be a nearly singular point that precedes the singular point L. In order to determine the singular point, the displacements and reactions at A are computed with the procedure described in chapter 6. The displacements due to the pattern load and the error vector are computed with the secant matrix for the step from A to L. The displacement increments from the nearly singular point A to the singular point L are expressed as functions of the unknown load factor increment from A to L with (6.46):

$$\Delta \mathbf{d}_{1s} = \Delta \lambda \mathbf{d}_{1t} + \mathbf{f}_1 \quad (7.44)$$

$$\Delta \mathbf{d}_{2s} = \Delta \lambda \mathbf{d}_{2t}$$

$\Delta \lambda$ load factor increment from nearly singular point A to singular point L

$\Delta \mathbf{d}_{1s}$ free system displacement increment from A to L

$\Delta \mathbf{d}_{2s}$ prescribed displacement increment from A to L

\mathbf{d}_{1t} free displacement due to the load pattern

\mathbf{f}_1 displacement correction due to the error vector at point A

\mathbf{d}_{2t} prescribed displacement pattern at the supports

The derivatives of the displacement increments are substituted into the expressions for the tangent element stiffness matrices and for the rotation matrices, as shown for trusses in section 7.4.2 and for frames in section 7.4.3. The resulting expressions for the tangent element stiffness matrices are nonlinear functions of the load factor increment $\Delta \lambda$. The tangent stiffness matrix at point L, which is assembled from the element matrices, is also a nonlinear function of the load factor increment $\Delta \lambda$. The value of $\Delta \lambda$ that drives the determinant of the tangent stiffness matrix at point L to null is determined by iteration.

In iteration 0, the terms that are nonlinear in $\Delta \lambda$ are neglected. The value of $\Delta \lambda$, for which the determinant of the approximate tangent stiffness matrix at point L is null, is determined by solving for the smallest eigenvalue of the general eigenvalue problem (7.46):

$$\det(\mathbf{K}_{s0} + \Delta \lambda \mathbf{K}_{s1}) = 0 \quad (7.45)$$

$$\mathbf{K}_{s0} \mathbf{x} = \Delta \lambda (-\mathbf{K}_{s1}) \mathbf{x} \quad (7.46)$$

\mathbf{K}_{s0} residual stiffness matrix

\mathbf{K}_{s1} modification stiffness matrix

Matrices \mathbf{K}_{s_0} and \mathbf{K}_{s_1} contain rows and columns associated with prescribed displacements and rows associated with free displacements. If the eigenvalue problem (7.46) is solved for the complete stiffness matrix, the eigenstates corresponding to rigid body motions are included in the eigenvalue spectrum. Such eigenstates are not of interest for the stability analysis. The purpose of the stability analysis is to find an eigenstate with eigenvalue null at the singular point L, whose eigenvector is not a rigid body motion of the structure as a whole. The problem is solved by restricting the vector iteration to those rows and columns of the stiffness matrix \mathbf{K}_s^t which form the submatrix \mathbf{K}_{11} in (6.39) and correspond to the free displacement coordinates at the nodes.

Since the eigenstate \mathbf{x} with the smallest absolute eigenvalue is used to drive the determinant of \mathbf{K}_s^t to null, the general eigenvalue problem (7.46) is solved by inverse vector iteration, as described in section 7.8. If state A, for which matrices \mathbf{K}_{s_0} and \mathbf{K}_{s_1} are determined, is close to the singular state L, the smallest eigenvalue $\Delta\lambda$ is small compared to the next larger eigenvalue. Therefore, the iteration for the eigenvalue converges fast.

In cycle 0 of the iteration for the step from the nearly singular point A to the limit point L, the secant stiffness matrix is approximated by the tangent stiffness matrix at A. In the succeeding cycles of the iteration, the displacement increment of cycle $m-1$ is used to approximate the secant stiffness matrix for cycle m , as shown in chapter 6.

In cycle 0 for the step from A to L, the tangent stiffness matrix at L is approximated by the terms that are linear in the load factor increment $\Delta\lambda$. In the succeeding cycles of the iteration, the nonlinear terms in the expressions for the tangent stiffness matrix are approximated with linear terms by using the approximate value of the load factor increment that was determined in the preceding cycle of iteration.

The change in the approximation of the load factor increment changes the coefficients of the modification matrix \mathbf{K}_{s_1} . The algorithm for the solution of the eigenvalue problem is the same as that in cycle 0 of the iteration. The procedure is terminated when the change in the load factor increment is less than a preset limit δ .

Since the displacement increment $\Delta\mathbf{d}$ is approximated in expression (7.44) by a linear function of the load factor increment $\Delta\lambda$, the computed point L is not the exact singular point of the structure. The procedure is repeated with the computed point L as the new nearly singular point. The algorithm terminates when the change in the load factor increment is less than a prescribed limit value.

7.4.2 Singular Points for Trusses

The residual matrix \mathbf{K}_{s_0} and the modification matrix \mathbf{K}_{s_1} in (7.46) will now be determined for a typical bar of a plane truss. The increment in the element displacement vector due to the displacement (7.44) is given by (4.37):

$$\Delta\mathbf{v}_e = \mathbf{R}_{esy}^T \mathbf{T}_e \Delta\mathbf{d}_s \quad (7.47)$$

$\Delta\mathbf{v}_e$ increment of the element displacement vector in reference space

The derivatives of the displacement increment $\Delta \mathbf{v}_e$ with respect to the axial coordinate y_1 are obtained by substitution of (6.46) and (7.47) into expressions (4.64):

$$\Delta v_{m,1} = \frac{1}{a} \mathbf{g}_m^T \mathbf{R}_{\text{esy}}^T \mathbf{T}_e (\Delta \lambda \mathbf{d}_t + \mathbf{f})$$

$$\Delta v_{m,1} = \Delta \lambda d_{m1} + d_{m2} \quad (7.48)$$

$$d_{m1} := \frac{1}{a} \mathbf{g}_m^T \mathbf{R}_{\text{esy}}^T \mathbf{T}_e \mathbf{d}_t \quad (7.49)$$

$$d_{m2} := \frac{1}{a} \mathbf{g}_m^T \mathbf{R}_{\text{esy}}^T \mathbf{T}_e \mathbf{f} \quad (7.50)$$

The derivatives of the node displacement of the truss at the singular point with respect to y_1 are denoted by $\bar{v}_{m,1}$ and expressed with (7.48) as functions of the load factor increment $\Delta \lambda$:

$$\bar{v}_{m,1} = v_{m,1} + \Delta v_{m,1} = d_{m0} + \Delta \lambda d_{m1} \quad (7.51)$$

$$d_{m0} = v_{m,1} + d_{m2} \quad (7.52)$$

$\bar{v}_{m,1}$ derivative of v_m with respect to y_1 at the singular point

The components $\bar{\mathbf{K}}_d$ and $\bar{\mathbf{K}}_g$ of the tangential stiffness matrix $\bar{\mathbf{K}}_e$ at the singular point are determined by substitution of (7.51) into (4.92) and (4.93):

$$\bar{\mathbf{K}}_d = \frac{AE}{a} (c_1 \mathbf{G}_{11} + c_2 \mathbf{G}_{22} + c_3 \mathbf{G}_{12}) \quad (7.53)$$

$$c_1 = (d_{10} + \Delta \lambda d_{11})(2 + d_{10} + \Delta \lambda d_{11})$$

$$c_2 = (d_{20} + \Delta \lambda d_{21})^2$$

$$c_3 = (d_{20} + \Delta \lambda d_{21})(1 + d_{10} + \Delta \lambda d_{11})$$

$$\bar{\mathbf{K}}_g = \frac{AE}{a} c_4 (\mathbf{G}_{11} + \mathbf{G}_{22}) \quad (7.54)$$

$$c_4 = (d_{10} + \Delta \lambda d_{11}) + \frac{1}{2}(d_{10} + \Delta \lambda d_{11})^2 + \frac{1}{2}(d_{20} + \Delta \lambda d_{21})^2$$

The components \mathbf{K}_m and \mathbf{K}_n in (4.94) and (4.95) are not part of the tangential element stiffness matrix, which is a quadratic function of the load factor increment $\Delta \lambda$:

$$\bar{\mathbf{K}}_e = \mathbf{K}_{e0} + \Delta \lambda \mathbf{K}_{e1} + \Delta \lambda^2 \mathbf{K}_{e2} \quad (7.55)$$

$$\mathbf{K}_{e0} = \mathbf{K}_c + \mathbf{K}_{d0} + \mathbf{K}_{g0}$$

$$\mathbf{K}_{e1} = \mathbf{K}_{d1} + \mathbf{K}_{g1}$$

$$\mathbf{K}_{e2} = \mathbf{K}_{d2} + \mathbf{K}_{g2}$$

The deviatoric stiffness matrix $\bar{\mathbf{K}}_d$ contributes the following terms to $\bar{\mathbf{K}}_e$:

$$\mathbf{K}_{d0} = \frac{AE}{a} (b_1 \mathbf{G}_{11} + b_2 \mathbf{G}_{22} + b_3 \mathbf{G}_{12}) \quad (7.56)$$

$$b_1 = d_{10}(2 + d_{10}) \quad b_2 = d_{20}^2 \quad b_3 = d_{20}(1 + d_{10})$$

$$\mathbf{K}_{d1} = \frac{AE}{a}(b_4 \mathbf{G}_{11} + b_5 \mathbf{G}_{22} + b_6 \mathbf{G}_{12}) \quad (7.57)$$

$$b_4 = 2d_{11}(1+d_{10}) \quad b_5 = 2d_{20}d_{21} \quad b_6 = d_{21}(1+d_{10}) + d_{11}d_{20}$$

$$\mathbf{K}_{d2} = \frac{AE}{a}(b_7 \mathbf{G}_{11} + b_8 \mathbf{G}_{22} + b_9 \mathbf{G}_{12}) \quad (7.58)$$

$$b_7 = d_{11}^2 \quad b_8 = d_{21}^2 \quad b_9 = d_{11}d_{21}$$

The geometric stiffness matrix $\bar{\mathbf{K}}_g$ contributes the following terms to $\bar{\mathbf{K}}_e$:

$$\mathbf{K}_{g0} = \frac{AE}{a}e_0(\mathbf{G}_{11} + \mathbf{G}_{22}) \quad (7.59)$$

$$e_0 = d_{10} + \frac{1}{2}d_{10}^2 + \frac{1}{2}d_{20}^2$$

$$\mathbf{K}_{g1} = \frac{AE}{a}e_1(\mathbf{G}_{11} + \mathbf{G}_{22}) \quad (7.60)$$

$$e_1 = d_{11} + d_{10}d_{11} + d_{20}d_{21}$$

$$\mathbf{K}_{g2} = \frac{AE}{a}e_2(\mathbf{G}_{11} + \mathbf{G}_{22}) \quad (7.61)$$

$$e_2 = \frac{1}{2}(d_{11}^2 + d_{21}^2)$$

Expressions (4.89) and (7.53) to (7.61) lead to the components of $\bar{\mathbf{K}}_e$:

$$\mathbf{K}_{e0} = \frac{EA}{a} \begin{array}{|c|c|c|c|} \hline k_{01} & k_{03} & -k_{01} & -k_{03} \\ \hline k_{03} & k_{02} & -k_{03} & -k_{02} \\ \hline -k_{01} & -k_{03} & k_{01} & k_{03} \\ \hline -k_{03} & -k_{02} & k_{03} & k_{02} \\ \hline \end{array} \quad (7.62)$$

$$k_{01} = 1 + b_1 + e_0 \quad k_{02} = b_2 + e_0 \quad k_{03} = b_3$$

$$\mathbf{K}_{e1} = \frac{EA}{a} \begin{array}{|c|c|c|c|} \hline k_{11} & k_{13} & -k_{11} & -k_{13} \\ \hline k_{13} & k_{12} & -k_{13} & -k_{12} \\ \hline -k_{11} & -k_{13} & k_{11} & k_{13} \\ \hline -k_{13} & -k_{12} & k_{13} & k_{12} \\ \hline \end{array} \quad (7.63)$$

$$k_{11} = b_4 + e_1 \quad k_{12} = b_5 + e_1 \quad k_{13} = b_6$$

$$\mathbf{K}_{e2} = \frac{EA}{a} \begin{array}{|c|c|c|c|} \hline k_{21} & k_{23} & -k_{21} & -k_{23} \\ \hline k_{23} & k_{22} & -k_{23} & -k_{22} \\ \hline -k_{21} & -k_{23} & k_{21} & k_{23} \\ \hline -k_{23} & -k_{22} & k_{23} & k_{22} \\ \hline \end{array} \quad (7.64)$$

$$k_{21} = b_7 + e_2 \quad k_{22} = b_8 + e_2 \quad k_{23} = b_9$$

The component system matrices \mathbf{K}_i for the incremental system stiffness matrix are assembled from the element matrices, as described in chapter 6:

$$\mathbf{K}_i = \sum_e \mathbf{T}_e^T \mathbf{R}_{esy} \mathbf{K}_{ei} \mathbf{R}_{esy}^T \mathbf{T}_e \quad i \in \{0,1,2\} \quad (7.65)$$

The residual and modification stiffness matrices in (7.46) are computed with the approximate load factor increment $\Delta\bar{\lambda}$:

$$\mathbf{K}_{s0} = \mathbf{K}_0 \quad \mathbf{K}_{s1} = \mathbf{K}_1 + \Delta\bar{\lambda} \mathbf{K}_2 \quad (7.66)$$

The load factor increment for the singular point is determined iteratively, as follows:

Cycle 0: Only the terms that are linear in $\Delta\lambda^{(0)}$ are retained in the system stiffness matrix:

$$\bar{\mathbf{K}}^{(0)} = \mathbf{K}_0 + \Delta\lambda^{(0)} \mathbf{K}_1^{(0)} \quad \text{with } \mathbf{K}_1^{(0)} = \mathbf{K}_1 \quad (7.67)$$

$$\mathbf{K}_1 = \sum_e \mathbf{T}_e^T \mathbf{R}_{esy} \mathbf{K}_{e1} \mathbf{R}_{esy}^T \mathbf{T}_e$$

The eigenvalue problem (7.46) is solved for the matrix pair $(\mathbf{K}_0, \mathbf{K}_1^{(0)})$ to yield $\Delta\lambda^{(0)}$.

Cycle m: The nonlinear term $\Delta\lambda^{(m)} \Delta\lambda^{(m)}$ is approximated by $\Delta\lambda^{(m)} \Delta\lambda^{(m-1)}$:

$$\bar{\mathbf{K}}^{(m)} = \mathbf{K}_0 + \Delta\lambda^{(m)} \mathbf{K}_1^{(m)} \quad \text{with } \mathbf{K}_1^{(m)} = \mathbf{K}_1 + \Delta\lambda^{(m-1)} \mathbf{K}_2 \quad (7.68)$$

$$\mathbf{K}_2 = \sum_e \mathbf{T}_e^T \mathbf{R}_{esy} \mathbf{K}_{e2} \mathbf{R}_{esy}^T \mathbf{T}_e$$

The eigenvalue problem (7.46) is solved for the matrix pair $(\mathbf{K}_0, \mathbf{K}_1^{(m)})$ to yield $\Delta\lambda^{(m)}$.

State of the truss at the singular point: The iteration is terminated if the absolute value of the change in the load factor increment $\Delta\lambda$ in a cycle is less than a prescribed limit value. The load factor increment is then used to compute the displacements at the limit point:

$$\Delta \mathbf{d}^{(A)} = \Delta\lambda^{(A)} \mathbf{d}_t^{(A)} \quad (7.69)$$

$$\mathbf{d}^{(L)} = \mathbf{d}^{(A)} + \Delta \mathbf{d}^{(A)}$$

$\mathbf{d}^{(L)}$ system displacement vector at the singular point L

$\mathbf{d}^{(A)}$ system displacement vector at the nearly singular point A

$\mathbf{d}_t^{(A)}$ pattern displacement at the nearly singular A

$\Delta \mathbf{d}^{(A)}$ displacement increment from nearly singular point A to singular point L

This completes the stability analysis for trusses.

7.4.3 Singular Points for Frames

Residual matrix \mathbf{K}_{s0} and modification matrix \mathbf{K}_{s1} in (7.46) are determined as sums of the contributions of the members of a plane frame. The increment in the element displacement vector due to displacement (7.44) is given by (5.76) and (6.46):

$$\Delta \mathbf{w}_e = \mathbf{R}_{esz}^T \mathbf{T}_e \Delta \mathbf{d}_s = \mathbf{R}_{esz}^T \mathbf{T}_e (\Delta\lambda \mathbf{d}_t + \mathbf{f}) \quad (7.70)$$

$\Delta \mathbf{w}_e$ increment of the element displacement vector in instant space

The coefficients with indices 2 and 5 of the increment $\Delta \tilde{\mathbf{w}}_e$ of the modified element displacement vector in (5.158) may contain arbitrary values of the rotations. The derivatives of the increments of the chord displacement with respect to y_1 are obtained by substituting (7.70) into (5.160). The increments of the relative rotations at the nodes are the differences between the total rotation increments at the nodes and the increment $\Delta \theta = \Delta w_{c2,1}$ of the rotation of the chord.

$$\Delta \mathbf{w}_{c1,1} = \mathbf{d}_{11} \Delta \lambda + \mathbf{d}_{12} \quad (7.71)$$

$$\mathbf{d}_{11} = \frac{1}{a} \mathbf{g}_1^T \mathbf{R}_{esz}^T \mathbf{T}_e \mathbf{d}_t \quad \mathbf{d}_{12} = \frac{1}{a} \mathbf{g}_1^T \mathbf{R}_{esz}^T \mathbf{T}_e \mathbf{f}$$

$$\Delta \mathbf{w}_{c2,1} = \mathbf{d}_{21} \Delta \lambda + \mathbf{d}_{22} \quad (7.72)$$

$$\mathbf{d}_{21} = \frac{1}{a} \mathbf{g}_2^T \mathbf{R}_{esz}^T \mathbf{T}_e \mathbf{d}_t \quad \mathbf{d}_{22} = \frac{1}{a} \mathbf{g}_2^T \mathbf{R}_{esz}^T \mathbf{T}_e \mathbf{f}$$

$$\Delta \gamma_A = \Delta \beta_A - \Delta \theta = \mathbf{d}_{31} \Delta \lambda + \mathbf{d}_{32} \quad (7.73)$$

$$\Delta \gamma_B = \Delta \beta_B - \Delta \theta = \mathbf{d}_{41} \Delta \lambda + \mathbf{d}_{42}$$

$$\mathbf{d}_{31} = \mathbf{e}_3^T \mathbf{R}_{esz}^T \mathbf{T}_e \mathbf{d}_t - \mathbf{d}_{21} \quad \mathbf{d}_{32} = \mathbf{e}_3^T \mathbf{R}_{esz}^T \mathbf{T}_e \mathbf{f} - \mathbf{d}_{22}$$

$$\mathbf{d}_{41} = \mathbf{e}_6^T \mathbf{R}_{esz}^T \mathbf{T}_e \mathbf{d}_t - \mathbf{d}_{21} \quad \mathbf{d}_{42} = \mathbf{e}_6^T \mathbf{R}_{esz}^T \mathbf{T}_e \mathbf{f} - \mathbf{d}_{22}$$

The first derivatives of the chord displacement at the singular point are expressed with (7.71) to (7.73) as functions of the load factor increment $\Delta \lambda$:

$$\bar{\mathbf{w}}_{c1,1} = \mathbf{w}_{c1,1} + \Delta \mathbf{w}_{c1,1} = \mathbf{d}_{11} \Delta \lambda + \mathbf{d}_{10} \quad (7.74)$$

$$\bar{\mathbf{w}}_{c2,1} = \mathbf{w}_{c2,1} + \Delta \mathbf{w}_{c2,1} = \mathbf{d}_{21} \Delta \lambda + \mathbf{d}_{20} \quad (7.75)$$

$$\bar{\gamma}_A = \gamma_A + \Delta \gamma_A = \mathbf{d}_{31} \Delta \lambda + \mathbf{d}_{30} \quad (7.76)$$

$$\bar{\gamma}_B = \gamma_B + \Delta \gamma_B = \mathbf{d}_{41} \Delta \lambda + \mathbf{d}_{40}$$

$$\mathbf{d}_{10} = \mathbf{w}_{c1,1} + \mathbf{d}_{12} \quad \mathbf{d}_{20} = \mathbf{w}_{c2,1} + \mathbf{d}_{22} \quad \mathbf{d}_{30} = \gamma_A + \mathbf{d}_{32} \quad \mathbf{d}_{40} = \gamma_B + \mathbf{d}_{42}$$

The tangent stiffness matrix $\bar{\mathbf{K}}_e$ at the singular point follows from (5.203):

$$\bar{\mathbf{K}}_e = \bar{\mathbf{K}}_{ac} + \bar{\mathbf{K}}_{ag} + \bar{\mathbf{K}}_{bc} + \bar{\mathbf{K}}_{bg} \quad (7.77)$$

The components of the tangent stiffness matrix are determined by substitution of (7.74) to (7.76) into expressions (5.205) to (5.210). The nonlinear components \mathbf{K}_{am} and \mathbf{K}_{an} are not contained in the tangent stiffness matrix $\bar{\mathbf{K}}_e$:

$$\bar{\mathbf{K}}_{ac} = \frac{AE}{a} c_0 \mathbf{G}_{11} \quad (7.78)$$

$$c_0 = (1 - \mathbf{d}_{10} - \mathbf{d}_{11} \Delta \lambda)^2$$

$$\bar{\mathbf{K}}_{ag} = \frac{AE}{a}(c_1 \mathbf{G}_{11} + c_2 \mathbf{G}_{22}) \quad (7.79)$$

$$c_1 = -(d_{10} + \Delta\lambda d_{11}) + \frac{1}{2}(d_{10} + \Delta\lambda d_{11})^2 + \frac{1}{2}(d_{20} + \Delta\lambda d_{21})^2$$

$$c_2 = (1 - d_{10} - \Delta\lambda d_{11}) \left((d_{10} + \Delta\lambda d_{11}) - \frac{1}{2}(d_{10} + \Delta\lambda d_{11})^2 - \frac{1}{2}(d_{20} + \Delta\lambda d_{21})^2 \right)$$

$$\bar{\mathbf{K}}_{bc} = \mathbf{K}_{bc} \quad (7.80)$$

$$\bar{\mathbf{K}}_{bg} = \frac{JE}{a^3} c_{10} \mathbf{G}_{22} \quad (7.81)$$

$$c_{10} = -4 \left(\frac{\hat{a}}{a} \right)^2 \left((d_{30} + \Delta\lambda d_{31})^2 + (d_{30} + \Delta\lambda d_{31})(d_{40} + \Delta\lambda d_{41}) + (d_{40} + \Delta\lambda d_{41})^2 \right)$$

The tangent element stiffness matrix $\bar{\mathbf{K}}_e$ at the singular point is a polynomial in the load factor increment $\Delta\lambda$:

$$\bar{\mathbf{K}}_e = \mathbf{K}_{e0} + \Delta\lambda \mathbf{K}_{e1} + \Delta\lambda^2 \mathbf{K}_{e2} + \Delta\lambda^3 \mathbf{K}_{e3} \quad (7.82)$$

$$\mathbf{K}_{e0} = \mathbf{K}_{ac0} + \mathbf{K}_{ag0} + \mathbf{K}_{bc} + \mathbf{K}_{bg0}$$

$$\mathbf{K}_{e1} = \mathbf{K}_{ac1} + \mathbf{K}_{ag1} + \mathbf{K}_{bg1}$$

$$\mathbf{K}_{e2} = \mathbf{K}_{ac2} + \mathbf{K}_{ag2} + \mathbf{K}_{bg2}$$

$$\mathbf{K}_{e3} = \mathbf{K}_{ag3}$$

The term \mathbf{K}_{bc} is given by (5.210). The contribution of component $\bar{\mathbf{K}}_{ac}$ to matrix $\bar{\mathbf{K}}_e$ is:

$$\mathbf{K}_{ac0} = \frac{AE}{a} b_1 \mathbf{G}_{11} \quad b_1 = (1 - d_{10})^2 \quad (7.83)$$

$$\mathbf{K}_{ac1} = \frac{AE}{a} b_2 \mathbf{G}_{11} \quad b_2 = -2d_{11}(1 - d_{10})$$

$$\mathbf{K}_{ac2} = \frac{AE}{a} b_3 \mathbf{G}_{11} \quad b_3 = d_{11}^2$$

The contribution of component $\bar{\mathbf{K}}_{ag}$ to matrix $\bar{\mathbf{K}}_e$ is:

$$\mathbf{K}_{ag0} = \frac{AE}{a} (b_4 \mathbf{G}_{11} + b_5 \mathbf{G}_{22}) \quad (7.84)$$

$$b_4 = -d_{10} + \frac{1}{2}d_{10}^2 + \frac{1}{2}d_{20}^2$$

$$b_5 = -(1 - d_{10})b_4$$

$$\mathbf{K}_{ag1} = \frac{AE}{a} (b_6 \mathbf{G}_{11} + b_7 \mathbf{G}_{22}) \quad (7.85)$$

$$b_6 = -d_{11}(1 - d_{10}) + d_{20}d_{21}$$

$$b_7 = d_{11}b_4 - (1 - d_{10})b_6$$

$$\mathbf{K}_{g2} = \frac{AE}{a} (b_8 \mathbf{G}_{11} + b_9 \mathbf{G}_{22}) \quad (7.86)$$

$$b_8 = \frac{1}{2} d_{11}^2 + \frac{1}{2} d_{21}^2$$

$$b_9 = d_{11} b_6 - (1 - d_{10}) b_8$$

$$\mathbf{K}_{g3} = \frac{AE}{a} (b_{10} \mathbf{G}_{22}) \quad (7.87)$$

$$b_{10} = d_{11} b_8$$

The contribution of component $\bar{\mathbf{K}}_{bg}$ to matrix $\bar{\mathbf{K}}_e$ is:

$$\mathbf{K}_{bg0} = \frac{JE}{a^3} b_{11} \mathbf{G}_{22} \quad b_{11} = -4 \left(\frac{\hat{a}}{a} \right)^2 (d_{30}^2 + d_{30} d_{40} + d_{40}^2) \quad (7.88)$$

$$\mathbf{K}_{bg1} = \frac{JE}{a^3} b_{12} \mathbf{G}_{22} \quad b_{12} = -4 \left(\frac{\hat{a}}{a} \right)^2 (2d_{30} d_{31} + d_{30} d_{41} + d_{31} d_{40} + 2d_{40} d_{41})$$

$$\mathbf{K}_{bg2} = \frac{JE}{a^3} b_{13} \mathbf{G}_{22} \quad b_{13} = -4 \left(\frac{\hat{a}}{a} \right)^2 (d_{31}^2 + d_{31} d_{41} + d_{41}^2)$$

Expressions (7.83) to (7.88) lead to the components of $\bar{\mathbf{K}}_e$ in (7.82):

$$\mathbf{K}_{e0} = \mathbf{K}_{bc} + \begin{array}{|c|c|c|c|c|c|} \hline a_0 & 0 & 0 & -a_0 & 0 & 0 \\ \hline 0 & a_1 & 0 & 0 & -a_1 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 \\ \hline -a_0 & 0 & 0 & a_0 & 0 & 0 \\ \hline 0 & -a_1 & 0 & 0 & a_1 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 \\ \hline \end{array} \quad (7.89)$$

$$a_0 = \frac{AE}{a} (b_1 + b_4) \quad a_1 = \frac{AE}{a} b_5 + \frac{JE}{a^3} b_{11}$$

$$\mathbf{K}_{e1} = \begin{array}{|c|c|c|c|c|c|} \hline a_2 & 0 & 0 & -a_2 & 0 & 0 \\ \hline 0 & a_3 & 0 & 0 & -a_3 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 \\ \hline -a_2 & 0 & 0 & a_2 & 0 & 0 \\ \hline 0 & -a_3 & 0 & 0 & a_3 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 \\ \hline \end{array} \quad (7.90)$$

$$a_2 = \frac{AE}{a} (b_2 + b_6) \quad a_3 = \frac{AE}{a} b_7 + \frac{JE}{a^3} b_{12}$$

$$\mathbf{K}_{e2} = \begin{array}{|c|c|c|c|c|c|} \hline a_4 & 0 & 0 & -a_4 & 0 & 0 \\ \hline 0 & a_5 & 0 & 0 & -a_5 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 \\ \hline -a_4 & 0 & 0 & a_4 & 0 & 0 \\ \hline 0 & -a_5 & 0 & 0 & a_5 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 \\ \hline \end{array} \quad (7.91)$$

$$a_4 = \frac{AE}{a} (b_3 + b_8) \quad a_5 = \frac{AE}{a} b_9 + \frac{JE}{a^3} b_{13}$$

$$\mathbf{K}_{e3} = \begin{array}{|c|c|c|c|c|c|} \hline 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & a_6 & 0 & 0 & -a_6 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & -a_6 & 0 & 0 & a_6 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 \\ \hline \end{array} \quad (7.92)$$

$$a_6 = \frac{AE}{a} b_{10}$$

The component system matrices for the incremental stiffness matrix are assembled from the element matrices, as described in chapter 6. The contribution of the incremental stiffness matrix $\bar{\mathbf{K}}_e$ of the element to the incremental stiffness matrix $\bar{\mathbf{K}}_s$ of the frame at the singular point is given by (5.217):

$$d\bar{\mathbf{K}}_s = \mathbf{T}_e^T \bar{\mathbf{R}}_{esz} \bar{\mathbf{K}}_e \bar{\mathbf{R}}_{esz}^T \mathbf{T}_e \quad (7.93)$$

$\bar{\mathbf{R}}_{esz}$ element rotation matrix support - instant space at the singular point

The element rotation matrix $\bar{\mathbf{R}}_{esz}$ for the transformation between instant and support space is derived from the rotation matrices $\bar{\mathbf{R}}_{nsz}$ at nodes A and B of the member:

$$\bar{\mathbf{R}}_{esz} = \begin{array}{|c|c|} \hline \bar{\mathbf{R}}_{nsz}^{(A)} & \mathbf{0} \\ \hline \mathbf{0} & \bar{\mathbf{R}}_{nsz}^{(B)} \\ \hline \end{array} \quad (7.94)$$

$$\bar{\mathbf{R}}_{nsz} = \mathbf{R}_{nxs}^T \mathbf{R}_{nxy} \bar{\mathbf{R}}_{nyz} \quad (7.95)$$

$\bar{\mathbf{R}}_{nsz}$ node rotation matrix support - instant space at the singular point

\mathbf{R}_{nxs} node rotation matrix global - support space

\mathbf{R}_{nxy} node rotation matrix global - reference space

$\bar{\mathbf{R}}_{nyz}$ node rotation matrix reference - instant space at the singular point

The transformation between global and reference space is given by (5.18):

$$\mathbf{u}_n = \mathbf{R}_{nxy} \mathbf{v}_n \quad (7.96)$$

If the angle from the global axis x_1 to the support axis s_1 in figure 5.2 is denoted by ϕ , the transformation (5.21) between global to support space is given by:

$$\mathbf{u}_n = \mathbf{R}_{nxs} \mathbf{d}_n \quad (7.97)$$

$$\begin{bmatrix} u_1 \\ u_2 \\ \beta \end{bmatrix} = \begin{bmatrix} \cos \phi & -\sin \phi & 0 \\ \sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The coordinates of the transformation matrix $\bar{\mathbf{R}}_{nyz}$ between reference and instant space at the singular point are given in terms of the angle of rotation $\bar{\theta}$ by (5.50):

$$\mathbf{v}_n = \bar{\mathbf{R}}_{nyz} \bar{\mathbf{w}}_n \quad (7.98)$$

$$\begin{bmatrix} v_1 \\ v_2 \\ \beta \end{bmatrix} = \begin{bmatrix} \cos \bar{\theta} & -\sin \bar{\theta} & 0 \\ \sin \bar{\theta} & \cos \bar{\theta} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \bar{w}_1 \\ \bar{w}_2 \\ \beta \end{bmatrix}$$

$\bar{\theta}$ angle from axis y_1 to axis z_1 at the singular point

\bar{w}_i displacement coordinates in instant space at the singular point

The rotation matrices \mathbf{R}_{nxs} and \mathbf{R}_{nxy} are independent of the frame displacements. The variation of the rotation matrix \mathbf{R}_{nyz} with the rotation θ is approximated:

$$\bar{\mathbf{R}}_{nyz}(\theta + \Delta\theta) \cong \mathbf{R}_{nyz}(\theta) + \Delta\theta \frac{d\mathbf{R}_{nyz}(\theta)}{d\theta} + \frac{1}{2} \Delta\theta^2 \frac{d^2\mathbf{R}_{nyz}(\theta)}{d\theta^2} \quad (7.99)$$

$$\bar{\mathbf{R}}_{nyz}(\theta + \Delta\theta) = \left(1 - \frac{1}{2} \Delta\theta^2\right) \mathbf{R}_{n1} + \Delta\theta \mathbf{R}_{n2}$$

$$\mathbf{R}_{n1} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{R}_{n2} = \begin{bmatrix} -\sin \theta & -\cos \theta & 0 \\ \cos \theta & -\sin \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

θ angle from axis y_1 to axis z_1 at the nearly singular point

The approximation of the element rotation matrix is obtained by substituting (7.95) and (7.99) into the element rotation matrix (7.94):

$$\bar{\mathbf{R}}_{esz} = \left(1 - \frac{1}{2} \Delta\theta^2\right) \mathbf{R}_{e1} + \Delta\theta \mathbf{R}_{e2} \quad (7.100)$$

$$\mathbf{R}_{e1} = \begin{bmatrix} \mathbf{R}_{nxs}^{(A)T} \mathbf{R}_{nxy} \mathbf{R}_{n1} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_{nxs}^{(B)T} \mathbf{R}_{nxy} \mathbf{R}_{n1} \end{bmatrix}$$

$$\mathbf{R}_{e2} = \begin{bmatrix} \mathbf{R}_{nxs}^{(A)T} \mathbf{R}_{nxy} \mathbf{R}_{n2} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_{nxs}^{(B)T} \mathbf{R}_{nxy} \mathbf{R}_{n2} \end{bmatrix}$$

The change $\Delta\theta$ in the angle of chord rotation θ from the nearly singular point to the singular point is given by (5.78). The derivative $\Delta w_{c2,1}$ is substituted from (7.72):

$$\Delta\theta = \frac{\Delta w_{2B} - \Delta w_{2A}}{a} = \Delta w_{c2,1} \quad (7.101)$$

$$\Delta\theta = d_{21} \Delta\lambda + d_{22} \quad (7.102)$$

The increment $\Delta\theta$ is substituted from (7.102) into (7.100):

$$\begin{aligned}\bar{\mathbf{R}}_{\text{esz}} &= \left(1 - \frac{1}{2}(d_{21}\Delta\lambda + d_{22})^2\right)\mathbf{R}_{e1} + (d_{21}\Delta\lambda + d_{22})\mathbf{R}_{e2} \\ \bar{\mathbf{R}}_{\text{esz}} &= \mathbf{A}_0 + \Delta\lambda\mathbf{A}_1 + \Delta\lambda^2\mathbf{A}_2 \quad (7.103) \\ \mathbf{A}_0 &= \left(1 - \frac{1}{2}d_{22}^2\right)\mathbf{R}_{e1} + d_{22}\mathbf{R}_{e2} \\ \mathbf{A}_1 &= -d_{21}d_{22}\mathbf{R}_{e1} + d_{21}\mathbf{R}_{e2} \\ \mathbf{A}_2 &= -\frac{1}{2}d_{21}^2\mathbf{R}_{e1}\end{aligned}$$

The load factor increment for the singular point is determined by iteration, as follows:

Cycle 0:

Only the linear terms in $\Delta\lambda$ are retained in the expressions:

$$\bar{\mathbf{K}}_e^{(0)} = \mathbf{K}_{e0} + \Delta\lambda^{(0)}\mathbf{K}_{e1}^{(0)} \quad \text{with} \quad \mathbf{K}_{e1}^{(0)} = \mathbf{K}_{e1} \quad (7.104)$$

$$\bar{\mathbf{R}}_{\text{esz}}^{(0)} = \mathbf{A}_0 + \Delta\lambda^{(0)}\mathbf{A}_1^{(0)} \quad \text{with} \quad \mathbf{A}_1^{(0)} = \mathbf{A}_1 \quad (7.105)$$

$$\bar{\mathbf{R}}_{\text{esz}}^{(0)} \bar{\mathbf{K}}_e^{(0)} (\bar{\mathbf{R}}_{\text{esz}}^{(0)})^T = (\mathbf{A}_0 + \Delta\lambda^{(0)}\mathbf{A}_1^{(0)}) (\mathbf{K}_{e0} + \Delta\lambda^{(0)}\mathbf{K}_{e1}^{(0)}) (\mathbf{A}_0 + \Delta\lambda^{(0)}\mathbf{A}_1^{(0)})^T \quad (7.106)$$

Only the terms linear in $\Delta\lambda$ are retained in the product (7.106):

$$\bar{\mathbf{R}}_{\text{esz}}^{(0)} \bar{\mathbf{K}}_e^{(0)} (\bar{\mathbf{R}}_{\text{esz}}^{(0)})^T = \mathbf{H}_0 + \Delta\lambda^{(0)}\mathbf{H}_1^{(0)} \quad (7.107)$$

$$\mathbf{H}_0 = \mathbf{A}_0 \mathbf{K}_{e0} \mathbf{A}_0^T$$

$$\mathbf{H}_1^{(0)} = \mathbf{A}_1 \mathbf{K}_{e0} \mathbf{A}_0^T + \mathbf{A}_0 \mathbf{K}_{e0} \mathbf{A}_1^T + \mathbf{A}_0 \mathbf{K}_{e1} \mathbf{A}_0^T$$

The element matrices are assembled as described in chapter 6 to determine the system matrix $\bar{\mathbf{K}}_s$ at the singular point as a function of the load factor increment $\Delta\lambda$. The general eigenvalue problem (7.46) is solved for the load factor increment $\Delta\lambda^{(0)}$.

Cycle m:

The nonlinear term $(\Delta\lambda^{(m)})^k$ is approximated by the term $\Delta\lambda^{(m)} (\Delta\lambda^{(m-1)})^{k-1}$ which is linear in $\Delta\lambda^{(m)}$. The value of $\Delta\lambda^{(m-1)}$ is known from the preceding cycle of iteration.

$$\bar{\mathbf{K}}_e^{(m)} = \mathbf{K}_{e0} + \Delta\lambda^{(m)}\mathbf{K}_{e1}^{(m)} \quad \text{with} \quad \mathbf{K}_{e1}^{(m)} = \mathbf{K}_{e1} + \Delta\lambda^{(m-1)}(\mathbf{K}_{e2} + \Delta\lambda^{(m-1)}\mathbf{K}_{e3}) \quad (7.108)$$

$$\bar{\mathbf{R}}_{\text{esz}}^{(m)} = \mathbf{A}_0 + \Delta\lambda^{(m)}\mathbf{A}_1^{(m)} \quad \text{with} \quad \mathbf{A}_1^{(m)} = \mathbf{A}_1 + \Delta\lambda^{(m-1)}\mathbf{A}_2 \quad (7.109)$$

$$\bar{\mathbf{R}}_{\text{esz}}^{(m)} \bar{\mathbf{K}}_e^{(m)} (\bar{\mathbf{R}}_{\text{esz}}^{(m)})^T = (\mathbf{A}_0 + \Delta\lambda^{(m)}\mathbf{A}_1^{(m)}) (\mathbf{K}_{e0} + \Delta\lambda^{(m)}\mathbf{K}_{e1}^{(m)}) (\mathbf{A}_0 + \Delta\lambda^{(m)}\mathbf{A}_1^{(m)})^T \quad (7.110)$$

Terms of the product (7.110) that are nonlinear in $\Delta\lambda^{(m)}$ are approximated in the same manner:

$$\bar{\mathbf{R}}_{esz}^{(m)} \bar{\mathbf{K}}_e^{(m)} (\bar{\mathbf{R}}_{esz}^{(m)})^T = \mathbf{H}_0 + \Delta\lambda^{(m)} \mathbf{H}_1^{(m)} \quad (7.111)$$

$$\mathbf{H}_1^{(m)} = \mathbf{H}_{10}^{(m)} + \Delta\lambda^{(m-1)} \mathbf{H}_{11}^{(m)} + (\Delta\lambda^{(m-1)})^2 \mathbf{H}_{12}^{(m)} \quad (7.112)$$

$$\mathbf{H}_{10}^{(m)} = \mathbf{A}_1^{(m)} \mathbf{K}_{e0} \mathbf{A}_0^T + \mathbf{A}_0 \mathbf{K}_{e0} \mathbf{A}_1^{(m)T} + \mathbf{A}_0 \mathbf{K}_{e1}^{(m)} \mathbf{A}_0^T$$

$$\mathbf{H}_{11}^{(m)} = \mathbf{A}_0 \mathbf{K}_{e1}^{(m)} \mathbf{A}_1^{(m)T} + \mathbf{A}_1^{(m)} \mathbf{K}_{e0} \mathbf{A}_1^{(m)T} + \mathbf{A}_1^{(m)} \mathbf{K}_{e1}^{(m)} \mathbf{A}_0^T$$

$$\mathbf{H}_{12}^{(m)} = \mathbf{A}_1^{(m)} \mathbf{K}_{e1}^{(m)} \mathbf{A}_1^{(m)T}$$

The element matrices are assembled as in iteration 0 and the general eigenvalue problem (7.46) is solved for the load factor increment $\Delta\lambda^{(m)}$.

7.5 BIFURCATION THEORY

7.5.1 Scope of the Theory

The mathematical methods applied to determine singular configurations of structures form part of bifurcation theory. This branch of mathematics is highly developed. It deals with systems of nonlinear equations that contain one or several parameters whose values determine the type (quality) of the solution of the equation. For a specific set of values of the parameters, the type of the solution suddenly changes. Such a set of values is called a bifurcation point of the set of equations.

Bifurcation theory treats the determination of bifurcation points for systems of algebraic equations, as well as ordinary and partial differential equations with one or several dependent variables. It develops theoretical and computational methods for the continuation of the solutions on branches (surfaces) originating at bifurcation points. The classification of bifurcation points contributes significantly to the understanding of the physical phenomena described by the equations.

A well-known example of the bifurcation phenomenon is the buckling of an Euler column. The parameter in the governing equations for this problem is the axial load. While the load is less than critical, there is a unique solution for the displacements of the cross-sections of the column for each value of the load. The character of the solution suddenly changes when the axial load reaches the critical value (buckling load). The cross-sections of the column can now displace in three different directions (on three distinct branches of the load path): increasing axial displacement and displacements in the direction of the positive or the negative normal to the axis. For a load larger than the buckling load, the column can be in any one of three equilibrium configurations. This type of bifurcation is called pitchfork bifurcation. The example in section 2.4 is another case of pitchfork bifurcation.

The following sections are devoted to those aspects of bifurcation theory which are relevant to the analysis of the stability of structures.

7.5.2 Classification of Singular Points

Consider the set of equilibrium equations of a structure:

$$\mathbf{g}(\mathbf{w}, \lambda) = \mathbf{0} \quad (7.113)$$

- \mathbf{g} vector of smooth functions
- \mathbf{w} displacement vector of the structure
- λ load factor

Assume that \mathbf{w} and λ are smooth functions of a parameter z :

$$\mathbf{g}(z) = \mathbf{g}(\mathbf{w}(z), \lambda(z)) = \mathbf{0} \quad (7.114)$$

The derivative of vector \mathbf{g} with respect to the parameter z is determined by differentiation of equation (7.114):

$$\mathbf{t} := \frac{d\mathbf{g}}{dz} = \sum_{i=1}^m \frac{\partial \mathbf{g}}{\partial w_i} \frac{dw_i}{dz} + \frac{\partial \mathbf{g}}{\partial \lambda} \frac{d\lambda}{dz} = \mathbf{0} \quad (7.115)$$

- m dimension of the displacement vector \mathbf{w}

Singular point: The derivatives of the function \mathbf{g} with respect to the displacement coordinates w_i are collected in a matrix \mathbf{K} . The coefficients of \mathbf{K} are functions of the displacement \mathbf{w} . A point (\mathbf{w}, λ) that satisfies equation (7.114) is called regular if the matrix \mathbf{K} is regular for the displacement at that point. The point is called singular if the matrix \mathbf{K} is singular for the displacement \mathbf{w} at the point. Consider the special eigenvalue problem for matrix \mathbf{K} :

$$\mathbf{K}\mathbf{x} = \mu\mathbf{x} \quad \text{with} \quad \mathbf{x}^T\mathbf{x} = 1 \quad (7.116)$$

$$\mathbf{y}^T\mathbf{K} = \mu\mathbf{y}^T \quad \text{with} \quad \mathbf{y}^T\mathbf{y} = 1 \quad (7.117)$$

$$\mathbf{K} = \begin{array}{|c|c|c|} \hline \frac{\partial \mathbf{g}}{\partial w_1} & \dots & \frac{\partial \mathbf{g}}{\partial w_m} \\ \hline \end{array}$$

μ eigenvalue of matrix \mathbf{K}

\mathbf{x} right eigenvector of matrix \mathbf{K}

\mathbf{y} left eigenvector of matrix \mathbf{K}

Let the value of matrix \mathbf{K} at a singular point be denoted by \mathbf{K}_0 . The singular point is called simple if exactly one of the eigenvalues of \mathbf{K}_0 is null. In the following, only simple singular points will be considered. Let the right and left eigenvectors corresponding to the eigenvalue $\mu = 0$ of \mathbf{K}_0 be denoted by \mathbf{x}_0 and \mathbf{y}_0 . Equation (7.117) has the following form for the eigenstate $(0, \mathbf{y}_0)$:

$$\mathbf{y}_0^T \mathbf{K}_0 = \mathbf{0}^T \quad (7.118)$$

$$\mathbf{y}_0^T \frac{\partial \mathbf{g}_0}{\partial w_i} = 0 \quad w_i \in \{1, \dots, m\} \quad (7.119)$$

\mathbf{K}_0 value of \mathbf{K} at the singular point

$\frac{\partial \mathbf{g}_0}{\partial w_i}$ partial derivative of \mathbf{g} with respect to displacement w_i at the singular point

Let the value of vectors \mathbf{w} and \mathbf{t} at a singular point be denoted by \mathbf{w}_0 and \mathbf{t}_0 . The scalar product of the vectors \mathbf{y}_0 and \mathbf{t}_0 is null since \mathbf{t}_0 is null in (7.115). Substitute the expression for vector \mathbf{t}_0 from (7.115) into the scalar product:

$$\mathbf{y}_0^T \mathbf{t}_0 = \sum_{i=1}^m \left(\mathbf{y}_0^T \frac{\partial \mathbf{g}_0}{\partial w_i} \right) \frac{dw_i}{dz} + \left(\mathbf{y}_0^T \frac{\partial \mathbf{g}_0}{\partial \lambda} \right) \frac{d\lambda}{dz} = 0 \quad (7.120)$$

$\frac{\partial \mathbf{g}_0}{\partial \lambda}$ partial derivative of \mathbf{g} with respect to load factor λ at the singular point

Substitution of the scalar product from equation (7.119) into equation (7.120) yields the following expression:

$$\left(\mathbf{y}_0^T \frac{\partial \mathbf{g}_0}{\partial \lambda} \right) \frac{d\lambda}{dz} = 0 \quad (7.121)$$

Classification: Singular points are classified according to the value of the scalar product in equation (7.121). If the product differs from null, the singular point is called a turning point (snap-through point). If the product equals null, the singular point is called a bifurcation point.

$$\mathbf{y}_0^T \frac{\partial \mathbf{g}_0}{\partial \lambda} \neq 0 \rightarrow (\mathbf{w}_0, \lambda_0) \text{ is a turning point} \quad (7.122)$$

$$\mathbf{y}_0^T \frac{\partial \mathbf{g}_0}{\partial \lambda} = 0 \rightarrow (\mathbf{w}_0, \lambda_0) \text{ is a bifurcation point} \quad (7.123)$$

Expressions (7.121) and (7.122) show that the derivative of the load factor with respect to parameter z is null at a turning point:

$$(\mathbf{w}_0, \lambda_0) \text{ is a turning point} \rightarrow \frac{d\lambda_0}{dz} = 0 \quad (7.124)$$

7.5.3 Load Path at a Turning Point

Consider equation (7.115) for the special case of a turning point and substitute property (7.124) of turning points:

$$\frac{d\mathbf{g}_0}{dz} = \sum_{i=1}^m \frac{\partial \mathbf{g}_0}{\partial w_i} \frac{dw_i}{dz} = \mathbf{0} \quad (7.125)$$

A comparison of equations (7.116) and (7.125) shows that the derivative of the displacement vector \mathbf{w} with parameter z is proportional to the eigenvector \mathbf{x}_0 :

$$\frac{d\mathbf{w}}{dz} = \theta \mathbf{x}_0 \quad (7.126)$$

\mathbf{x}_0 eigenvector of \mathbf{K}_0 corresponding to $\mu_0 = 0$
 θ factor $\in \mathbb{R}$

At a turning point, the load path points in the direction of the eigenvector \mathbf{x}_0 of matrix \mathbf{K}_0 which is associated with the eigenvalue null. The load factor on this path is expressed as a Taylor series at the turning point:

$$\lambda(z) = \lambda_0 + \frac{d\lambda_0}{dz} dz + \frac{1}{2} \frac{d^2\lambda_0}{dz^2} dz^2 + \frac{1}{6} \frac{d^3\lambda_0}{dz^3} dz^3 + \dots \quad (7.127)$$

Equation (7.124) shows that the first derivative in expression (7.127) is null. In order to determine the change in the load factor on the load path, the second derivative of the load factor is determined by differentiation of equation (7.115):

$$\begin{aligned} \frac{d^2\mathbf{g}}{dz^2} &= \sum_{i=1}^m \sum_{k=1}^m \frac{\partial^2 \mathbf{g}}{\partial w_i \partial w_k} \frac{dw_i}{dz} \frac{dw_k}{dz} + 2 \sum_{i=1}^m \frac{\partial^2 \mathbf{g}}{\partial w_i \partial \lambda} \frac{dw_i}{dz} \frac{d\lambda}{dz} + \sum_{i=1}^m \frac{\partial \mathbf{g}}{\partial w_i} \frac{d^2 w_i}{dz^2} + \frac{d^2 \mathbf{g}}{d\lambda^2} \left(\frac{d\lambda}{dz} \right)^2 + \frac{d\mathbf{g}}{d\lambda} \frac{d^2 \lambda}{dz^2} \\ &= \mathbf{0} \end{aligned} \quad (7.128)$$

Property (7.124) of the turning points is substituted into equation (7.128):

$$\sum_{i=1}^m \sum_{k=1}^m \frac{\partial^2 \mathbf{g}}{\partial w_i \partial w_k} \frac{dw_i}{dz} \frac{dw_k}{dz} + \sum_{i=1}^m \frac{\partial \mathbf{g}}{\partial w_i} \frac{d^2 w_i}{dz^2} + \frac{d\mathbf{g}}{d\lambda} \frac{d^2 \lambda}{dz^2} = \mathbf{0} \quad (7.129)$$

Equation (7.129) is multiplied from the left with eigenvector \mathbf{y}_0^T . The scalar product $\mathbf{y}_0^T \mathbf{g}_0$ is denoted by a_0 .

$$\sum_{i=1}^m \sum_{k=1}^m \frac{\partial^2 a_0}{\partial w_i \partial w_k} \frac{dw_i}{dz} \frac{dw_k}{dz} + \sum_{i=1}^m \frac{\partial a_0}{\partial w_i} \frac{d^2 w_i}{dz^2} + \frac{da_0}{d\lambda} \frac{d^2 \lambda}{dz^2} = \mathbf{0} \quad (7.130)$$

$$a_0 = \mathbf{y}_0^T \mathbf{g}_0 \quad (7.131)$$

The derivatives of product a with respect to the displacement coordinates w_i are determined with expression (7.119):

$$\frac{\partial a_0}{\partial w_i} = \frac{\partial (\mathbf{y}_0^T \mathbf{g}_0)}{\partial w_i} = \mathbf{y}_0^T \frac{\partial \mathbf{g}_0}{\partial w_i} = 0 \quad (7.132)$$

Substitution of the derivatives (7.132) into equation (7.130) yields:

$$\sum_{i=1}^m \sum_{k=1}^m \frac{\partial^2 a_0}{\partial w_i \partial w_k} \frac{dw_i}{dz} \frac{dw_k}{dz} + \frac{\partial a_0}{\partial \lambda} \frac{d^2 \lambda_0}{dz^2} = 0 \quad (7.133)$$

This equation is solved for the second derivative of the load factor. The derivatives of the displacement coordinates are substituted from expression (7.126) and the double sum is denoted by c_0 :

$$\frac{d^2 \lambda_0}{dz^2} = -\theta^2 c_0 \left(\frac{\partial a_0}{\partial \lambda} \right)^{-1} \quad (7.134)$$

$$c_0 = \sum_{i=1}^m \sum_{k=1}^m \frac{\partial^2 a_0}{\partial w_i \partial w_k} x_{i0} x_{k0}$$

x_{i0} coordinate in row i of eigenvector \mathbf{x}_0

The Taylor series for the load factor follows by substitution of (7.134) into (7.127):

$$\lambda = \lambda_0 - \frac{1}{2} \theta^2 c_0 \left(\frac{\partial a_0}{\partial \lambda} \right)^{-1} dz^2 + \frac{1}{6} \frac{d^3 \lambda_0}{dz^3} dz^3 \quad (7.135)$$

The second derivative of the load factor in equation (7.135) determines the change of the load factor on the load path at the turning point. The following cases occur:

Case T1: Simple quadratic turning point: Maximum of the load factor

Equation (7.124) shows that the first derivative of the load factor is null at a turning point. If the second derivative of the load factor differs from null, the turning point is called a simple quadratic turning point. The Taylor series (7.135) shows that the load factor at the turning point is a maximum if the second derivative of the load factor in expression (7.134) is negative. Equation (7.126) shows that the increment of the displacement vector at the turning point is proportional to the eigenvector \mathbf{x}_0 :

$$\frac{\partial a_0}{\partial \lambda} \neq 0 \quad \wedge \quad \frac{d\lambda_0}{dz} = 0 \quad \wedge \quad \frac{d^2 \lambda_0}{dz^2} < 0 \quad \Rightarrow \quad (\mathbf{w}_0, \lambda_0) \text{ is a maximal point}$$

$$\frac{d\mathbf{w}_0}{dz} = \theta \mathbf{x}_0 \quad (7.136)$$

Case T2: Simple quadratic turning point: Minimum of the load factor

If the second derivative of the load factor at a simple quadratic turning point is positive, the load factor at the turning point is a minimum. The increment of the displacement vector at the turning point is proportional to the eigenvector \mathbf{x}_0 :

$$\frac{\partial \mathbf{a}_0}{\partial \lambda} \neq 0 \quad \wedge \quad \frac{d\lambda_0}{dz} = 0 \quad \wedge \quad \frac{d^2\lambda_0}{dz^2} > 0 \quad \Rightarrow \quad (\mathbf{w}_0, \lambda_0) \text{ is a minimal point}$$

$$\frac{d\mathbf{w}_0}{dz} = \theta \mathbf{x}_0 \tag{7.137}$$

Case T3: Simple cubic turning point: Saddle point of the load factor

If the parameter c_0 in expression (7.134) is null, the second derivative of the load factor at the turning point is null. The Taylor series (7.127) shows that the incremental change of the load factor at the turning point then depends on the third derivative of the load factor. If it is not null, the turning point is called a simple cubic turning point. The increment of the displacement vector at the turning point is proportional to the eigenvector \mathbf{x}_0 :

$$\frac{\partial \mathbf{a}_0}{\partial \lambda} \neq 0 \quad \wedge \quad \frac{d\lambda_0}{dz} = 0 \quad \wedge \quad \frac{d^2\lambda_0}{dz^2} = 0 \quad \wedge \quad \frac{d^3\lambda_0}{dz^3} \neq 0 \quad \Rightarrow \quad (\mathbf{w}_0, \lambda_0) \text{ is a saddle point}$$

$$\frac{d\mathbf{w}_0}{dz} = \theta \mathbf{x}_0 \tag{7.138}$$

If the third derivative of the load factor is null, higher order derivatives of the load factor must be considered. The three types of turning point are illustrated in figure 7.3 which shows the variation of the load factor with the parameter z .

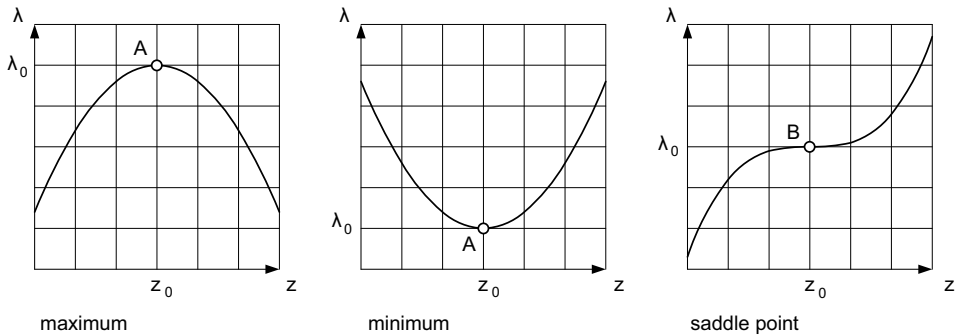


Figure 7.3 Classification of simple turning points
 A simple quadratic turning point
 B simple cubic turning point

7.5.4 Load Paths at a Bifurcation Point

Consider equation (7.115) for the special case of a bifurcation point and treat it as a system of linear equations with the derivatives of the displacements w_i and of the load factor λ as unknowns. In order to solve these equations, the derivative of function \mathbf{g}_0 with respect to the load factor λ is decomposed into components parallel and normal to eigenvector \mathbf{y}_0 :

$$\frac{\partial \mathbf{g}_0}{\partial \lambda} = \mathbf{c} \mathbf{y}_0 + \mathbf{n}_0 \quad \text{with} \quad \mathbf{y}_0^T \mathbf{n}_0 = 0 \quad (7.139)$$

\mathbf{n}_0 vector normal to eigenvector \mathbf{y}_0

Equation (7.139) is multiplied from the left with \mathbf{y}_0^T . The resulting value of coefficient \mathbf{c} is substituted into (7.139) to obtain the normal \mathbf{n}_0 :

$$\mathbf{c} = \mathbf{y}_0^T \frac{\partial \mathbf{g}_0}{\partial \lambda} \quad (7.140)$$

$$\mathbf{n}_0 = (\mathbf{I} - \mathbf{y}_0 \mathbf{y}_0^T) \frac{\partial \mathbf{g}_0}{\partial \lambda} \quad (7.141)$$

The relationship between the normal \mathbf{n}_0 and the derivatives of the displacement coordinates w_{i0} is obtained by multiplying equation (7.115) from the left with the matrix $\mathbf{I} - \mathbf{y}_0 \mathbf{y}_0^T$:

$$\sum_{i=1}^m (\mathbf{I} - \mathbf{y}_0 \mathbf{y}_0^T) \frac{\partial \mathbf{g}_0}{\partial \mathbf{w}_i} \frac{dw_{i0}}{dz} = -(\mathbf{I} - \mathbf{y}_0 \mathbf{y}_0^T) \frac{\partial \mathbf{g}_0}{\partial \lambda} \frac{d\lambda}{dz} = -\mathbf{n}_0 \frac{d\lambda}{dz} \quad (7.142)$$

Expression (7.119) is substituted into equation (7.142):

$$\sum_{i=1}^m \frac{\partial \mathbf{g}_0}{\partial \mathbf{w}_i} \frac{dw_{i0}}{dz} = -\frac{\partial \mathbf{g}_0}{\partial \lambda} \frac{d\lambda}{dz} = -\mathbf{n}_0 \frac{d\lambda}{dz} \quad (7.143)$$

A linear combination of an unknown vector \mathbf{z}_0 and the eigenvector \mathbf{x}_0 is chosen as ansatz for the derivative of the displacement vector \mathbf{w} at the bifurcation point:

$$\frac{d\mathbf{w}_0}{dz} = \theta_1 \mathbf{z}_0 + \theta_2 \mathbf{x}_0 \quad (7.144)$$

$$\frac{dw_{i0}}{dz} = \theta_1 z_{i0} + \theta_2 x_{i0} \quad (7.145)$$

θ_i coefficients $\in \mathbb{R}$

z_{i0} coordinate in row i of vector \mathbf{z}_0

The ansatz (7.144) is substituted into equation (7.143):

$$\sum_{i=1}^m \frac{\partial \mathbf{g}_0}{\partial \mathbf{w}_i} (\theta_1 z_{i0} + \theta_2 x_{i0}) = -\frac{\partial \mathbf{g}_0}{\partial \lambda} \frac{d\lambda}{dz} \quad (7.146)$$

Due to equation (7.119), the sum of the terms containing x_{i0} on the left-hand side of equation (7.146) is identically null. The coefficient θ_2 can therefore be chosen freely. Choose coefficient θ_1 equal to the derivative of the load factor. Equation (7.146) is satisfied if vector \mathbf{z}_0 satisfies equation (7.147):

$$\sum_{i=1}^m \frac{\partial \mathbf{g}_0}{\partial \mathbf{w}_i} z_{i0} = \mathbf{K}_0 \mathbf{z}_0 = -\frac{\partial \mathbf{g}_0}{\partial \lambda} \quad (7.147)$$

Matrix \mathbf{K}_0 is singular so that the vector \mathbf{z}_0 cannot be determined directly with this equation. Vector \mathbf{z}_0 must, however, satisfy the equation. Once \mathbf{z}_0 has been determined, the derivative of the displacement vector \mathbf{w} at the bifurcation point follows from ansatz (7.144):

$$\frac{d\mathbf{w}_0}{dz} = \frac{d\lambda}{dz} \mathbf{z}_0 + \theta_2 \mathbf{x}_0 \quad (7.148)$$

θ_2 free parameter

Classification of bifurcation points: In order to distinguish among various types of bifurcation points, the second derivative (7.128) of the vector \mathbf{g} is considered. Equation (7.128) is multiplied from the left with eigenvector \mathbf{y}_0^T . The scalar product $\mathbf{y}_0^T \mathbf{g}_0$ is denoted by a_0 , as defined in equation (7.131). Its derivative with respect to the load factor vanishes due to definition (7.123) of the bifurcation point. Expressions (7.132) and (7.149) are substituted into the equation to yield (7.150):

$$\frac{\partial a_0}{\partial \lambda} = \frac{\partial (\mathbf{y}_0^T \mathbf{g}_0)}{\partial \lambda} = \mathbf{y}_0^T \frac{\partial \mathbf{g}_0}{\partial \lambda} = 0 \quad (7.149)$$

$$\sum_{i=1}^m \sum_{k=1}^m \frac{\partial^2 a_0}{\partial \mathbf{w}_i \partial \mathbf{w}_k} \frac{d\mathbf{w}_i}{dz} \frac{d\mathbf{w}_k}{dz} + 2 \sum_{i=1}^m \frac{\partial^2 a_0}{\partial \mathbf{w}_i \partial \lambda} \frac{d\mathbf{w}_i}{dz} \frac{d\lambda}{dz} + \frac{\partial^2 a_0}{\partial \lambda^2} \left(\frac{d\lambda}{dz} \right)^2 = 0 \quad (7.150)$$

The derivatives of the displacement coordinate w_{i0} at the bifurcation point are substituted from equation (7.148) into equation (7.150):

$$\begin{aligned} & \sum_{i=1}^m \sum_{k=1}^m \frac{\partial^2 a_0}{\partial \mathbf{w}_i \partial \mathbf{w}_k} z_{i0} z_{k0} \left(\frac{d\lambda}{dz} \right)^2 + 2\theta_2 \sum_{i=1}^m \frac{\partial^2 a_0}{\partial \mathbf{w}_i \partial \mathbf{w}_k} z_{i0} \mathbf{x}_{k0} \frac{d\lambda}{dz} + \theta_2^2 \sum_{i=1}^m \frac{\partial^2 a_0}{\partial \mathbf{w}_i \partial \mathbf{w}_k} \mathbf{x}_{i0} \mathbf{x}_{k0} + \\ & 2 \sum_{i=1}^m \frac{\partial^2 a_0}{\partial \mathbf{w}_i \partial \lambda} z_{i0} \left(\frac{d\lambda}{dz} \right)^2 + 2\theta_2 \sum_{i=1}^m \frac{\partial^2 a_0}{\partial \mathbf{w}_i \partial \lambda} \mathbf{x}_{i0} \frac{d\lambda}{dz} + \frac{\partial^2 a_0}{\partial \lambda^2} \left(\frac{d\lambda}{dz} \right)^2 = 0 \end{aligned} \quad (7.151)$$

This expression is a quadratic equation for the derivative of the load factor and is called the bifurcation equation:

$$c_2 \left(\frac{\partial \lambda}{dz} \right)^2 + 2\theta_2 c_1 \frac{\partial \lambda}{dz} + \theta_2^2 c_0 = 0 \quad (7.152)$$

$$c_0 = \sum_{i=1}^m \sum_{k=1}^m \frac{\partial^2 a_0}{\partial \mathbf{w}_i \partial \mathbf{w}_k} \mathbf{x}_{i0} \mathbf{x}_{k0} \quad (7.153)$$

$$c_1 = \sum_{i=1}^m \sum_{k=1}^m \frac{\partial^2 a_0}{\partial \mathbf{w}_i \partial \mathbf{w}_k} z_{i0} \mathbf{x}_{k0} + \sum_{i=1}^m \frac{\partial^2 a_0}{\partial \mathbf{w}_i \partial \lambda} \mathbf{x}_{i0} \quad (7.154)$$

$$c_2 = \sum_{i=1}^m \sum_{k=1}^m \frac{\partial^2 a_0}{\partial \mathbf{w}_i \partial \mathbf{w}_k} z_{i0} z_{k0} + 2 \sum_{i=1}^m \frac{\partial^2 a_0}{\partial \mathbf{w}_i \partial \lambda} z_{i0} + \frac{\partial^2 a_0}{\partial \lambda^2} \quad (7.155)$$

The roots of the quadratic equation (7.152) depend on the sign of its discriminant, which is proportional to the parameter d :

$$d := c_1^2 - c_0 c_2 \quad (7.156)$$

Branches of a bifurcation path: The two roots of the bifurcation equation (7.152) correspond to two branches of the load curve at the bifurcation point. The bifurcation point is approached in the direction of the primary branch, as shown in figure 7.4. The load path can be continued in the direction of the primary branch, or in the direction of one of the two secondary paths associated with the other root.

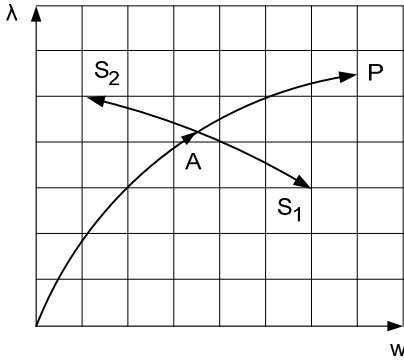


Figure 7.4 Primary and secondary branches of a load path at a bifurcation point

- A primary branch entering the bifurcation point
- P primary branch leaving the bifurcation point
- S1 secondary branch leaving the bifurcation point
- S2 secondary branch leaving the bifurcation point

Cases: In addition to the factor d , the coefficients c_0, c_1 and c_2 influence the type of the bifurcation point. The cases in the following table are considered:

Table 7.1 Classification of Bifurcation Points

case	c_0	c_1	c_2	d	type
B1	$\neq 0$	$\neq 0$	$\neq 0$	> 0	asymmetric
B2	$\neq 0$	$\neq 0$	0	> 0	asymmetric
B3	0	$\neq 0$	$\neq 0$	> 0	symmetric
B4	0	$\neq 0$	0	> 0	symmetric
B5	$\neq 0$	$\neq 0$	$\neq 0$	0	cuspl

There are additional cases that are not contained in table 7.1. If, for example, $d < 0$ so that the discriminant is negative, the roots of equation (7.152) are complex and the solution for the path consists of a single point that satisfies the system equation $\mathbf{g} = \mathbf{0}$. This solution is called an isola formation point.

Case B1: Asymmetric bifurcation point: $c_0 \neq 0 \wedge c_1 \neq 0 \wedge c_2 \neq 0 \wedge d > 0$

If all parameters c_i differ from 0 and the discriminant of the bifurcation equation is positive, the equation has two real roots for the derivative of the load factor:

$$\frac{d\lambda_1}{dz} = \frac{\theta_2}{c_2}(-c_1 + \sqrt{d}) \quad (7.157)$$

$$\frac{d\lambda_2}{dz} = \frac{\theta_2}{c_2}(-c_1 - \sqrt{d}) \quad (7.158)$$

Equation (7.148) shows that roots (7.157) and (7.158) of (7.152) correspond to the following tangents to the branches of the load path at the bifurcation point:

$$\frac{dw_{i0}}{dz} = \theta_2 \left(\frac{z_{i0}}{c_2}(-c_1 + \sqrt{d}) + x_{i0} \right) \quad i \in \{1, \dots, m\} \quad (7.159)$$

$$\frac{dw_{i0}}{dz} = \theta_2 \left(\frac{z_{i0}}{c_2}(-c_1 - \sqrt{d}) + x_{i0} \right) \quad (7.160)$$

The singular point $(\mathbf{w}_0, \lambda_0)$ is called an asymmetric (simple transcritical) bifurcation point.

Case B2: Asymmetric bifurcation point with $c_2 = 0$

Since parameter c_2 is null, the bifurcation equation (7.152) is linear and contains the parameter θ_2 as a factor. For solution $\theta_2 = 0$, a real load factor increment is chosen freely and the tangent (7.162) to the path is determined with equation (7.148). If the second factor on the left-hand side of equation (7.161) is set to null, the secondary branch (7.163) of the bifurcation path is obtained.

$$\theta_2 \left(2c_1 \frac{d\lambda}{dz} + \theta_2 c_0 \right) = 0 \quad (7.161)$$

$$\text{primary branch:} \quad \frac{d\lambda_1}{dz} \in \mathbb{R} \quad \frac{d\mathbf{w}_0}{dz} = \frac{d\lambda_1}{dz} \mathbf{z}_0 \quad (7.162)$$

$$\text{secondary branches:} \quad \frac{d\lambda_1}{dz} = -\frac{c_0}{2c_1} \theta_2 \quad \frac{d\mathbf{w}_0}{dz} = \theta_2 \left(-\frac{c_0}{2c_1} \mathbf{z}_0 + \mathbf{x}_2 \right) \quad (7.163)$$

Case B3: Symmetric bifurcation point: $c_0 = 0 \wedge c_1 \neq 0 \wedge c_2 \neq 0 \wedge d > 0$

If the parameter c_0 is null while the other parameters differ from null, and the discriminant of the bifurcation equation (7.152) is positive, the equation has two real roots for the derivative of the load factor, one of which equals null:

$$\frac{d\lambda}{dz} \left(c_2 \frac{d\lambda}{dz} + 2\theta_2 c_1 \right) = 0 \quad (7.164)$$

$$\frac{d\lambda_1}{dz} = -2\theta_2 \frac{c_1}{c_2} \quad (7.165)$$

$$\frac{d\lambda_2}{dz} = 0 \quad (7.166)$$

Equation (7.144) shows that the two roots correspond to the following tangents to the branches of the load path at the bifurcation point:

$$\text{primary branch: } \frac{dw_i}{dz} = \theta_2 \left(-2z_{i0} \frac{c_1}{c_2} + x_{i0} \right) \quad i \in \{1, \dots, m\} \quad (7.167)$$

$$\text{secondary branches: } \frac{dw_i}{dz} = \theta_2 x_{i0} \quad (7.168)$$

The singular point $(\mathbf{w}_0, \lambda_0)$ is called a symmetric (simple pitchfork) bifurcation point. The change in the load factor in the secondary directions (7.168), for which the load factor derivative is null, depends on the second derivative of the load factor. If this derivative is positive, the symmetric bifurcation point is called stable since the load factor increases as the load path is traversed near the bifurcation point. If the derivative is negative, the symmetric bifurcation point is called unstable.

The second derivative of the load factor with respect to the parameter z will now be determined for the secondary branch (7.168) of the load path, for which equations (7.166), (7.132), (7.149) and the assumption $c_0 = 0$ hold:

$$\frac{d\lambda_0}{dz} = \frac{da_0}{dw_i} = \frac{da_0}{d\lambda} = 0 \quad (7.169)$$

$$c_0 = \sum_{i=1}^m \sum_{k=1}^m \frac{\partial^2 a_0}{\partial w_i \partial w_k} x_{i0} x_{k0} = 0$$

Equation (7.128) contains the second derivative of the load factor. If this equation is multiplied from the left with eigenvector \mathbf{y}_0^T , the term containing the second derivative is multiplied with the derivative of the scalar product a_0 with respect to the load factor, which is null. The second derivative thus cannot be derived directly from equation (7.128).

If equation (7.128) is differentiated with respect to parameter z and then multiplied from the left with eigenvector \mathbf{y}_0^T , the following expression is obtained for the third derivative of the scalar product a_0 after equations (7.169) have been substituted:

$$\begin{aligned} \frac{d^3 a_0}{dz^3} &= \sum_{i=1}^m \sum_{k=1}^m \sum_{n=1}^m \frac{\partial^3 a_0}{\partial w_i \partial w_k \partial w_n} \frac{dw_i}{dz} \frac{dw_k}{dz} \frac{dw_n}{dz} + 3 \sum_{i=1}^m \sum_{k=1}^m \frac{\partial^2 a_0}{\partial w_i \partial w_k} \frac{dw_i}{dz} \frac{d^2 w_k}{dz^2} + \\ &3 \sum_{i=1}^m \frac{\partial^2 a_0}{\partial w_i} \frac{dw_i}{dz} \frac{d^2 \lambda}{dz^2} = 0 \end{aligned} \quad (7.170)$$

The second derivatives of the displacement coordinates w_k in equation (7.170) are derived from equation (7.115) using properties (7.169):

$$\sum_{i=1}^m \frac{\partial \mathbf{g}_0}{\partial w_i} \frac{d^2 w_i}{dz^2} = - \frac{d\mathbf{g}_0}{d\lambda} \frac{d^2 \lambda}{dz^2} - \sum_{i=1}^m \sum_{k=1}^m \frac{\partial^2 \mathbf{g}_0}{\partial w_i \partial w_k} \frac{dw_i}{dz} \frac{dw_k}{dz} \quad (7.171)$$

Equation (7.171) is multiplied from the left with matrix $\mathbf{I} - \mathbf{y}_0 \mathbf{y}_0^T$ derived in equation (7.141). The derivatives of the displacement coordinates w_i are replaced by the coordinates x_{i0} of the eigenvector using expressions (7.144) and (7.169):

$$\sum_{i=1}^m \frac{\partial \mathbf{g}_0}{\partial w_i} \frac{d^2 w_i}{dz^2} = - \frac{d\mathbf{g}_0}{d\lambda} \frac{d^2 \lambda}{dz^2} - \theta_2^2 \sum_{i=1}^m \sum_{k=1}^m \frac{\partial^2 \mathbf{g}_0}{\partial w_i \partial w_k} x_{i0} x_{k0} + \mathbf{y}_0 \sum_{i=1}^m \sum_{k=1}^m \frac{\partial^2 a_0}{\partial w_i \partial w_k} \frac{dw_i}{dz} \frac{dw_k}{dz} \quad (7.172)$$

Since the coefficient c_0 is null in expressions (7.169), the last term of equation (7.172) is null. Matrix \mathbf{K}_0 is substituted into equation (7.172) from equation (7.118) and vector \mathbf{z}_0 is substituted from equation (7.147):

$$\mathbf{K}_0 \frac{d^2 \mathbf{w}_0}{dz^2} = -\frac{d\mathbf{g}_0}{d\lambda} \frac{d^2 \lambda_0}{dz^2} - \theta_2^2 \sum_{i=1}^m \sum_{k=1}^m \frac{\partial^2 \mathbf{g}_0}{\partial w_i \partial w_k} x_{i0} x_{k0}$$

$$\frac{d^2 \mathbf{w}_0}{dz^2} = \mathbf{z}_0 \frac{d^2 \lambda_0}{dz^2} + \theta_2^2 \mathbf{v}_0 \quad (7.173)$$

$$\mathbf{v}_0 = -\mathbf{K}_0^{-1} \sum_{i=1}^m \sum_{k=1}^m \frac{\partial^2 \mathbf{g}_0}{\partial w_i \partial w_k} x_{i0} x_{k0} \quad (7.174)$$

The second derivative of the displacement is substituted from equation (7.173) into expression (7.170) for the third derivative of the scalar product a_0 . The equation is solved for the second derivative of the load factor λ :

$$\frac{\partial^2 \lambda_0}{\partial z^2} = -\frac{\theta_2^2 \sum_{i=1}^m \sum_{k=1}^m \sum_{n=1}^m \frac{\partial^3 a_0}{\partial w_i \partial w_k \partial w_n} x_{i0} x_{k0} x_{n0} + 3 \sum_{i=1}^m \sum_{k=1}^m \frac{\partial^2 a_0}{\partial w_i \partial w_k} x_{i0} v_{k0}}{\sum_{i=1}^m \sum_{k=1}^m \frac{\partial^2 a_0}{\partial w_i \partial w_k} x_{i0} z_{k0} + \sum_{i=1}^m \frac{\partial^2 a_0}{\partial w_i \partial \lambda} x_{i0}} \quad (7.175)$$

The sign of the second derivative of the load factor at the bifurcation point is used to distinguish between stable and unstable symmetric bifurcation points:

$$\frac{\partial^2 \lambda_0}{\partial z^2} > 0 \Rightarrow \text{stable symmetric bifurcation} \quad (7.176)$$

$$\frac{\partial^2 \lambda_0}{\partial z^2} < 0 \Rightarrow \text{unstable symmetric bifurcation} \quad (7.177)$$

If the bifurcation is stable, the load factor increases as the secondary load path is traversed in the vicinity of the symmetric bifurcation point.

Case B4: Symmetric bifurcation point with $c_2 = 0$

Since parameter c_2 is null, the bifurcation equation (7.152) is linear and contains the parameter θ_2 , as well as the load factor derivative as factors on the left-hand side. For solution $\theta_2 = 0$, a real load factor increment is chosen freely and the tangent to the primary load path is determined with equation (7.148). If the load factor derivative is set to null, equation (7.148) shows that the tangent to the secondary load path is parallel to the eigenvector \mathbf{x}_0 of the tangent stiffness matrix at the bifurcation point:

$$2\theta_2 c_1 \frac{d\lambda}{dz} = 0 \quad (7.178)$$

$$\text{primary branch : } \frac{d\lambda_1}{dz} \in \mathbb{R} \quad \frac{d\mathbf{w}_0}{dz} = \frac{d\lambda_1}{dz} \mathbf{z}_0 \quad (7.179)$$

$$\text{secondary branches: } \frac{d\lambda_1}{dz} = 0 \quad \frac{d\mathbf{w}_0}{dz} = \theta_2 \mathbf{x}_0 \quad (7.180)$$

Case B5: Cusp bifurcation point: $d=0$

Since the discriminant of the bifurcation equation (7.152) is null, the equation has two equal real roots for the derivative of the load factor. The tangent to the load path at the bifurcation point is given by equation (7.148):

$$c_2 \frac{d\lambda_0}{dz} = -\theta_2 c_1 \quad (7.181)$$

$$\frac{d\mathbf{w}_0}{dz} = \frac{d\lambda_0}{dz} \mathbf{z}_0 + \theta_2 \mathbf{x}_0 \quad (7.182)$$

$$c_1^2 - c_0 c_2 = 0 \quad (7.183)$$

The singular point $(\mathbf{w}_0, \lambda_0)$ is called a cusp bifurcation point. Equation (7.183) shows that if one of the parameters c_i equals null, then at least one other parameter c_k also equals null. Consider the case $c_0 = c_1 = 0$ and $c_2 \neq 0$. It follows from (7.181) and (7.182) that the derivative of the load factor is null, and that the tangents to the two branches of the load path at the bifurcation point are parallel to the eigenvector \mathbf{x}_0 :

$$\begin{aligned} \frac{d\lambda_0}{dz} &= 0 \\ \frac{d\mathbf{w}_0}{dz} &= \theta_2 \mathbf{x}_0 \end{aligned} \quad (7.184)$$

The change of the load factor as the load path is traversed in the vicinity of the cusp point depends on the higher order derivatives of the load factor.

Figure 7.5 shows examples of the variation of the load factor on branches of load paths near bifurcation points. The secondary branches are shown for the two symmetric bifurcation points.

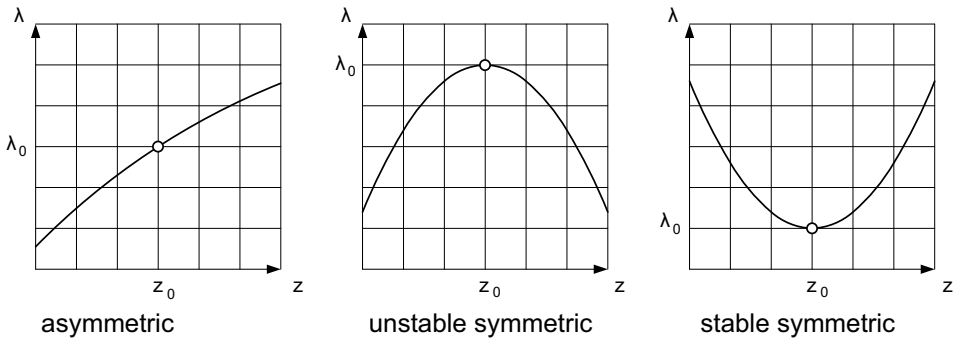


Figure 7.5 Classification of bifurcation points

7.5.5 Bifurcation Theory for Finite Element Formulations

Assume that equation (7.113) has the following special properties:

- System vector \mathbf{w} contains the free displacement coordinates of the nodes of a finite element model.
- Vector \mathbf{g} contains the equilibrium equations of the finite element model.
- The load terms are linear functions λp_i of the coefficients p_i of a pattern load and of the load factor λ .
- The load coefficients p_i are independent of the displacement \mathbf{w} .

These properties of the finite element formulation lead to the following simplifications of the general bifurcation theory presented in the preceding sections:

(a) The matrix \mathbf{K} in equation (7.116) is symmetric. Its left and right eigenvectors are therefore identical. In the following derivations, the distinction between left and right eigenvectors is dropped and the eigenvectors are denoted with \mathbf{x} .

(b) The derivative of the vector \mathbf{g} with respect to the load factor λ is the negative pattern load vector \mathbf{p} . Since the pattern load is assumed to be independent of the load factor and of the displacements, the second derivatives of the vector \mathbf{g} , which contain a differentiation with respect to the load factor λ , are null.

$$\frac{\partial \mathbf{g}}{\partial \lambda} = -\mathbf{p} \quad (7.185)$$

$$\frac{\partial^2 \mathbf{g}}{\partial \mathbf{w}_i \partial \lambda} = \frac{\partial^2 \mathbf{g}}{\partial \lambda^2} = \mathbf{0} \quad (7.186)$$

Due to simplifications (7.185) and (7.186), the following general expressions of the bifurcation theory are replaced by special expressions for the finite element formulation:

$$(7.116): \quad \mathbf{K}^t \mathbf{x} = \mu \mathbf{x} \quad (7.187)$$

\mathbf{K}^t tangent stiffness matrix

$$(7.118): \quad \mathbf{K}^t \mathbf{x}_0 = \mathbf{0} \quad (7.188)$$

\mathbf{x}_0 eigenvector corresponding to $\mu_0 = 0$

$$(7.122): \quad \mathbf{x}_0^T \mathbf{p} \neq 0 \rightarrow (\mathbf{w}_0, \lambda_0) \text{ is a turning point} \quad (7.189)$$

$$(7.123): \quad \mathbf{x}_0^T \mathbf{p} = 0 \rightarrow (\mathbf{w}_0, \lambda_0) \text{ is a bifurcation point} \quad (7.190)$$

$$(7.134): \quad \frac{d^2 \lambda}{dz^2} = \frac{\theta^2 c_0}{\mathbf{x}_0^T \mathbf{p}} \quad (7.191)$$

$$(7.153): \quad c_0 = \sum_{i=1}^m \sum_{k=1}^m \frac{\partial^2 \mathbf{a}_0}{\partial \mathbf{w}_i \partial \mathbf{w}_k} x_{i0} x_{k0} \quad (7.192)$$

$$(7.154): \quad c_1 = \sum_{i=1}^m \sum_{k=1}^m \frac{\partial^2 \mathbf{a}_0}{\partial \mathbf{w}_i \partial \mathbf{w}_k} z_{i0} x_{k0} \quad (7.193)$$

$$(7.155): \quad c_2 = \sum_{i=1}^m \sum_{k=1}^m \frac{\partial^2 \mathbf{a}_0}{\partial \mathbf{w}_i \partial \mathbf{w}_k} z_{i0} z_{k0} \quad (7.194)$$

7.5.6 Classification of the Singular Points of a 2-Bar Truss

In order to classify the singular points of the 2-bar truss in section 2.1, equilibrium equations (2.18) and (2.19) for the truss are arranged in a vector \mathbf{g} that is a special case of the vector \mathbf{g} in equation (7.113):

$$\mathbf{g}(\mathbf{w}, \lambda) = \begin{array}{|c|} \hline w_1(2m^2 + w_1^2 + w_2(2 + w_2)) - \lambda p_1 \\ \hline (1 + w_2)(w_1^2 + w_2(2 + w_2)) - \lambda p_2 \\ \hline \end{array} \quad (7.195)$$

The first partial derivatives of the vector \mathbf{g} are:

$$\frac{\partial \mathbf{g}}{\partial w_1} = \begin{array}{|c|} \hline 2m^2 + 3w_1^2 + w_2(2 + w_2) \\ \hline 2w_1(1 + w_2) \\ \hline \end{array} \quad (7.196)$$

$$\frac{\partial \mathbf{g}}{\partial w_2} = \begin{array}{|c|} \hline 2w_1(1 + w_2) \\ \hline w_1^2 + 3w_2^2 + 6w_2 + 2 \\ \hline \end{array} \quad (7.197)$$

$$\frac{\partial \mathbf{g}}{\partial \lambda} = \begin{array}{|c|} \hline -p_1 \\ \hline -p_2 \\ \hline \end{array} \quad (7.198)$$

The second partial derivatives of the vector \mathbf{g} are:

$$\frac{\partial^2 \mathbf{g}}{\partial w_1^2} = \begin{array}{|c|} \hline 6w_1 \\ \hline 2(1 + w_2) \\ \hline \end{array} \quad (7.199)$$

$$\frac{\partial^2 \mathbf{g}}{\partial w_1 \partial w_2} = \begin{array}{|c|} \hline 2(1 + w_2) \\ \hline 2w_1 \\ \hline \end{array} \quad (7.200)$$

$$\frac{\partial^2 \mathbf{g}}{\partial w_2^2} = \begin{array}{|c|} \hline 2w_1 \\ \hline 6(1 + w_2) \\ \hline \end{array} \quad (7.201)$$

Turning point: Consider the singular point defined by equation (2.49). The displacements and the load at this point are:

$$w_{10} = 0 \quad (7.202)$$

$$w_{20} = -1 + \frac{1}{\sqrt{3}}$$

$$p_{10} = 0$$

$$p_{20} = -\frac{2}{3\sqrt{3}}$$

The load (p_{10}, p_{20}) is arbitrarily chosen as the pattern load, so that the load factor at the singular point is $\lambda_0 = 1.0$. The tangent matrix \mathbf{K}^t in equation (7.187) contains the vectors (7.196) and (7.197), evaluated at the singular point (7.202):

$$\mathbf{K}_0 = \begin{array}{|cc|} \hline 2m^2 - 0.6667 & 0 \\ \hline 0 & 0 \\ \hline \end{array} \quad (7.203)$$

The eigenstates of matrix \mathbf{K}_0 are determined as follows:

$$\mu_0 = 0: \quad \mathbf{x}_0 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (7.204)$$

$$\mu_1 = 2m^2 - 0.6667: \quad \mathbf{x}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad (7.205)$$

The scalar product $\mathbf{x}_0^T \mathbf{p}$ is different from null and satisfies condition (7.189). The singular point is therefore a turning point (snap-through point). The type of this turning point depends on the second derivative of the load factor λ with the parameter z . The second derivatives (7.199) to (7.201) of \mathbf{g} are evaluated at the turning point:

$$\frac{\partial^2 \mathbf{g}_0}{\partial w_1^2} = \begin{bmatrix} 0 \\ 2(1+w_{20}) \end{bmatrix} \quad \frac{\partial^2 \mathbf{g}_0}{\partial w_1 \partial w_2} = \begin{bmatrix} 2(1+w_{20}) \\ 0 \end{bmatrix} \quad \frac{\partial^2 \mathbf{g}_0}{\partial w_2^2} = \begin{bmatrix} 0 \\ 6(1+w_{20}) \end{bmatrix} \quad (7.206)$$

The second derivative of the load factor λ is determined by substituting (7.204) and (7.206) into expression (7.191):

$$\begin{aligned} \frac{d^2 \lambda}{dz^2} &= \frac{\theta^2}{p_{20}} \left(\frac{\partial^2 (\mathbf{x}_0^T \mathbf{g}_0)}{\partial w_1^2} x_{10}^2 + 2 \frac{\partial^2 (\mathbf{x}_0^T \mathbf{g}_0)}{\partial w_1 \partial w_2} x_{10} x_{20} + \frac{\partial^2 (\mathbf{x}_0^T \mathbf{g}_0)}{\partial w_2^2} x_{20}^2 \right) \\ \frac{d^2 \lambda}{dz^2} &= \frac{6\theta^2(1+w_{20})}{p_{20}} < 0 \end{aligned} \quad (7.207)$$

Since $p_{20} < 0$ and $|w_{20}| < 1$, the second derivative of λ satisfies condition (7.136). The singular point is thus a simple quadratic turning point where the value of the load factor is a local maximum. This result is in agreement with figure 2.4, which shows the variation of the load with the displacement for the 2-bar truss.

Bifurcation point: Consider the singular point defined by the displacements and loads in equation (2.48):

$$\begin{aligned} w_{10} &= 0 \\ w_{20} &= -1 + \sqrt{1-2m^2} \\ p_{10} &= 0 \\ p_{20} &= -2m^2 \sqrt{1-2m^2} \end{aligned} \quad (7.208)$$

The load (p_{10}, p_{20}) is arbitrarily chosen as pattern load, so that the load factor at the singular point is $\lambda_0 = 1.0$. The tangent matrix \mathbf{K}^t in equation (7.187) contains the vectors (7.196) and (7.197), evaluated at the singular point (7.208):

$$\mathbf{K}_0 = \begin{bmatrix} 0 & 0 \\ 0 & 2-6m^2 \end{bmatrix} \quad (7.209)$$

The eigenstates of matrix \mathbf{K}_0 are determined as follows:

$$\mu_0 = 0: \quad \mathbf{x}_0 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad (7.210)$$

$$\mu_1 = 2-6m^2: \quad \mathbf{x}_1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (7.211)$$

The scalar product $\mathbf{x}_0^T \mathbf{p}$ equals null and satisfies condition (7.190). The singular point is therefore a bifurcation point. In order to determine the type of this bifurcation point, the second derivatives (7.199) to (7.201) are evaluated at the bifurcation point:

$$\frac{\partial^2 \mathbf{g}_0}{\partial \mathbf{w}_1^2} = \begin{bmatrix} 0 \\ 2(1+w_{20}) \end{bmatrix} \quad \frac{\partial^2 \mathbf{g}_0}{\partial \mathbf{w}_1 \partial \mathbf{w}_2} = \begin{bmatrix} 2(1+w_{20}) \\ 0 \end{bmatrix} \quad \frac{\partial^2 \mathbf{g}_0}{\partial \mathbf{w}_2^2} = \begin{bmatrix} 0 \\ 6(1+w_{20}) \end{bmatrix} \quad (7.212)$$

Consider equation (7.147) for the determination of the vector \mathbf{z}_0 :

$$\begin{bmatrix} 0 & 0 \\ 0 & 2-6m^2 \end{bmatrix} \begin{bmatrix} z_{10} \\ z_{20} \end{bmatrix} = \begin{bmatrix} 0 \\ p_{20} \end{bmatrix} \quad (7.213)$$

The value of z_{20} is determined by the second equation in (7.213), whereas z_{10} can be chosen freely. Since ansatz (7.144) contains the eigenvector \mathbf{x}_0 , the vector \mathbf{z}_0 is chosen orthogonally to the eigenvector \mathbf{x}_0 . The solution for equation (7.213) is then:

$$z_{10} = 0 \quad (7.214)$$

$$z_{20} = \frac{p_{20}}{2-6m^2}$$

The coefficients c_0 to c_2 can now be evaluated:

$$\begin{aligned} c_0 &= \frac{\partial^2(\mathbf{x}_0^T \mathbf{g}_0)}{\partial \mathbf{w}_1^2} x_{10} x_{10} + \frac{\partial^2(\mathbf{x}_0^T \mathbf{g}_0)}{\partial \mathbf{w}_1 \partial \mathbf{w}_2} (x_{10} x_{20} + x_{20} x_{10}) + \frac{\partial^2(\mathbf{x}_0^T \mathbf{g}_0)}{\partial \mathbf{w}_2^2} x_{20} x_{20} \\ c_1 &= \frac{\partial^2(\mathbf{x}_0^T \mathbf{g}_0)}{\partial \mathbf{w}_1^2} x_{10} z_{10} + \frac{\partial^2(\mathbf{x}_0^T \mathbf{g}_0)}{\partial \mathbf{w}_1 \partial \mathbf{w}_2} (x_{10} z_{20} + x_{20} z_{10}) + \frac{\partial^2(\mathbf{x}_0^T \mathbf{g}_0)}{\partial \mathbf{w}_2^2} x_{20} z_{20} \\ c_2 &= \frac{\partial^2(\mathbf{x}_0^T \mathbf{g}_0)}{\partial \mathbf{w}_1^2} z_{10} z_{10} + \frac{\partial^2(\mathbf{x}_0^T \mathbf{g}_0)}{\partial \mathbf{w}_1 \partial \mathbf{w}_2} (z_{10} z_{20} + z_{20} z_{10}) + \frac{\partial^2(\mathbf{x}_0^T \mathbf{g}_0)}{\partial \mathbf{w}_2^2} z_{20} z_{20} \end{aligned}$$

The coordinates x_{10}, x_{20}, z_{10} and z_{20} are substituted:

$$\begin{aligned} c_0 &= c_2 = 0 \quad (7.215) \\ c_1 &= -\frac{2m^2(1-2m^2)}{1-3m^2} \end{aligned}$$

Comparison of parameters (7.215) with expressions (7.178) to (7.180) for case B4 at bifurcation points shows that the bifurcation point is symmetric with $c_2 = 0$:

$$2\theta_2 c_1 \frac{d\lambda}{dz} = 0 \quad \wedge \quad c_1 \neq 0 \quad \rightarrow \quad \frac{d\lambda}{dz} = 0 \quad \vee \quad \theta_2 = 0 \quad (7.216)$$

Equation (7.144) shows that the load path following the bifurcation point is either tangential to eigenvector \mathbf{x}_0 if the derivative of the load factor is null, or tangential to vector \mathbf{z}_0 if the parameter θ_2 is null.

7.6 CONTINUATION OF LOAD PATHS AT TURNING POINTS

7.6.1 Continuation Concepts

Continuation: Assume that a singular point has been determined with the method described in section 7.4. The determination of the load path beyond the singular point is called a continuation of the load path at the singular point. The load factor can increase or decrease on the continuation of the load path. The structure is called unstable if the load factor decreases on the continuation. The analysis of the continuation is therefore a significant part of the stability analysis of the structure.

The constant arc method described in chapter 6 cannot be used directly to continue the load path beyond a singular point, since the tangent stiffness matrix cannot be inverted at the singular point. The first cycle of iteration in the load step following the singular point can therefore not be performed. In addition, a closer investigation will reveal additional numerical problems at the start of a continuation which are not immediately obvious.

Type of singularity: Assume that a singular point has been reached on the load path of a truss or frame. Let \mathbf{x}_0 be the eigenvector associated with the eigenvalue null of the tangent stiffness matrix at the singular point. In order to determine the nature of the singular point, the value of the scalar product $\mathbf{x}_0^T \mathbf{p}_0$ of the eigenvector \mathbf{x}_0 with the pattern load \mathbf{p}_0 is computed. If it is null, the singular point is a bifurcation point. Otherwise the singular point is a turning point.

The bifurcation theory in section 7.5 shows that for simple singular points there is a single continuation of the load path beyond a turning point. The load path beyond a bifurcation point can follow one of several branches. In the numerical analysis, this branch must be chosen explicitly. In nature, imperfections of the geometry and variations in the material properties of the structure, as well as perturbations of the load, will favour one of the branches which is then followed.

The numerical properties of continuations at turning points and bifurcation points, as well as the level of difficulty of the required concepts and algorithms, differ significantly. Since the continuation of load paths beyond turning points is less difficult, it is treated first in this section. The more difficult continuation of load paths beyond bifurcation points is treated in section 7.7.

7.6.2 Continuation at Turning Points of a 2-Bar Truss

Turning point: Consider a 2-bar truss with aspect ratio $m > 0.7071$ subjected to a load (p_1, p_2) with $p_2 = 0$. As shown in figure 7.1 and discussed in section 7.2.2, this truss has a turning point C, but does not have any bifurcation points. Let the aspect ratio be $m = 0.90$. The displacement state $(w_{1c}, w_{2c}) = (0.0, -0.422650)$ at the turning point is determined with (7.31). The tangent stiffness matrix \mathbf{K}_1^t at the turning point follows from (7.14) and (7.31):

$$\mathbf{K}_1^t \Delta \mathbf{w} = \Delta \mathbf{p} \quad (7.217)$$

0.953333	0	ΔW_1	=	0
0	0	ΔW_2		Δp_2

Consider an infinitesimal displacement step (dw_1, dw_2) beyond the turning point C. Equation (7.217) shows that the infinitesimal displacement dw_1 must be null at C, and that the infinitesimal displacement dw_2 is accompanied by an infinitesimal load $dp_2 = 0$. The direction of the displacement beyond the singular point is the same as the direction of the displacement before the singular point. The tangent to the load curve at the turning point is parallel to that eigenvector of \mathbf{K}_1^1 which corresponds to the eigenvalue 0.

First step beyond the turning point: The first step of the algorithm for the continuation at the turning point does not lead to an infinitesimal displacement increment (dw_1, dw_2) , but rather to a finite displacement increment $(\Delta w_1, \Delta w_2)$. Assume that the load step is taken in the direction of the eigenvector associated with eigenvalue 0 at the turning point so that the displacement at the end of the load step is $(0.0, w_2 + \Delta w_2)$. The displacement increment is determined with the secant matrix for the step, whose coefficients are given by expressions (7.24) with $w_1 = \Delta w_1 = 0$:

$$\begin{array}{|c|c|c|} \hline 2m^2 + w_{2c}(2 + w_{2c}) + (1 + w_{2c})\Delta w_{2c} + \Delta w_{2c}^2 & 0 & 0 \\ \hline 0 & 3(1 + w_{2c})\Delta w_{2c} + \Delta w_{2c}^2 & \Delta w_2 \\ \hline \end{array} = \begin{array}{|c|} \hline 0 \\ \hline \Delta p_2 \\ \hline \end{array} \quad (7.218)$$

Equation (7.218) shows that the coefficient k_{22} of the secant stiffness matrix at the turning point is null for $\Delta w_{2c} = 0$. The secant matrix is therefore singular in cycle 0 of the iteration for the load step following the turning point. It cannot be inverted as required for application of the constant arc increment method.

The displacement increment in cycle 0 of the iteration is therefore set equal to a fraction of the eigenvector \mathbf{x}_0 . The scale factor is chosen so that the chord length in the step equals that of the constant arc increment method. The displacement increment leads to a reaction increment, even if the load does not change.

Consider a truss with aspect ratio $m = 0.9$. Let the displacement increment at the turning point in the first cycle of iteration be $\Delta w_{2c} = -0.05$. The secant matrix for the second cycle of the iteration is computed with expression (7.218):

$$\mathbf{K}^{(s,1)} = \begin{array}{|c|c|} \hline 0.926965 & 0.0 \\ \hline 0.0 & -0.084103 \\ \hline \end{array} \quad (7.219)$$

The secant matrix (7.219) for the second cycle of the iteration in the load step is no longer singular. The iteration can therefore be continued as described in section 6.4 for the constant arc increment method. The iteration is terminated when the error norm in the trial state at the end of the load step is less than a prescribed limit value.

7.6.3 Nearly Singular Systems of Equations

Concept: The tangent stiffness matrix at a turning point is by definition singular. The infinitesimal displacement increment is parallel to the eigenvector \mathbf{x}_0 of the tangent matrix associated with the eigenvalue null. The incremental governing equations for the load path beyond the turning point are nearly singular in the vicinity of the turning point. A numerically stable solution of these equations must converge to a fraction of the eigenvector \mathbf{x}_0 at the turning point because the location on the load path converges towards the turning point. This requirement is not satisfied if the incremental equations are solved with the decomposition algorithm in section 6.6.

Solutions for nearly singular governing equations can be determined by two methods: deflation and stabilisation. The concepts of the two methods are presented in this section. It is shown that the deflation method is preferable for the continuation algorithm, since it converges to the solution at the turning point, whereas the stabilisation method does not converge to that solution.

Deflation: Consider a nearly singular system of linear incremental equations with a symmetric coefficient matrix \mathbf{K} , displacement increment $\Delta \mathbf{d}$ and load increment $\Delta \mathbf{p}$:

$$\mathbf{K} \Delta \mathbf{d} = \Delta \mathbf{p} \quad (7.220)$$

Let the set of orthonormal eigenstates of matrix \mathbf{K} be (ω_i, \mathbf{x}_i) with $i = 0, \dots, n-1$, where n is the dimension of the matrix:

$$\mathbf{K} = \sum_{i=0}^{n-1} \omega_i \mathbf{x}_i \mathbf{x}_i^T \quad (7.221)$$

$$\mathbf{x}_i^T \mathbf{x}_m = \delta_{im}$$

ω_i eigenvalue of matrix \mathbf{K}

\mathbf{x}_i eigenvector of matrix \mathbf{K}

All eigenvalues of the nearly singular matrix \mathbf{K} differ from null. The displacement and load increments are expressed as linear combinations of the eigenvectors:

$$\Delta \mathbf{d} = \sum_{k=0}^{n-1} c_k \mathbf{x}_k \quad (7.222)$$

$$\Delta \mathbf{p} = \sum_{m=0}^{n-1} \theta_m \mathbf{x}_m \quad (7.223)$$

Expressions (7.221) to (7.223) are substituted into equation (7.220):

$$\sum_{i=0}^{n-1} \omega_i \mathbf{x}_i \mathbf{x}_i^T \sum_{k=0}^{n-1} c_k \mathbf{x}_k = \sum_{m=0}^{n-1} \theta_m \mathbf{x}_m \quad (7.224)$$

Due to the orthonormality of the eigenvectors, equation (7.224) simplifies to:

$$\sum_{i=0}^{n-1} \omega_i c_i \mathbf{x}_i = \sum_{m=0}^{n-1} \theta_m \mathbf{x}_m \quad (7.225)$$

A comparison of the coefficients of a given eigenvector on the two sides of equation (7.225) yields an expression that relates the coefficients for an eigenstate:

$$\omega_i c_i = \theta_i \quad i \in \{0, \dots, n-1\} \quad (7.226)$$

Let the eigenstate (ω_0, \mathbf{x}_0) with the smallest eigenvalue $|\omega_0|$ be known. The coefficients c_0 and ω_0 for this eigenstate can be computed without knowledge of the other eigenstates of matrix \mathbf{K} . Coefficient θ_0 is obtained from equation (7.223) by multiplying with the eigenvector \mathbf{x}_0^T from the left:

$$\theta_0 = \mathbf{x}_0^T \Delta \mathbf{p} \quad (7.227)$$

The coefficient c_0 follows from expression (7.226):

$$c_0 = \frac{\theta_0}{\omega_0} \quad (7.228)$$

If the eigenstate (ω_0, \mathbf{x}_0) dominates the solution of equation (7.220), it is advantageous to remove the known contribution of this eigenstate from the solution by removing the term $c_0 \mathbf{x}_0$ in equation (7.232) by means of the following identity:

$$c_0 \mathbf{K} \mathbf{x}_0 = c_0 \omega_0 \mathbf{x}_0 = \theta_0 \mathbf{x}_0 \quad (7.229)$$

Equation (7.229) is subtracted from equation (7.224):

$$\sum_{i=1}^{n-1} \omega_i \mathbf{x}_i \mathbf{x}_i^T \sum_{k=1}^{n-1} c_k \mathbf{x}_k = \sum_{m=1}^{n-1} \theta_m \mathbf{x}_m \quad (7.230)$$

The remaining sums in equation (7.230) are denoted by $\Delta \mathbf{z}_0$ and $\Delta \mathbf{p}_0$:

$$\Delta \mathbf{z}_0 = \sum_{k=1}^{n-1} c_k \mathbf{x}_k = \Delta \mathbf{d} - c_0 \mathbf{x}_0 \quad (7.231)$$

$$\Delta \mathbf{p}_0 = \sum_{m=1}^{n-1} \theta_m \mathbf{x}_m = \Delta \mathbf{p} - \theta_0 \mathbf{x}_0 \quad (7.232)$$

Expressions (7.231) and (7.232) are substituted into equation (7.230):

$$\mathbf{K} \Delta \mathbf{z}_0 = \Delta \mathbf{p}_0 \quad (7.233)$$

Since matrix \mathbf{K} is assumed to be regular, the equation can be solved for the deflated displacement increment $\Delta \mathbf{z}_0$. Substitution of the increment $\Delta \mathbf{z}_0$ into equation (7.231) yields the solution $\Delta \mathbf{d}$ for equation (7.220).

The theory shows that nearly singular equations, whose solution is dominated by the eigenstate (ω_0, \mathbf{x}_0) with the smallest eigenvalue, can be solved as follows:

- (1) Compute the coefficient θ_0 with expression (2.227). Use equation (7.232) to sweep the load vector of the component which is proportional to the eigenvector.
- (2) Solve equation (7.233) for the deflated displacement increment $\Delta \mathbf{z}_0$.
- (3) Compute the coefficient c_0 with expression (7.228). Substitute the coefficient and the deflated displacement increment into equation (7.231) to obtain the total displacement increment $\Delta \mathbf{d}$.

Equation (7.228) shows that the solution of equation (7.220) converges to the eigenvector \mathbf{x}_0 as the tangent matrix \mathbf{K} becomes singular at the bifurcation point, since c_0 becomes large and dominates in expression (7.222).

Example of deflation: Consider the following incremental governing equations that are nearly singular and ill conditioned:

$$\begin{bmatrix} 0.147431000 & 14.469023000 \\ 14.469023000 & 1327.394235000 \end{bmatrix} \begin{bmatrix} \Delta u_0 \\ \Delta u_1 \end{bmatrix} = \begin{bmatrix} -0.078864000 \\ -3.617205000 \end{bmatrix}$$

The eigenstates of the coefficient matrix are:

$$\omega_0 = -0.010284771 \quad \omega_1 = 1327.551950770$$

$$\mathbf{x}_0 = \begin{bmatrix} -0.999940598 \\ 0.010899589 \end{bmatrix} \quad \mathbf{x}_1 = \begin{bmatrix} 0.010899589 \\ 0.999940598 \end{bmatrix}$$

The singular matrix in the vicinity of the given matrix equals $\omega_1 \mathbf{x}_1 \mathbf{x}_1^T$:

$$\mathbf{K}_0 = \begin{bmatrix} 0.157714549 & 14.468910907 \\ 14.468910907 & 1327.394236222 \end{bmatrix}$$

The parameter θ_0 in equation (2.227) and the right-hand side of equation (7.232) are:

$$\theta_0^{(0)} = \mathbf{x}_0^T \Delta \mathbf{p} = -0.0394332680$$

$$\Delta \mathbf{p} - \theta_0^{(0)} \mathbf{x}_0 = \begin{bmatrix} -0.039433074 \\ -3.617634806 \end{bmatrix}$$

Solution of equation (7.233) yields an improved deflated displacement increment:

$$\Delta \mathbf{d}_0^{(1)} = \begin{bmatrix} -0.000029704 \\ -0.002725042 \end{bmatrix}$$

The coefficient c_0 is given by equation (7.228):

$$c_0 = \frac{-0.0394332680}{-0.010284771} = 3.834141567$$

The displacement increment follows from equation (7.231):

$$\Delta \mathbf{d} = \begin{bmatrix} -0.000029704 \\ -0.002725042 \end{bmatrix} + 3.834141567 \begin{bmatrix} -0.999940598 \\ 0.010899589 \end{bmatrix} = \begin{bmatrix} 3.833884301 \\ -0.044515611 \end{bmatrix}$$

The error in the governing equations is of order $O[10^{-9}]$ relative to the load vector.

Extended deflation: Consider a system of linear equations that contains free displacements \mathbf{d}_1 , prescribed displacements \mathbf{d}_2 , loads \mathbf{q}_1 and \mathbf{q}_2 and reactions \mathbf{r}_2 . Let the equations be structured as shown in equation (7.234).

$$\mathbf{Kd} = \mathbf{q} + \mathbf{r} \tag{7.234}$$

$$\begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} \\ \mathbf{K}_{21} & \mathbf{K}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{d}_1 \\ \mathbf{d}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{q}_1 \\ \mathbf{q}_2 \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{r}_2 \end{bmatrix}$$

Let the submatrix \mathbf{K}_{11} be nearly singular and let $(\omega_0, \mathbf{x}_{10})$ be the eigenstate of \mathbf{K}_{11} with the smallest absolute value of the eigenvalue ω_0 .

$$\mathbf{K}_{11}\mathbf{x}_{10} = \omega_0\mathbf{x}_{10} \quad \wedge \quad \omega_0 \neq 0 \quad \wedge \quad \mathbf{x}_{10}^T\mathbf{x}_{10} = 1 \quad (7.235)$$

$(\omega_0, \mathbf{x}_{10})$ eigenstate of matrix \mathbf{K}_{11}

The displacement increment \mathbf{x}_{10} causes a reaction increment \mathbf{r}_{20} which equals the product of the tangent stiffness submatrix \mathbf{K}_{21} with eigenvector \mathbf{x}_{10} . The eigenvector \mathbf{x}_{10} is arranged in the upper rows of a vector \mathbf{x}_0 whose dimension equals the dimension of matrix \mathbf{K} . The reaction vector \mathbf{r}_{20} is arranged in the lower rows of a vector \mathbf{r}_0 whose dimension also equals the dimension of matrix \mathbf{K} .

$$\begin{aligned} \mathbf{K}\mathbf{x}_0 &= \omega_0\mathbf{x}_0 + \mathbf{r}_0 \\ \mathbf{r}_{20} &= \mathbf{K}_{21}\mathbf{x}_{10} \end{aligned} \quad (7.236)$$

$$\begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} & \mathbf{x}_{10} \\ \mathbf{K}_{21} & \mathbf{K}_{22} & \mathbf{0} \end{bmatrix} = \omega_0 \begin{bmatrix} \mathbf{x}_{10} \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{r}_{20} \end{bmatrix}$$

\mathbf{r}_{20} reaction due to displacement \mathbf{x}_{10}

\mathbf{x}_0 system eigen-displacement

\mathbf{r}_0 system eigen-reaction

Let the load subvector \mathbf{q}_1 and the displacement subvector \mathbf{d}_1 in equation (7.234) be deflated as shown in equations (7.227) to (7.232):

$$\begin{aligned} \bar{\mathbf{q}}_1 &= \mathbf{q}_1 - \theta_0\mathbf{x}_{10} & \bar{\mathbf{d}}_1 &= \mathbf{d}_1 - c_0\mathbf{x}_0 \\ \theta_0 &= \mathbf{x}_{10}^T\mathbf{q}_1 & \theta_0 &= c_0\omega_0 \end{aligned} \quad (7.237)$$

The system load vector \mathbf{q} and the system displacement vector \mathbf{d} are deflated by replacing the subvectors \mathbf{q}_1 and \mathbf{d}_1 with their deflated counterparts:

$$\bar{\mathbf{q}} = \mathbf{q} - \theta_0\mathbf{x}_0 \quad \bar{\mathbf{d}} = \mathbf{d} - c_0\mathbf{x}_0 \quad (7.238)$$

$$\bar{\mathbf{q}} = \begin{bmatrix} \bar{\mathbf{q}}_1 \\ \mathbf{q}_2 \end{bmatrix} \quad \bar{\mathbf{d}} = \begin{bmatrix} \bar{\mathbf{d}}_1 \\ \mathbf{d}_2 \end{bmatrix}$$

$\bar{\mathbf{q}}$ deflated load vector

$\bar{\mathbf{d}}$ deflated displacement vector

Equation (7.234) is deflated by multiplying equation (7.236) with c_0 and subtracting it from equation (7.234):

$$\mathbf{K}(\mathbf{d} - c_0\mathbf{x}_0) = (\mathbf{q} - \theta_0\mathbf{x}_0) + (\mathbf{r} - c_0\mathbf{r}_0) \quad (7.239)$$

$$\mathbf{K}\bar{\mathbf{d}} = \bar{\mathbf{q}} + \bar{\mathbf{r}} \quad (7.240)$$

$$\bar{\mathbf{r}} = \mathbf{r} - c_0\mathbf{r}_0 \quad (7.241)$$

$\bar{\mathbf{r}}$ deflated reaction vector

The total displacements and reactions follow from equations (7.238) and (7.241):

$$\mathbf{d} = \bar{\mathbf{d}} + c_0\mathbf{x}_0 \quad (7.242)$$

$$\mathbf{r} = \bar{\mathbf{r}} + c_0\mathbf{r}_0 \quad (7.243)$$

This completes the solution of the nearly singular equations (7.234).

Stabilisation: Consider a nearly singular system of linear incremental equations with a symmetric coefficient matrix \mathbf{K} , displacement increment $\Delta \mathbf{d}$ and load increment $\Delta \mathbf{p}$:

$$\mathbf{K} \Delta \mathbf{d} = \Delta \mathbf{p} \quad (7.244)$$

Let the diagonal coefficient in row i go to null when the matrix \mathbf{K} is decomposed in the form of expression (6.80) with diagonal matrix \mathbf{D} . Let a matrix \mathbf{K}_a be called the stabilised matrix associated with matrix \mathbf{K} in equation (7.244) if their diagonal coefficients differ by a specified constant c :

$$\mathbf{K}_a = \mathbf{K} + c \mathbf{e}_i \mathbf{e}_i^T \quad (7.245)$$

\mathbf{K}_a stabilised coefficient matrix

\mathbf{e}_i unit vector with coefficient 1 in column i

The term $c \mathbf{e}_i \mathbf{e}_i^T \Delta \mathbf{d}$ is added on both sides of equation (7.244):

$$\mathbf{K}_a \Delta \mathbf{d} = \Delta \mathbf{p} + c \mathbf{e}_i \mathbf{e}_i^T \Delta \mathbf{d} \quad (7.246)$$

The parameter c is chosen so that matrix \mathbf{K}_a is regular. Equation (7.246) is multiplied from the left with the inverse of matrix \mathbf{K}_a :

$$\Delta \mathbf{d} = \mathbf{K}_a^{-1} \Delta \mathbf{p} + c (\mathbf{e}_i^T \Delta \mathbf{d}) \mathbf{K}_a^{-1} \mathbf{e}_i \quad (7.247)$$

The vectors on the right-hand side of equation (7.247) are denoted by $\Delta \mathbf{v}_1$ and $\Delta \mathbf{v}_2$:

$$\begin{aligned} \mathbf{K}_a \Delta \mathbf{v}_1 &= \Delta \mathbf{p} & \Delta \mathbf{v}_1 &= \mathbf{K}_a^{-1} \Delta \mathbf{p} \\ \mathbf{K}_a \Delta \mathbf{v}_2 &= \mathbf{e}_i & \Delta \mathbf{v}_2 &= \mathbf{K}_a^{-1} \mathbf{e}_i \\ \Delta \mathbf{d} &= \Delta \mathbf{v}_1 + c (\mathbf{e}_i^T \Delta \mathbf{d}) \Delta \mathbf{v}_2 \end{aligned} \quad (7.248)$$

Equation (7.248) is multiplied from the left with the transposed unit vector \mathbf{e}_i^T and the resulting equation is solved for the scalar product $\mathbf{e}_i^T \Delta \mathbf{d}$:

$$\mathbf{e}_i^T \Delta \mathbf{d} = \frac{\mathbf{e}_i^T \Delta \mathbf{v}_1}{1 - c (\mathbf{e}_i^T \Delta \mathbf{v}_2)} \quad (7.249)$$

Substitution of expression (7.249) into equation (2.247) leads to the increment of the displacement in equation (7.244):

$$\Delta \mathbf{d} = \Delta \mathbf{v}_1 + \frac{c \mathbf{e}_i^T \Delta \mathbf{v}_1}{1 - c (\mathbf{e}_i^T \Delta \mathbf{v}_2)} \Delta \mathbf{v}_2 \quad (7.250)$$

Equation (2.250) shows that as parameter c goes to null, the stabilised matrix \mathbf{K}_a goes to the almost singular matrix \mathbf{K} and the stabilised solution $\Delta \mathbf{d}$ goes to:

$$\Delta \mathbf{v}_1 = \mathbf{K}^{-1} \Delta \mathbf{p} \quad (7.251)$$

The derivation of the stabilisation method shows that the solution does not converge to a multiple of the eigenvector associated with the lowest eigenvalue of the tangent matrix as the singular point is approached. The stabilisation method is therefore not used in the computation of load paths following a turning point.

7.6.4 First Cycle of Iteration in the First Step at a Turning Point

The tangent stiffness matrix at the start of the first cycle of iteration in the first load step of the continuation beyond a turning point is singular. A special procedure is therefore required for this one cycle of iteration.

Let s be the index of the first load step of a continuation beyond a turning point. Assume that the error vector in the turning configuration is negligible, so that governing equation (6.45) for the step reduces to equation (7.252). The equations and variables in equation (7.252) are ordered as in equation (6.45) so that the free displacement coordinates precede the prescribed displacement coordinates:

$$\mathbf{K}_0 \Delta \mathbf{d}^{(s)} = \Delta \lambda^{(s)} \mathbf{q}_t + \Delta \mathbf{r}^{(s)} \quad (7.252)$$

$$\begin{array}{|c|c|} \hline \mathbf{K}_{11} & \mathbf{K}_{12} \\ \hline \mathbf{K}_{21} & \mathbf{K}_{22} \\ \hline \end{array} \begin{array}{|c|} \hline \Delta \mathbf{d}_1 \\ \hline \Delta \lambda \mathbf{d}_{2t} \\ \hline \end{array} = \Delta \lambda \begin{array}{|c|} \hline \mathbf{q}_{1t} \\ \hline \mathbf{q}_{2t} \\ \hline \end{array} + \begin{array}{|c|} \hline \mathbf{0} \\ \hline \Delta \mathbf{r}_2 \\ \hline \end{array}$$

\mathbf{K}_0 tangent stiffness matrix at the turning point

The stiffness submatrix \mathbf{K}_{11} is singular and therefore has an eigenstate (ω_0, \mathbf{x}_0) with eigenvalue $\omega_0 = 0$:

$$\mathbf{K}_{11} \mathbf{x}_{10} = \omega_0 \mathbf{x}_{10} \quad \wedge \quad \omega_0 = 0 \quad \wedge \quad \mathbf{x}_{10}^T \mathbf{x}_{10} = 1 \quad (7.253)$$

ω_0 smallest eigenvalue of matrix \mathbf{K}_{11}

\mathbf{x}_{10} eigenvector of matrix \mathbf{K}_{11} corresponding to eigenvalue ω_0

The load path theory for turning points in section 7.5.3 shows that the infinitesimal displacement at the turning point is parallel to the eigenvector \mathbf{x}_{10} . The approximate displacement increment $\Delta \mathbf{d}_1^{(s,0)}$ for cycle 0 of the iteration in step s is therefore set equal to a fraction μ of \mathbf{x}_{10} :

$$\Delta \mathbf{d}_1^{(s,0)} = \mu \mathbf{x}_{10} \quad (7.254)$$

$\Delta \mathbf{d}_1^{(s,0)}$ free displacement increment in cycle 0

μ scale factor

In order to avoid continuation of the load path in the direction from which the turning point was reached, the direction of the displacement increment $\Delta \mathbf{d}_1^{(s,0)}$ is chosen so that it makes the smallest possible angle θ with the system displacement increment $\Delta \mathbf{d}_1^{(s-1)}$ preceding the turning point. The sign of the factor μ is chosen according to the following rule:

$$\left| \Delta \mathbf{d}_1^{(s,0)} \right| \left| \Delta \mathbf{d}_1^{(s-1)} \right| \cos \theta = \mu \mathbf{x}_0^T \Delta \mathbf{d}_1^{(s-1)} \\ \text{signum}(\mu) = \text{signum}(\mathbf{x}_0^T \Delta \mathbf{d}_1^{(s-1)}) \quad (7.255)$$

The load factor increment $\Delta \lambda$ in cycle 0 of the first step following the turning point is null. The displacement increment \mathbf{x}_{10} causes a reaction increment \mathbf{r}_{20} which is given by equation (7.236). The system eigenvector \mathbf{x}_0 and the system eigenreaction \mathbf{r}_0 are defined in equation (7.236). Since the eigenvalue has the special value null at the turning point, equation (7.236) is replaced by equation (7.256):

$$\mathbf{K}_0 \mathbf{x}_0 = \mathbf{r}_0 \quad (7.256)$$

$$\begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} \\ \mathbf{K}_{21} & \mathbf{K}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{x}_{10} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{r}_{20} \end{bmatrix}$$

- \mathbf{x}_0 system eigendisplacement
- \mathbf{r}_0 system eigenreaction
- \mathbf{r}_{20} reaction $\mathbf{K}_{21}\mathbf{x}_{10}$ due to displacement \mathbf{x}_{10}

The scale factor μ is chosen so that the chord length for the first step on the load path equals the chord length of the constant arc increment method. This is a special case of the load factor determination treated in section 6.5:

- The system correction vector and the load increment are null.
- The pattern displacement vector is replaced by the eigendisplacement \mathbf{x}_0 .
- The pattern reaction vector is replaced by the eigenreaction \mathbf{r}_0 .

The first trial state following the turning point is given by:

$$\mathbf{d}^{(s+1,0)} = \mathbf{d}^{(s)} + \mu \mathbf{x}_0 \quad (7.257)$$

$$\mathbf{r}^{(s+1,0)} = \mathbf{r}^{(s)} + \mu \mathbf{r}_0 \quad (7.258)$$

$$\mathbf{q}^{(s+1,0)} = \mathbf{q}^{(s)} \quad (7.259)$$

7.6.5 General Cycle of Iteration in the First Step at a Turning Point

The displacement increment computed in cycle $m-1$ of the iteration is used to approximate the secant matrix $\mathbf{K}^{(s,m)}$ for cycle m of the iteration as in the constant arc increment method. For the second cycle of iteration, this is the displacement increment in equation (7.254). The secant matrix is then used to continue the iteration as in the constant arc increment method. Since the linear system of equations is nearly singular, it is solved with the deflation method of section 7.6.3. Consider the governing equations (6.45) for a load step:

$$\mathbf{K}_\Delta \mathbf{d} = \Delta\lambda \mathbf{q}_t + \Delta \mathbf{r} + \mathbf{e} \quad (7.260)$$

$$\begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} \\ \mathbf{K}_{21} & \mathbf{K}_{22} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{d}_1 \\ \Delta\lambda \mathbf{d}_{2t} \end{bmatrix} = \Delta\lambda \begin{bmatrix} \mathbf{q}_{1t} \\ \mathbf{q}_{2t} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \Delta \mathbf{r}_2 \end{bmatrix} + \begin{bmatrix} \mathbf{e}_1 \\ \mathbf{e}_2 \end{bmatrix}$$

- \mathbf{e}_1 error vector for the free nodes at the start of the step
- \mathbf{e}_2 error vector for the supports at the start of the step

Submatrix \mathbf{K}_{11} of the secant matrix \mathbf{K} in (7.260) is assumed to be regular relative to the accuracy of the computer, so that all of its eigenvalues differ from null. Let ω_0 be the eigenvalue with the smallest absolute value. The extended theory of deflation in section 7.6.3 is extended for equation (7.260) by replacing the load \mathbf{q} in equation (7.234) with the load increment $\Delta\lambda \mathbf{q}_t + \mathbf{e}$ in equation (7.260). Since the load is now a function of the load factor increment $\Delta\lambda$, the deflation coefficient θ_0 in equation (7.237) is also a function of the load factor increment:

$$\begin{aligned}\theta_0 &= \mathbf{x}_{10}^T(\Delta\lambda\mathbf{q}_{1t} + \mathbf{e}_1) \\ \theta_0 &= \Delta\lambda\theta_{10} + \theta_{20}\end{aligned}\quad (7.261)$$

$$\theta_{10} = \mathbf{x}_{10}^T\mathbf{q}_{1t} \quad \theta_{20} = \mathbf{x}_{10}^T\mathbf{e}_1$$

The deflated load increments are derived from equation (7.238):

$$\bar{\mathbf{q}} = \Delta\lambda(\mathbf{q}_t - \theta_{10}\mathbf{x}_0) + (\mathbf{e} - \theta_{20}\mathbf{x}_0) \quad (7.262)$$

The deflated governing equations follow from equations (7.242), (7.260) and (7.262):

$$\mathbf{K}\Delta\bar{\mathbf{d}} = \Delta\bar{\mathbf{q}} + \Delta\bar{\mathbf{r}} \quad (7.263)$$

$$\begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} \\ \mathbf{K}_{21} & \mathbf{K}_{22} \end{bmatrix} \begin{bmatrix} \Delta\bar{\mathbf{d}}_1 \\ \Delta\lambda\mathbf{d}_{2t} \end{bmatrix} = \Delta\lambda \begin{bmatrix} \mathbf{q}_{1t} - \theta_{10}\mathbf{x}_{10} \\ \mathbf{q}_{2t} \end{bmatrix} + \begin{bmatrix} \mathbf{e}_1 - \theta_{20}\mathbf{x}_{10} \\ \mathbf{e}_2 \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \Delta\bar{\mathbf{r}}_2 \end{bmatrix}$$

Equation (7.263) is solved for the deflated displacements and reactions. The solution is split into three parts: the solution for the deflated pattern load, the solution for the deflated error load and the determination of the load factor increment. The deflated pattern state is defined as follows:

$$\mathbf{K}\bar{\mathbf{v}}_t = \bar{\mathbf{q}}_t + \bar{\mathbf{w}}_t \quad (7.264)$$

$$\begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} \\ \mathbf{K}_{21} & \mathbf{K}_{22} \end{bmatrix} \begin{bmatrix} \bar{\mathbf{v}}_{1t} \\ \mathbf{d}_{2t} \end{bmatrix} = \begin{bmatrix} \mathbf{q}_{1t} - \theta_{10}\mathbf{x}_{10} \\ \mathbf{q}_{2t} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \bar{\mathbf{w}}_{2t} \end{bmatrix}$$

Since submatrix \mathbf{K}_{11} is assumed to be regular, the first equation can be solved for the deflated pattern displacement $\bar{\mathbf{v}}_{1t}$ and the second for the deflated pattern reaction $\bar{\mathbf{w}}_{2t}$:

$$\bar{\mathbf{v}}_{1t} = \mathbf{K}_{11}^{-1}(\mathbf{q}_{1t} - \mathbf{K}_{12}\mathbf{d}_{2t} - \theta_{10}\mathbf{x}_{10}) \quad (7.265)$$

$$\bar{\mathbf{w}}_{2t} = \mathbf{K}_{21}\bar{\mathbf{v}}_{1t} + \mathbf{K}_{22}\mathbf{d}_{2t} - \mathbf{q}_{2t} \quad (7.266)$$

The deflated error state is defined as follows:

$$\mathbf{K}\bar{\mathbf{f}} = \bar{\mathbf{e}} + \bar{\mathbf{t}} \quad (7.267)$$

$$\begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} \\ \mathbf{K}_{21} & \mathbf{K}_{22} \end{bmatrix} \begin{bmatrix} \bar{\mathbf{f}}_1 \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{e}_1 - \theta_{20}\mathbf{x}_{20} \\ \mathbf{e}_2 \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \bar{\mathbf{t}}_2 \end{bmatrix}$$

The first of equations (7.267) is solved for the deflated correction displacement $\bar{\mathbf{f}}_1$, the second for the deflated reaction correction $\bar{\mathbf{t}}_2$:

$$\bar{\mathbf{f}}_1 = \mathbf{K}_{11}^{-1}(\mathbf{e}_1 - \theta_{20}\mathbf{x}_{20}) \quad (7.268)$$

$$\bar{\mathbf{t}}_2 = \mathbf{K}_{21}\bar{\mathbf{f}}_1 - \mathbf{e}_2 \quad (7.269)$$

The solution to equation (7.263) is obtained by multiplying the solution to equation (7.264) by the load factor increment $\Delta\lambda$ and adding the solution to equation (7.267):

$$\Delta\bar{\mathbf{d}}_{1t} = \Delta\lambda\bar{\mathbf{v}}_{1t} + \bar{\mathbf{f}}_1 \quad (7.270)$$

$$\Delta\bar{\mathbf{r}}_{2t} = \Delta\lambda\bar{\mathbf{w}}_{2t} + \bar{\mathbf{t}}_2$$

The load factor increment $\Delta\lambda$ is computed as for the constant arc increment method so that the chord length of the load step in the load path equals that of the first load step of the analysis. The total displacements and reactions for the solution of equation (7.260) are computed with equations (7.242) and (7.243):

$$\Delta\mathbf{d}_1 = \Delta\lambda \bar{\mathbf{d}}_{1t} + \mathbf{c}_0 \mathbf{x}_0 \quad (7.271)$$

$$\Delta\mathbf{r}_2 = \Delta\lambda \bar{\mathbf{r}}_{2t} + \mathbf{c}_0 \mathbf{r}_0 \quad (7.272)$$

$$\mathbf{c}_0 = \frac{1}{\omega_0} (\Delta\lambda \theta_{10} + \theta_{20})$$

The increments of the state variables are used to compute the trial state at the end of the load step. If the norm of the trial error is less than a prescribed limit, the step is terminated. Otherwise a new secant matrix is computed and the iteration in the load step is repeated.

7.6.6 General Load Step of the Continuation at a Turning Point

General load step: The special procedure for cycle 0 of the iteration is used only for the first load step beyond the turning point. In all other steps of the continuation, the tangential stiffness matrix is used to determine the displacement and reaction increments in cycle 0, as described for the constant arc increment method. The deflation method is used to solve the system of equations.

In cycle 1 and all following cycles of iteration, the approximate secant stiffness matrix is used as described in section 7.6.5 to determine the displacement and reaction increments. The deflation method is used to solve the system of equations.

Termination of the continuation: As the load path is continued beyond the turning point, the coefficients of the tangent and secant stiffness matrices change so that the numerical problems due to the nearly singular and ill-conditioned equations usually disappear. The continuation is terminated when the displacement norm increment in a load step is sufficiently small compared to the force norm increment:

$$|m(a_2 - a_1)| \leq |\eta(b_2 - b_1)| \quad (7.273)$$

a_1, a_2 displacement norms at the start and end of the step

b_1, b_2 force norms at the start and end of the step

m initial slope of the load curve

η scale factor

After the continuation has been terminated, the next part of the load path is determined again with the constant arc increment method, using the method of chapter 6 to solve the system of linear equations.

7.7 CONTINUATION OF LOAD PATHS AT BIFURCATION POINTS

7.7.1 Approximation of the Governing Equations

Consistent linearisation: The constant arc increment method is based on the assumption that an iterative solution of the governing equations for nonlinear frame behaviour can be computed by consistent linearisation of these equations. This assumption is valid if the nonlinear terms of the Taylor series for the dependent behavioural variables remain small during a load step. It will be shown for the example of the 2-bar truss that this condition is not always satisfied in the neighbourhood of a bifurcation point. As a result, the numerical algorithms based on linearisation become highly sensitive to the chosen step size, and a suitable step size can in many cases only be determined by numerical experiment.

Total stiffness: The continuation of load paths in the neighbourhood of bifurcation points can be based on the total stiffness matrix of the structure. This is the essential feature of the continuation algorithm that is developed in the following sections. By basing the computation on the total stiffness matrix, which does not become singular at the bifurcation point, the serious numerical difficulties associated with incremental solutions using tangent and secant stiffness matrices are avoided. This leads to a robust continuation algorithm.

Numerical properties: In order to gain insight into the numerical properties of the mathematical model of frames in the neighbourhood of bifurcation points, the loading condition $p_1 = 0$ for the solution is dropped. This extension permits investigation of the properties of equilibrium configurations in the neighbourhood of the load path for the exact solution. If the properties change rapidly in the vicinity of the load path, the properties of the trial solutions of the iterative methods can differ significantly from the solutions on the path, which can lead to divergence of the iteration process. Such numerical behaviour has been observed for the algorithms based on the linearisation of the governing equations.

7.7.2 Properties at Bifurcation Points of a 2-Bar Truss

Tangent matrix: Consider a 2-bar truss with aspect ratio $m = 0.40$ subjected to a load (p_1, p_2) with $p_1 = 0$. As shown in figure 7.1 and discussed in section 7.2.2, this truss is in a bifurcation state at point B on the load path. The load path can be continued beyond point B either on curve BC for solution 1 or on curve BE for solution 2. The displacement state $(w_{1c}, w_{2c}) = (0.0, -0.175379)$ at the bifurcation point is determined with (7.32). The tangent stiffness matrices for solutions 1 and 2 are equal at the bifurcation point. They follow from (7.15) and (7.32):

$$\mathbf{K}_1^t \Delta \mathbf{w} = \mathbf{K}_2^t \Delta \mathbf{w} = \Delta \mathbf{p} \quad (7.274)$$

$$\begin{bmatrix} 0 & 0 \\ 0 & 1.0400 \end{bmatrix} \begin{bmatrix} \Delta w_1 \\ \Delta w_2 \end{bmatrix} = \begin{bmatrix} 0 \\ \Delta p_2 \end{bmatrix}$$

Since the tangent matrix is singular, it cannot be used in the constant arc method for the first iteration in the load step following the bifurcation point. Equation (7.274) can, however, be used to study the properties of an infinitesimal continuation.

Consider an infinitesimal displacement step (dw_1, dw_2) beyond the bifurcation point B. Equation (7.274) shows that the infinitesimal displacement increment dw_1 can be chosen freely, whereas the infinitesimal displacement increment dw_2 depends on the load increment dp_2 . If dw_1 is set to null and dp_2 differs from null, the continuation is on the primary branch of the load path. If the infinitesimal load increment dp_2 is set to null and dw_1 differs from null, the continuation is on a secondary branch of the load path. The tangent to the secondary branch of the load path at the bifurcation point is parallel to the eigenvector of \mathbf{K}_1^t which corresponds to eigenvalue 0.

Total stiffness matrix: The total stiffness matrix of the truss at the bifurcation point is computed with (7.10) using the displacement state $(w_{1c}, w_{2c}) = (0.0, -0.175379)$:

$$\begin{array}{|c|c|c|} \hline 0.144621 & 0 & 0 \\ \hline 0 & 1.504621 & -0.175379 \\ \hline \end{array} = \begin{array}{|c|} \hline 0 \\ \hline -0.263879 \\ \hline \end{array} \quad (7.275)$$

Expression (7.275) shows that the total stiffness matrix at the bifurcation point is regular and well conditioned. At a neighbouring point Q with the displacement state $(w_1, w_2) = (0.05, -0.18)$, substitution into (7.10) yields the following equation:

$$\begin{array}{|c|c|c|} \hline 0.142500 & -0.041000 & 0.050000 \\ \hline -0.041000 & 1.492400 & -0.180000 \\ \hline \end{array} = \begin{array}{|c|} \hline 0.014505 \\ \hline -0.270682 \\ \hline \end{array} \quad (7.276)$$

Equation (7.276) shows that the total stiffness matrix remains regular in the neighbourhood of the bifurcation point, and that the load vector changes gradually. The equilibrium at point Q does, however, require the load $p_1 = 0.014505$, whereas the problem solution is to be found for $p_1 = 0$. The continuation algorithm will be designed so that the load p_1 can be driven to null.

Taylor series approximations: Consider the governing equations (2.18) and (2.19) of a 2-bar truss which express the load coordinates p_1 and p_2 as functions of the displacement coordinates w_1 and w_2 . The bifurcation point of a truss with aspect ratio $m = 0.2$ is $(w_{1c}, w_{2c}) = (0.0, -0.040833)$ with load $(p_1, p_2) = (0.0, -0.076733)$. Let the increments of the load coordinates for displacement (w_1, w_2) relative to the load coordinates at the bifurcation point be denoted by $(\Delta p_1, \Delta p_2)$. Figure 7.6 shows iso-lines for Δp_1 and Δp_2 which are computed with the exact equations (2.18) and (2.19).

In order to investigate the approximation in the first step of the continuation, load increments Δp_1 and Δp_2 are computed by means of the Taylor series for the loads p_1 and p_2 at the bifurcation point. Since the derivatives of higher than third order are null, the Taylor series with pivot (w_1, w_2) has the following general form:

$$\begin{aligned} f(w_1 + \Delta w_1, w_2 + \Delta w_2) = & f(w_1, w_2) + f_1 \Delta w_1 + f_2 \Delta w_2 + \\ & \frac{1}{2} (f_{11} \Delta w_1^2 + 2f_{12} \Delta w_1 \Delta w_2 + f_{22} \Delta w_2^2) + \\ & \frac{1}{6} (f_{111} \Delta w_1^3 + 3f_{112} \Delta w_1^2 \Delta w_2 + 3f_{122} \Delta w_1 \Delta w_2^2 + f_{222} \Delta w_2^3) \end{aligned} \quad (7.277)$$

w_1, w_2 pivot of the Taylor series

Δw_i increment of w_i

f_m derivative of function f with respect to coordinate w_m

Consider the nonlinear equilibrium equations (2.18) and (2.19) of the truss:

$$p_1 = w_1(2m^2 + w_1^2 + w_2(2 + w_2)) \quad (7.278)$$

$$p_2 = (1 + w_2)(w_1^2 + w_2(2 + w_2)) \quad (7.279)$$

The functions are differentiated to obtain the first derivatives for the Taylor series:

$$\frac{\partial p_1}{\partial w_1} = 2m^2 + 3w_1^2 + w_2(2 + w_2) \quad \frac{\partial p_2}{\partial w_1} = 2w_1(1 + w_2) \quad (7.280)$$

$$\frac{\partial p_1}{\partial w_2} = 2w_1 + 2w_1w_2 \quad \frac{\partial p_2}{\partial w_2} = w_1^2 + 2 + 6w_2 + 3w_2^2$$

The second derivatives of the load coordinates are:

$$\frac{\partial^2 p_1}{\partial w_1^2} = 6w_1 \quad \frac{\partial^2 p_2}{\partial w_1^2} = 2(1 + w_2) \quad (7.281)$$

$$\frac{\partial^2 p_1}{\partial w_1 \partial w_2} = 2(1 + w_2) \quad \frac{\partial^2 p_2}{\partial w_1 \partial w_2} = 2w_1$$

$$\frac{\partial^2 p_1}{\partial w_2^2} = 2w_1 \quad \frac{\partial^2 p_2}{\partial w_2^2} = 6(1 + w_2)$$

The third derivatives of the load coordinates are constant:

$$\frac{\partial^3 p_1}{\partial w_1^3} = 6 \quad \frac{\partial^3 p_2}{\partial w_1^2 \partial w_2} = 2 \quad (7.282)$$

$$\frac{\partial^3 p_1}{\partial w_1 \partial w_2^2} = 2 \quad \frac{\partial^3 p_2}{\partial w_2^3} = 6$$

$$\frac{\partial^3 p_1}{\partial w_2^3} = 2$$

Figure 7.7 shows the approximation of the isolines for the load increments Δp_1 and Δp_2 which is obtained if the Taylor series is written at the bifurcation point and truncated after the linear term. The load increment Δp_1 is null in the plane, and the isolines for Δp_2 are parallel to axis w_1 . The isolines for $\Delta p_1 = 0$ and $\Delta p_2 = 0$, which nearly coincide in figure 7.6, are not adequately approximated in figure 7.7. Since the continuation follows the isoline $\Delta p_1 = 0$ and the increment Δp_2 is very small, a solution cannot be obtained with the Taylor series approximation in figure 7.7.

Figure 7.8 shows the approximation of the isolines for the load increments Δp_1 and Δp_2 which is obtained if the quadratic terms are retained in the Taylor series at the bifurcation point. While the approximation of the isolines for Δp_2 is improved, the isoline for $\Delta p_1 = 0$ is parallel to axis w_1 and lies far from the isoline for $\Delta p_2 = 0$. The approximation in figure 7.8 is also not suitable for the continuation of the load path.

Figure 7.9 shows the approximation of the isolines for the load increments Δp_1 and Δp_2 which is obtained if the cubic terms are retained in the Taylor series at the bifurcation point. Both sets of isolines are now well approximated and the isolines for $\Delta p_1 = 0$ and $\Delta p_2 = 0$ nearly coincide. The approximation in figure 7.8 is suitable for the continuation of the load path.

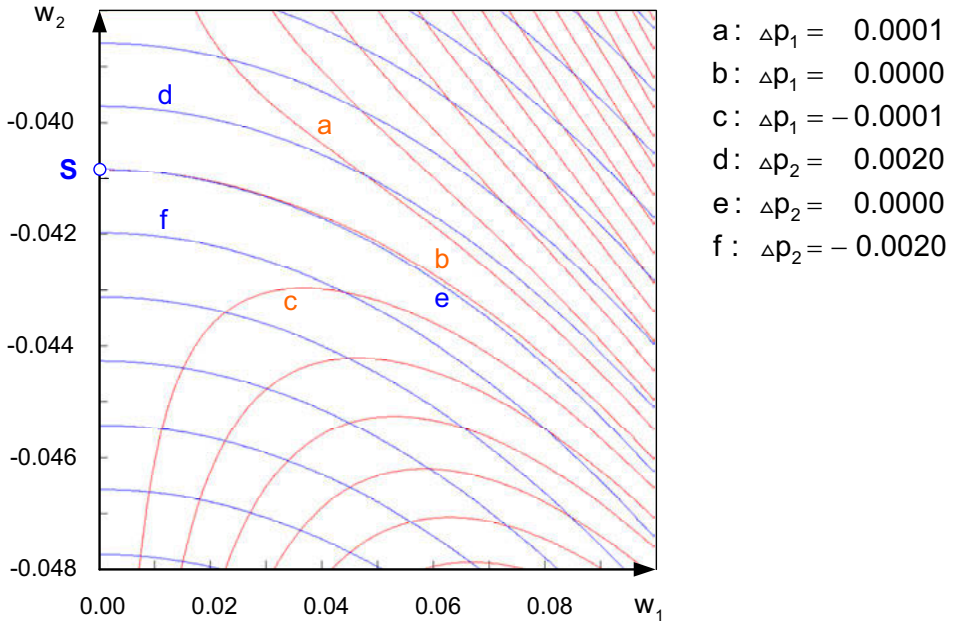


Figure 7.6 : Isolines of the exact load increments relative to the loads at the bifurcation point S

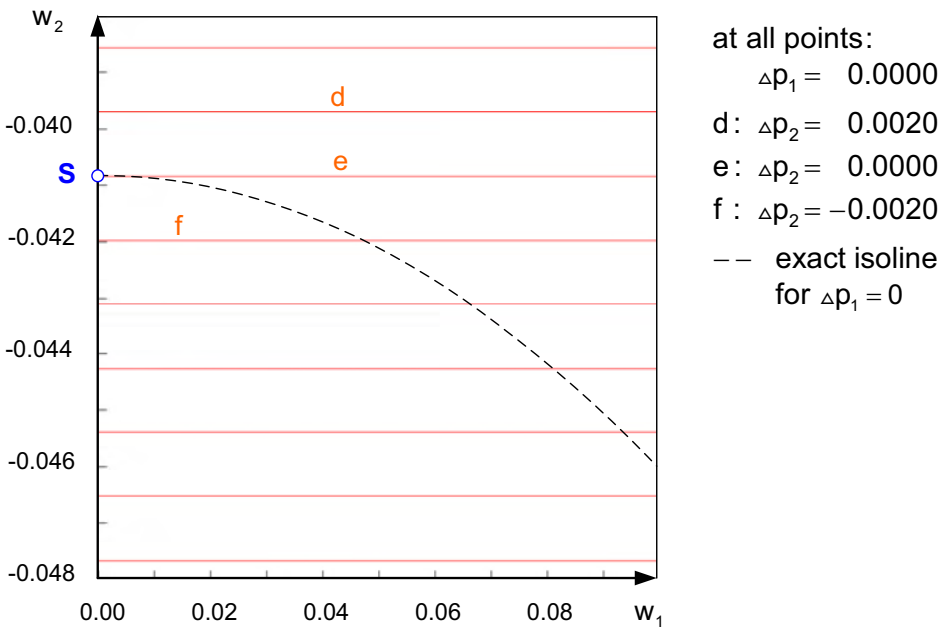


Figure 7.7 : Isolines of the approximate load increments for the linear Taylor series at the bifurcation point S

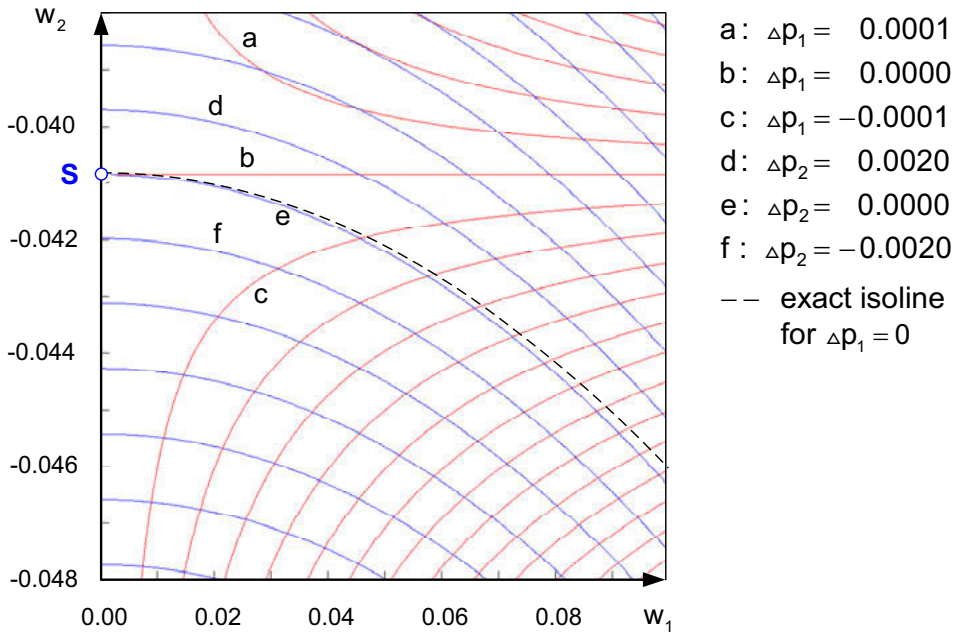


Figure 7.8 : Isolines of the approximate load increments for the quadratic Taylor series at bifurcation point S

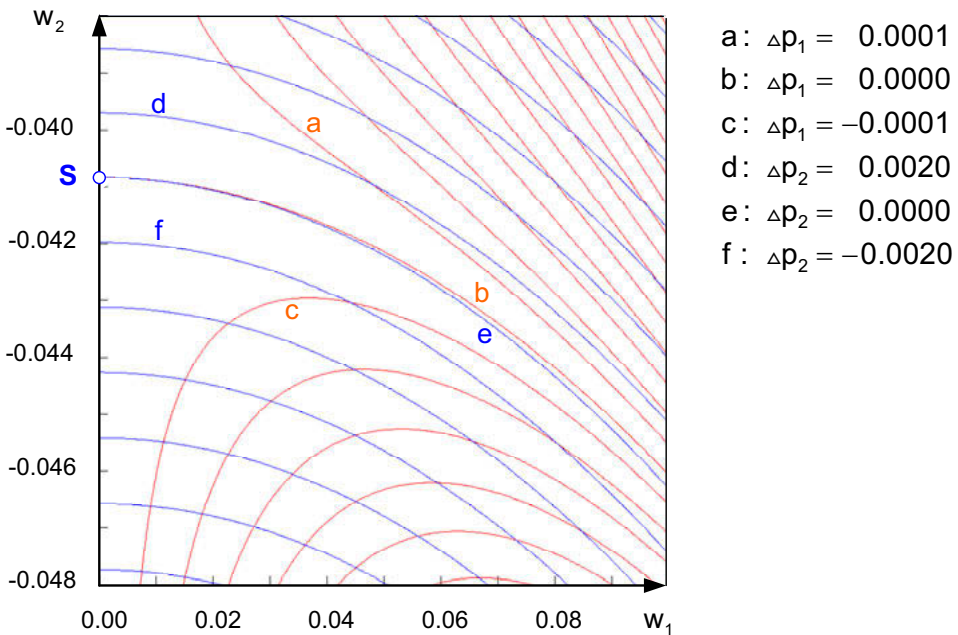


Figure 7.9 : Isolines of the approximate load increments for the cubic Taylor series at the bifurcation point S

In order to investigate the approximation in the second step of the continuation, load increments Δp_1 and Δp_2 are computed by means of the Taylor series for the loads p_1 and p_2 at a pivot P on the load path with displacement state $(0.03, -0.041303)$. Figure 7.10 shows the approximation of the isolines for the load increments Δp_1 and Δp_2 which is obtained if the Taylor series is written at the pivot P and truncated after the linear term. The isolines for Δp_1 are parallel to the tangent to the load path at P . The directions of the isolines at P differ slightly. The exact isoline $\Delta p_1 = 0$ deviates significantly from its linear approximation.

Figure 7.11 shows the approximation of the isolines for the load increments Δp_1 and Δp_2 which is obtained if the quadratic terms are retained in the Taylor series at the pivot P . The approximation of the isoline $\Delta p_1 = 0$ is improved significantly over the linear approximation. The quadratic approximation in figure 7.11 is suited for the continuation of the path with a small load step in the immediate vicinity of pivot P .

Figure 7.12 shows the approximation of the isolines for the load increments Δp_1 and Δp_2 which is obtained if the cubic terms are retained in the Taylor series at pivot P . While the isolines for Δp_2 do not change significantly relative to figure 7.11, the isolines for Δp_1 differ significantly at distances $\Delta w_1 > 0.03$.

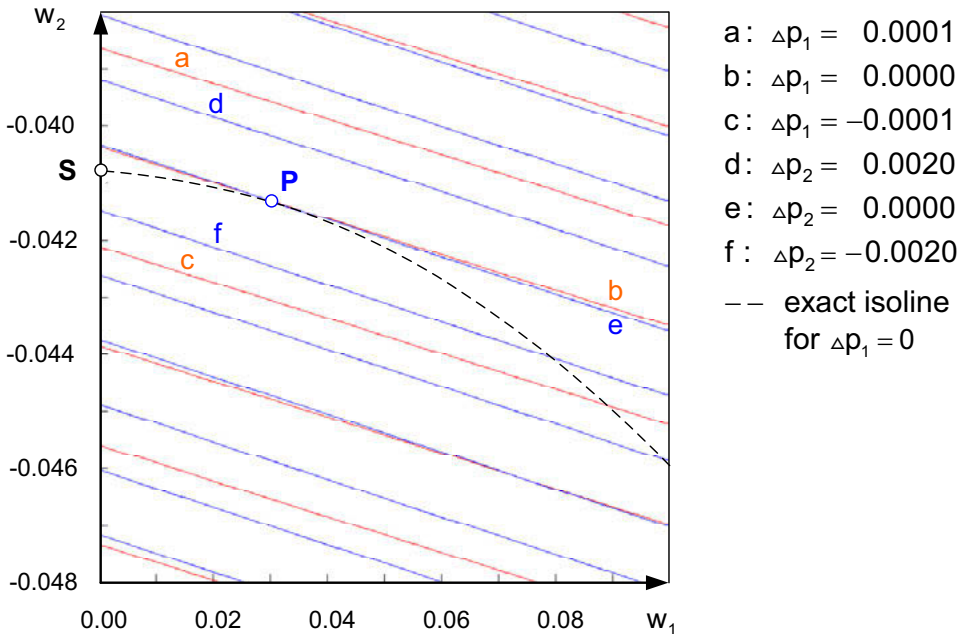


Figure 7.10 : Isolines of the approximate increments for the linear Taylor series at pivot $P = (0.03, -0.041303)$

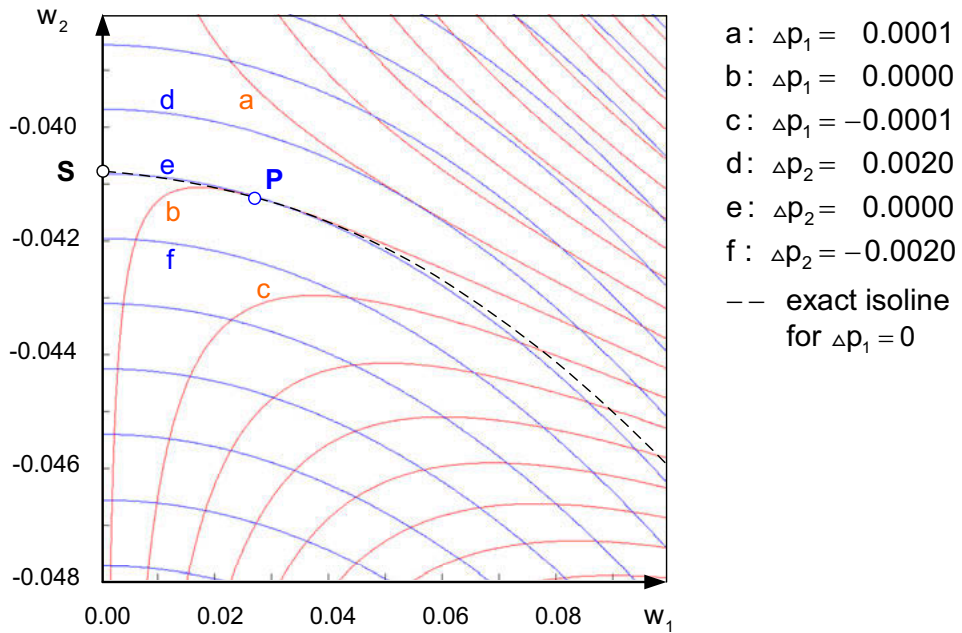


Figure 7.11 : Isolines of the approximate increments for the quadratic Taylor series at pivot $P = (0.03, -0.041303)$

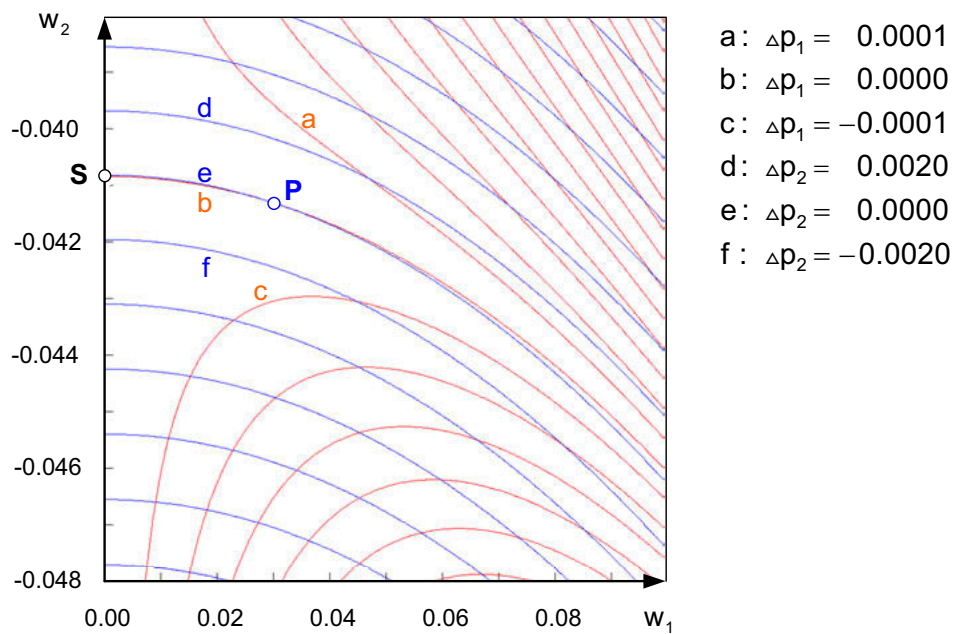


Figure 7.12 : Isolines of the approximate increments for the cubic Taylor series at pivot $P = (0.03, -0.041303)$

In order to investigate the approximation in the steps of the continuation that are distant from the bifurcation point S, load increments Δp_1 and Δp_2 are computed by means of the Taylor series for the loads p_1 and p_2 at a pivot Q on the load path with displacement state $(0.10, -0.046061)$. Figure 7.13 shows the approximation of the isolines for the load increments Δp_1 and Δp_2 which is obtained if the Taylor series is written at the pivot Q and truncated after the linear term. The isolines for Δp_1 are parallel to the tangent to the load path at Q. The directions of the isolines at Q differ slightly. The approximate isoline $\Delta p_1 = 0$ is a good approximation of the exact isoline. It is suited for the continuation of the load path.

Figure 7.14 shows the approximation of the isolines for the load increments Δp_1 and Δp_2 which is obtained if the quadratic terms are retained in the Taylor series at the pivot Q. The approximate isoline $\Delta p_1 = 0$ is a very good approximation of the exact isoline.

Figure 7.15 shows the approximation of the isolines for the load increments Δp_1 and Δp_2 which is obtained if the cubic terms are retained in the Taylor series at pivot Q. The isolines do not change significantly relative to figure 7.14.

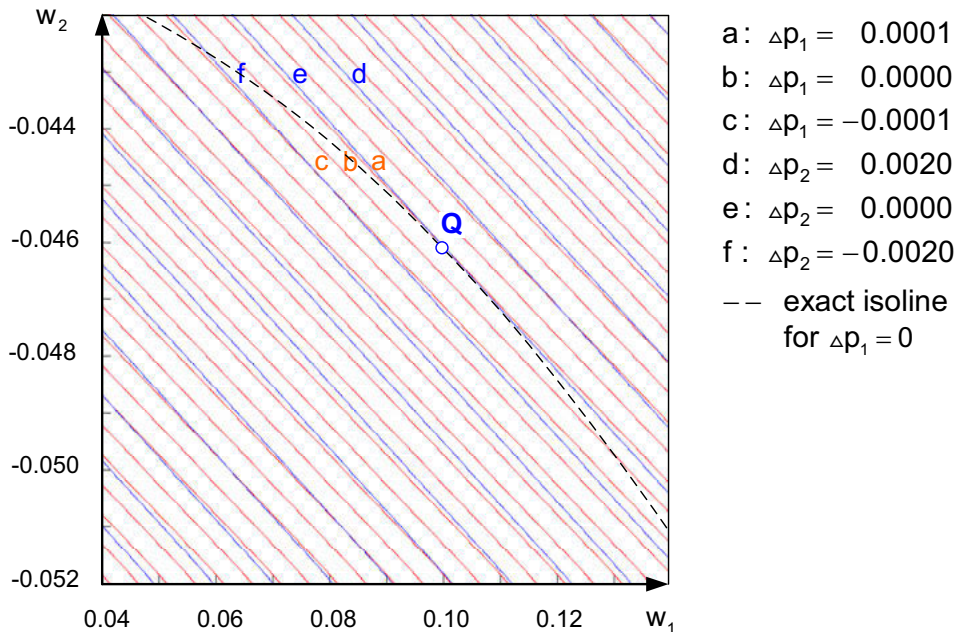


Figure 7.13 : Isolines of the approximate increments for the linear Taylor series at pivot Q = $(0.10, -0.046061)$

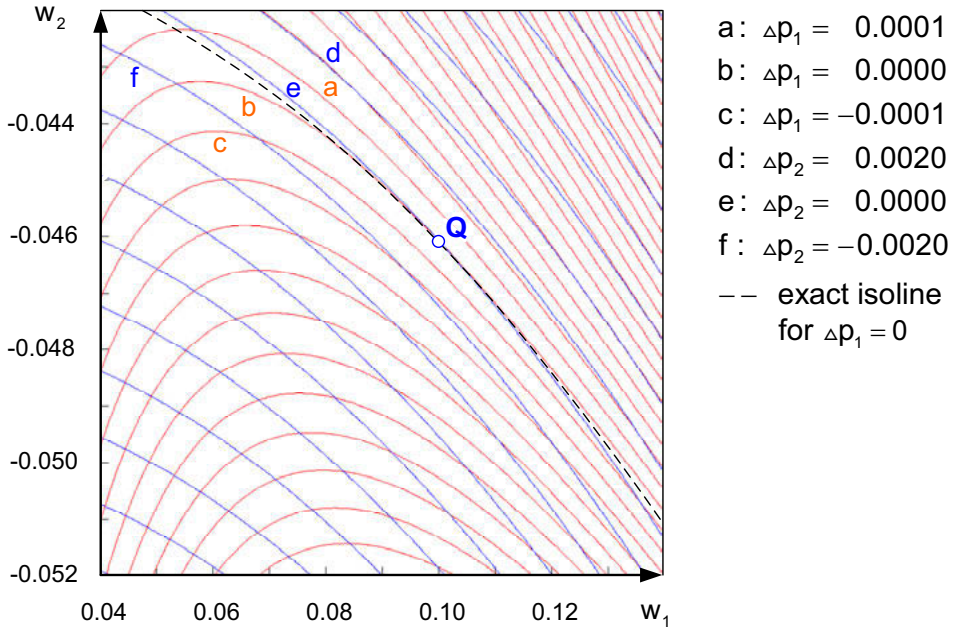


Figure 7.14 : Isolines of the approximate increments for the quadratic Taylor series at pivot $Q = (0.10, -0.046061)$

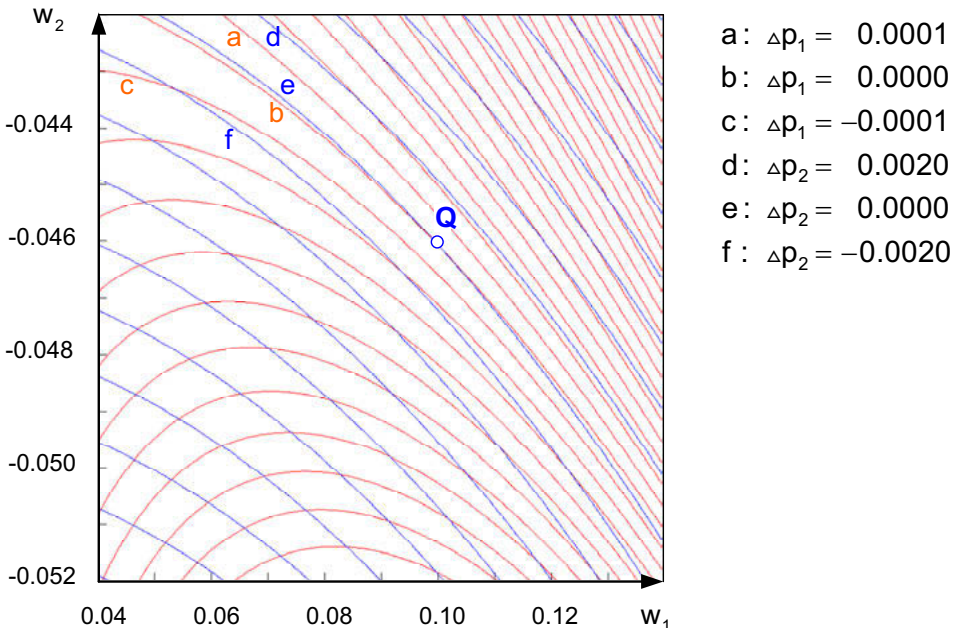


Figure 7.15 : Isolines of the approximate increments for the cubic Taylor series at pivot $Q = (0.10, -0.046061)$

Determinant of the tangent matrix: The iterative solution of the governing equations for a continuation depends on the sign and magnitude of the determinant of the tangential stiffness matrix in the neighbourhood of the load path. Curve “a” in figure 7.16 shows the continuation of the load path of the normalised 2-bar truss with shape factor $m = 0.2$ at a bifurcation. Curves “b”, “c” and “d” show the isolines for the determinant of the tangential matrix of the truss in the vicinity of the load path. The absolute value of 0.001 of the determinant on isolines “c” and “d” is significantly less than the determinant $m^2 = 0.160$ of the linear stiffness matrix of the truss.

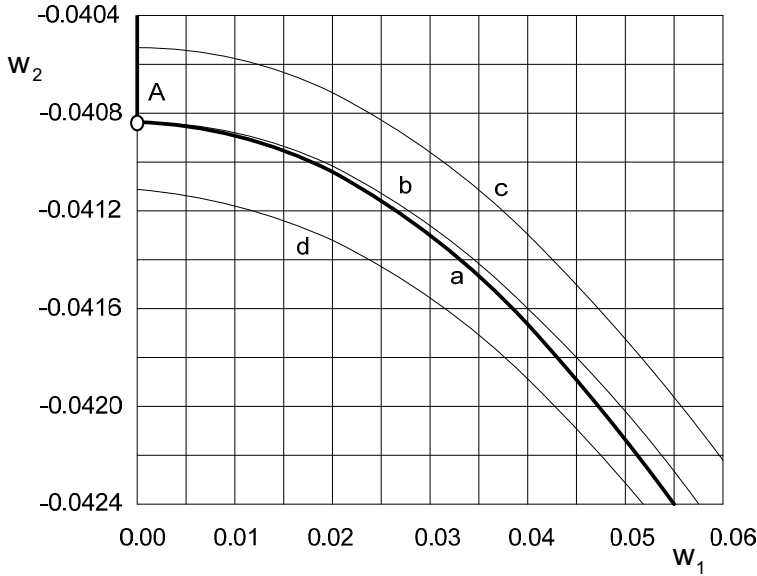


Figure 7.16 Load path and isolines of the determinant at a bifurcation point A

- a continuation of the load path
- b isoline for determinant = 0.000
- c isoline for determinant = 0.001
- d isoline for determinant = -0.001

Table 7.2 Comparison of the Load Path and the Null-Line of the Determinant

w_1	w_2 for load path	w_2 for null line
0.000	-0.0408337	-0.0408337
0.010	-0.0408858	-0.0408811
0.020	-0.0410422	-0.0410233
0.030	-0.0413030	-0.0412603
0.040	-0.0416681	-0.0415922
0.050	-0.0421378	-0.0420192
0.060	-0.0427122	-0.0425413

The proximity of curves “a” and “b” in figure 7.16 has a significant influence on the stability of the iteration procedures for the determination of the load path. Since the determinant is small, small unbalanced loads lead to large displacements. Even small displacements are sufficient to shift the trial points between the zones with positive and negative values of the determinant. The iteration tends to diverge.

7.7.3 Continuation beyond the Bifurcation Point of a 2-Bar Truss

The investigation in section 7.7.2 shows that a linear approximation of the governing equations of a 2-bar truss at a bifurcation point is not a suitable method of solution. Alternate numerical methods of continuation for 2-bar trusses are considered in this section. These methods consider trial solutions that are not located on the exact load path. Since the approximate trial configurations are in equilibrium, load component p_1 may differ from null. Therefore the general equilibrium equations (2.18) and (2.19) must be considered:

$$\begin{aligned} p_1 &= w_1(2m^2 + w_1^2 + w_2(2 + w_2)) = w_1(w_1^2 + x^2 + 2m^2 - 1) \\ p_2 &= (1 + w_2)(w_1^2 + w_2(2 + w_2)) = x(w_1^2 + x^2 - 1) \\ w &:= 1 + w_2 \end{aligned}$$

These equations contain the variables w_1, w_2, p_1 and p_2 . There are 6 choices for the two independent variables.

Analytical solution: Analytical solutions are available for 5 of the 6 choices of the independent variables. Choices (d) and (e) require the solution of cubic equations.

$$\begin{aligned} \text{(a) } w_1, w_2: \quad p_1 &= w_1(w_1^2 + x^2 + 2m^2 - 1) & (7.284) \\ p_2 &= x(w_1^2 + x^2 - 1) \end{aligned}$$

$$\begin{aligned} \text{(b) } w_1, p_1: \quad x^2 &= 1 - 2m^2 - w_1^2 + \frac{p_1}{w_1} & (7.284) \\ p_2 &= x(w_1^2 + x^2 - 1) \end{aligned}$$

$$\begin{aligned} \text{(c) } w_2, p_2: \quad w_1^2 &= 1 - x^2 + \frac{p_2}{x} & (7.285) \\ p_1 &= w_1(w_1^2 + x^2 + 2m^2 - 1) \end{aligned}$$

$$\begin{aligned} \text{(d) } w_1, p_2: \quad x^3 + (w_1^2 - 1)x - p_2 &= 0 & (7.286) \\ p_1 &= w_1(w_1^2 + x^2 + 2m^2 - 1) \end{aligned}$$

$$\begin{aligned} \text{(e) } w_2, p_1: \quad w_1^3 + (x^2 + 2m^2 - 1)w_1 - p_1 &= 0 & (7.287) \\ p_2 &= x(w_1^2 + x^2 - 1) \end{aligned}$$

An analytical solution is not available if the load components p_1 and p_2 are chosen as independent variables.

Numerical solution: Numerical solutions for the continuation are determined with the total stiffness matrix in (7.1), the tangential stiffness matrix in (7.3) or the secant stiffness matrix in (7.4). Consider the governing equations (7.10) containing the total stiffness matrix. The matrix is computed for specified values of the displacements. As in the analytical method, 2 of the variables w_1, w_2, p_1, p_2 are independent variables and the other 2 are computed with the equations. If the displacements are chosen as

independent variables, the result obtained is exact. In all other cases, at least one displacement w_k is computed with the equations. Since the computed displacement will in general differ from the displacement for which the total stiffness matrix was computed, the computed dependent variables are not exact and must be improved by iteration. In contrast to the analytical solution, approximate numerical solutions are possible if the load components are chosen as independent variables.

Iteration: A great variety of iteration schemes can be designed for the numerical computation of the continuation using the different stiffness matrices mentioned above. These schemes differ in the type of the stiffness matrix, the choice of the independent variables and the strategy for the recomputation of the stiffness matrix in the course of the iteration. Convergence of the iteration procedures depends on the choice of the initial values of the independent variables. Many of the potential iteration schemes diverge near bifurcation points. In the case of the 2-bar truss, this is due to the dominance of the third derivatives of the loads with respect to the displacements in the Taylor series expansion for the load components.

The displacement increment at a bifurcation point follows from expression (7.148) of bifurcation theory. It consists of two components: a component proportional to the eigenvector \mathbf{x}_0 corresponding to eigenvalue 0 of the tangent stiffness matrix at the bifurcation point, and a component \mathbf{z}_0 that satisfies the governing equations at the bifurcation point. The eigenvector \mathbf{x}_0 is known at the bifurcation point. It can be used in the first cycle of iteration of the first load step beyond the bifurcation point to set the trial displacement increment equal to a fraction of \mathbf{x}_0 . The component \mathbf{z}_0 cannot be determined directly with the singular governing equations. Two approaches can be used to circumvent this problem. The incremental stiffness matrix can be computed for the trial configuration, or the total stiffness matrix can be used in the iteration as it is not singular near the bifurcation point. After the load path has left the vicinity of the singular point, it can be continued with the constant arc increment method.

Example: Consider the 2-bar truss with aspect ratio $m = 0.20$ and bifurcation point $(w_1, w_2) = (0.0, -0.040833)$ for which the Taylor series at different pivots and the determinant of the tangent stiffness matrix were studied in section 7.7.2. The product of the total stiffness matrix with the total displacement at the bifurcation point equals the total load at the bifurcation point:

$$\mathbf{K}_s \mathbf{w}_s = \mathbf{p}_s \tag{7.288}$$

0.039166304	0	0	=	0
0	1.879166306	-0.040833695		-0.076733303

- \mathbf{K}_s total stiffness matrix at the singular point
- \mathbf{w}_s total displacement at the singular point
- \mathbf{p}_s total load at the singular point

The first cycle of iteration of the first load step of the continuation consists of the following operations:

(a) The displacement increment is set equal to a fraction 0.03 of the eigenvector at the bifurcation point. This increment leads to a trial displacement $\mathbf{w}_a = (w_{1s}, w_{2s}) =$

(0.03, -0.040833695). The total stiffness matrix \mathbf{K}_a is computed for displacement state \mathbf{w}_a . Load \mathbf{p}_a is set equal to the total load \mathbf{p}_s at the beginning of the cycle. The equations are solved for the total displacement \mathbf{w}_b :

$$\mathbf{K}_a \mathbf{w}_b = \mathbf{p}_a \quad (7.289)$$

0.040066305	0.028774989	0.029652215
0.028774989	1.879166305	-0.041287749

$$= \begin{matrix} 0 \\ -0.076733304 \end{matrix}$$

$\mathbf{K}_a(\mathbf{w}_a)$ approximate total stiffness matrix in the trial state

\mathbf{p}_a total load at the start of the cycle

\mathbf{w}_b approximate total displacement at the end of the cycle

(b) The total stiffness matrix \mathbf{K}_b is computed for the displacement state \mathbf{w}_b determined in operation (a). Matrix \mathbf{K}_b is multiplied with displacement \mathbf{w}_b to determine the load \mathbf{p}_b for which the frame is in equilibrium:

$$\mathbf{K}_b \mathbf{w}_b = \mathbf{p}_b$$

0.039591505	0.028427938	0.029652211
0.028427938	1.877841432	-0.041287749

$$= \begin{matrix} 0.000000250 \\ -0.076688894 \end{matrix}$$

$\mathbf{K}_b(\mathbf{w}_b)$ total stiffness matrix at the end of operation (a)

\mathbf{p}_b total load at the end of operation (b)

\mathbf{w}_b total displacement at the end of operation (a)

(c) The component p_{b1} of load \mathbf{p}_b contradicts the specified load $p_1 = 0$. In order to correct the load, the displacement $\Delta \mathbf{w}_c$ due to the load $\Delta \mathbf{p}_c = (p_{b1}, 0)$ is computed with the total stiffness matrix \mathbf{K}_b :

$$\mathbf{K}_b \Delta \mathbf{w}_c = \Delta \mathbf{p}_c$$

0.039591505	0.028427938	0.000006386
0.028427938	1.877841432	-0.000000097

$$= \begin{matrix} 0.000000250 \\ 0.000000000 \end{matrix}$$

$\Delta \mathbf{p}_b$ total load correction

$\Delta \mathbf{w}_b$ total displacement correction

The corrected displacements and loads at the end of operation (c) are:

$$\mathbf{w}_c = \mathbf{w}_b - \Delta \mathbf{w}_c = \begin{matrix} 0.029652211 \\ -0.041287749 \end{matrix} - \begin{matrix} 0.000006386 \\ -0.000000097 \end{matrix} = \begin{matrix} 0.029645825 \\ -0.041287652 \end{matrix}$$

$$\mathbf{p}_c = \mathbf{p}_b - \Delta \mathbf{p}_c = \begin{matrix} 0.000000250 \\ -0.076688894 \end{matrix} - \begin{matrix} 0.000000250 \\ 0.000000000 \end{matrix} = \begin{matrix} 0.000000000 \\ -0.076688894 \end{matrix}$$

(d) Since the total stiffness matrix was computed for displacement \mathbf{w}_b , the truss is not in equilibrium for load \mathbf{p}_c and displacement \mathbf{w}_c . The unbalanced forces $\Delta \mathbf{p}_e$ are computed with the equilibrium equations (7.278) and (7.279). The displacement $\Delta \mathbf{w}_e$ due to the unbalanced forces is determined with the total stiffness \mathbf{K}_b :

$$\mathbf{K}_b \Delta \mathbf{w}_e = \Delta \mathbf{p}_e$$

$$\begin{bmatrix} 0.039591505 & 0.028427938 \\ 0.028427938 & 1.877841432 \end{bmatrix} \begin{bmatrix} -0.000006101 \\ 0.000004452 \end{bmatrix} = \begin{bmatrix} -0.000000244 \\ 0.000008094 \end{bmatrix}$$

$\Delta \mathbf{p}_e$ unbalanced forces

$\Delta \mathbf{w}_e$ displacement correction for the unbalanced forces

The corrected displacements are:

$$\mathbf{w}_s = \mathbf{w}_c - \Delta \mathbf{w}_e = \begin{bmatrix} 0.029645825 \\ -0.041287652 \end{bmatrix} - \begin{bmatrix} -0.000006101 \\ 0.000004452 \end{bmatrix} = \begin{bmatrix} 0.029651926 \\ -0.041292104 \end{bmatrix}$$

The total stiffness matrix is recomputed and multiplied with the computed displacement. The resulting load is nearly proportional to the pattern load:

$$\mathbf{K}_s \mathbf{w}_s = \mathbf{p}_s \quad (7.290)$$

$$\begin{bmatrix} 0.039587133 & 0.028427536 \\ 0.028427536 & 1.877828726 \end{bmatrix} \begin{bmatrix} 0.029651926 \\ -0.041292104 \end{bmatrix} = \begin{bmatrix} 0.000000002 \\ -0.076696568 \end{bmatrix}$$

7.7.4 First Step beyond a Bifurcation Point

The method of analysis presented in the example of section 7.7.3 is generalised for the continuation of load paths beyond bifurcation points of frames. The load p_2 acting on the 2-bar truss is replaced by the scaled pattern load $\lambda \mathbf{p}_t$. The load p_1 that is null for the 2-bar truss is replaced by the load component normal to \mathbf{q}_t that is null for the frame. The displacement component u_1 for the truss is replaced by the eigendisplacement $\mu \mathbf{x}_0$ of the frame, the displacement component u_2 for the truss by the displacement component normal to \mathbf{x}_0 in the frame.

Consider a frame with a pattern load \mathbf{q}_t and assume that state s of the frame with displacement $\mathbf{d}^{(s)}$, reaction $\mathbf{r}^{(s)}$ and load $\mathbf{q}^{(s)}$ is a bifurcation point. The first step of the load path continuation is treated in this section, the general step in section 7.7.5

Iteration cycle 0: The displacement increment in cycle 0 is set equal to a prescribed fraction μ of eigendisplacement \mathbf{x}_0 . The corresponding reaction increment is $\mu \mathbf{r}_0$. The load $\mathbf{q}^{(s)}$ is kept constant. The state of the frame at the end of cycle 0 is:

$$\mathbf{d}^{(s+1,0)} = \mathbf{d}^{(s)} + \mu \mathbf{x}_0 \quad (7.291)$$

$$\mathbf{r}^{(s+1,0)} = \mathbf{r}^{(s)} + \mu \mathbf{r}_0 \quad (7.292)$$

$$\mathbf{q}^{(s+1,0)} = \mathbf{q}^{(s)} \quad (7.293)$$

Iteration cycle m : Let the displacement variables and the system equations be ordered so that the free displacements precede the prescribed displacements as in equation (6.39). The general cycle m consists of the following operations:

(a) The approximate total stiffness matrix \mathbf{K}_a is computed for the displacement state at the end of the previous cycle of iteration. Expressions for the total stiffness matrices for trusses are given in section 7.7.6 and expressions for frames in section 7.7.7. The approximate total displacements \mathbf{d}_b and reactions \mathbf{r}_b for state $s+1$ are computed for the load \mathbf{q}_a acting at the end of the previous cycle $m-1$:

$$\mathbf{K}_a \mathbf{d}_b = \mathbf{q}_a + \mathbf{r}_b \quad (7.294)$$

$$\begin{bmatrix} \mathbf{K}_{a11} & \mathbf{K}_{a12} & \mathbf{d}_{b1} \\ \mathbf{K}_{a21} & \mathbf{K}_{a22} & \mathbf{d}_{b2} \end{bmatrix} = \begin{bmatrix} \mathbf{q}_{a1} \\ \mathbf{q}_{a2} \end{bmatrix} + \begin{bmatrix} 0 \\ \mathbf{r}_{b2} \end{bmatrix}$$

$$\mathbf{K}_a = \mathbf{K}(\mathbf{d}_a)$$

$$\mathbf{q}_a = \mathbf{q}^{(s+1, m-1)}$$

(b) The total stiffness matrix \mathbf{K}_b is computed for the displacement state \mathbf{d}_b determined in operation (a). Matrix \mathbf{K}_b is multiplied with displacement \mathbf{d}_b to determine the total equilibrium load \mathbf{q}_b for which the frame is in equilibrium:

$$\mathbf{q}_b = \mathbf{K}_b \mathbf{d}_b - \mathbf{r}_b \quad (7.295)$$

$$\begin{bmatrix} \mathbf{q}_{b1} \\ \mathbf{q}_{b2} \end{bmatrix} = \begin{bmatrix} \mathbf{K}_{b11} & \mathbf{K}_{b12} & \mathbf{d}_{b1} \\ \mathbf{K}_{b21} & \mathbf{K}_{b22} & \mathbf{d}_{b2} \end{bmatrix} - \begin{bmatrix} 0 \\ \mathbf{r}_{b2} \end{bmatrix}$$

$$\mathbf{K}_b = \mathbf{K}(\mathbf{d}_b)$$

(c) The load \mathbf{q}_b in (7.295) is not proportional to the pattern load \mathbf{q}_t . It is decomposed into a component $\theta \mathbf{q}_t$ and a component $\Delta \mathbf{q}_b$ which is normal to the pattern load \mathbf{q}_t :

$$\mathbf{q}_b = \theta \mathbf{q}_t + \Delta \mathbf{q}_b \quad (7.296)$$

$$\mathbf{q}_t^T \Delta \mathbf{q}_b = 0 \quad \rightarrow \quad \theta = \frac{\mathbf{q}_t^T \mathbf{q}_b}{\mathbf{q}_t^T \mathbf{q}_t}$$

The displacement correction $\Delta \mathbf{d}_b$ and the reaction correction $\Delta \mathbf{r}_b$ due to the load correction $\Delta \mathbf{q}_b$ are determined with the total stiffness matrix \mathbf{K}_b :

$$\mathbf{K}_b \Delta \mathbf{d}_b = \Delta \mathbf{q}_b + \Delta \mathbf{r}_b \quad (7.297)$$

$$\begin{bmatrix} \mathbf{K}_{b11} & \mathbf{K}_{b12} & \Delta \mathbf{d}_{b1} \\ \mathbf{K}_{b21} & \mathbf{K}_{b22} & \Delta \mathbf{d}_{b2} \end{bmatrix} = \begin{bmatrix} \Delta \mathbf{q}_{b1} \\ \Delta \mathbf{q}_{b2} \end{bmatrix} + \begin{bmatrix} 0 \\ \Delta \mathbf{r}_{b2} \end{bmatrix}$$

The corrected displacements, reactions and loads are computed by subtracting (7.297) from (7.295):

$$\mathbf{K}_b (\mathbf{d}_b - \Delta \mathbf{d}_b) = (\mathbf{q}_b - \Delta \mathbf{q}_b) + (\mathbf{r}_b - \Delta \mathbf{r}_b) \quad (7.298)$$

$$\mathbf{K}_b \mathbf{d}_c = \mathbf{q}_c + \mathbf{r}_c \quad (7.299)$$

$$\mathbf{d}_c = \mathbf{d}_b - \Delta \mathbf{d}_b \quad (7.299)$$

$$\mathbf{r}_c = \mathbf{r}_b - \Delta \mathbf{r}_b \quad (7.300)$$

$$\mathbf{q}_c = \mathbf{q}_b - \Delta \mathbf{q}_b \quad (7.301)$$

(d) Let the exact total stiffness matrix, displacements and reactions for load \mathbf{q}_c be \mathbf{K}_e , \mathbf{d}_e and \mathbf{r}_e :

$$\mathbf{K}_e \mathbf{d}_e = \mathbf{q}_c + \mathbf{r}_e \quad (7.302)$$

The unbalanced forces \mathbf{e} in state $(\mathbf{d}_c, \mathbf{r}_c)$ are computed with the methods in sections 4.3.3 and 5.3.4:

$$\mathbf{e} = \mathbf{q}_c + \mathbf{r}_c - \mathbf{K}_e \mathbf{d}_c \quad (7.303)$$

It is assumed that the unbalanced forces are due to the use of the approximate total stiffness matrix \mathbf{K}_b instead of the exact total stiffness matrix \mathbf{K}_e in equation (7.302):

$$\mathbf{K}_b \mathbf{d}_e = \mathbf{q}_c + \mathbf{r}_e + \mathbf{e} \quad (7.304)$$

Equation (7.298) is subtracted from (7.304) to obtain the correction $\Delta \mathbf{d}_c$ of the total displacement and $\Delta \mathbf{r}_c$ of the total reaction for the unbalanced forces \mathbf{e} :

$$\mathbf{K}_b \Delta \mathbf{d}_c = \Delta \mathbf{r}_c + \mathbf{e} \quad (7.305)$$

$$\Delta \mathbf{d}_c = \mathbf{d}_e - \mathbf{d}_c$$

$$\Delta \mathbf{r}_c = \mathbf{r}_e - \mathbf{r}_c$$

The increments (7.305) are used to compute the trial state m for the load step:

$$\mathbf{d}^{(s+1,m)} = \mathbf{d}_c + \Delta \mathbf{d}_c \quad (7.306)$$

$$\mathbf{r}^{(s+1,m)} = \mathbf{r}_c + \Delta \mathbf{r}_c$$

$$\mathbf{q}^{(s+1,m)} = \mathbf{q}_c$$

Termination of the iteration: State (7.306) of the frame is not an exact equilibrium state, since the displacement correction $\Delta \mathbf{d}_c$ and the reaction correction $\Delta \mathbf{r}_c$ are computed with the total stiffness matrix \mathbf{K}_b that corresponds to the displacement state \mathbf{d}_b . If the error norm $|\mathbf{e}|$ exceeds a specified limit, the iteration is continued with cycle $m + 1$. Otherwise the state at the end of cycle m is state $s + 1$ of the frame.

7.7.5 General Step of the Continuation at a Bifurcation Point

General load step: The tangent stiffness matrix $\mathbf{K}^{t(s+1)}$ at the end of the first step of the continuation at a bifurcation point is not singular, but the poor approximation of the equilibrium equations by the tangent matrix persists. The smallest eigenvalue $\omega_0^{(s+1)}$ and the corresponding eigendisplacement $\mathbf{x}_0^{(s+1)}$ of matrix $\mathbf{K}^{t(s+1)}$ as well as the associated eigenreaction $\mathbf{r}_0^{(s+1)}$ are computed. The vectors $\mathbf{x}_0^{(s+1)}$ and $\mathbf{r}_0^{(s+1)}$ replace the vectors \mathbf{x}_0 and \mathbf{r}_0 used in the first step of the continuation.

$$\mathbf{K}^{t(s+1)} \mathbf{x}_0^{(s+1)} = \omega_0^{(s+1)} \mathbf{x}_0^{(s+1)} + \mathbf{r}_0^{(s+1)} \quad (7.307)$$

Termination of the continuation: As the load path is continued beyond the bifurcation point, the error in the steps of the continuation increases. The displacement increment in the step is no longer proportional to the eigendisplacement $\mathbf{x}_0^{(s+1)}$. At the same time, the coefficients of the tangent and secant stiffness matrices change so that the numerical problems due to the nearly singular and ill-conditioned equations are reduced. The continuation is terminated when the norm of the unbalanced forces at the end of a load step exceeds a preset limit:

$$|\mathbf{e}| \geq \text{limit } |\mathbf{e}| \Rightarrow \text{return to the constant arc increment method} \quad (7.308)$$

$|\mathbf{e}|$ norm of the unbalanced forces at the end of the load step

$\text{limit } |\mathbf{e}|$ preset limit value of the norm

After the continuation has been terminated, the next part of the load path is again determined with the constant arc increment method described in chapter 6.

7.7.6 Total Stiffness Matrix of a Truss

The total system stiffness matrix \mathbf{K}_s of a truss is derived from the governing equation (4.52) for the error term $\delta e = 0$:

$$\sum_e \int_{C_e} \delta \varepsilon^0 s dv = \delta \mathbf{d}_s^T \mathbf{K}_s \mathbf{d}_s \quad (7.309)$$

\mathbf{K}_s total stiffness matrix of the truss

\mathbf{d}_s system displacement vector

The axial strain ε is substituted from expression (4.32), the variation $\delta \varepsilon^0$ from (4.48):

$$\varepsilon = v_{1,1} + \frac{1}{2}(v_{1,1}^2 + v_{2,1}^2)$$

$$\delta \varepsilon^0 = \delta v_{1,1} + v_{1,1} \delta v_{1,1} + v_{2,1} \delta v_{2,1}$$

v_m displacement coordinates in reference space

$v_{m,1}$ derivative of v_i with respect to reference coordinate y_1

The terms of the product $\varepsilon \delta \varepsilon$ are arranged so that the product leads to a symmetric total stiffness matrix:

$$\varepsilon^0 \delta \varepsilon = c_1 v_{1,1} \delta v_{1,1} + c_2 (v_{1,1} \delta v_{2,1} + v_{2,1} \delta v_{1,1}) + c_3 v_{2,1} \delta v_{2,1} \quad (7.310)$$

$$c_1 = 1 + \frac{3}{2}v_{1,1} + \frac{1}{2}v_{1,1}^2 + \frac{1}{2}v_{2,1}^2 \quad c_2 = \frac{1}{2}v_{2,1} \quad c_3 = \frac{1}{2}v_{1,1} + \frac{1}{2}v_{1,1}^2 + \frac{1}{2}v_{2,1}^2$$

The derivatives of the bar displacements are interpolated with expression (4.64):

$$v_{m,1} = \frac{1}{a} \mathbf{g}_m^T \mathbf{v}_e \quad m \in \{1,2\} \quad (7.311)$$

a length of the bar in the reference configuration

\mathbf{g}_m displacement derivative interpolation vector

\mathbf{v}_e bar displacement vector referred to reference space

The displacement derivatives are constant over the length of a bar. The integration over the volume in (7.309) is performed analytically:

$$\int_{C_e} \delta \varepsilon^0 s dv = \delta \mathbf{v}_e^T \mathbf{K}_e \mathbf{v}_e \quad (7.312)$$

$$\mathbf{K}_e = \frac{EA}{a} (c_1 \mathbf{g}_1 \mathbf{g}_1^T + c_2 (\mathbf{g}_1 \mathbf{g}_2^T + \mathbf{g}_2 \mathbf{g}_1^T) + c_3 \mathbf{g}_2 \mathbf{g}_2^T)$$

The differentiation vectors \mathbf{g}_m are substituted from (4.64):

$$\mathbf{K}_e = \frac{EA}{a} \begin{bmatrix} c_1 & c_2 & -c_1 & -c_2 \\ c_2 & c_3 & -c_2 & -c_3 \\ -c_1 & -c_2 & c_1 & c_2 \\ -c_2 & c_3 & c_2 & c_3 \end{bmatrix} \quad (7.313)$$

The bar displacement vector \mathbf{v}_e is expressed in terms of the system displacement vector \mathbf{d}_s with equation (4.37) and substituted into (7.312). The contributions of the bars are added to obtain the total system stiffness matrix \mathbf{K}_s :

$$\delta \mathbf{d}_s^T \mathbf{K}_s \mathbf{d}_s = \sum_e \delta \mathbf{v}_e^T \mathbf{K}_e \mathbf{v}_e$$

$$\mathbf{K}_s = \sum_e \mathbf{T}_e^T \mathbf{R}_{esy} \mathbf{K}_e \mathbf{R}_{esy}^T \mathbf{T}_e \quad (7.314)$$

\mathbf{T}_e topology matrix of bar e

\mathbf{R}_{esy} element rotation matrix support-reference space

The total system stiffness matrix is assembled with the method described in chapter 6 for incremental stiffness matrices.

7.7.7 Total Stiffness Matrix of a Frame

The total system stiffness matrix \mathbf{K}_s of a frame is derived from governing equation (5.77) for frame equilibrium:

$$\sum_e \int_{C_e} \delta \varepsilon^0 s dv = \delta \mathbf{d}_s^T \mathbf{K}_s \mathbf{d}_s \quad (7.315)$$

\mathbf{K}_s global stiffness matrix of the frame

\mathbf{d}_s system displacement vector

The strain ε is substituted from expression (5.81):

$$\varepsilon = \varepsilon_a + y_2 \varepsilon_b \quad (7.316)$$

$$\varepsilon_a = w_{c,1} - \frac{1}{2} w_{c,1}^2 - \frac{1}{2} w_{c,2,1}^2$$

$$\varepsilon_b = -w_{2,11}$$

ε_a axial strain

ε_b bending strain derivative with respect to y_2

w_{cm} coordinate of the chord displacement in instant space

w_2 displacement of the axis of the member relative to the chord

$w_{cm,1}$ 1.derivative of w_{cm} with respect to the reference coordinate y_1

$w_{2,11}$ 2.derivative of w_2 with respect to the reference coordinate y_1

The variation of the strain follows from (5.107) and (5.110):

$$\delta \varepsilon_a^0 = (1 - w_{c,1}) \delta w_{c,1} \quad (7.317)$$

$$\delta \varepsilon_b^0 = -\delta w_{2,11} \quad (7.318)$$

The terms in the product $\varepsilon \delta \varepsilon^0$ that contain y_2 are discarded because the integral of y_2 over the cross-section of the member is null. The axial and bending strains can therefore be treated independently. The product $\varepsilon_a \delta \varepsilon_a^0$ is not symmetrical in the indices 1 and 2 and therefore does not lead to a symmetric stiffness matrix:

$$\varepsilon_a \delta \varepsilon_a^0 = c_1 w_{c1,1} \delta w_{c1,1} + c_2 w_{c2,1} \delta w_{c1,1} \quad (7.319)$$

$$c_1 = (1 - w_{c1,1}) \left(1 - \frac{1}{2} w_{c1,1}\right) \quad c_2 = -\frac{1}{2} (1 - w_{c1,1}) w_{c2,1}$$

The derivatives of the chord displacements are interpolated with expression (5.150):

$$w_{cm,1} = \frac{1}{a} \mathbf{g}_m^T \mathbf{w}_e \quad (7.320)$$

a length of the member in the reference configuration

\mathbf{g}_m displacement derivative interpolation vector

\mathbf{w}_e element displacement vector in instant space

The displacement derivatives are constant over the length of the member. The integration over the volume in (7.315) is performed analytically:

$$\int_{C_e} \delta \varepsilon_a^0 s_a dv = \delta \mathbf{w}_e^T \mathbf{K}_a \mathbf{w}_e \quad (7.321)$$

$$\mathbf{K}_a = \frac{EA}{a} (c_1 \mathbf{g}_1 \mathbf{g}_1^T + c_2 \mathbf{g}_2 \mathbf{g}_2^T)$$

The differentiation vectors \mathbf{g}_m are substituted from (5.149):

$$\mathbf{K}_a = \frac{EA}{a} \begin{array}{|c|c|c|c|c|c|} \hline c_1 & c_2 & 0 & -c_1 & -c_2 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 \\ \hline -c_1 & -c_2 & 0 & c_1 & c_2 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 \\ \hline \end{array} \quad (7.322)$$

The contribution of the bending strains to the product $\varepsilon \delta \varepsilon^0$ is:

$$\varepsilon_b \delta \varepsilon_b^0 = y_2^2 w_{2,11} \delta w_{2,11} \quad (7.323)$$

The 2. derivative of the relative displacement is interpolated with expression (5.155):

$$w_{2,11} = \frac{\hat{a}}{a^2} ((-1+3z)\gamma_A + (1+3z)\gamma_B) \quad (7.324)$$

$$\gamma_A = \beta_A - \theta \quad (7.325)$$

$$\gamma_B = \beta_B - \theta \quad (7.326)$$

β_A, β_B rotation of nodes A and B of the member

θ rotation of the chord of the member

\hat{a} instant length of the member

z normalised coordinate

Substitute (7.325) and (7.326) into the interpolation (7.324):

$$w_{2,11} = \frac{\hat{a}}{a^2} ((-1+3z)\beta_A + (1+3z)\beta_B - 6\theta z) \quad (7.327)$$

The chord rotation θ is given by expression (5.60):

$$\sin \theta = \frac{w_{2B} - w_{2A}}{a} \quad (7.328)$$

Since the total stiffness matrix is computed for given values of w_{2A} and w_{2B} , the chord rotation can be determined with (7.328) and expressed in the following form:

$$\theta = \frac{r}{a}(w_{2B} - w_{2A}) \quad (7.329)$$

r nonlinearity factor, 1.0 for small rotations

Substitute expression (7.329) into the interpolation (7.327):

$$w_{2,11} = \mathbf{k}^T \mathbf{w}_e \quad (7.330)$$

$$\mathbf{k}^T = \frac{\hat{a}}{a^2} \begin{bmatrix} 0 & \frac{6r}{a}z & -1+3z & 0 & -\frac{6r}{a}z & 1+3z \end{bmatrix}$$

$$\mathbf{w}_e^T = \begin{bmatrix} w_{1A} & w_{2A} & \beta_A & w_{1B} & w_{2B} & \beta_B \end{bmatrix}$$

The variation of the second derivative of the relative displacement w_2 with respect to the reference coordinate y_1 follows from (7.330):

$$\delta w_{2,11} = \mathbf{k}^T \delta \mathbf{w}_e \quad (7.331)$$

The second derivative of the relative displacement varies over the length of the member. The contribution of the bending strain to the integral (7.315) is determined analytically:

$$\int_{c_e} \delta \varepsilon_b s_b dv = \delta \mathbf{w}_e^T \mathbf{K}_b \mathbf{w}_e \quad (7.332)$$

$$\mathbf{K}_b = \frac{EJ}{a^3} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & c_4 & c_5 & 0 & -c_4 & c_5 \\ 0 & c_5 & c_6 & 0 & -c_5 & c_7 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -c_4 & -c_5 & 0 & c_4 & -c_5 \\ 0 & c_5 & c_7 & 0 & -c_5 & c_6 \end{bmatrix}$$

$$c_4 = 12 \left(\frac{\hat{a}r}{a} \right)^2 \quad c_5 = 6\hat{a} \left(\frac{\hat{a}r}{a} \right)$$

$$c_6 = 4\hat{a}^2 \quad c_7 = 2\hat{a}^2$$

The member displacement vector \mathbf{w}_e is expressed in terms of the system displacement vector \mathbf{d}_s with equation (5.76) and substituted into (7.314). The contributions of the members are added to obtain the total system stiffness matrix \mathbf{K}_s :

$$\delta \mathbf{d}_s^T \mathbf{K}_s \mathbf{d}_s = \sum_e \delta \mathbf{w}_e^T \mathbf{K}_e \mathbf{w}_e$$

$$\mathbf{K}_s = \sum_e \mathbf{T}_e^T \mathbf{R}_{esz} \mathbf{K}_e \mathbf{R}_{esz}^T \mathbf{T}_e \quad (7.333)$$

$$\mathbf{K}_e = \mathbf{K}_a + \mathbf{K}_b$$

\mathbf{T}_e topology matrix of bar e

\mathbf{R}_{esz} element rotation matrix support-instant space

7.8 SOLUTION OF THE GENERAL EIGENVALUE PROBLEM

Concept: Consider the general eigenvalue problem (7.46):

$$\mathbf{A} \mathbf{x}_i = \lambda_i \mathbf{B} \mathbf{x}_i \quad i = 1, \dots, n \quad (7.334)$$

A real regular symmetric coefficient matrix of dimension n

B real regular symmetric coefficient matrix of dimension n

λ_i eigenvalue of the i-th eigenstate

\mathbf{x}_i eigenvector of the i-th eigenstate, $\mathbf{x}_i^T \mathbf{B} \mathbf{x}_k = \delta_{ik}$

Assume that the eigenvalue λ_1 with the smallest absolute value is to be determined by vector iteration. The start vector $\mathbf{u}_0 \neq \mathbf{0}$ of the iteration is chosen freely. In step s of the iteration, the vector \mathbf{u}_{s-1} is used in equation (7.335) to compute an auxiliary vector \mathbf{y}_s which is then normalised with respect to **B** in (7.336) to yield the iterated vector \mathbf{u}_s . The iteration converges to the eigenstate $(\lambda_1, \mathbf{x}_1)$.

$$\text{Compute } \mathbf{y}_s : \quad \mathbf{A} \mathbf{y}_s = \mathbf{B} \mathbf{u}_{s-1} \quad (7.335)$$

$$\text{Scale with } m_s : \quad \mathbf{u}_s = m_s \mathbf{y}_s \quad \text{so that} \quad \mathbf{u}_s^T \mathbf{B} \mathbf{u}_s = \delta_s \in \{-1, 1\} \quad (7.336)$$

Convergence to the eigenvector: The start vector \mathbf{u}_0 of the iteration is a linear combination $\sum_k c_k \mathbf{x}_k$ of the eigenvectors. It is modified as follows by the iteration, since none of the eigenvalues equals null:

$$\begin{aligned}
\text{Step 1: } \mathbf{A}\mathbf{y}_1 &= \mathbf{B}\mathbf{u}_0 &= \sum_{k=1}^n \frac{\mathbf{C}_k}{\lambda_k} \mathbf{A}\mathbf{x}_k &\Rightarrow \mathbf{y}_1 = \sum_{k=1}^n \frac{\mathbf{C}_k}{\lambda_k} \mathbf{x}_k \\
m_1^{-2} &= \frac{1}{\delta_1} \mathbf{y}_1^T \mathbf{B}\mathbf{y}_1 &= \sum_{k=1}^n \left(\frac{\mathbf{C}_k}{\lambda_k} \right)^2 \frac{\delta_k}{\delta_1} &\Rightarrow \mathbf{u}_1 = m_1 \sum_{k=1}^n \frac{\mathbf{C}_k}{\lambda_k} \mathbf{x}_k \\
\text{Step 2: } \mathbf{A}\mathbf{y}_2 &= \mathbf{B}\mathbf{u}_1 &= m_1 \sum_{k=1}^n \frac{\mathbf{C}_k}{\lambda_k^2} \mathbf{A}\mathbf{x}_k &\Rightarrow \mathbf{y}_2 = m_1 \sum_{k=1}^n \frac{\mathbf{C}_k}{\lambda_k^2} \mathbf{x}_k \\
m_2^{-2} &= \frac{1}{\delta_2} \mathbf{y}_2^T \mathbf{B}\mathbf{y}_2 &= m_1^2 \sum_{k=1}^n \left(\frac{\mathbf{C}_k}{\lambda_k^2} \right)^2 \frac{\delta_k}{\delta_2} &\Rightarrow \mathbf{u}_2 = m_1 m_2 \sum_{k=1}^n \frac{\mathbf{C}_k}{\lambda_k^2} \mathbf{x}_k \\
\text{Step } s: \mathbf{A}\mathbf{y}_s &= \mathbf{B}\mathbf{u}_{s-1} &= m_1 \dots m_{s-1} \sum_{k=1}^n \frac{\mathbf{C}_k}{\lambda_k^s} \mathbf{A}\mathbf{x}_k &\Rightarrow \mathbf{y}_s = m_1 \dots m_{s-1} \sum_{k=1}^n \frac{\mathbf{C}_k}{\lambda_k^s} \mathbf{x}_k \\
m_s^{-2} &= \frac{1}{\delta_s} \mathbf{y}_s^T \mathbf{B}\mathbf{y}_s &= (m_1 \dots m_{s-1})^2 \sum_{k=1}^n \left(\frac{\mathbf{C}_k}{\lambda_k^s} \right)^2 \frac{\delta_k}{\delta_s} &\Rightarrow \mathbf{u}_s = m_1 \dots m_s \sum_{k=1}^n \frac{\mathbf{C}_k}{\lambda_k^s} \mathbf{x}_k
\end{aligned}$$

The product $m_1 \dots m_s$ is eliminated between the last two equations to yield:

$$\mathbf{u}_s = \frac{\sum_{k=1}^n \frac{\mathbf{C}_k}{\lambda_k^s} \mathbf{x}_k}{\sqrt{\sum_{k=1}^n \left(\frac{\mathbf{C}_k}{\lambda_k^s} \right)^2 \frac{\delta_k}{\delta_s}}} \quad (7.337)$$

Since λ_1 is the smallest eigenvalue, the product $(\lambda_1/\lambda_k)^s$ will tend to null as the power s increases, unless there are multiple eigenvalues with absolute value $|\lambda_1|$. The term with index $k=1$ is therefore taken out of the summations in (7.337):

$$\mathbf{u}_s = \frac{\mathbf{c}_1 \mathbf{x}_1 + \sum_{k=2}^n \left(\frac{\lambda_1}{\lambda_k} \right)^s \mathbf{c}_k \mathbf{x}_k}{\sqrt{\mathbf{c}_1^2 + \sum_{k=2}^n \left(\frac{\lambda_1}{\lambda_k} \right)^{2s} \mathbf{c}_k^2 \frac{\delta_k}{\delta_s}}} \quad (7.338)$$

As the power s increases, the root in (7.338) tends towards \mathbf{c}_1 . If \mathbf{c}_1 differs from null, the limit value of the iterated vector \mathbf{u}_s is the eigenvalue \mathbf{x}_1 :

$$\lim_{s \rightarrow \infty} (\mathbf{u}_s) = \mathbf{x}_1 \quad (7.339)$$

Rate of convergence to the eigenvector: The rate of convergence k_u of the eigenvector \mathbf{x}_1 is defined as the limit of the ratio of the \mathbf{B} -quadratic form of the error in the iterated vector for steps s and $s-1$:

$$k_u = \lim_{s \rightarrow \infty} \frac{(\mathbf{u}_s - \mathbf{x}_1)^T \mathbf{B}(\mathbf{u}_s - \mathbf{x}_1)}{(\mathbf{u}_{s-1} - \mathbf{x}_1)^T \mathbf{B}(\mathbf{u}_{s-1} - \mathbf{x}_1)} \quad (7.340)$$

Substitution of (7.337) into (7.340) leads to:

$$\mathbf{u}_s - \mathbf{x}_1 = \frac{c_1 \mathbf{x}_1 + c_2 \left(\frac{\lambda_1}{\lambda_2} \right)^s \frac{\delta_2}{\delta_s} \mathbf{x}_2 + \sum_{k=3}^n \left(\frac{\lambda_1}{\lambda_k} \right)^s c_k \mathbf{x}_k \frac{\delta_k}{\delta_s}}{\sqrt{c_1^2 + \sum_{k=2}^n \left(\frac{\lambda_1}{\lambda_k} \right)^{2s} c_k^2 \frac{\delta_k}{\delta_s}}} - \mathbf{x}_1$$

$$\lim_{s \rightarrow \infty} (\mathbf{u}_s - \mathbf{x}_1) = \frac{c_2}{c_1} \left(\frac{\lambda_1}{\lambda_2} \right)^s \frac{\delta_2}{\delta_s} \mathbf{x}_2 \quad (7.341)$$

Substitution of (7.341) into (7.340) shows that the rate of convergence of the iteration for the eigenvector equals the ratio of the two eigenvalues with the smallest absolute value:

$$k_u = \frac{\lambda_1}{\lambda_2} \quad (7.342)$$

Special eigenvalues: All eigenvalues of (7.334) differ from null. If one of the eigenvalues were null, the determinant of \mathbf{A} would be null. This contradicts the assumption that \mathbf{A} is regular. If the smallest eigenvalue λ_m is multiple, equation (7.338) is replaced by:

$$\mathbf{u}_s = \frac{c_1 \mathbf{x}_1 + c_2 \mathbf{x}_2 + \sum_{k=3}^n \left(\frac{\lambda_m}{\lambda_k} \right)^s \frac{\delta_k}{\delta_s} c_k \mathbf{x}_k}{\sqrt{c_1^2 + c_2^2 + \sum_{k=3}^n \left(\frac{\lambda_m}{\lambda_k} \right)^{2s} \frac{\delta_k}{\delta_s} c_k^2}} \quad (7.343)$$

The iterated vector now converges to:

$$\lim_{s \rightarrow \infty} (\mathbf{u}_s) = \frac{c_1 \mathbf{x}_1 + c_2 \mathbf{x}_2}{c_1^2 + c_2^2} \quad (7.344)$$

The real values c_1 and c_2 are arbitrary. Let the limit value in (7.344) be \mathbf{w} . The second eigenvector corresponding to λ_m is determined by choosing a start vector that is \mathbf{B} -orthogonal to \mathbf{w} . This is analogous to the determination of the neighbouring eigenstates.

If there are two eigenvalues of equal absolute value but opposite sign, they are orthogonal and equation (7.338) is replaced by:

$$\mathbf{u}_s = \frac{c_1 \mathbf{x}_1 + (-1)^s c_2 \mathbf{x}_2 + \sum_{k=3}^n \left(\frac{\lambda_1}{\lambda_k} \right)^s \frac{\delta_k}{\delta_s} c_k \mathbf{x}_k}{\sqrt{c_1^2 + c_2^2 + \sum_{k=3}^n \left(\frac{\lambda_1}{\lambda_k} \right)^{2s} \frac{\delta_k}{\delta_s} c_k^2}} \quad (7.345)$$

The iterated vector now converges to:

$$\lim_{s \rightarrow \infty} (\mathbf{u}_s) = \frac{c_1 \mathbf{x}_1 + (-1)^s c_2 \mathbf{x}_2}{c_1^2 + c_2^2} \quad (7.346)$$

The iterated vector oscillates between the two values on the right-hand side of (7.346). Successive limit values yield the two eigenvectors. Their Rayleigh quotients yield the corresponding eigenvalues.

Convergence of the Rayleigh quotient to the eigenvalue: The eigenvalue for an iterated vector \mathbf{u}_s is estimated by means of the Rayleigh quotient $r(\mathbf{u}_s)$:

$$r(\mathbf{u}_s) = \frac{\mathbf{u}_s^T \mathbf{A} \mathbf{u}_s}{\mathbf{u}_s^T \mathbf{B} \mathbf{u}_s} \quad (7.347)$$

Expression (7.337) for the iterated vector \mathbf{u}_s is substituted into (7.347) in order to obtain an expression for $r(\mathbf{u}_s)$ in terms of the eigenvectors:

$$\begin{aligned} r(\mathbf{u}_s) &= \frac{\left(\sum_{k=1}^n \frac{C_k}{\lambda_k^s} \mathbf{x}_k^T \right) \mathbf{A} \left(\sum_{m=1}^n \frac{C_m}{\lambda_m^s} \mathbf{x}_m \right)}{\left(\sum_{k=1}^n \frac{C_k}{\lambda_k^s} \mathbf{x}_k^T \right) \mathbf{B} \left(\sum_{m=1}^n \frac{C_m}{\lambda_m^s} \mathbf{x}_m \right)} = \frac{\left(\sum_{k=1}^n \frac{C_k}{\lambda_k^s} \mathbf{x}_k^T \right) \mathbf{B} \left(\sum_{m=1}^n \frac{C_m}{\lambda_m^{s-1}} \mathbf{x}_m \right)}{\left(\sum_{k=1}^n \frac{C_k}{\lambda_k^s} \mathbf{x}_k^T \right) \mathbf{B} \left(\sum_{m=1}^n \frac{C_m}{\lambda_m^s} \mathbf{x}_m \right)} = \frac{\sum_{k=1}^n \frac{C_k^2}{\lambda_k^{2s-1}} \delta_k}{\sum_{k=1}^n \frac{C_k^2}{\lambda_k^{2s}} \delta_k} \\ r(\mathbf{u}_s) &= \frac{\lambda_1^{2s} c_1^2 \delta_1 + \sum_{k=2}^n \left(\frac{\lambda_1}{\lambda_k} \right)^{2s-1} c_k^2 \delta_k}{\lambda_1^{2s-1} c_1^2 \delta_1 + \sum_{k=2}^n \left(\frac{\lambda_1}{\lambda_k} \right)^{2s} c_k^2 \delta_k} \end{aligned} \quad (7.348)$$

As the power s increases, the sums go to null since the ratio $|\lambda_1/\lambda_k|$ is less than 1.0. The limit value of $r(\mathbf{u}_s)$ is therefore the eigenvalue λ_1 :

$$\lim_{s \rightarrow \infty} r(\mathbf{u}_s) = \lambda_1 \quad (7.349)$$

Rate of convergence of the Rayleigh quotient to the eigenvalue: The rate of convergence k_r to the Rayleigh quotient is defined as the limit of the ratio of the difference between the Rayleigh quotient and the eigenvector in steps s and $s-1$ of the iteration:

$$k_r = \lim_{s \rightarrow \infty} \frac{r(\mathbf{u}_s) - \lambda_1}{r(\mathbf{u}_{s-1}) - \lambda_1} \quad (7.350)$$

Expression (7.337) for the iterated vector \mathbf{u}_s is substituted into (7.350) in order to obtain an expression k_r in terms of the eigenvectors:

$$\begin{aligned}
r(\mathbf{u}_s) - \lambda_1 &= \frac{\sum_{k=1}^n \frac{c_k^2}{\lambda_k^{2s-1}} \delta_k - \lambda_1 \sum_{k=1}^n \frac{c_k^2}{\lambda_k^{2s}} \delta_k}{\sum_{k=1}^n \frac{c_k^2}{\lambda_k^{2s}} \delta_k} = \frac{\sum_{k=2}^n \frac{c_k^2}{\lambda_k^{2s-1}} (1 - \frac{\lambda_1}{\lambda_k}) \delta_k}{\sum_{k=1}^n \frac{c_k^2}{\lambda_k^{2s}} \delta_k} \\
&= \frac{\lambda_1^{2s} \left(c_2^2 (1 - \frac{\lambda_1}{\lambda_k}) \delta_2 + \sum_{k=3}^n c_k^2 (1 - \frac{\lambda_1}{\lambda_k}) \delta_k \left(\frac{\lambda_2}{\lambda_k} \right)^{2s-1} \right)}{\lambda_2^{2s-1} \left(c_1^2 \delta_1 + \sum_{k=2}^n c_k^2 \left(\frac{\lambda_1}{\lambda_k} \right)^{2s} \delta_k \right)} \\
\lim_{s \rightarrow \infty} (r(\mathbf{u}_s) - \lambda_1) &= \frac{\lambda_1^{2s} \left(c_2^2 (1 - \frac{\lambda_1}{\lambda_k}) \delta_2 \right)}{\lambda_2^{2s-1} c_1^2 \delta_1} \tag{7.351}
\end{aligned}$$

An analogous formula is derived for $r(\mathbf{u}_{s-1}) - \lambda_1$:

$$\lim_{s \rightarrow \infty} (r(\mathbf{u}_{s-1}) - \lambda_1) = \frac{\lambda_1^{2s-2} \left(c_2^2 (1 - \frac{\lambda_1}{\lambda_k}) \delta_2 \right)}{\lambda_2^{2s-3} c_1^2 \delta_1} \tag{7.352}$$

Substitution of (7.351) and (7.352) into (7.350) shows that the rate of convergence to the eigenvalue equals the square of the ratio of the eigenvalues with the smallest absolute value:

$$k_r = \lim_{s \rightarrow \infty} \frac{r(\mathbf{u}_s) - \lambda_1}{r(\mathbf{u}_{s-1}) - \lambda_1} = \frac{\lim_{s \rightarrow \infty} r(\mathbf{u}_s) - \lambda_1}{\lim_{s \rightarrow \infty} r(\mathbf{u}_{s-1}) - \lambda_1} = \left(\frac{\lambda_1}{\lambda_2} \right)^2 \tag{7.353}$$

Neighbouring eigenstates: Assume that the eigenstate $(\lambda_1, \mathbf{x}_1)$ of (7.334) with the smallest absolute value of the eigenvalue is known. The eigenstate $(\lambda_2, \mathbf{x}_2)$ with the second smallest absolute value of the eigenvalue is to be determined by vector iteration. This is achieved by making the start vector \mathbf{u}_0 **B**-orthogonal to \mathbf{x}_1 . Let $\hat{\mathbf{u}}_0$ be an arbitrary vector that is not **B**-orthogonal to \mathbf{x}_1 and let the component of $\hat{\mathbf{u}}_0$ that is **B**-orthogonal to \mathbf{x}_1 be \mathbf{u}_0 :

$$\begin{aligned}
\hat{\mathbf{u}}_0 &= c_1 \mathbf{x}_1 + \dots + c_n \mathbf{x}_n \\
\mathbf{u}_0 &= \hat{\mathbf{u}}_0 - c_1 \mathbf{x}_1 = c_2 \mathbf{x}_2 + \dots + c_n \mathbf{x}_n \\
\mathbf{u}_0^T \mathbf{B} \mathbf{x}_1 &= 0 \quad \wedge \quad \mathbf{x}_1^T \mathbf{B} \mathbf{x}_1 = \delta_1 \quad \Rightarrow \quad c_1 = \delta_1 \hat{\mathbf{u}}_0^T \mathbf{B} \mathbf{x}_1 \quad \delta_1 \in \{-1, 1\} \tag{7.354}
\end{aligned}$$

The first step of the iteration now yields:

$$\mathbf{A} \mathbf{y}_1 = \mathbf{B} \mathbf{u}_0 = \sum_{k=2}^n c_k \mathbf{B} \mathbf{x}_k \quad \Rightarrow \quad \mathbf{y}_1 = \sum_{k=2}^n \frac{c_k}{\lambda_k} \mathbf{x}_k$$

Since \mathbf{u}_0 does not contain the eigenvector \mathbf{x}_1 , the iteration will converge to the eigenvector \mathbf{x}_2 corresponding to the remaining eigenvalue with the smallest absolute value. Due to the round-off in the numerical solution, the iterated vector \mathbf{u}_s must periodically be made **B**-orthogonal to the eigenvector \mathbf{x}_1 . In order to be on the safe

side, this can be done after each cycle of iteration. Additional neighbouring eigenstates can be determined in analogous fashion:

$$\begin{aligned}\hat{\mathbf{u}}_0 &= c_1 \mathbf{x}_1 + \dots + c_n \mathbf{x}_n \\ \mathbf{u}_0 &= \hat{\mathbf{u}}_0 - \sum_{i=1}^m c_i \mathbf{x}_i = c_{m+1} \mathbf{x}_{m+1} + \dots + c_n \mathbf{x}_n \\ i \in \{1, \dots, m\}: \quad \mathbf{u}_0^T \mathbf{B} \mathbf{x}_i &= 0 \quad \wedge \quad \mathbf{x}_i^T \mathbf{B} \mathbf{x}_j = \delta_{ij} \quad \Rightarrow \quad c_i = \delta_i \hat{\mathbf{u}}_0^T \mathbf{B} \mathbf{x}_i\end{aligned}\quad (7.355)$$

Algorithm for the vector iteration: The iteration rules (7.335) and (7.336) are not implemented directly in the algorithm. The repetition of computations is avoided by introducing the following auxiliary vectors:

$$\mathbf{u}_s = m_s \mathbf{y}_s \quad (7.356)$$

$$\mathbf{z}_s = \mathbf{B} \mathbf{y}_s \quad (7.357)$$

$$\mathbf{w}_s = \mathbf{B} \mathbf{u}_s \quad (7.358)$$

The algorithm for the computation of the eigenvalue with the smallest absolute value of the general eigenvalue problem (7.334) consists of the following steps:

1. Choose a start vector \mathbf{w}_0 and set $s = 1$.
2. Decompose \mathbf{A} : $\mathbf{A} = \mathbf{L} \mathbf{D} \mathbf{L}^T$ (7.359)
3. Compute \mathbf{y}_s : $\mathbf{A} \mathbf{y}_s = \mathbf{B} \mathbf{u}_{s-1} = \mathbf{w}_{s-1}$
 $\mathbf{y}_s = \mathbf{L}^{-T} \mathbf{D}^{-1} \mathbf{L}^{-1} \mathbf{w}_{s-1}$ (7.360)
4. Compute \mathbf{z}_s : $\mathbf{z}_s = \mathbf{B} \mathbf{y}_s$ (7.361)
5. Compute a_s : $a_s = \mathbf{y}_s^T \mathbf{B} \mathbf{y}_s = \mathbf{y}_s^T \mathbf{z}_s$ (7.362)
6. Compute r_s : $r(\mathbf{u}_s) = \frac{\mathbf{u}_s^T \mathbf{A} \mathbf{u}_s}{\mathbf{u}_s^T \mathbf{B} \mathbf{u}_s} = \frac{\mathbf{y}_s^T \mathbf{A} \mathbf{y}_s}{\mathbf{y}_s^T \mathbf{B} \mathbf{y}_s} = \frac{\mathbf{y}_s^T \mathbf{w}_{s-1}}{a_s}$ (7.363)
7. Compute \mathbf{w}_s : $\mathbf{w}_s = m_s \mathbf{z}_s$ with $m_s = \frac{1}{\sqrt{|a_s|}}$ (7.364)
8. Repeat from step 3 if $|r_s - r_{s-1}| > \varepsilon$, otherwise set $\mathbf{u}_s = m_s \mathbf{y}_s$ and terminate.

For the determination of the second eigenstate, an initial value $\hat{\mathbf{w}}_0 = \mathbf{B} \hat{\mathbf{u}}_0$ is chosen for cycle 1. The \mathbf{B} -orthogonalisation of $\hat{\mathbf{u}}_0$ is replaced by corresponding operations on $\hat{\mathbf{w}}_0$:

$$\begin{aligned}\hat{\mathbf{w}}_0 &= \mathbf{B} \hat{\mathbf{u}}_0 = \mathbf{B}(c_1 \mathbf{x}_1 + \dots + c_n \mathbf{x}_n) \\ \mathbf{w}_0 &= \mathbf{B} \mathbf{u}_0 = \mathbf{B}(\hat{\mathbf{u}}_0 - c_1 \mathbf{x}_1) = \hat{\mathbf{w}}_0 - c_1 \mathbf{p}_1 \\ \mathbf{p}_1 &:= \mathbf{B} \mathbf{x}_1 \\ \mathbf{u}_0^T \mathbf{B} \mathbf{x}_1 &= \mathbf{w}_0^T \mathbf{x}_1 = 0 \quad \wedge \quad \mathbf{x}_1^T \mathbf{B} \mathbf{x}_1 = \mathbf{p}_1^T \mathbf{x}_1 = \delta_1 \quad \Rightarrow \quad c_1 = \delta_1 \hat{\mathbf{w}}_0^T \mathbf{x}_1 \\ \mathbf{w}_0 &= \hat{\mathbf{w}}_0 - c_1 \mathbf{p}_1\end{aligned}\quad (7.365)$$

Due to round-off, the value of \mathbf{w}_{s-1} computed in cycle $s-1$ is no longer \mathbf{B} -orthogonal to \mathbf{x}_1 and is therefore denoted by $\hat{\mathbf{w}}_{s-1}$. The \mathbf{B} -orthogonalisation of \mathbf{u}_{s-1} is replaced by corresponding operations on $\hat{\mathbf{w}}_{s-1}$:

$$\begin{aligned}
\hat{\mathbf{w}}_{s-1} &= \mathbf{B}\hat{\mathbf{u}}_{s-1} = \mathbf{B}(a_1\mathbf{x}_1 + \dots + a_n\mathbf{x}_n) \\
\mathbf{w}_{s-1} &= \mathbf{B}\mathbf{u}_{s-1} = \mathbf{B}(\hat{\mathbf{u}}_{s-1} - a_1\mathbf{x}_1) = \hat{\mathbf{w}}_{s-1} - a_1\mathbf{p}_1 \\
\mathbf{u}_{s-1}^T \mathbf{B}\mathbf{x}_1 &= \mathbf{w}_{s-1}^T \mathbf{x}_1 = 0 \quad \wedge \quad \mathbf{x}_1^T \mathbf{B}\mathbf{x}_1 = \mathbf{p}_1^T \mathbf{x}_1 = \delta_1 \quad \Rightarrow \quad a_1 = \delta_1 \hat{\mathbf{w}}_{s-1}^T \mathbf{x}_1 \\
\mathbf{w}_{s-1} &= \hat{\mathbf{w}}_{s-1} - a_1\mathbf{p}_1
\end{aligned} \tag{7.366}$$

Step 7 is extended to include the **B**-orthogonalisation of \mathbf{w}_{s-1} according to (7.366). If more than two eigenstates are determined, the **B**-orthogonalisation is extended to all eigenvectors that have been determined.

8 SOFTWARE FOR NONLINEAR STRUCTURAL ANALYSIS

8.1 SCOPE OF THE IMPLEMENTATION

Software implementation of geometrically nonlinear frame and truss analysis can have different purposes, for example to develop a commercially successful product or to create a tool for research and development. The criteria for the design and evaluation of the software will usually differ significantly for the two cases.

The initial implementation of a commercial software product requires a comparatively small portion of the total effort spent on the product during its lifetime. Other activities, such as documentation, advertising, distribution, sales, training, support service, extension and updating as well as reengineering, have a great influence on the commercial success of the product. The software will therefore be designed with the focus on properties such as broad functionality, high reliability, great simplicity, comfort of handling, efficiency in execution and adaptability to changes in the state of the art. A greater effort in the initial implementation of the software is acceptable if it leads to a product with good properties for the other phases of the life cycle. The user is in most cases expected to apply the software as a black box with specified external properties.

The effort devoted to the implementation of software that is to serve as a research tool is frequently a significant portion of the total effort of the research project. The software is constructed so that it permits scientific evaluation of the different components and aspects of the methods, models and processes being studied, as well as their interaction. The software is expected to be suitable as a test bed. In most cases, the user is familiar with the data base and the algorithms of the software. The development effort is reduced as far as possible, for instance by placing less emphasis on factors such as simplicity, comfort and adaptability than would be done for commercial software. The life span of research tools tends to be short compared to the decades that are typical for successful commercial products.

This chapter describes a software implementation of the geometrically nonlinear analysis of trusses and frames that is to serve as a research tool. This tool is used for several purposes, as set out below.

Test bed for the truss and frame theories: The theory derived in chapters 4 and 5 contains hypotheses for truss and frame behaviour. The frame theory is based on the assumption of small strains. The finite element formulation contains trial functions that do not satisfy the governing equations exactly. The research tool provides a test bed for testing the consequences of these approximations. Chapter 9 contains examples in which exact solutions for simple structures are compared to results obtained with the theories in chapters 4 and 5.

Test bed for the numerical method of solution: The nonlinear behaviour of trusses and frames is treated in chapters 6 and 7 as an initial value problem. The nonlinear terms in the governing equations are approximated by iteration in each load step. The numerical algorithm contains parameters such as the arc length increment for a step or the limit value for the norm of the unbalanced forces at the end of the load step. The determination of singular points and the continuation of the

load path depend on additional parameters. The research tool provides a test bed for studying the influence of different choices of these parameters.

Test bed for nonlinear behaviour of typical structures: The nonlinear behaviour of typical structures depends on the geometry of the nodes, as well as the topology and properties of the members. The research tool is used for enumerative studies on the influence of diverse parameters on the behaviour of types of structures such as portal frames and parallel chord trusses. This leads to an insight into favourable design strategies and suitable profile ranges for selected spans. This know-how is the background for good engineering design.

Complexity: The smaller volume of a research tool implementation, as compared to the volume of a commercial product, does not imply that the implementation of the research tool is less refined. On the contrary, the need to relate closely to the formulated theory and methods, as well as the requirement that the influence of diverse parameters on the results should be identifiable, can only be satisfied with well-structured software suited to the complexity of nonlinear behaviour.

Components of the software: The implementation of the research tool for nonlinear analysis contains all the main components typical for application software in civil engineering: a general data base that describes the structure and its behaviour, a structured implementation of the algorithms of the finite element method presented in chapters 4 to 7 for the nonlinear analysis of trusses and frames, and an interactive graphical user interface for specification of the structure, control of the workflow and presentation of the results of the analysis.

8.2 OBJECT STRUCTURE OF INFORMATION

8.2.1 Objects and Classes of Objects

Consider a structure mapped to a computer and the behaviour of this structure that is analysed with the computer. The ordered pair consisting of the mapping and the behaviour of the structure is called a computer application.

The data and algorithms that are part of an application are called the substance of the application. The object-oriented approach is used to structure the application by decomposing its substance into objects. Each object consists of an identifier and a substance. The intersection of the substance of any two objects is empty. Every identifier is unique in the application. Typical objects are nodes of the structure, bars of trusses, members of frames, as well as supports and loads of the structure.

The substance of an object consists of attributes and methods. The attributes specify the properties of the object at a point in time. The methods contain rules that permit the computation of attributes of this and other objects at a future point in time, given the attributes of this and other objects at previous points in time.

If an equivalence relation can be defined for the attributes and methods of a set of objects, this set is called a class of objects. The methods of the class and the attributes that describe properties of the class rather than properties of individual objects are collected in a class object. A distinction is made between methods that are called for a specific object and implicitly operate on the attributes of that object, and methods that are called for the class and operate on the attributes of the class.

If the programming environment supports reflection, the class object can be used to query the names and data types of the attributes and of the methods of the objects of the class, as well as the size and data types of the parameter lists of the signatures of its methods.

The decomposition of the objects of an application into classes is called the class structure of the application. The objects of the various classes that are constructed in the application, and their mutual references, are called the object structure of the application.

In the most general case, the class structure, the object structure and the substance of the objects of an application change with time. It is not necessary to consider this most general case for the nonlinear analysis of trusses and frames. The class structure and the methods in the classes for nonlinear truss and frame analysis do not change between applications. They are therefore defined during the implementation of the software and remain unchanged over the applications.

The number of objects in the classes of the implementation and the values of their attributes are, however, dependent on the structure under consideration. They vary from application to application, and change during the execution of an application. The implementation must therefore provide aggregations for the management of the current object sets. These aggregations are part of the model structure of the application. The methods with which the aggregations are administered are called the model management of the application.

8.2.2 Identification of Objects

Before an operation can be performed on an object, the object must be identified. Significant differences exist between the identification of objects inside the computer and the identification of objects by the external user. The aim of internal identification is to find the storage locations that contain the attributes and the methods of the object. External identification can be alphanumeric or geometric.

Let an application have a set C of objects, and let the identifiers of these objects be a set S of symbols. The injective mapping of the symbol set S to the object set C is called identification of the objects of the application:

$$\text{id} : S \rightarrow C \quad \text{with} \quad \text{id}(s) = c \tag{8.1}$$

- id identification mapping
- S set of identification symbols
- C set of objects of the application
- $s \in S$ identifier of object $c \in C$

Two types of identifiers are used for the internal identification of objects by the processor of the computer: transient and persistent identifiers. A transient identifier is an identifier whose validity is restricted to a computer session or to part of a session in which the application is used. In the programming language C, transient identifiers are called pointers and are managed by the programmer who allocates and deletes storage space. In the programming language Java, the transient identifiers are called references and cannot be managed by the programmer, although they can be com-

pared and set equal. The management of the working storage in Java is the privilege of the runtime system.

The persistent identifier of an object corresponds to the name of a person. It remains unchanged throughout the life-time of the object. It is convenient to use strings as persistent identifiers. If an algorithm of the application operates on the object during a computer session, the persistent identifier is mapped to the storage location that currently contains the attributes and the methods of the object.

Two methods can be implemented for the identification of objects in the user interface of the application. If the object has a persistent string identifier, the string can be typed on the keyboard. If the object is presented as a shape on the graphics display of the computer, the shape can be clicked with the mouse to identify the object.

8.2.3 Attributes

An attribute of an object is an ordered pair consisting of a data type and a data value. The attribute can have additional properties that determine, for instance, its accessibility (public or private) and its lifetime (transient or persistent).

The data type determines the syntax and semantics of the mapping of an attribute to the source code and the binary code of the program. Every programming language contains primitive data types. The characteristic of a primitive data type is that only the binary value of the attribute is stored. Primitive data types of Java are:

integer numbers : byte, short, int, long
 floating point numbers : float, double
 truth values : boolean
 characters : char

Objects can also have other objects as attributes. In order to provide this feature, the storage space of an object consists of two separated sections: the reference and the body of the object. All references have the same binary length, just as all values of a specific primitive data type have the same binary length. The bodies of objects of different classes usually differ in length. Figure 8.1 shows that if object B is an attribute of object A, a reference to B is stored in the body of A. The object B can also be an attribute of other objects besides A, for instance of C in figure 8.1. If D is an attribute of A, A can at the same time be an attribute of D, as shown in figure 8.1.

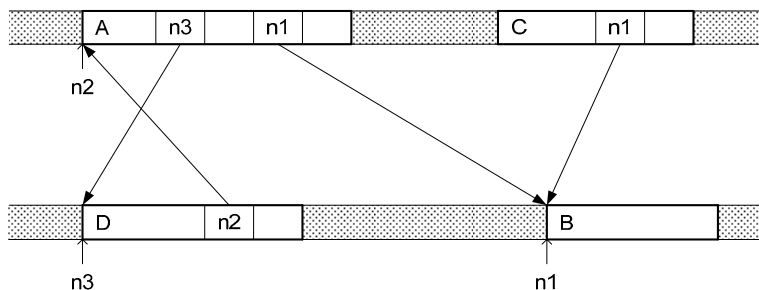


Figure 8.1 Attributes that are objects

8.2.4 Classes of Objects for Trusses and Frames

The following classes and attributes of objects are defined in the implementation of the geometrically nonlinear analysis of trusses and frames. Only attributes used to map the components of the structure to the computer are treated here. Objects and attributes for aggregations of objects, for structural behaviour and for the user interface will be considered in later sections of this chapter.

Node: An end point of a bar or frame member is called a node. The node identifier, sequence number, rotation of the local coordinate system and identifiers of the load and support objects for one loading condition are stored. If a variable is not prescribed, its value is null.

String	nodeName	persistent identifier of the node
int	nodeNumber	sequence number of the node in the system
double	xUser1,xUser2	node coordinates in the user coordinate system
double	theta	angle from axis x1 to axis s1 (support space)
String	loadNameS1	prescribed force in the direction s1, or null
String	loadNameS2	prescribed force in the direction s2, or null
String	loadNameRot	prescribed moment about axis s3, or null
String	supportNameS1	prescribed displacement in direction s1, or null
String	supportNameS2	prescribed displacement in direction s2, or null
String	supportNameRot	prescribed rotation about axis s3, or null

Bar: An element of a frame that can resist axial load but not bending moment is called a bar. The identifiers of the bar and its start and end node objects, the type code that distinguishes bars from frame members, the bar area and its modulus of elasticity are stored. Computed properties such as the bar length and the trigonometric ratios of reference space are considered to be part of the method of solution and are not treated here.

String	barName	persistent identifier of the bar
String	startNodeName	persistent identifier of the start node of the bar
String	endNodeName	persistent identifier of the end node of the bar
double	barArea	area of the cross-section of the bar in user units
double	modulusOfElasticity	modulus of elasticity of the bar in user units

Member: An element of a frame that can resist axial force and bending moment is called a member. The identifiers of the member and its start and end node objects, the type code that distinguishes frame members from bars, the area and moment of inertia of the cross-section, the modulus of elasticity of the member and the load intensities of the line load for one loading condition are stored. Computed properties such as the member length and the trigonometric ratios of reference and instant spaces are considered to be part of the method of solution and are not treated here.

String	memberName	persistent identifier of the member
String	startNodeName	persistent identifier of the start node of the member
String	endNodeName	persistent identifier of the end node of the member

double	memberArea	area of the cross-section in user units
double	memberInertia	moment of inertia of the cross-section in user units
double	modulusOfElasticity	modulus of elasticity of the bar in user units
Boolean	hasLineLoad	true: the member is loaded
double	p1A,p2A	load intensity at the start node in user units
double	p1B,p2B	load intensity at the end node in user units

Support: A prescribed displacement in the direction of one of the axes of a node coordinate system or rotation about the axis normal to the plane of the frame is called a support. The identifiers of the support and the node object at which it acts, as well as the magnitude of the prescribed displacement or rotation, are stored.

String	supportName	persistent identifier of the support
String	nodeName	persistent identifier of the supported node
double	displacement	value of the prescribed displacement

Load: A prescribed force in the direction of one of the axes of a node coordinate system or a prescribed moment about the axis normal to the plane of the frame is called a load. The identifiers of the load and the node object at which it acts, as well as the magnitude of the prescribed force or moment, are stored.

String	loadName	persistent identifier of the load
String	nodeName	persistent identifier of the loaded node
double	force	value of the prescribed force

8.2.5 Aggregation of Objects

Collection: The modelling of structures requires the aggregation of the objects whose attributes describe the structure and its behaviour. An aggregation of objects is called a collection. A collection is specified by enumeration of the identifiers of its objects in arbitrary sequence. It can contain more than one reference to the same object.

Set: If a collection contains exactly one reference to each of its objects, the collection is called a set. A set can consist of objects of the same class, or it may contain objects that are instances of different classes. The number of objects in the set can be constant or it may vary with time. A set does not have any structure. If it is traversed, the order in which its objects are encountered is stochastic.

Subset: A set of objects is structured by defining subsets of its elements. Each subset is an enumeration of the identifiers of the objects that are the elements of the subset. An object of the set can be an element of more than one subset of the set. If the union of the subsets equals the set A and if the pair-wise intersections of all subsets of A are empty, the subsets are called a decomposition of set A.

Set object: A set or subset is itself an object with a unique identifier. The substance of the set object is the enumeration of the identifiers of its objects. A set or subset may not contain itself as one of its elements.

Ordered pair: Two elements, whose sequence is essential, are called an ordered pair. The ordered pair is denoted by the ordered names of its elements, enclosed in round brackets. The elements can be members of different sets. Two ordered pairs are equal if their first elements are equal and their second elements are equal. An ordered pair is an object with a reference.

Cartesian product: Consider sets A and B. The set of all ordered pairs (a,b) that can be formed with elements $a \in A$ and $b \in B$, are called the Cartesian product of the sets A and B. The Cartesian product is denoted by $A \times B$.

Relation: A subset of the Cartesian product $A \times B$ is called a relation on A and B. The expression aRb has the value true if (a,b) is a member of the relation, and the value false if it is not. The relation is defined by enumerating the references of its ordered pairs. A relation R in a set S can have the following properties:

$$R := \{(a,b) \in S \times S \mid (aRb)\} \quad (8.2)$$

$$R \text{ is reflexive} \quad := \quad \forall a \in S: aRa$$

$$R \text{ is symmetric} \quad := \quad \forall a \in S, \forall b \in S: aRb \Rightarrow bRa$$

$$R \text{ is transitive} \quad := \quad \forall a \in S, \forall b \in S, \forall c \in S: aRb \wedge bRc \Rightarrow aRc$$

$$R \text{ is identitive} \quad := \quad \forall a \in S, \forall b \in S: aRb \wedge bRa \Rightarrow a = b$$

$$R \text{ is linear} \quad := \quad \forall a \in S, \forall b \in S: aRb \vee bRa$$

Equivalence relation: A relation E with the properties (8.3) is called an equivalence relation. Elements a and b of a set E are called equivalent if E is an equivalence relation. The objects of a class are equivalent, which is denoted by $a \sim b$.

$$E \text{ is reflexive} \quad a \sim a \quad (8.3)$$

$$E \text{ is symmetric} \quad a \sim b \Rightarrow b \sim a$$

$$E \text{ is transitive} \quad a \sim b \wedge b \sim c \Rightarrow a \sim c$$

Order relation: A relation with the properties (8.4) is called an order relation. The set is called a partially ordered set. Ordered sets are frequently denoted by symbols such as \leq :

$$\leq \text{ is reflexive} \quad a \in S \quad \Rightarrow \quad a \leq a \quad (8.4)$$

$$\leq \text{ is identitive} \quad a \leq b \wedge b \leq a \quad \Rightarrow \quad a = b$$

$$\leq \text{ is transitive} \quad a \leq b \wedge b \leq c \quad \Rightarrow \quad a \leq c$$

If the relation in (8.4) is linear, the relation is called a total order relation. The set is called a totally ordered set. Two arbitrarily selected members of a totally ordered set may be compared, whereas they may not be comparable in an ordered set.

$$\leq \text{ is linear} : \quad a, b \in S \quad \Rightarrow \quad a \leq b \vee b \leq a \quad (8.5)$$

Strict order relation: A relation with the properties (8.6) is called a strict order relation. The set is called a partially strictly ordered set and denoted by $<$:

$$< \text{ is antireflexive: } a \in S \quad \Rightarrow \quad \neg(a < a) \quad (8.6)$$

$$< \text{ is asymmetric: } a < b \quad \Rightarrow \quad \neg(b < a)$$

$$< \text{ is transitive} \quad : \quad a < b \wedge b < c \quad \Rightarrow \quad a < c$$

If the relation (8.6) is total, the relation is called a total strict order relation. The set is called a totally strictly ordered set or a list.

$$< \text{ is total : } a, b \in S \Rightarrow (a \neq b \Rightarrow a < b \vee b < a) \quad (8.7)$$

Mapping: A relation $f \subseteq A \times Z$ is called a mapping if it is left total and right unique:

$$f : A \rightarrow B \quad \text{with} \quad f(a) = b \quad (8.8)$$

A definition set of the mapping with $a \in A$

Z image set of the mapping with $b \in B$

A is left total $\forall (a \in A) \exists (b \in B): f(a) = b$

B is right unique $\forall (a \in A) \forall (b \in B) \forall (c \in C): f(a) = b \wedge f(a) = c \Rightarrow b = c$

The physical structure is mapped to its model in the computer. The computer model is the image of the physical structure.

Storage structures for set objects: The mapping of the identifiers of the objects of a set to locations in the computer memory depends on the use of the set. Different mappings, as shown in figure 8.2, are efficient for different purposes.

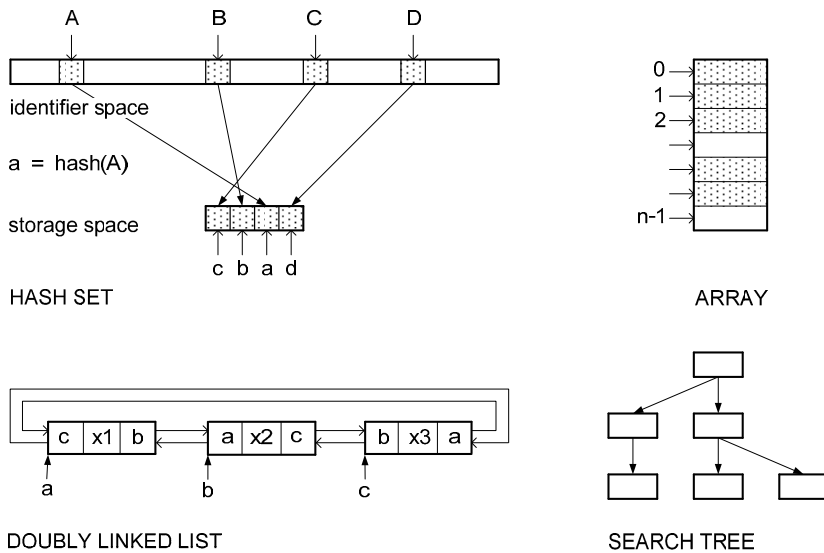


Figure 8.2 Storage structures for aggregations

– If the set is not ordered, the identifier space is hashed to a smaller address space in the memory. The members of the set can only be read by traversal of the set.

– If the set is ordered and speed of access is essential, the identifiers are stored in order in an array. The location of a member in the array is determined by its index. The members can be located either very fast by index calculation, or fast by means of bisection if their identifier but not their index is known. The entry of a new object in the array may require rearrangement of the array with a large number of operations.

- If the set is ordered and frequent rearrangement due to addition and removal of members is to be performed efficiently, a singly or doubly connected list is used. Each element of the list knows the storage address of its predecessor and of its successor. Access by identifier is slow since it requires a linear search in the list.
- If the set is totally ordered and changed frequently, the identifiers of the objects are stored in a search tree.

8.2.6 Aggregations in Java

Java interfaces for aggregations: The package `java.util` maps the mathematical structure of an aggregation type to a Java interface with a specified set of methods. An interface can be used as a data type. The following interfaces are defined:

Collection	enumeration of identifiers
Set	enumeration of unique identifiers
List	ordered sequence of identifiers
Map	mapping of keys to identifiers of objects
SortedSet	ordered sequence of identifiers of a totally ordered set
SortedMap	mapping from totally ordered set of keys to identifiers of objects

Java classes for aggregations: The identifiers of the Java classes for set objects contain two names. The first name specifies the storage structure of the set. The second name specifies the mathematical structure of the set.

HashSet	an unordered set of objects whose identifiers are mapped with a hash function to an address space for storage locations
TreeSet	a totally ordered set of objects whose identifiers are mapped to a search tree
LinkedHashSet	a HashSet of objects whose identifiers are traversed in the order in which they were entered in the set
ArrayList	a totally ordered set of objects whose identifiers are mapped to an array
LinkedList	a totally ordered set of objects whose identifiers are mapped to a doubly connected list
HashMap	an unordered set of ordered pairs whose identifiers are mapped with a hash function to an address space for storage locations
TreeMap	a totally ordered set of ordered pairs whose identifiers are mapped to a search tree
LinkedHashMap	a HashMap of ordered pairs whose identifiers are traversed in the order in which they were entered in the set

Traversal: A procedure in which every member of a stored set is visited exactly once is called a traversal of the set. If the set is not ordered, its members are visited in arbitrary sequence. If the set is ordered, its members are visited in ascending or in descending order. It is advantageous that the identifiers of the objects in the set need not be known for a traversal of the set.

If a set contains objects of the same class, the class is specified in the construction of the set. The set is conveniently traversed with a for-statement, as shown in the following example for a linked list of nodes. At the start of the first loop of the for-statement, the reference of the first node in the list is assigned to the variable "node". At the start of each succeeding loop, the reference of the next node in the list is assigned to variable "node". The variable "node" is then used in the block of the for-statement.

```
LinkedList <Node> nodeList = new LinkedList <Node> ();
for (Node node: nodeList) { ... }
```

If a set contains objects that are instances of different classes, the set is traversed with a special object called an iterator, which is obtained by calling method iterator() with the set object. The iterator is used to call the methods of interface Iterator, which test whether there are objects that have not yet been visited, and to read the reference of the next object. The reference must be cast to the appropriate data type. The iterator can also be used to remove objects from the set.

```
Bar      bar      ;
Member  member   ;
Object  object   ;
HashSet componentSet ;

while (componentSet.hasNext())
{ object = componentSet.next();
  if (object instanceof (Bar)    { bar = (Bar)object ; ... }
  if (object instanceof (Member) { member = (Member)object ; ... }
}
```

If the set is totally ordered, it can be traversed in the forward and in the backward direction with a list iterator, which is obtained by calling method listIterator() for the set object. The list iterator manages a pointer that points to the location between two objects of the set. At the start of the traversal, it points to the location preceding the first object of the set. The objects of the ordered set are indexed sequentially. The list iterator object is used to call the methods of Interface ListIterator:

boolean	hasNext()	true if the list contains an object after the pointer
boolean	hasPrevious()	true if the list contains an object before the pointer
Object	next()	returns the reference of the object after the pointer
Object	previous()	returns the reference of the object before the pointer
Int	nextIntIndex()	returns the index of the object after the pointer
Int	previousIndex()	returns the index of the object before the pointer
void	add(object)	adds an object to the set
void	set(object)	replaces an object in the set
void	remove()	removes an object from the set

These methods are used as in the preceding example.

8.2.7 Named Objects

Name: In the Java implementation of the geometrically nonlinear analysis of trusses and frames, each object is identified by a unique String that is defined when the object is constructed. The String is called the name of the object. The name cannot be changed during the lifespan of the object. Objects whose persistent identifier is a String are called named objects.

The use of named objects permits the specification of an object before it has been constructed in the computer. The concept is illustrated in the next example.

Location: An object can be identified by the location of the first byte of its body in a storage medium. It is assumed that the medium consists of a list of bytes that are indexed consecutively.

If an object is located in working storage, the byte index in the computer memory is used as transient identifier of the object. In Java, this identifier is managed by the run time system and called the reference of the object. The reference is transient since it may change during the session, for instance if the memory is reorganised.

If the object is stored in a file, the byte index in the byte stream of the file is used as identifier of the object. The byte index is managed by the user of the file, who must record the byte indices of his objects. While a file is closed or opened for reading only, the byte index is a persistent identifier.

Instead of recording the byte indices of his objects, the programmer may choose to write the objects into the file in a fixed order, and to read them in the same order. The byte length of the body of the object or the byte index of the next object must be stored with each object. This technique is used in Java serialisation. Since it permits only serial and not random access to the individual objects, the method is restrictive.

Mapping: If an object is used in the computer memory during a session, its name must be mapped to its reference in working storage. This is achieved by means of the hash map in figure 8.3, which is called an object table. The name of the object is the search key in the table. The current reference of the object in working storage is the value in the table.

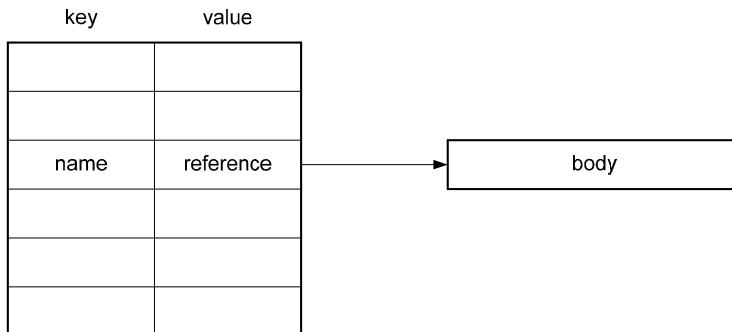


Figure 8.3 Object table for mapping of identifiers

When an object is constructed in working storage, the name and the reference of the object are entered in the object table. If an object A refers to an object B, the name of object B is stored as a String attribute of object A. In order to read attributes of B at run time, A first reads the reference of B in the object table and then uses this reference to access the body of B.

The use of named objects is essential for the construction of object-oriented application software and offers the following benefits:

- (a) An object A can refer to an object B before the object B exists in the computer. The string "B" can be stored in the body of A even if the reference and the body of B do not yet exist. This gives the user freedom of choice in the sequence in which he defines the objects.
- (b) The application can be stored in a file which permits random access to the objects of the application [7]. When an object is read from the file, its references to other objects of the application can be resolved by means of the object table.

Example: Figure 8.4 illustrates the construction of the object table for a bar named B whose end nodes are named S and T. The table is filled in when the objects are constructed during the execution of the following statements:

```
Bar bar = new Bar ("B", "S", "T") ;
Node nodeS = new Node ("S", xS, yS) ;
Node nodeT = new Node ("T", xT, yT) ;
```

When the bar object is constructed at run time, the run time system assigns storage location b to the body of the bar object. The reference to this location is stored in the variable called bar in the Java program. The constructor of class Bar stores the ordered pair (B,b) in the object table. Similarly, the bodies of the nodes are assigned to storage locations s and t, which are stored in the variables called nodeS and nodeT in the Java program. The constructor of class Node stores the ordered pairs (S,s) and (T,t) in the object table.

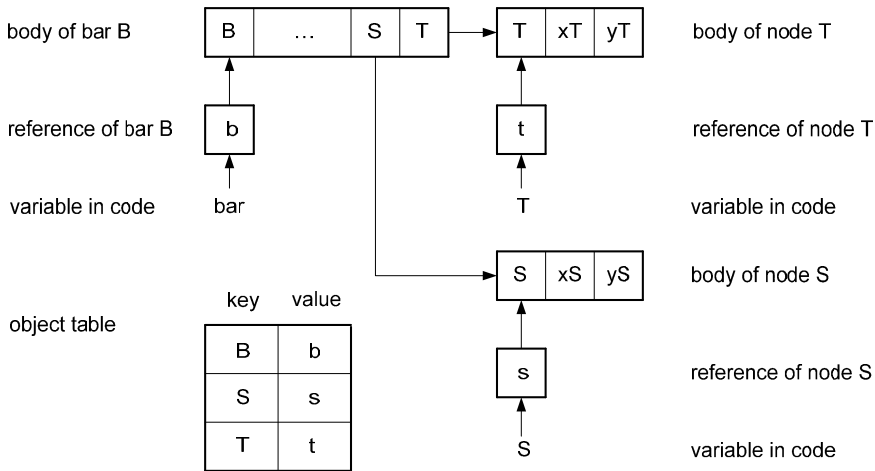


Figure 8.4 Identifiers of a bar B with nodes S and T

If a method in class Bar wishes to access the attribute x of node S, it calls the object management with the name "S" of the node as parameter. The object management looks up the identifier in the object table and returns the reference s of the body of node object S. This reference is then used to access the attribute x. This procedure is coded with the following statements:

```
Node node = (Node) base.getObject("S") ;
double x   = node.getX( )                ;
```

Method getObject() returns a value of data type Object, which is cast to an object of class Node, whose reference is stored in the variable node.

Class Base: The named objects of the implementation are managed by means of an object of class Base. The name of an object is either chosen by the user or provided by the base. The base object contains the object table of figure 8.4 and supports the following methods for object management:

```

        Base( )                                // constructor
String  autoName( )                            // returns a unique name
boolean putObject (Object object)
Object  getObject (String objectName)
boolean containsObject (String objectName)
boolean removeObject (String objectName)
void    clear( )
```

This set of functions is adequate for research software. A more elaborate object management [7] is required for commercial software. Class Base is replaced as a whole by the class that implements the extended object management.

8.2.8 Data Bases

The data structure, in which the persistent information of an application is stored and managed, is called the data base of the application. The following properties of data bases are desirable for applications in civil engineering [7, p. 28]:

(a) Utilisation of the functionality of the programming platform

The data base should directly use the attribute structure of its objects as specified in the definition of the class of the object. This implies that the data base should use reflection of the class object to determine the sequence and the data type of the attributes of an object written to storage or read from storage. The data base management should not contain duplicate functions for this purpose. The aggregation of the objects in the application should be retained automatically in the data base.

(b) Efficient and flexible management of the objects

The average time of access to an object in the base should be of the same order of magnitude as the access time in a Java method. The programmer should be able to determine the point in time at which an object is transferred between working and file storage. He should also be able to control which part of the object tree, to which the

object refers, is transferred. The mapping between transient and persistent identifiers of objects should be efficient.

(c) Control of data flow

Objects are stored persistently in files. The programmer should be able to control the allocation of each object to a specific file. He should also be able to control the flow of data so that only the required working set of objects is stored in working storage at each point in time. If the programmer does not place an object into working storage as required by an executing method, the object should be loaded automatically if it is available in an open file.

Many technologies have been developed for data bases, for instance:

- serialisation concept of Java
- relational data banks
- object-oriented data banks
- XML data banks (extensible mark-up language)

The suitability of these technologies for the implementation of persistent information structures in civil engineering has been studied in depth by I. Bilchuk [7]. Their suitability for information structures in civil engineering was evaluated and it was concluded that none of the technologies is fully suited. A new data base concept was developed and its performance was tested with two typical examples from civil engineering. The requirements of the persistent information structure of commercial software for nonlinear truss and frame analysis, as outlined above, can be met with this new data base technology. For the research tool, the Java serialisation concept is adequate as data base.

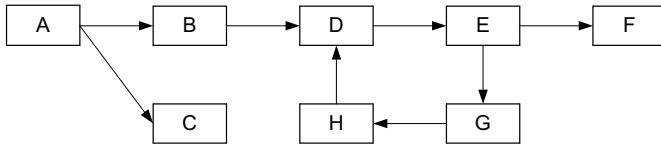
Serialisation: Java implements the Serialisation concept for the persistent storage of objects. It is assumed that the objects are written sequentially to the file and read sequentially in the same order from the file. Objects of classes that implement the interface `Serializable` are written to a file with methods of interface `ObjectOutput`, which is implemented in the class `ObjectOutputStream`:

```
FileOutputStream fos = new FileOutputStream("project") ;
ObjectOutputStream oos = new ObjectOutputStream(fos) ;
oos.writeObject( bar) ;
```

The objects are read as follows from the file with the methods of interface `ObjectInput`, which is implemented in class `ObjectInputStream`:

```
FileInputStream fis = new FileInputStream("project") ;
ObjectInputStream ois = new ObjectInputStream(fis) ;
Bar bar = (Bar)ois.readObject( ) ;
```

If an object A that is written to a file contains references to other objects such as B, C, ... which implement interface `Serializable`, these objects are automatically also written to the file. If the objects B,C, ... contain further references, the process is continued for these references. Object A is the root of an object graph, as shown in figure 8.5, which is written to the file as a whole. The object graph is also read from the file as a whole and the new references are automatically entered in the objects.



object graph in working storage



objects of the graph stored in a file

Figure 8.5 Storage of an object graph in a file

Figure 8.5 shows that an object graph can contain cycles such as D-E-G-H-D. These cycles are detected by the run time system of Java and interrupted, for instance at D in figure 8.5, when the graph is written to a file.

Implementation: The frame model contains a large number of objects. Only the single object of class Base that contains the object table of the application is written explicitly to the file. Since the object table contains the references of all objects of the application, the Base object is the root of an object graph that contains all objects that are to be written to the file. The entire object graph is automatically written to the file.

In order to read the objects of an application from a file, only the object of class Base is explicitly read from the file. Since it contains the object table with all other objects of the application, these are automatically read from the file and their references are entered in the object table. These references will differ from the references that were valid before the objects were written to the file.

Evaluation: The implemented data base has the following disadvantages:

- (a) The complete set of objects of the application must be written to file or read from file at the same time. The object flow between working storage and file therefore cannot be suitably spread over the execution of the tasks.
- (b) It is not possible to have random access to individual objects in the file.
- (c) The complete set of objects is written to file or read from file although it is frequently not necessary to write or read the complete set of objects since the file or the working storage already contains a subset of objects.
- (d) If the Base object is written to the same file twice, only those objects are written to the file in the second operation that are not contained in the object table in the first operation. If attributes of an object are changed between the two operations, the attributes of these objects in the file are not updated in the second operation.

The data base has the advantages of simplicity and automatic execution by the Java run time system.

8.3 STRUCTURE OF FRAME MODELS

8.3.1 Contents of Frame Models

Model: A frame model is the mapping of a physical frame and its behaviour to the storage spaces, processors and user interfaces of a computer. The model consists of data and procedures.

The data of the model describe the frame itself, the external influences acting on the frame and the behaviour of the frame when it is subjected to these influences. They are stored persistently in the data base of the model. During a session, the data required by the procedures are kept in working storage.

The procedures of the model use the data that describe the properties of the frame to compute the data that describe the behaviour of the frame under the external influences. These procedures implement the algorithms for the analysis of the deformation and the stability of frames, which are constructed on the basis of the nonlinear frame theory presented in chapters 4 to 7.

Interface: The interaction of the user with the model is supported by a graphical user interface consisting of a display, a mouse and a keyboard. The data and procedures for this interface are part of the frame model.

The data of the interface specify the attributes of the mappings of the internal frame model to various presentation models on the display. They also specify properties of the user actions in the interface and of associated internal reactions of the computer.

The procedures of the interface map the internal model to the presentation model on the display. They also monitor and interpret user input with the mouse and with the keyboard, and trigger the execution of procedures of the frame model.

Object structure: The data and the procedures of the frame model and its interface are not treated as independent entities. Instead, data and procedures are combined in the object structure described in section 8.2.

During the design of the frame model, equivalence relations are established for subsets of the data set. This process structures the data of the model. The structure of the frame model is, however, not unique. The suitability of the chosen equivalence relations influences the quality of the model significantly. Each equivalence class is implemented in Java as a class of objects.

Each procedure of the frame model is implemented as a method. The method contains sequences of statements that specify the operations of the procedure, as well as control statements that influence the sequence in which these operations are performed. Each method is implemented in a specific class and is called as attribute of an object on whose attributes the method operates implicitly. Attributes of other objects are addressed explicitly. The methods also contain local variables. Some methods are called as attributes of a class object (static methods in Java).

Data of the model that describe an object of a class are stored as attributes of the object. Data of the model that describe the class are stored as attributes of the class. Transient data used in a method are defined as local variables.

8.3.2 Class Structure of Frame Models

The data and procedures of the frame model are arranged in 45 classes. In order to maintain an overview of this large number of classes, the classes are arranged in groups. Objects of the classes of a group are used for related tasks:

(a) General classes

Session	: start and coordination of the computer session
Base	: data base for all objects of the model
NamedObject	: interface for objects with String identifiers
Frame	: model data structure
Z	: formatted alphanumeric input and output
Key	: enumeration of symbolic constants

(b) Components of the frame

Node	: node of a truss or frame
Bar	: element of a truss
Member	: element of a frame
Load	: external force influence on the frame
Support	: external displacement influence on the frame
Element1D	: interface for bars and members

(c) Methods of analysis

Deformation	: nonlinear analysis of the frame deformation under load
Stability	: determination of singular points and continuation of load paths
Path	: constant arc increment method for load paths

(d) Mathematical tools

Algebra	: vector and matrix algebra
Equation	: linear equations with profile structure and status
Eigen	: general eigenvalue problem for profile matrices
Special	: special eigenvalue problem for profile matrices
Jacobi	: special eigenvalue problem for full matrices
Complex	: complex number algebra
Cubic	: solution of cubic equations
Quartic	: solution of quartic equations

(e) Graphical user interface

Display	: overall structure of the presentation of the interface
Board	: panel of the display
Control	: menu for overall control of the interface
Surface	: mapping of the frame model to the presentation model
Formater	: formatting of variables for graphics display
Diagram	: display of a function on a grid
MessageField	: messages for the user

(f) Presentation of frame behaviour

- GraphicPanel : presentation of the frame for input of components
- FramePanel : presentation of computed frame behaviour
- ResultPanel : presentation of displacement or reaction history at a node
- NodeResult : displacement and reaction at a node for a state of the frame
- BarResult : axial strain, stress and force in a bar for a state of the frame
- MemberResult : axial force, shear and bending moment diagrams of a member

(f) Editors

- BasicEditor : common data and methods for all editors
- AnalysisEditor : control parameters for the analysis of the frame
- ElementEditor : specification of bars and members
- FormatEditor : specification of the format of graphical presentations
- LoadEditor : specification of loads acting at nodes
- ModelEditor : definition of models and persistent storage
- NodeEditor : specification of nodes of the frame
- ResultEditor : specification of the results to be presented
- SupportEditor : specification of supports acting at nodes

The global class structure of a frame model is shown in figure 8.6:

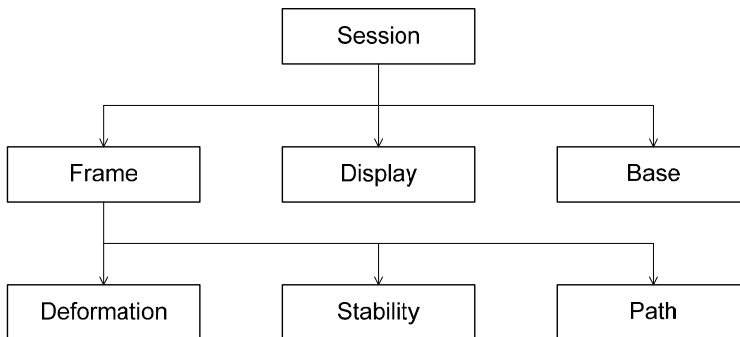


Figure 8.6 Global class structure of a frame model

The interaction between the classes in which the frame model and its behaviour are treated, is described in more detail in section 8.5. The interaction between the classes in which the graphical user interface of the frame model is treated, is described in more detail in section 8.6.

8.4 DATA STRUCTURE OF A FRAME MODEL

8.4.1 Classes Containing the Data Structure

The implementation of the data structure of frame models is distributed over several classes. Data associated with the frame as a whole are treated in class `Frame`. Data associated with specific nodes, supports, bars and members or with the graphical interface are treated in the classes `Node`, `Support`, `Bar`, `Member`, `Stability` and `GraphicPanel`.

All variables of the data structure of the frame model exist only once. Their value is not copied to other locations. Objects of any class that wish to read or write in a variable of the data structure must do so by calling the object that contains the unique copy of this variable.

8.4.2 Data Structure in Class `Frame`

Each `Frame` object contains the following data aggregations:

(a) Names of the components of the frame model

`HashSet <String> nodeNameSet`

`HashSet <String> elementNameSet`

`HashSet <String> loadNameSet`

`HashSet <String> supportNameSet`

(b) History of frame variables over the load steps

`ArrayList <Double> loadFactorHistory`

`ArrayList <Double> displacementNormHistory`

`ArrayList <Double> specialDisplacementNormHistory`

`ArrayList <Double> forceNormHistory`

`ArrayList <Double> specialForceNormHistory`

(c) Pattern variables that do not depend on the instant configuration

`double[] nodeLoadPattern`

`double[] nodeDisplacementPattern`

(d) Pattern variables that depend on the instant configuration

`double[] patternLoad`

`double[] patternDisplacement`

`double[] patternReaction`

(e) State of the frame at the start of the current load step

`double[] stateLoad`

`double[] stateDisplacement`

`double[] stateReaction`

(f) State increments in the current load step

double[] incrementLoad

double[] incrementDisplacement

double[] incrementReaction

(g) Trial state of the frame at the end of the current load step

double[] trialLoad

double[] trialDisplacement

double[] trialReaction

(h) Error in the state of the frame at the start of the current load step

double[] stateErrorLoad

double[] stateErrorDisplacement

double[] stateErrorReaction

(i) Error in the trial state of the frame at the end of the current load step

double[] trialErrorLoad

double[] trialErrorDisplacement

double[] trialErrorReaction

(j) State of the frame at the start of the last load step

double[] previousLoad

double[] previousDisplacement

double[] previousReaction

(k) State of the frame at the start of the next to last load step

double[] pivotLoad

double[] pivotDisplacement

double[] pivotReaction

Each Frame object contains the following scalar variables:

Equation	equation	: system equations with profile structure
double	dimension	: of the system equations
boolean	isAnalysed	: nonlinear behaviour of the frame
boolean	hasTensionOnlyBars	: in the frame
int	numberOfLoadSteps	: in the analysis
double	initialLoadFactor	: for the load path
double	initialSlope	: for the load path
double	errorTestFactor	: for the error limit
double	deformationErrorLimit	: for the deformation analysis
double	stabilityErrorLimit	: for the stability analysis
double	stateErrorNorm	: at the start of the current load step
double	trialErrorNorm	: at the end of the current load step

int	stepAtSignChange	: sign of the force norm
int	decompositionCode	: for decomposition of the stiffness matrix
double	loadFactorIncrement	: for the current load step
double	loadFactor	: at the end of the current load step
double	chordLength	: for the arc increment method
double	reducedChordLength	: for the arc increment method
double	trialChordLength	: for tension-only bars
double	calibrationFactor	: for the displacement norm
double	dimensionFactor	: for scaling of rotations and moments
double	displacementNorm	: for the load path
double	specialDisplacementNorm	: for the load path
double	linearDisplacementNorm	: for the load path
double	forceNorm	: for the load path
double	specialForceNorm	: for the load path
double	linearForceNorm	: for the load path

8.4.3 Data Structure in Class Node

Class object Node contains the following data aggregations for the current node and the current load step of the analysis:

		pattern displacement
static double	supportPatternDisplacement[3]	support space coordinates
static double	globalPatternDisplacement[3]	global space coordinates
		state displacement
static double	supportDisplacement[3]	support space coordinates
static double	globalDisplacement[3]	global space coordinates
		displacement increment
static double	supportDisplacementIncrement[3]	support space coordinates
static double	globalDisplacementIncrement[3]	global space coordinates
		displacement correction
static double	bsSupport[3]	support space coordinates
static double	bsGlobal [3]	global space coordinates

Each Node object contains the following data aggregations:

ArrayList <Double[]>	displacementHistory	: of the node
HashSet	elementNameSet	: incident elements
int[]	indexVector0	: indices of system variables
int[]	indexVector1	: indices of system variables
int[]	indexVector2	: indices of system variables

Each Node object contains the following scalars in addition to those listed in section 8.2.4:

int	nodeNumber	: node sequence for indexing
int	dimensionGroup0	: for indices of system variables
int	dimensionGroup1	: for indices of system variables
int	dimensionGroup2	: for indices of system variables
int	numberOfVariables	: at this node

8.4.4 Data Structure in Class Support

Each Support object contains the following data aggregation:

ArrayList <Double> reactionHistory : of the support

Each Support object contains two scalars in addition to those listed in section 8.2.4:

int	groupNumber	: of the index group at the node
int	localIndex	: in the index group

8.4.5 Data Structure in Class Bar

Class object Bar contains the following data aggregations for the current bar and the current load step of the analysis:

static double	stiffness[4][4]	: stiffness matrix in support space
static double	kReference[4][4]	: stiffness matrix in reference space
static double	save[4][4]	: auxiliary matrix for transformations
static double	endForces[4]	: forces at the end nodes of the bar
		pattern displacement
static double	upSupport[4]	: support space coordinates
static double	upGlobal[4]	: global space coordinates
static double	vpReference[4]	: reference space coordinates
		state displacement at start of step
static double	uSupport[4]	: support space coordinates
static double	uGlobal[4]	: global space coordinates
static double	vReference[4]	: reference space coordinates
		trial displacement at end of step
static double	vTrial[4]	: reference space coordinates
		displacement increment in step
static double	uSupportIncrement[4]	: support space coordinates
static double	uGlobalIncrement [4]	: global space coordinates
static double	vReferenceIncrement [4]	: reference space coordinates
		displacement correction
static double	bsSupport[4]	: support space coordinates
static double	bsGlobal[4]	: global space coordinates
static double	bsReference[4]	: reference space coordinates

Each Bar object contains the following data aggregations:

int[]	systemIndex	: of the element variables
int[]	localStartIndex	: in the index groups at the nodes
int[]	localEndIndex	: in the index groups at the nodes

Each Bar object contains the following scalars in addition to those listed in section 8.2.4:

double	cosReference,sinReference	: reference configuration
double	cosInstant,sinInstant	: configuration at the start of the step
double	cosTrial,sinTrial	: configuration at the end of the step
boolean	isStiff	: for tension-only members
double	referenceLength	: in the reference configuration
double	instantLength	: configuration at start of load step
double	instantLengthTrial	: configuration at end of load step
double	stateStress	: axial stress at the start of the step
double	stateStrain	: axial strain at the start of the step
double	trialStress	: axial stress at the end of the step
double	trialStrain	: axial strain at the end of the step

8.4.6 Data Structure in Class Member

Class object Member contains the following data aggregations for the current member, load step of the analysis and instant configuration:

static double	stiffness[6][6]	: stiffness matrix in support space
static double	klInstant[6][6]	: stiffness matrix in instant space
static double	save[6][6]	: auxiliary stiffness matrix
static double	hold[6][6]	: auxiliary stiffness matrix
static double	endForces[6]	: forces at end nodes of member
static double	load[6]	: member load vector
pattern displacement		
static double	upSupport[6]	: support space coordinates
static double	upGlobal[6]	: global space coordinates
static double	vpReference[6]	: reference space coordinates
static double	wpInstant[6]	: instant space coordinates
state displacement at start of step		
static double	uSupport[6]	: support space coordinates
static double	uGlobal[6]	: global space coordinates
static double	vReference[6]	: reference space coordinates
static double	wInstant[6]	: instant space coordinates
trial displacement at end of step		
static double	vTrial[6]	: reference space coordinates

		displacement increment in step
static double	uSupportIncrement[6]	: support space coordinates
static double	uGlobalIncrement [6]	: global space coordinates
static double	vReferenceIncrement [6]	: reference space coordinates
static double	wInstantIncrement [6]	: instant space coordinates
		displacement correction
static double	bsSupport[6]	: support space coordinates
static double	bsGlobal[6]	: global space coordinates
static double	bsReference[6]	: reference space coordinates
static double	bsInstant[6]	: instant space coordinates
		interpolation vectors
static double	s1Inter[6]	: instant displacement w_1
static double	s2Inter[6]	: instant displacement w_2
static double	g1Inter[6]	: instant displacement derivative $w_{1,1}$
static double	g2Inter[6]	: instant displacement derivative $w_{2,1}$
static double	h2Inter[6]	: instant displacement derivative $w_{2,11}$

Each Member object contains the following data aggregations:

int[]	systemIndex	: of the element variables
int[]	localStartIndex	: in the index groups at the nodes
int[]	localEndIndex	: in the index groups at the nodes

Member objects contain the following scalars in addition to those in section 8.2.4:

boolean	hasLineLoad	
double	p1A, p2A, p1B, p2B	: load coordinates in reference space
double	cosReference	: reference configuration
double	sinReference	: $\mathbf{x} = \mathbf{a} + \mathbf{R}_{xy} \mathbf{y}$
double	referenceLength	
double	cosInstantState	: instant configuration at start of step
double	sinInstantState	: $\mathbf{x} = \mathbf{b} + \mathbf{R}_{xz} \mathbf{z}$
double	instantLengthState	
double	cosLocalState	: increment of the configuration
double	sinLocalState	: $\mathbf{y} = \mathbf{c} + \mathbf{R}_{yz} \mathbf{z}$
double	thetaLocalState	: instant chord rotation
double	gammaStateA	: relative rotation of node A
double	gammaStateB	: relative rotation of node B
double	axialStrainState	: state at start of step
double	axialStressState	
double	strainDerivativeState	

double	stressDerivativeState	
double	cosInstantTrial	: instant configuration at end of step
double	sinInstantTrial	: $\mathbf{x} = \mathbf{b} + \mathbf{R}_{xz} \mathbf{z}$
double	instantLengthTrial	
double	cosLocalTrial	: increment of the configuration
double	sinLocalTrial	: $\mathbf{y} = \mathbf{c} + \mathbf{R}_{yz} \mathbf{z}$
double	thetaLocalTrial	: instant chord rotation
double	gammaTrialA	: relative rotation of node A
double	gammaTrialB	: relative rotation of node B
double	axialStrainTrial	: state at start of step
double	axialStressTrial	
double	strainDerivativeTrial	
double	stressDerivativeTrial	

8.4.7 Data Structure in Class Stability

Each Stability object contains the following data aggregations:

double[]	eigenReaction	: reaction due to eigendisplacement
double[]	testDirection	: displacement increment leading to a singular point

Each Stability object contains the references of the following objects:

Eigen	eigen	: general eigenvalue problem for the determination of singular points
Special	special	: special eigenvalue problem for the continuation of a load path at a bifurcation point

8.4.8 Data Structure in Class GraphicPanel

Each GraphicPanel object contains the following scalars:

int	milliToPixel	: number of pixels per millimetre
double	userToMilli	: number of millimetres per user unit
double	userToPixel	: number of pixels per user unit
double	pixelToMilli	: number of millimetres per pixel
double	milliToUser	: number of user units per millimetre
double	pixelToUser	: number of user units per pixel
int	pixelPerMesh	: number of pixels per mesh of the grid
int	meshesPerBlock	: number of meshes per block of the grid
int	widthPixel	: width of the graphics panel, in pixels
int	heightPixel	: height of the graphics panel, in pixels
int	pixelPerBlock	: number of pixels per block of the mesh
int	userPerBlock	: number of user units per block of the grid

int	marginPixel	: width of the margin of the grid in pixels
int	cellPixel	: parameter for annotation
int	rimPixel	: width of the rim of the grid, in pixels
int	offsetPixel	: parameter for the annotation
int	xOriginPixel	: panel coordinate x of the origin of the grid
int	yOriginPixel	: panel coordinate y of the origin of the grid
int	nDecimals	: format parameter for real numbers
int	xMinPixel	: panel coordinate x for left edge of grid
int	yMinPixel	: panel coordinate y for top edge of grid
int	xMaxPixel	: panel coordinate x for right edge of grid
int	yMaxPixel	: panel coordinate y for bottom edge of grid
double	x1MinUser	: x coordinate of left bottom corner in user units
double	x2MinUser	: y coordinate of left bottom corner in user units
int	xPixelInput	: panel coordinate x for mouse click
int	yPixelInput	: panel coordinate y for mouse click
double	xMilliInput	: coordinate x for mouse click in millimetre space
double	yMilliInput	: coordinate y for mouse click in millimetre space
double	x1UserInput	: coordinate x for mouse click in user space
double	x2UserInput	: coordinate y for mouse click in user space
double	x1UserInputGrid	: coordinate x of nearest grid point in user space
double	x2UserInputGrid	: coordinate y of nearest grid point in user space
boolean	markNode1	: has node1 been selected?
int	markedX1	: panel coordinate x of node 1, in pixels
int	markedY1	: panel coordinate y of node 1, in pixels
boolean	markNode2	: has node2 been selected?
int	markedX2	: panel coordinate x of node 2, in pixels
int	markedY2	: panel coordinate y of node 2, in pixels
boolean	drawNodes	: show the nodes in the frame diagram
boolean	drawElements	: show the elements in the frame diagram
boolean	drawLoads	: show the loads in the frame diagram
boolean	drawSupports	: show the supports in the frame diagram

Other classes contain some additional variables of the model data structure.

8.5 METHOD STRUCTURE OF A FRAME MODEL

8.5.1 Class Structure for Nonlinear Analysis

The algorithms for the geometrically nonlinear analysis of trusses and frames are implemented as methods on the Java platform. These methods are arranged in four main groups and used for the following tasks:

- (a) computation of the nonlinear deformation of a frame
- (b) computation of singular points and analysis of the stability of the frame
- (c) handling and presentation of the model in the graphical user interface
- (d) parameter-driven automatic generation of structural models

These methods are described in the following sections and implemented in the classes shown in figure. 8.7. The methods of group (a) for the computation of the nonlinear deformation of frames are described in section 8.5.2. The execution of these methods is coordinated in class Deformation. The methods of group (b) for the

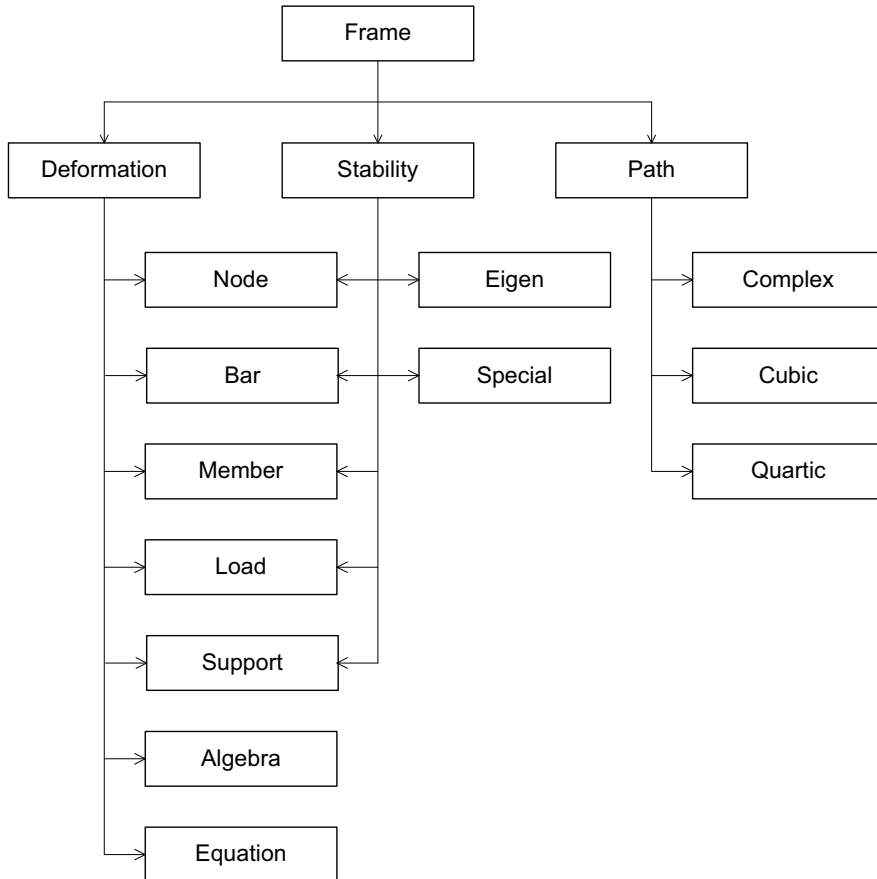


Figure 8.7 Class structure for nonlinear analysis

computation of singular points and the analysis of stability are described in section 8.5.3. The execution of these methods is coordinated in class *Stability*. Both the deformation analysis and the stability analysis call methods of class *Path* which compute load factor increments as well as general and special norms. The methods of group (d) are not covered in this publication. Any frame generated with software can also be input manually at the user interface of the application.

The methods for nonlinear frame analysis call methods in mathematical classes such as *Equation*, *Eigen* and *Special*. These classes are not specific classes for nonlinear structural analysis. They perform general numerical tasks such as matrix operations, the solution of linear equations and the solution of eigenproblems.

8.5.2 Methods for Deformation Analysis

The workflow of the nonlinear analysis of frame deformation is shown in figure 8.8. The procedure contains three nested loops. The outer loop treats the load steps of the initial value problem sequentially. The middle loop treats the reduction of the chord length of the constant arc increment method if the iteration for a larger chord length fails to converge. The inner loop iterates on the secant stiffness matrix. In cycle 0 of the iteration, the secant matrix is approximated by the tangent stiffness matrix at the start of the load step. In subsequent cycles, the secant matrix is approximated using the displacement increment that was computed in the preceding cycle of iteration. The iteration is terminated if the norm of the error in the trial state at the end of the load step is less than a specified limit.

The secant matrix is assembled at the start of each loop by traversing the elements of the frame and adding their contributions. It is then decomposed as the product of a left triangular matrix, a diagonal matrix and a right triangular matrix, as described in chapter 6. If the number of negative diagonal elements changes between the start and the end of the step, the step contains a singular point. A stability analysis is performed that determines the singular point and continues the load path as described in section 8.5.3. The deformation analysis is continued after the stability analysis has been completed.

At the start of the load step, the decomposed secant matrix is used to compute the displacements and the reactions due to the pattern loads and prescribed pattern displacements. The trial state at the end of the load step is then computed in phases. It is initially set equal to the state at the start of the load step. The unbalanced forces at the start of the step and the resulting displacement and force increments are computed with the decomposed secant matrix and added to the trial state. A method of class *Path* is then called to compute the load factor increment for the load step. The increment state for the load step equals the pattern state, multiplied by the load factor increment. The increment state is added to the trial state which becomes the approximate state at the end of the load step.

The unbalanced forces for the trial state are determined. The displacements and reactions due to these forces are computed with the decomposed secant matrix and added to the trial state to obtain the modified trial state. If the norm of the error vector

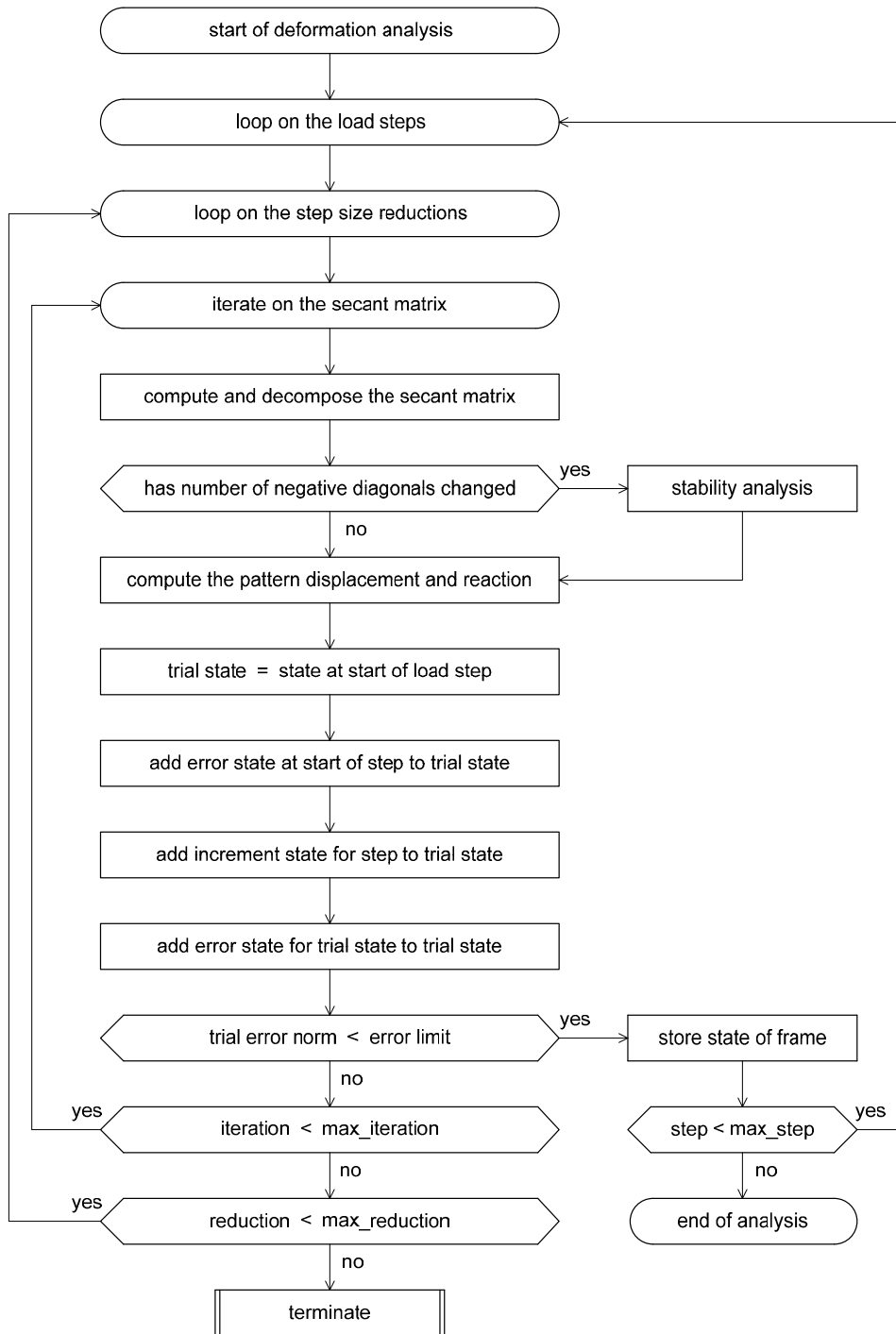


Figure 8.8 Work flow of nonlinear deformation analysis

for the modified trial state is less than a prescribed limit, the iteration in the load step is terminated and the state at the end of the step is stored in the history of the analysis. If the iteration does not converge after a specified number of cycles, the algorithm back-steps to the beginning of the procedure for the load step and the chord length is reduced. The computations for the load step are repeated with the new chord length. If a specified number of reductions does not lead to convergence of the iteration, the analysis of the frame is terminated.

The methods for deformation analysis are arranged in the classes, as follows. Only the main methods are listed. Methods for data management are not listed.

(a) **Class Frame**

```
void      allocateArrays (systemDimension)
void      initialiseComponentHistory (numberOfSteps)
void      saveStateOfFrame (state)
void      restoreStateOfFrame (state)
void      backStep (step,iteration)
boolean   computeReferenceSpace ()
boolean   computeStateSpace ()
boolean   computeTrialSpace ()
```

(b) **Class Deformation**

```
void      analyse()
void      computeAndDecomposeStiffnessMatrix (state,iteration)
void      computePatternState (state,iteration)
void      computeTrialState (step)
void      computeStateErrorState (state,iteration)
void      computeTrialErrorState (state,iteration)
void      computeIncrementState (step,iteration)
void      prepareSystemEquations ()
void      computeNodeSystemIndices ()
boolean   computeElementSystemIndices ()
void      computeNodeLoadPattern ()
void      computeNodeDisplacementPattern ()
void      computePatternLoad (state)
void      computeStateLoad (state)
void      enterPatternDisplacement ()
void      computeStateErrorLoad (state)
void      computeTrialErrorLoad ()
void      computeTangentStiffnessMatrix (state)
void      computeSecantStiffnessMatrix (state)
void      computeTotalStiffnessMatrix (state)
```

(c) Class Bar

boolean computeReferenceSpace ()
boolean computeInstantSpace ()
double[][] computeStiffnessMatrix (state)
double[][] computeKReference (state)
void rotateStiffnessMatrixToSupportSpace (matrix,sin,cos)
void computeStress (state)
double computeAxialForce (state)
double[] computeEndForces (state)

(d) Class Member

boolean computeReferenceSpace ()
boolean computeInstantSpace (int state)
void initialiseAnalysis ()
double[] getNodalElementDisplacement (state)
double[] getGlobalElementDisplacement (state)
double[] getReferenceElementDisplacement (state)
double[] getInstantElementDisplacement (state)
double computeRelativeRotation (beta,theta)
void computeRelativeNodeRotations (state)
void computeStress (zPoint,state)
void interpolateDisplacement (zPoint,state)
void interpolateDisplacementDerivatives (zPoint,state)
void interpolateDisplacementIncrement (zPoint,state)
void interpolateDisplacementDerivativeIncrements (zPoint,state)
double[] computeEndForcesInSupportSpace (state)
void computeInstantEndForces (state)
double[][] computeStiffnessMatrix (state)
void computeInstantStiffnessMatrix (state)
double[] computeStateLoadVector (state)
double[] computeTrialLoadVector ()
double[] computeLoadVector (instantLength,sinInstant,cosInstant,
sinLocal,cosLocal)
void computeInstantLoad (state)
void rotateStiffnessMatrixToSupportSpace (matrix,sin,cos,res)
void rotateVectorToSupportSpace (vector,sin,cos)

(e) Class Path

```
void      computeErrorLimit ()
void      computeNorms ()
double    computeDisplacementNorm (vector[])
double    computeSpecialDisplacementNorm (vector[])
double    computeUncalibratedDisplacementNorm (vector[])
double    computeForceNorm (load[], reaction[])
double    computeSpecialForceNorm (load[])
double    computeChordLength (uA[], uB[], qA[], qB[], rA[], rB[])
double    computeSpecialChordLength (uA[], uB[], qA[])
double    computeLoadFactorIncrement (chordLength, displacement[],
                                       load[], reaction[])

void      computeParameters (displacement[], load[], reaction[])
double    computeApproximateIncrement (chordLength)
double    computePivotDistance (increment)
void      iterateLoadFactorIncrement (chordLength)
double    computeFactorAtSingularPoint ()
```

8.5.3 Methods for Stability Analysis

The workflow of the stability analysis of frames is shown in figure 8.9. The analysis is triggered when the frame reaches a nearly singular configuration. The procedure contains two sections. The first section determines the singular point. The second section distinguishes between turning and bifurcation points and branches according to the type of the singular point to compute the continuation of the load path.

Steps to a singular point: The first section of the stability analysis contains two loops. The outer loop treats the load steps from the nearly singular point to the singular point. The inner loop iterates on the secant stiffness matrix. In cycle 0 of the iteration the secant matrix is approximated by the tangent stiffness matrix at the start of the load step. In subsequent cycles the secant matrix is approximated using the displacement increment computed in the preceding cycle of iteration.

The secant matrix for the load step is computed and decomposed at the beginning of each cycle of the iteration. It is used to compute the displacement and the reaction due to the pattern load, as well as the unbalanced forces (error vector) at the start of the load step. The coefficient matrices of the general eigenvalue problem for the determination of the singular point are assembled and the eigenvalue problem is solved for the load factor increment. This factor is used to compute an approximate value for the displacement increment for the load step. The iteration in the inner loop is terminated when the change in the load factor increment during the last cycle of iteration is less than a prescribed limit. If a specified number of iteration cycles is exceeded, the analysis is terminated.

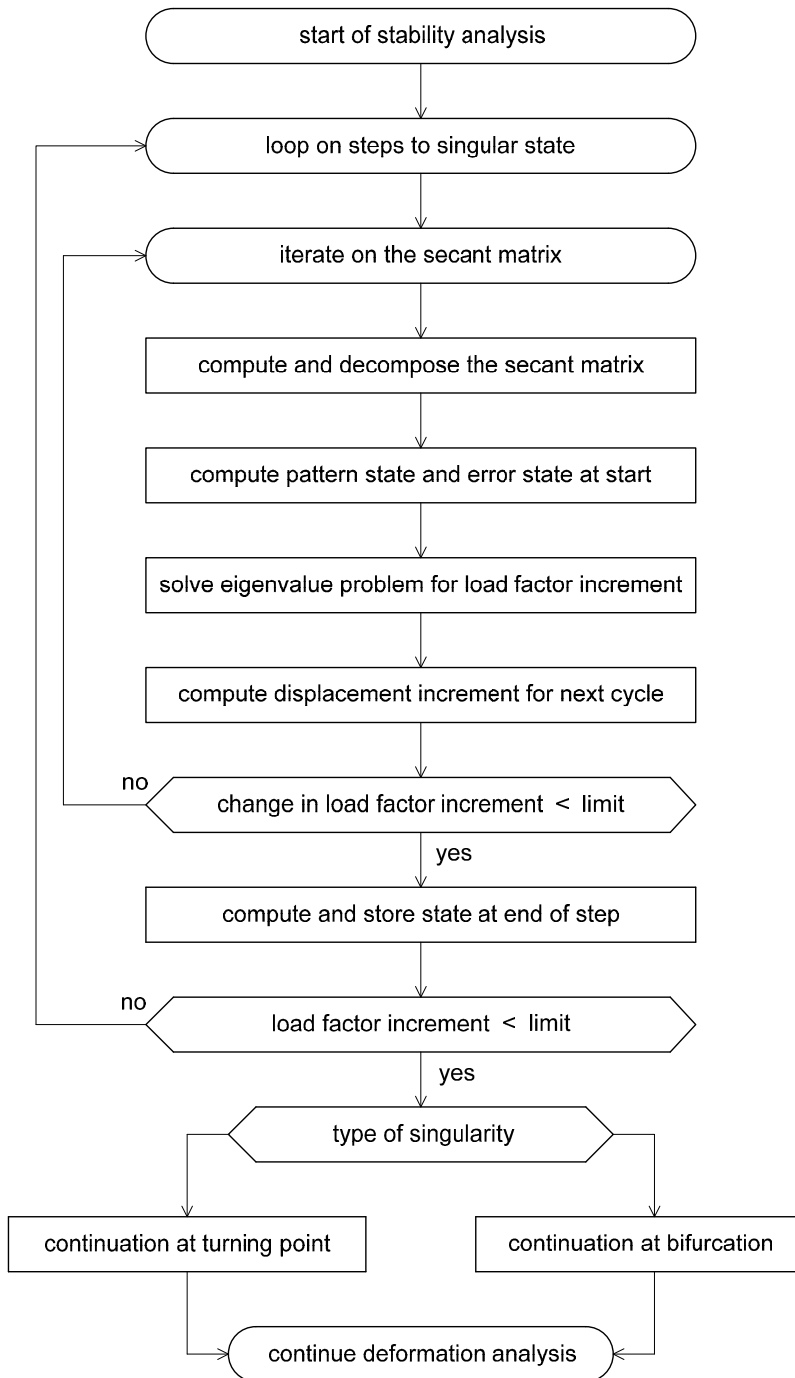


Figure 8.9 Work flow of the stability analysis

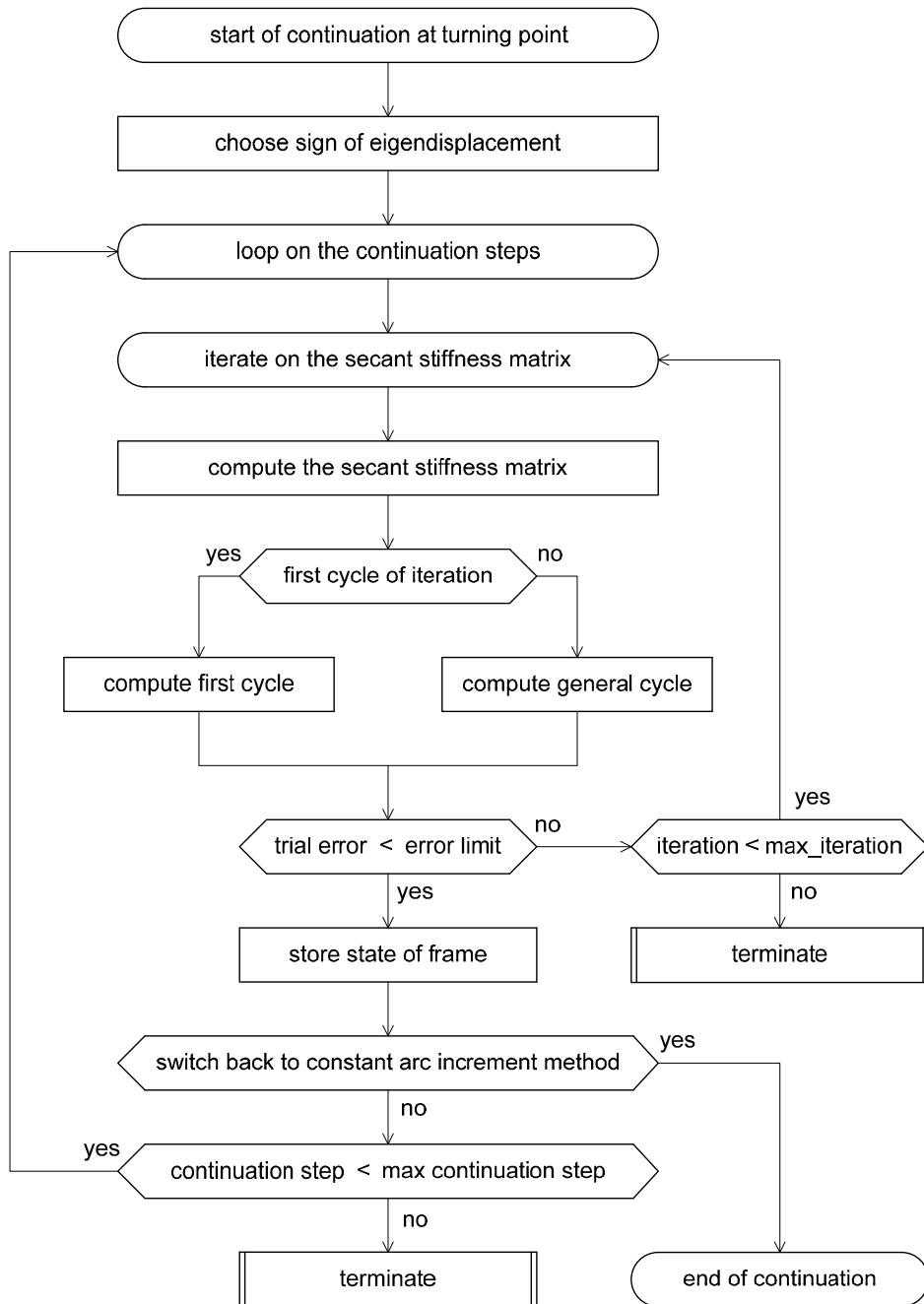


Figure 8.10 Work flow of the continuation at a turning point

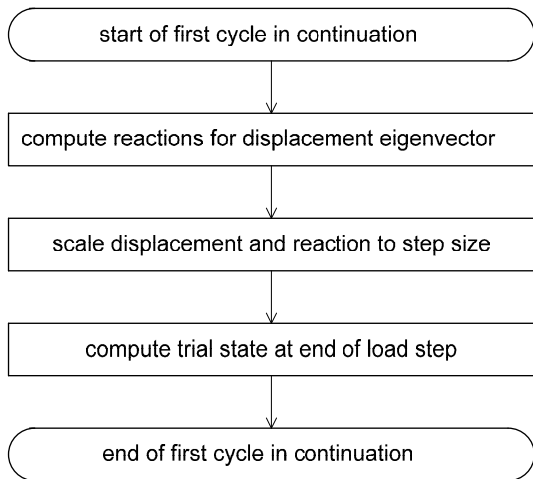


Figure 8.11 Work flow in the first cycle of a continuation

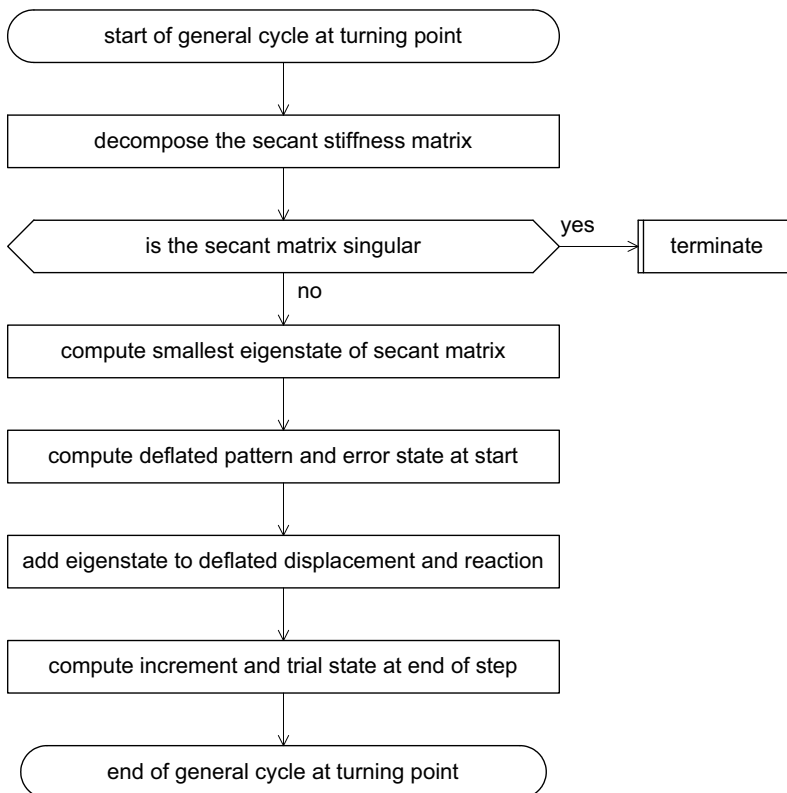


Figure 8.12 Work flow of the general iteration cycle at a turning point

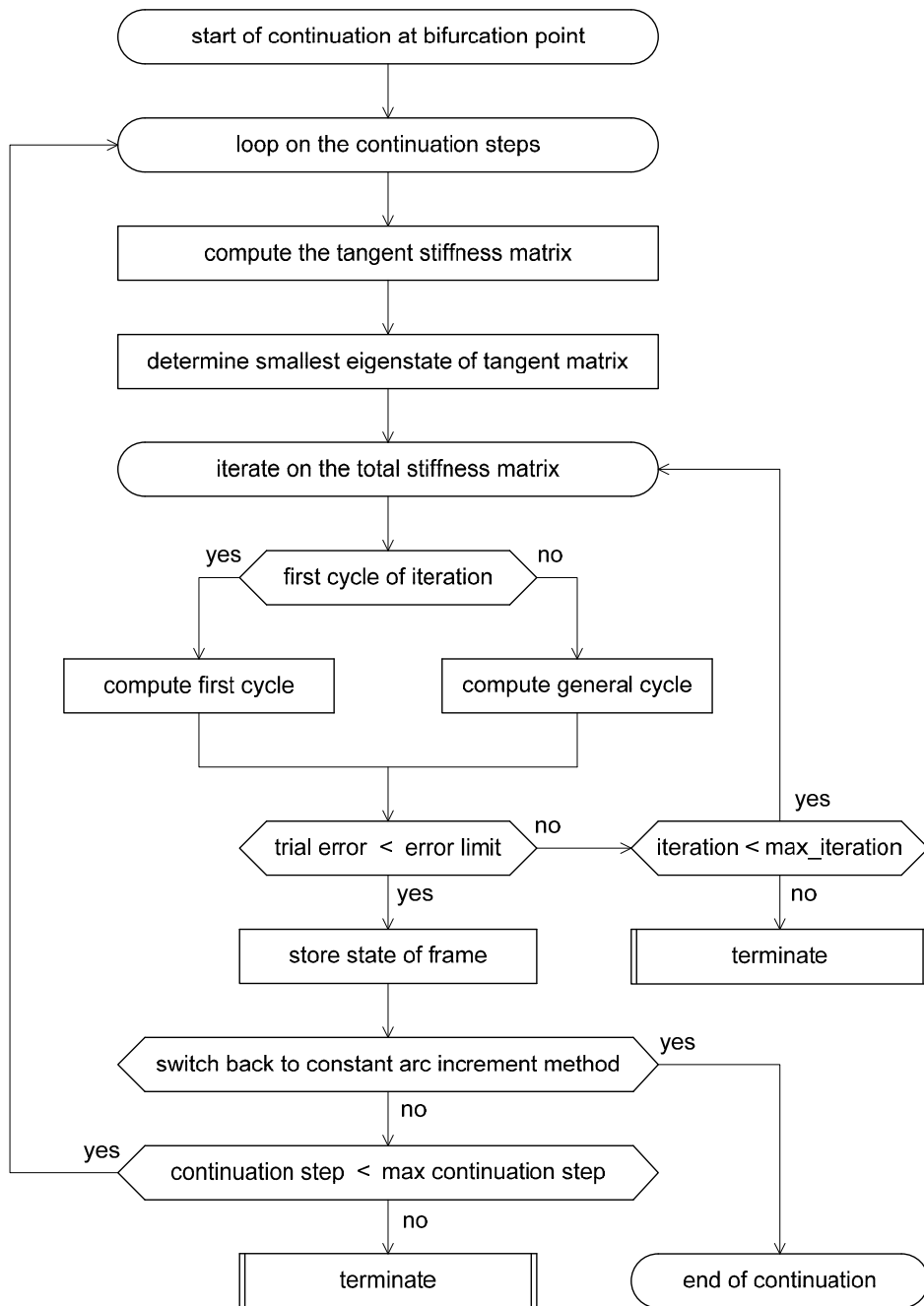


Figure 8.13 Workflow of the continuation at a bifurcation point

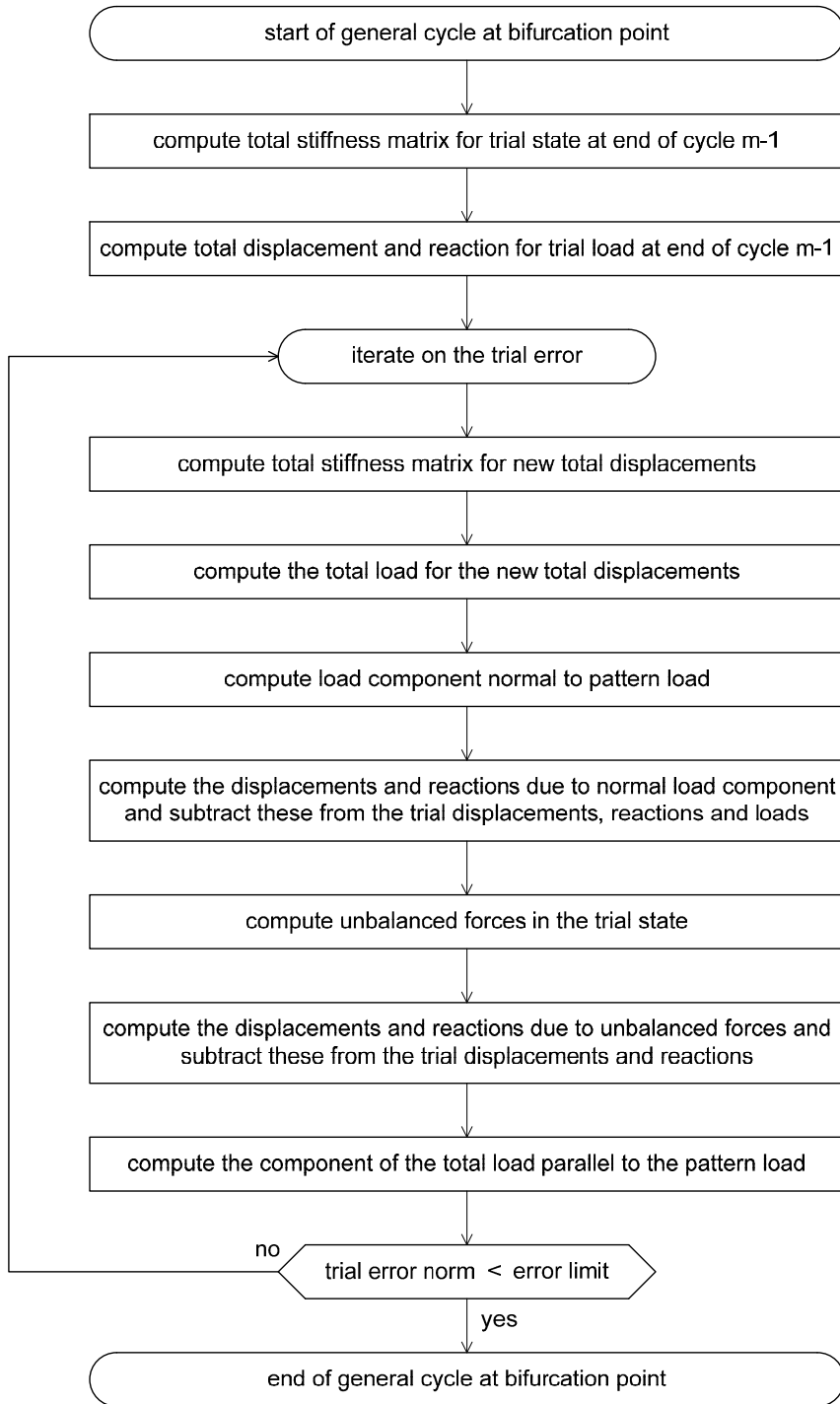


Figure 8.14 Workflow of the general iteration cycle at a bifurcation point

If the iteration in the inner loop has converged, the state of the frame at the end of the load step is stored. If the absolute value of the load factor increment in the load step is less than a prescribed limit, the iteration in the outer loop is terminated and the singular point has been reached. If a specified number of load steps from the nearly singular point to the singular point is exceeded, the analysis is terminated.

Continuation: In the second section of the stability analysis, the load path is continued beyond the singular point. The type of the singular point is determined. If it is a turning point, the work flow of figures 8.10, 8.11 and 8.12 yields the continuation. If the singular point is a bifurcation point, the continuation is computed with the work flow of figures 8.13, 8.11 and 8.14. The stability analysis ends with the termination of the computation of the continuation. The algorithm returns to the deformation analysis described in section 8.5.2.

Continuation at a turning point: The continuation of the load path at a turning point is treated in figure 8.10. It is assumed that the eigenvector associated with the smallest eigenvalue of the tangent stiffness matrix at the singular point is known. The sign of the eigenvector is chosen so that the vector makes the smallest possible angle with the displacement increment that led from the nearly singular to the singular point.

The work flow then enters a double loop. The outer loop treats the continuation steps, the inner loop iterates on the secant stiffness matrix. In the first cycle of the inner loop, the secant matrix is set equal to the tangent matrix. In the subsequent cycles, the displacement increment determined in the preceding cycle is used to approximate the secant matrix for the load step. Different work flows are executed for the first and the other cycles of iteration to determine a trial state at the end of the load step.

If the norm of the trial error vector is less than a specified limit, the inner loop has converged to the state of the frame at the end of the load step. This state is stored in the history of the frame. The change in the force norm in the load step is then compared to the change in the displacement norm. If the ratio exceeds a preset value, the mode of analysis is switched back from stability analysis to nonlinear deformation analysis. Otherwise the next step of the continuation is computed. If the inner or the outer loop does not converge in a specified number of cycles, the analysis is terminated.

First cycle of the first step: The work flow for the first cycle of the first step of a continuation at a turning point is shown in figure 8.11. The eigenvector for the singular point is substituted as displacement vector into the system equations to obtain the corresponding system reaction. The increments in the calibrated displacement norm and the force norm in the first step are scaled so that the chord for the step in the load path has a preset length. The scaled displacement and reaction vectors are added to the displacement and reaction state at the start of the load step to obtain the trial state at the end of the load step.

General cycle of the continuation at a turning point: The work flow for a general cycle of the iteration in a load step of the continuation at a turning point is shown in figure 8.12. The secant stiffness matrix is decomposed at the start of the cycle. The eigenvector associated with the smallest eigenvalue of the secant matrix is determined. The pattern load and the unbalanced forces (error vector) at the start of the load step are deflated with the eigenvector and the deflated displacement and reaction vectors for the two loadings are computed with the secant matrix. The contribution of the eigenvector is then added to the deflated displacement and reaction vectors. The load factor increment in the load step is determined so that it yields a preset chord length in the load step. The load factor increment is used to compute the increments of the displacement and the reaction in the load step, as well as the resulting trial state at the end of the load step.

Continuation at a bifurcation point: The work flow for the continuation of the load path at a bifurcation point is shown in figure 8.13. It contains two loops. The outer loop treats the continuation steps while the inner loop iterates on the total stiffness matrix. At the start of the loop for a continuation step, the tangent stiffness matrix at the start of the step is computed. The eigenvector corresponding to the smallest eigenvalue of the tangent matrix is determined. Then the iteration in the inner loop is started. In the first cycle of this loop, the trial state is computed as in the continuation at a turning point. In the general cycle, a special procedure for bifurcation points is executed as described below to obtain the trial state. The test of the error norm in the trial state and the subsequent operations are the same as at a turning point (see figure 8.10).

General cycle of the continuation at a bifurcation point: The work flow for a general cycle of the iteration in a load step of the continuation at a bifurcation point is shown in figure 8.14. The total stiffness matrix at the end of the load step is computed with the trial displacement computed in the preceding cycle of iteration. It is used to compute the total displacements and reactions for the trial load at the end of the load step. The new values of the total displacements are used to compute a new total stiffness matrix at the end of the step.

The system equations containing the total displacement and the total stiffness matrix at the end of the load step are used to compute the total reactions and the total load at the end of the load step. The component of the total load vector that is normal to the pattern load is determined. The displacements and reactions due to this normal load component are computed and subtracted from the total displacements and reactions to yield a trial state at the end of the load step.

The unbalanced forces in the trial state are computed. The displacements and the reactions due to these forces are computed with the total stiffness matrix and subtracted from the trial displacements and reactions. Finally the component of the total load parallel to the pattern load is computed.

The methods for stability analysis are contained in class `Stability`. In the following, only the main methods are listed. Methods for data management are not listed.

Class Stability

```
void analyse ()
void allocateArrays ()
void saveTestDirection ()
void computeStepsToSingularity ()
void computeSingularState ()
void computeTypeOfSingularity (eigenVector[])
void computeContinuationAtTurningPoint (vector[])
void doFirstCycleAtContinuation (vector[])
void doGeneralCycleAtTurningPoint ()
void computeContinuationAtBifurcationPoint (vector[])
void doGeneralCycleAtBifurcation ()
```

8.5.4 System Indices of the Node Displacements

The system indices of the node displacements are determined with a special concept because the elements that are incident at a node can be of different types:

- frame members subjected to bending and axial force with two displacement coordinates and one rotation coordinate at each end node of the member
- bars subjected to axial force with two displacement coordinates at each end of the bar

The elements at a node can be connected in different ways:

- all elements are connected so that their nodes undergo the same generalised displacements (their system translation and rotation coordinates are equal)
- the elements are connected in groups so that the value of a displacement coordinate (for instance the rotation) is equal for all elements of a group, but differs from the value of this displacement coordinate for the other group

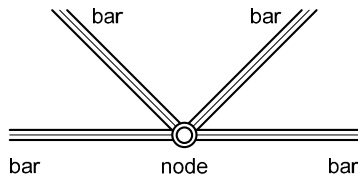
The different element and connection types lead to many different combinations of node variables, as shown in figure 8.15. In order to permit the analysis of frames with all possible combinations of element and connection types, the system variables at a node are decomposed into three groups that are associated with the axes of the support coordinate system:

group 0: translation in the direction of axis s_1 in figure 5.2

group 1: translation in the direction of axis s_2 in figure 5.2

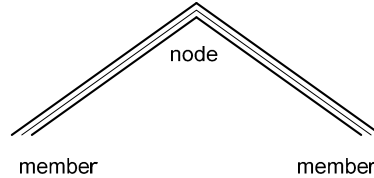
group 2: rotation about the axis normal to the plane of the frame

The number of variables in a group is called the dimension of the group. The dimension of the group can be null. For example, the dimension of group 2 is null at a node where all incident elements are bars. At a standard node of a frame, the dimension of all three groups is 1. If there are four members that are fully connected for forces but only connected pair-wise for moments, the dimension of groups 0 and 1 is 1, but the dimension of group 2 is 2 since the two pairs of members rotate independently.



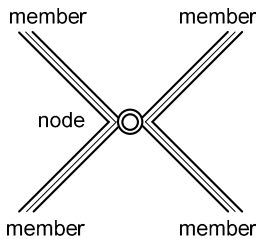
dimension group 0 = 1
 dimension group 1 = 1
 dimension group 2 = 0

STANDARD TRUSS NODE



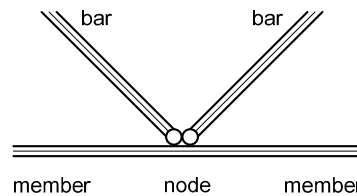
dimension group 0 = 1
 dimension group 1 = 1
 dimension group 2 = 1

STANDARD FRAME NODE



dimension group 0 = 1
 dimension group 1 = 1
 dimension group 2 = 2

HINGE IN FRAME



dimension group 0 = 1
 dimension group 1 = 1
 dimension group 2 = 1

HYBRID NODE

Figure 8.15 Combinations of elements and connections

The system variables in a group at a node have consecutive local indices, starting with 0. If the dimension of a group at a node exceeds 1, the user must specify for each element variable the local index of the corresponding system variable. The input consists of an index list for every element incident at the node. If two incident elements are connected for the displacement in group X at the node, they must be given the same local index in group X. In order to facilitate the user input for the specification of the structural model, the following conventions have been introduced:

(a) If the user does not specify any group dimensions at a node and only bars are incident at the node, a standard truss node with dimension 1 of groups 0 and 1, and dimension 0 of group 2, is assumed.

(b) If the user does not specify any group dimensions and at least one frame member is incident at the node, a standard frame node with three groups of equal dimension 1 is assumed.

If the user employs this group numbering scheme, the plausibility of his input is not checked in the implementation. It is his responsibility to assure that the input is structurally meaningful and complete. It is, for instance, not structurally meaningful to have more than one variable at a given node in both group 0 and group 1. At such locations, more than one node with equal coordinates should be defined in the structural model.

The following attributes of nodes, bars and members are used to manage the system indices in the implementation:

Node:	dimensionGroup0	number of variables in group 0
	dimensionGroup1	number of variables in group 1
	dimensionGroup2	number of variables in group 2
	indexVector0 []	system indices of the variables of group 0
	indexVector1 []	system indices of the variables of group 1
	indexVector2 []	system indices of the variables of group 2
Bar:	localStartIndex[2]	local indices of start node variables in the group
	localEndIndex[2]	local indices of end node variables in the group
	systemIndex[4]	system indices of the element variables
Member:	localStartIndex[3]	local indices of start node variables in the group
	localEndIndex[3]	local indices of end node variables in the group
	systemIndex[6]	system indices of the element variables

The variables dimensionGroup, localStartIndex [] and localEndIndex [] are specified by the user or set according to conventions (a) and (b). The variables indexVector [] and systemIndex [] are computed.

In order to compute the system indices at the nodes, a counter is set to null and the nodes of the frame are traversed in the order of their node numbers. Consecutive values are assigned to the indices in the three groups at a node and recorded in the arrays indexVector0 [] to indexVector2 []. The counter is incremented accordingly. The counter is passed to the next node, which uses the counter value as its first index value and proceeds to assign consecutive values to its system indices. The indexing ends when all nodes have been traversed.

In order to compute the index values for the elements of the frame, the elements are traversed in arbitrary order. The local indices of the variables at the nodes of the element are read in arrays localStartIndex [] and localEndIndex [] and used to read the corresponding system indices in arrays indexVector0 [] to indexVector2 []. The index values are stored in the array systemIndex [] of the element.

8.5.5 System Stiffness Matrix, Load Vector and Error Vector

The system stiffness matrices of the frame are computed as described in section 6.2. The methods that compute the tangent, secant and total stiffness matrices in class Deformation loop on the elements of the frame and call the element methods that compute the contribution of the bar or member. These methods compute the coefficients of the element stiffness matrix in the support coordinate spaces of the nodes of the element. Object equation of class Equation is called to add the element matrix to the stiffness matrix of the system.

In class Bar, the coefficients of the element stiffness matrix are computed in the reference space of the bar, transformed to the support spaces of the nodes of the bar with the rotation matrix \mathbf{R}_{esy} of the element and added to the system matrix with the mapping shown in figure 6.5.

In class Member, the coefficients of the element stiffness matrix are computed in the instant coordinate space of the member. The instant coordinate systems of the members are therefore updated for each trial state. The element matrix is then transformed to the support spaces of the nodes of the element with the rotation matrix \mathbf{R}_{esz} and added to the system matrix with the mapping shown in figure 6.5.

The incremental stiffness analysis is performed for the pattern load vector multiplied by the load factor increment. The coordinates of the node loads are specified by the user in the support spaces at the nodes and added directly to the pattern load vector. Member loads are specified for members only, not for bars. The load intensities are specified by the user in the reference space of the member and transformed to the instant space of the member before the coefficients of the element load vector in instant space are computed. The element load vector is transformed to the support spaces of the nodes of the element with the rotation matrix \mathbf{R}_{esz} before it is added to the system load vector.

The error vector for a configuration of the frame is computed in the support space of the nodes by copying the loads and reactions for the configuration and subtracting the end forces of the elements in the configuration. The algorithm traverses the elements of the frame and calls each element to compute its end forces. For bars, the axial end forces are computed in reference space and transformed to the support spaces of the nodes of the bar. For members, the end forces are computed in instant space and transformed to the support spaces of the nodes of the element.

8.5.6 Tension-only Analysis

Consider a frame that contains tension-only members. A tension-only member is defined by the following properties:

- (a) If the instant length of the member is less than its reference length, the member has no stiffness. The member is free of stress.
- (b) If the instant length of the member equals or exceeds its reference length, the member is stiff. The strain and stress in the member are computed as for a bar.

The condition of a tension-only member is defined as true if the member is stiff and as false if it is not. The solution algorithm accounts for changes in the condition of tension-only members by adjusting the chord length of the load step so that none of the tension-only members changes condition during a step. All condition changes occur at the stations between steps.

The changes in the condition of tension-only members are detected in the method `checkTensionOnlyCondition()` that is called after the trial configuration has been determined in the main method `analyse()` of class Deformation:

- (1) In the first cycle of iteration of load step 0 the tension-only members are stiff.
- (2) The trial configuration of the frame is determined for the stiffness of the frame at the beginning of the load step.

(3) Method `checkTensionOnlyCondition()` is called to traverse the elements of the frame in order to determine the condition of the tension-only members in the trial configuration. Method `checkMemberCondition()` is called for each tension-only member. It reads the stress s in the bar at the beginning of the load step and computes the stress t in the bar in the trial configuration. The change in state of the member is determined according to the following rules (tension positive, ε small):

– The member was stiff in the iteration cycle:

$$s \leq \varepsilon \wedge t \geq -\varepsilon: \quad \text{no change} \quad (8.9)$$

$$s \leq \varepsilon \wedge t < -\varepsilon: \quad \text{condition has changed}$$

$$s > \varepsilon \wedge t \geq \varepsilon: \quad \text{no change}$$

$$s > \varepsilon \wedge t < \varepsilon: \quad \text{compute new segment length}$$

– The member was flexible in the iteration cycle:

$$s \leq -\varepsilon \wedge t \leq -\varepsilon: \quad \text{no change} \quad (8.10)$$

$$s \leq -\varepsilon \wedge t > -\varepsilon: \quad \text{compute new segment length}$$

$$s > \varepsilon \wedge t \leq \varepsilon: \quad \text{no change}$$

$$s > \varepsilon \wedge t > \varepsilon: \quad \text{condition has changed}$$

The new chord length is computed so that the member changes condition at the end of the load step with the new chord length. The new chord length is recorded only if it is smaller than the current chord length.

(4) Method `setTensionOnlyCondition()` is called to make a second traversal of the members of the frame. Method `setMemberCondition()` is called for each tension-only member to determine whether the member is stiff or not in the next cycle of iteration of the tension-only analysis. This stiffness depends on the member stiffness in the preceding cycle of iteration, as well as the stress in the member at the beginning and at the end of the load step:

– The member was stiff in the iteration cycle:

$$s \leq \varepsilon \wedge t \geq -\varepsilon: \quad \text{stiff} \quad (8.11)$$

$$s \leq \varepsilon \wedge t < -\varepsilon: \quad \text{flexible}$$

$$s > \varepsilon \wedge t \geq \varepsilon: \quad \text{stiff}$$

$$s > \varepsilon \wedge t < \varepsilon: \quad \text{flexible}$$

– The member was flexible in the iteration cycle:

$$s \leq -\varepsilon \wedge t > -\varepsilon: \quad \text{stiff} \quad (8.12)$$

$$s \leq -\varepsilon \wedge t \leq -\varepsilon: \quad \text{flexible}$$

$$s > -\varepsilon \wedge t > \varepsilon: \quad \text{stiff}$$

$$s > -\varepsilon \wedge t \leq \varepsilon: \quad \text{flexible}$$

(5) If the condition of at least one of the tension-only members of the frame has changed during the iteration cycle, the computations in (2) to (4) are repeated with the revised chord length. If none of the tension-only members change condition, the tension-only analysis is terminated and the step-wise solution procedure is resumed. The full chord length is used for the first iteration in the next step.

8.6 GRAPHICAL USER INTERFACE

8.6.1 Functionality of the Interface

Figure 8.16 shows a screen shot of a user interface for nonlinear frame analysis. The components and functionality of the interface are described in this section. Examples of nonlinear analyses of plane frames by means of the interface are presented in chapter 9.

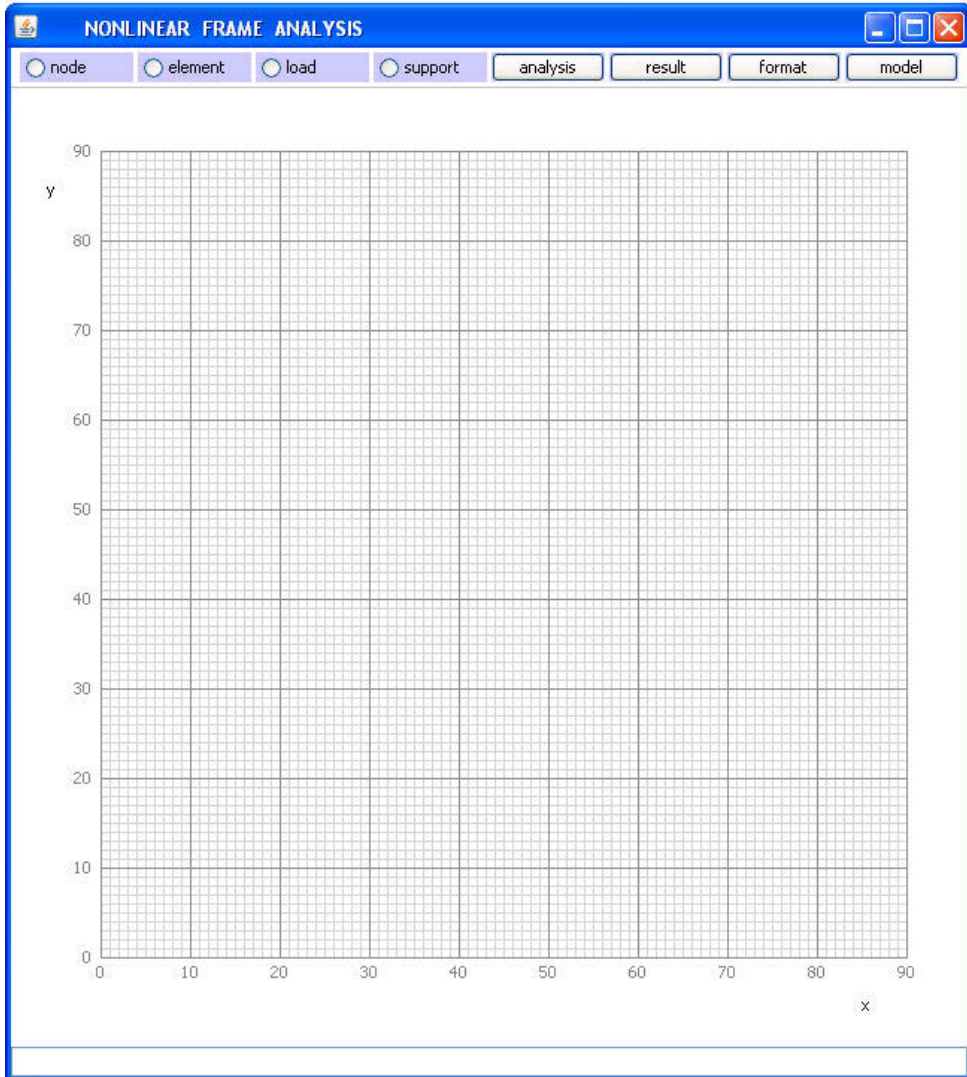


Figure 8.16 Graphical user interface for nonlinear frame analysis

The content pane of the window is decomposed into a control at its upper border, a message field at its lower border and a graphics panel that fills the space between the control and the message field. The control contains buttons with which the main functions of the interface are controlled. The message field is used to display text messages of the software to the user. The graphics panel is used to display the frame and to present its behaviour. As the analysis proceeds, additional editor panels for the specification of the frame and for work flow control, as well as panels for the display of the behaviour of the frame, are automatically added to the graphics panel.

If one of the radio buttons for nodes, elements, loads or supports is selected, the mouse actions in the graphics panel apply only to objects of the selected type. These actions can be the definition of a new component, the modification or deletion of an existing component, the query of user-specified attributes of an existing component, or the display of computed behavioural variables of the component.

The analysis button calls the analysis editor, which is used to set the parameters of the nonlinear analysis, for instance the number of steps in the solution procedure. The analysis is performed when the editor is closed. The computed behaviour is stored in the data structure of the application. The stored values are used in the presentation of the behaviour of the frame, which is triggered with the result button.

The result button calls the result editor, which is used to choose the component (node, member or complete frame) and the behavioural variables (displacement, reaction, axial force, or bending moment) that are to be displayed. The behaviour of nodes and members is presented for a selected range of load steps. The behaviour of the complete frame is presented for a specific configuration. The user can traverse the configurations of the solution with a selected step increment. He can also scale the variables in the presentation.

The format button calls the format editor. This editor is used to set the visibility of nodes, elements, loads and supports in the display. The user can also choose whether the stations of the load path are marked when the history of a variable is displayed in a diagram. The origin and the scale of the presentation of the frame in the graphics panel can be changed with this editor.

The model button calls the model editor. This editor is used to define a new frame. It can be used to store the current frame in a named file. It can also be used to replace the current frame with the frame that is stored in a named frame. The model editor offers a set of prefabricated test models for different aspects of the nonlinear behaviour of trusses and frames.

The interface in figure 8.16 is developed with the application programming interface API of the Java platform. The main packages used are `java.awt` and `java.swing` with their extensions. Mouse and key events on the display are treated with the Java listener concept. The keyboard input is controlled with a key event dispatcher. The rules of Java for graphical containers, visibility and component painting are utilised throughout the construction and operation of the interface.

8.6.2 Structure of the Interface

The class structure of the graphical user interface for nonlinear frame analysis is shown in figure 8.17. Class Session contains the method main() of the application, as well as the roots of the three main object graphs of the application: Frame for the model of the structural frame that is being analysed, Display for the presentation of this frame in the graphical user interface, and Base for the management of the named objects of the application.

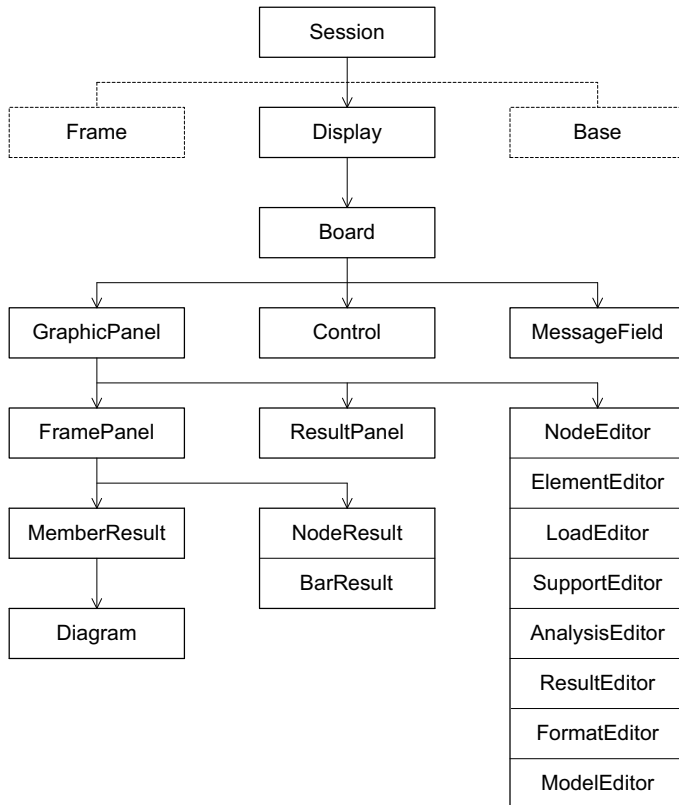


Figure 8.17 Class structure of the graphical user interface

Class Display is an extension of class JFrame. Its content pane is filled with an object of class Board which is the container for the objects of class GraphicPanel and MessageField, and sets their bounds when the components are painted. Each object of class GraphicPanel has the following set of initially invisible attributes:

- a complete set of editors
- one object each of classes FramePanel and ResultPanel
- one object each of classes NodeResult, BarResult and MemberResult
- three objects of class Diagram for an axial force diagram, a shear force diagram and a bending moment diagram

When the user calls an editor by selecting an existing component in the graphics panel or by pressing a control button, the application locates the editor near the selected component or button and makes it visible. When the user closes the editor, the application makes the editor invisible.

If the user requests the history of a node variable such as a displacement, rotation, force reaction or moment reaction in the result editor, the ResultPanel is set visible and the requested diagram is painted on this object. When the user closes the panel, the application makes the panel invisible.

If the user requests the history of an element variable such as the axial force, shear force or bending moment in a member, the ResultPanel is set visible and the requested diagram is painted on this object. When the user closes the panel, the application makes the panel invisible.

If the user requests the display of the displacements, axial forces, shear forces or bending moments for a specific configuration, the FramePanel is set visible and the elements of the frame, as well as the variation of the specified variable along the axes of the members, are displayed. When the user returns from the FramePanel to the GraphicPanel, the application makes the FramePanel invisible.

If the user picks a node in the reference configuration of the frame on the FramePanel, the application locates the NodeResult panel near the node and makes it visible. The panel shows the numerical values of the displacements and reactions at the node. If the user closes the NodeResult panel, the application makes the panel invisible.

If the user picks a bar in the reference configuration of the frame on the FramePanel, the application locates the BarResult panel near the bar and makes it visible. The panel shows the numerical values of the strain, stress and axial force of the bar.

If the user picks a member in the reference configuration of the frame on the FramePanel, the application locates the MemberResult panel near the member and makes it visible. The panel is subdivided into three zones. The zones show the variation of the axial force, shear force and bending moment along the axis of the member and are painted with one object each of class Diagram. If the user closes the member result panel, the graphics panel makes the result panel invisible.

8.6.3 Coordinate Systems

The implementation of the graphical user interface is based on three coordinate systems: the standard Java pixel coordinate system, the millimetre coordinate system of the graphics panel or the frame panel, and the user coordinate system of the frame. Each coordinate system consists of the location of its origin and the base vectors of its coordinate space. The user is concerned only with the user coordinate system. The painting routines are concerned only with the Java pixel coordinates. The millimetre system is the link between the user and Java coordinate systems. Transformations between these coordinate systems occur throughout the definition and operation of the graphical user interface.

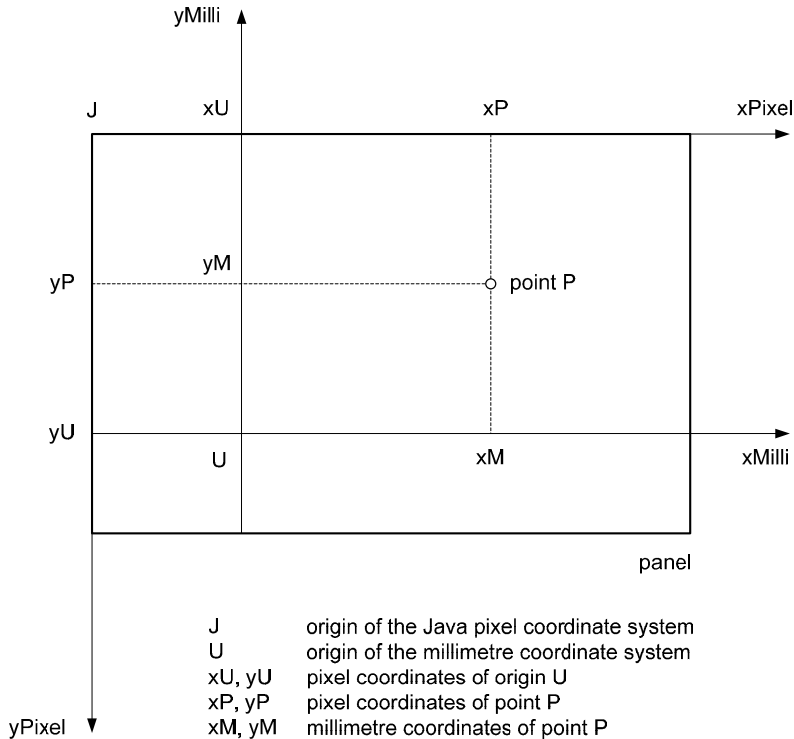


Figure 8.18 Coordinate systems of the graphical user interface

It is assumed that the following parameters for the transformation between the millimetre and pixel coordinate systems are known:

- milliToPixel number of pixels per millimetre
- xOriginPixel pixel x coordinate of the origin of the millimetre coordinate system
- yOriginPixel pixel y coordinate of the origin of the millimetre coordinate system

The transformation of the coordinates (xMilli, yMilli) of a point P in the millimetre coordinate system to its coordinates (xPixel, yPixel) in the pixel coordinate system is given by:

$$xPixel = xOriginPixel + xMilli * milliToPixel \quad (8.13)$$

$$yPixel = yOriginPixel - yMilli * milliToPixel \quad (8.14)$$

The inverse transformation is given by:

$$xMilli = (xPixel - xOriginPixel) / milliToPixel \quad (8.15)$$

$$yMilli = (-yPixel + yOriginPixel) / milliToPixel \quad (8.16)$$

The origin of the user coordinate system is mapped to the origin of the millimetre coordinate system. It is assumed that the following parameter is known for the mapping of the user coordinates to the millimetre coordinate system:

- userToMilli number of millimetres per user unit

The transformation of the coordinates (xUser,yUser) of point P in the user coordinate system to its coordinates (xMilli,yMilli) in the millimetre coordinate system is:

$$xMilli = xUser * userToMilli \quad (8.17)$$

$$yMilli = yUser * userToMilli \quad (8.18)$$

The transformation from user to pixel coordinates is obtained by substitution of (8.17) and (8.18) into (8.13) and (8.14):

$$xPixel = xOriginPixel + xUser * userToPixel \quad (8.19)$$

$$yPixel = yOriginPixel - yUser * userToPixel \quad (8.20)$$

$$userToPixel = userToMilli * MilliToPixel \quad (8.21)$$

The inverse transformation is given by:

$$xUser = (xPixel - xOriginPixel) / userToPixel \quad (8.22)$$

$$yUser = (-yPixel + yOriginPixel) / userToPixel \quad (8.23)$$

8.6.4 Coordinate Grid

It is assumed in the transformations that a millimetre is exactly equal to an integer number of pixels. This is usually not the case. The millimetre on the display is therefore not an exact unit of length. This convention is necessary to achieve a regular millimetre grid (meshes of equal size on the display) on the graphics, result and frame panels.

The grid in figure 8.19 consists of meshes bounded by fine lines. The meshes are grouped in blocks that are bounded by heavy lines. The following conventions are introduced for the grid:

- (a) The mesh size is one by one millimetres.
- (b) There are 10 meshes per block in each direction.
- (c) The number of blocks in the x and y directions depend on the panel size. They are chosen so that the margin is exact and the minimum rim size is observed.
- (d) The user coordinates of the bottom left corner of the grid are specified by the user. The default values are (0.0,0.0). The block size in user units is determined by the parameter userToPixel in (8.21).

The conventions are implemented with symbolic constants MESHES_PER_BLOCK, MARGIN_MILLI and RIM_MILLI. The parameters of the grid are:

$$\text{widthPixel} = \text{width of panel} \quad (8.24)$$

$$\text{heightPixel} = \text{height of panel}$$

$$\text{blockSizePixel} = \text{meshesPerBlock} * \text{milliToPixel}$$

$$\text{blockSizeUser} = \text{blockSizePixel} * \text{pixelToUser}$$

$$\text{marginPixel} = \text{MARGIN_MILLI} * \text{milliToPixel}$$

$$\text{rimPixel} = \text{RIM_MILLI} * \text{milliToPixel}$$

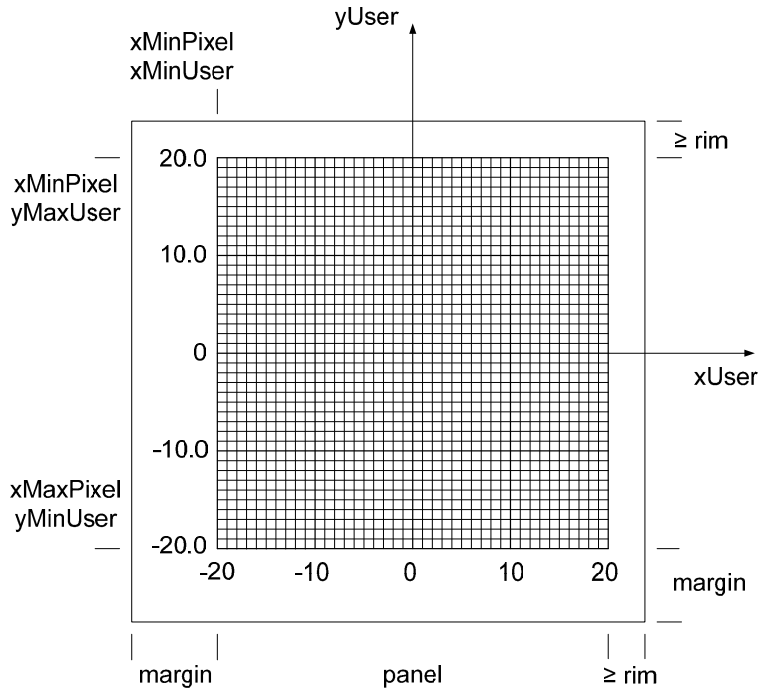


Figure 8.19 Layout of the coordinate grid

$$\begin{aligned}
 x\text{NumberOfBlocks} &= (\text{widthPixel} - \text{marginPixel} - \text{rimPixel}) / \text{blockSizePixel} \\
 y\text{NumberOfBlocks} &= (\text{widthHeight} - \text{marginPixel} - \text{rimPixel}) / \text{blockSizePixel} \\
 x\text{MinPixel} &= \text{marginPixel} & (8.25) \\
 x\text{MaxPixel} &= \text{marginPixel} + x\text{NumberOfBlocks} * \text{blockSizePixel} \\
 y\text{MaxPixel} &= \text{heightPixel} - \text{marginPixel} \\
 y\text{MinPixel} &= y\text{MaxPixel} - y\text{NumberOfBlocks} * \text{blockSizePixel} \\
 x\text{OriginPixel} &= x\text{MinPixel} + x\text{MinUser} * \text{userToPixel} \\
 y\text{OriginPixel} &= y\text{MaxPixel} - y\text{MinUser} * \text{userToPixel}
 \end{aligned}$$

The number of blocks in the x and y directions is rounded by discarding the fraction. The grid thus always consists of complete blocks. The unused space increases the width of the rim.

8.6.5 Paint Algorithm for the Graphics Panel

The graphics panel is painted when the display is constructed and repainted after the graphics operations of a transaction. The panel first paints itself according to the flow diagram in figure 8.20 and then follows the Java convention to paint the visible graphical components it contains. The components are called to paint themselves. The nodes and elements are painted in their reference location.

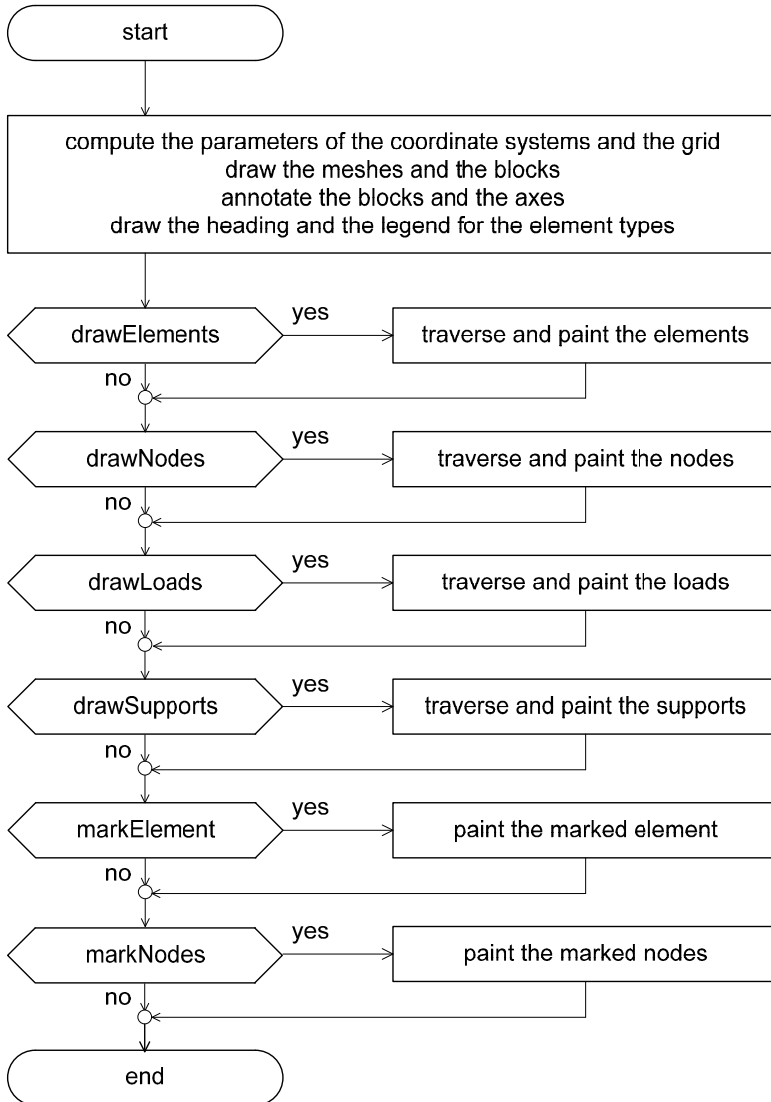
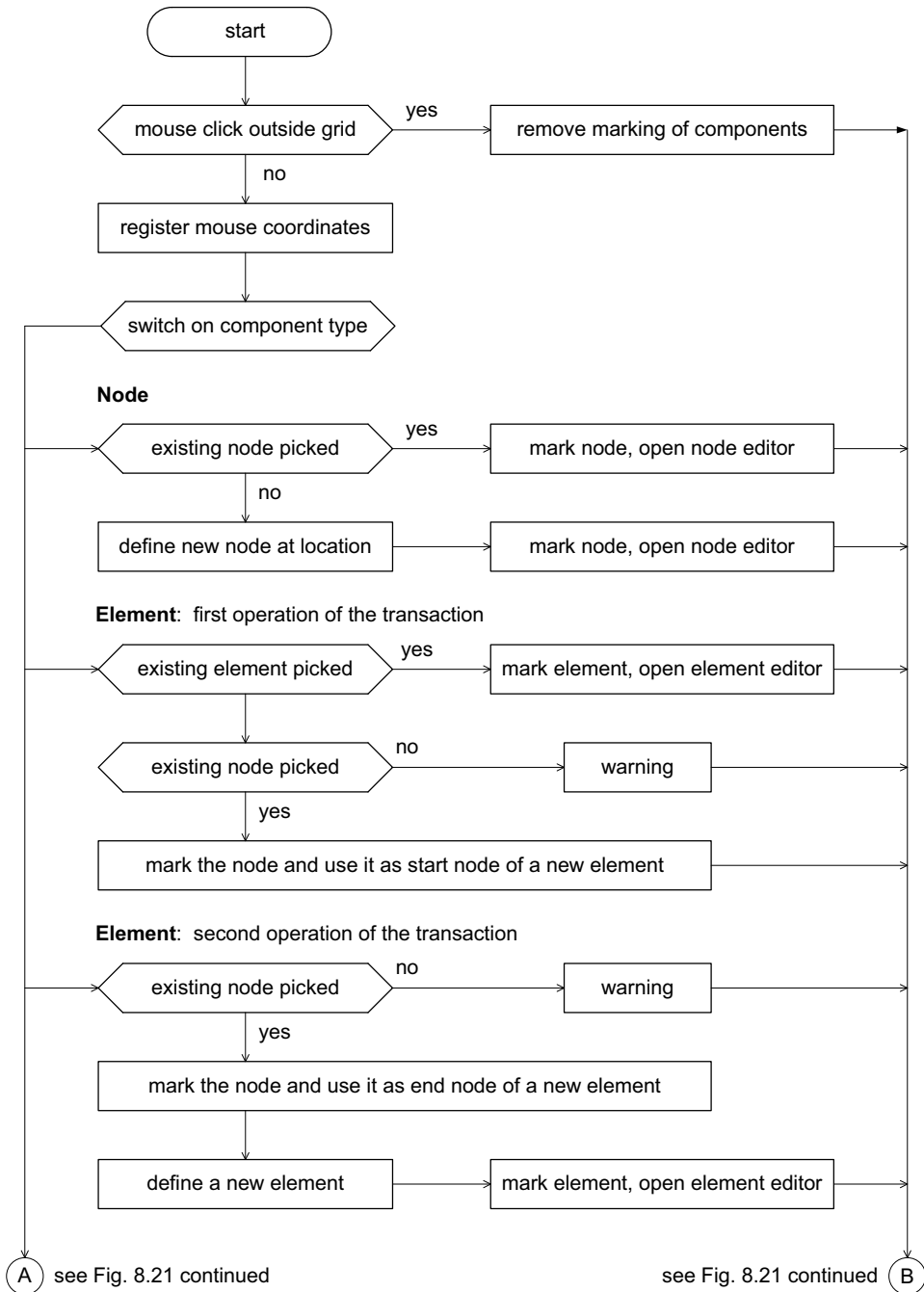


Figure 8.20 Flow diagram of the paint algorithm for the graphics panel

8.6.6 Action Algorithm for the Graphics Panel

The action algorithm, whose flow diagram is shown in figure 8.21, is triggered when the mouse is clicked in the graphics panel. The algorithm is implemented in method `mouseClicked` of class `GraphicPanel`.



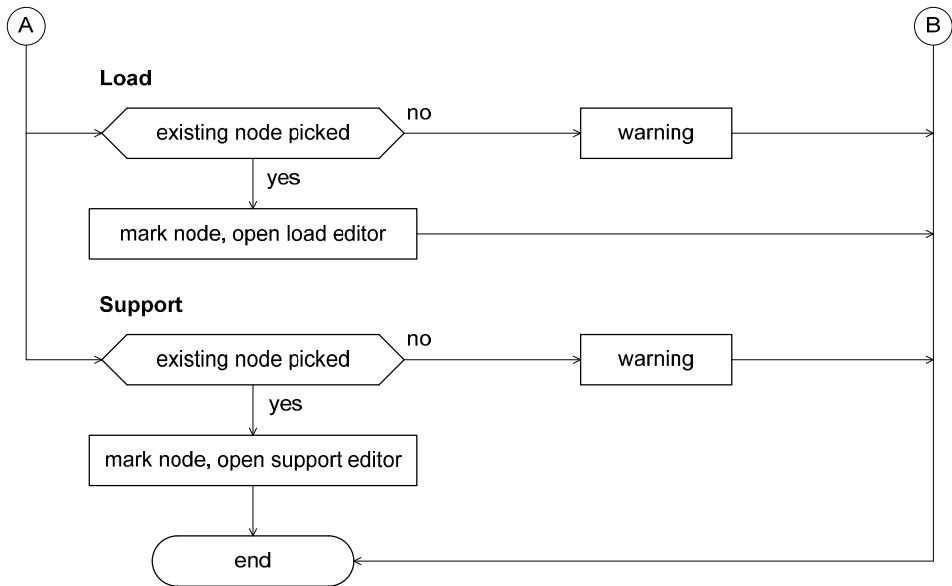


Figure 8.21 Flow diagram of the action algorithm for the graphics panel

The algorithm switches on the component type the user has activated in the control panel of the display. If the user selects an existing component, the editor for that component is displayed near the component and can be used to change the attributes of the component. If the mouse is clicked on a location of the grid which does not lie inside the catch zone of a component of the active type, the algorithm attempts to construct a new component of the type according to the following rules:

(a) **Node:** The location of the mouse is transformed to the user coordinate system and a new node with these coordinates is constructed. The name of the node is assigned by the data base. The marked node is painted and the node editor is opened next to the node. It can be used to specify or modify attributes of the node.

(b) **Element:** The transaction for the construction of a new element consists of two operations: the definition of the start node and the definition of the end node. The user must select existing nodes on the display. The nodes selected in the first and in the second operation are treated as start node and end node of the element respectively. After the second node has been selected, the element is constructed with the selected end nodes. The marked element with marked nodes is painted and the editor is opened next to the element. It can be used to specify the type and the attributes of the element, including the element load intensities.

(c) **Load:** In order to define a node load, the node is selected with the mouse on the graphics panel. The load editor is opened next to the node. It is used to specify the attributes of the load.

(d) **Support:** In order to define a support, the node is selected with the mouse on the graphics panel. The support editor is opened next to the node. It is used to specify the attributes of the support.

8.6.7 Algorithm for the Result Panel

The result panel shows the load factor on the frame (y axis) as a function of one of the following behavioural variables of a node or element (x axis):

- displacement or rotation of a node
- force or moment reaction at a node
- axial force in an element
- shear force in a member
- bending moment in a member

Reactions are only shown if the node is restrained. Displacements and force reactions at nodes have two components that are shown with curves of different colours. If the dimension of a variable group at a node is larger than 1, separate curves are drawn for the individual variables. The number of curves in the diagram can thus vary.

The algorithm for the presentation of node and member results on the result panel is decomposed into methods for the collection of the data from the data base, for the selection of the scales on the x and y axes and for the painting of the panel. The flow of operations is controlled in method `paintComponent()` of the result panel.

(a) **Painting:** The parameters for the coordinate systems and the grid are computed. The x coordinates of the stations on the curves in pixels are computed from the user coordinates and the scale for the x direction is set. The values of the y coordinates at the stations of the curve in user coordinates are also transformed to pixel coordinates and the scale for the y direction is set. The meshes and blocks are painted first. The method then annotates the axes and draws the heading that describes the contents of the diagram. The curves consist of polygons whose segments connect the stations of the curve.

(b) **x Coordinates:** The method switches according to the result type and collects the data for the component that the user has named in the result editor. The node displacement or reaction history is collected from the node object. The axial force history of a bar is collected from the bar object. The history of a stress resultant in a member is collected by calling the member, which collects the displacement history of its nodes and uses it to compute the history of the stress resultant. The scale on the x axis is set for the extreme values of the collected data.

(c) **y Coordinates:** The user specifies the load factor range for which the curves are to be shown by entering the indices of the first and last stations in the result editor. The value of the load factor at the stations is read from the load history. The scale on the y axis is set for the extreme values of the load factor.

(d) **Scale:** The number of user units per block of an axis is chosen so that it is 1.0, 2.0 or 5.0 times a power of 10. The choice is made so that the block size in user units is maximised.

8.6.8 Algorithm for the Frame Panel

The frame panel is used to display the history of the frame as a whole. The user specifies with the result editor whether the instant configuration, the axial force diagram, the shear force diagram or the bending moment diagram is to be shown. He also specifies the indices of the first and the last configuration for which the result is to be displayed, as well as the step increment for successive presentations. Each of the presentations consists of the reference configuration of the frame and the specified type of result.

The frame panel is opened with the initial configuration of the frame as specified in the result editor. When the "next"-button on the frame panel is pressed, the index of the configuration is increased by the step increment. If the incremented index of the configuration exceeds the last index specified in the result panel, the configuration with the last index is displayed. When the "return"-button in the frame panel is pressed, the frame panel is closed and the user can perform the next transaction in the graphics panel.

The user can select a node or member in the reference configuration of the frame on the frame panel by clicking the mouse. That element or node is then marked in the instant configuration of the frame. The results of the analysis for the node or member are displayed in a panel near the component in the instant configuration.

The frame panel is drawn in method `paintComponent()` of class `FramePanel`. It computes the parameters of the coordinate system and the grid. It draws the meshes and the blocks of the grid, as well as the reference configuration of the frame. The method then switches on the type of result that the user has requested in the result editor.

If the user has requested the deformed frame, the elements and the nodes of the frame are traversed and instructed to paint their displaced instant configuration. If the user has requested a diagram of the axial forces, shear forces or bending moments in the frame, the elements are traversed and each element is instructed to paint its own diagram. The computation of the variation of the stress resulting along the axis of the member is implemented in the component classes. Finally, the heading that describes the contents of the diagram, as well as the annotations of the axes, are drawn and the components selected by the user are marked.

User actions in the frame panel are reported to method `mouseClicked()` of class `FramePanel`. If the user clicks the button "next" or "previous", the index of the configuration is modified with the step increment and the panel is redrawn. If he clicks the button "return", the frame panel is closed and the graphics panel is activated.

If the mouse click is inside the grid, the coordinates of the mouse location are stored and transformed to user space. The nodes are traversed. If the mouse is inside the catch zone of a node, that node is selected and marked. The `NodeResult` panel is shown next to the node. If none of the nodes was selected, the members of the frame are traversed. If the mouse is inside the catch zone of a member, that member is selected and marked. The `ElementResult` panel is shown next to the element. The node and element result panels are closed by clicking a button. The next input is expected on the frame panel.

8.6.9 Construction of the Editors

Scope: The application contains a separate editor class for each component class of the frame. In addition, it contains editors for control of the work flow. All editors of the application are extensions of class BasicEditor that provides general services:

- position the editor relative to points and rectangles on the graphics panel
- move the dragged panel
- paint the basic editor frame
- construct editor components such as buttons and text fields
- parse primitive data types
- functions "do" and "cancel" of the editor

Layout: The general layout is the same for all editors of the implementation. It consists of an editor label, buttons D and C and a central panel, as shown in figure 8.22. The editor is constructed so that all input is stored temporarily. If button D for "do" is pressed, the input data are entered in the data base. If button C for "cancel" is pressed, the input data are discarded and the data base remains unchanged. The use of the central panel is determined in the extended editor class.

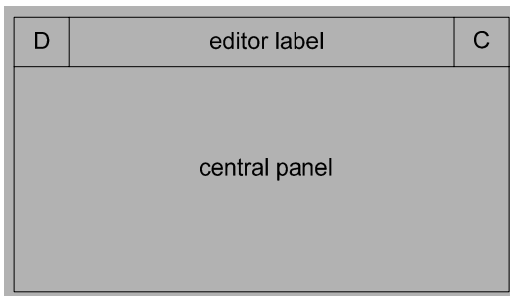


Figure 8.22 Basic layout of an editor

Workflow: The general workflow is the same for all editors of the implementation. The implementation of the methods is specific to the component or function controlled with the editor.

- The constructor of the editor makes the graphical components of the editor, such as buttons and text fields, and adds them to the panel. It sets the bounds of the components and the location of the annotations in millimetres.
- The editor contains a duplicate of every attribute of the component type. If the editor is called for a new component, the attribute duplicates are set to default values. If the user has selected an existing component in the panel, the attribute duplicates are set to the values of the attributes of that component.
- The attribute duplicates are formatted and written to the text fields of the editor panel.
- Method `paintComponent()` transforms the bounds of the components and the location of the annotation of the editor panel to pixels. It draws the annotation on the

panel. By Java convention, the run time system then calls the graphical components that were added to the editor in the constructor. These paint themselves.

– The editor is closed when the button D or C of the editor is pressed. If the input is cancelled with button C, the editor panel is set invisible and the interaction continues on the graphics panel. If the input is confirmed with button D, the text fields of the editor panel are parsed and the values are stored as the attributes of the component. The editor panel is then set invisible and the interaction continues on the graphical panel.

Location: The location at which an editor is opened on the graphics panel can be specified in different ways. After the editor has been opened, it can be dragged on the board with the mouse. The alternatives are:

- (a) The editor is centred on the panel.
- (b) The top left corner of the editor panel is located at a point whose coordinates are specified in the millimetre space of the panel.
- (c) The editor is located relative to a reference point on the panel.
- (d) The editor is located relative to a reference rectangle on the panel.

If a reference point is specified, the cases 1 to 12 in figure 8.23 are considered. The symbols are defined as follows:

- R reference point
- s half-height of the editor
- t half-width of the editor
- g gap between the reference point and an edge of the editor
- u distance from an edge of the editor to an edge of the panel

A case is considered possible if an editor that is located as shown in the diagram fits inside the graphics panel. The location of the reference point relative to the edges of the panel determines which of the cases are possible. The possible case with the lowest index is chosen.

If a reference rectangle is specified, the editor does not overlay the rectangle except for case 13 (editor at the centre of the panel). The midpoints A-D of the sides of the rectangle, as shown in figure 8.23, are chosen as reference points for the test cases 1 to 12:

- case 1 to case 3 : reference point A
- case 4 to case 6 : reference point B
- case 7 to case 9 : reference point C
- case 10 to case 12 : reference point D

Otherwise the choice is analogous to that for a specified reference point.

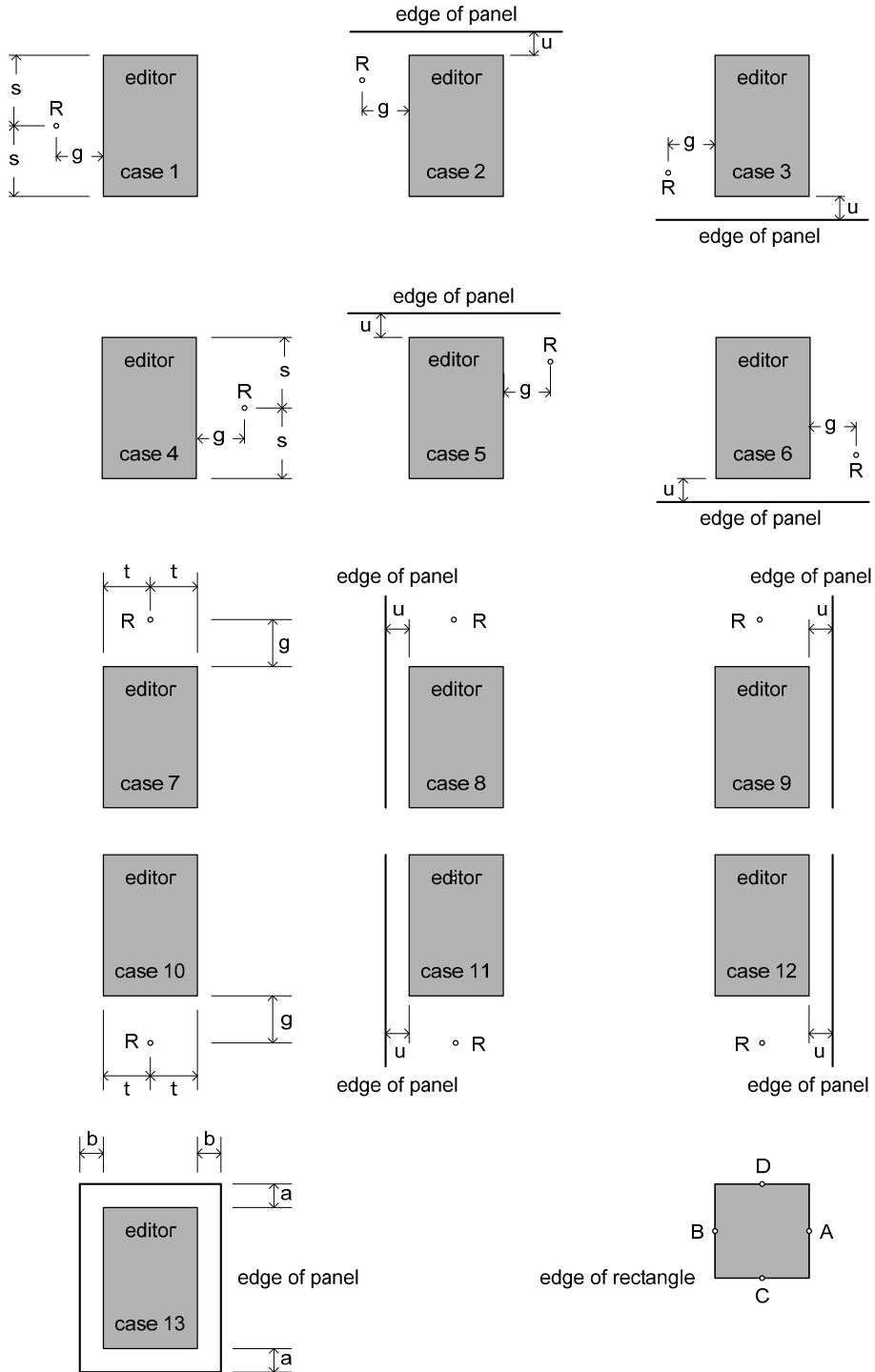


Figure 8.23 Locations of an editor on a panel

8.6.10 Screen Shots of Editors

The following screen shots illustrate the editors of the application:

D	Node Editor		C
node name			B
number			4
x1User			74.000
x2User			52.000
theta	deg		0.000
dimension	y1		1
	y2		1
	rot		0

D	Load Editor		C
node name			B
force	y1		
force	y2		-100.000
moment	m		
local index	y1		0
	y2		0
	m		0

D	Element Editor		C
<input checked="" type="radio"/> member	<input type="radio"/> bar	<input type="radio"/> cable	
element name			bar1
area			6.00
moment of inertia			4.58
E-modulus			20000.0
start load	p1		
	p2		
end load	p1		
	p2		
start node			B
local index	y1		0
	y2		0
	rot		0
end node			A
local index	y1		0
	y2		0
	rot		0

D	Support Editor		C
node name			A
displacement	y1		0.000
displacement	y2		0.000
rotation	a		
local index	y1		0
	y2		0
	a		0

D	Format Editor	C
<input checked="" type="radio"/>	show nodes	
<input checked="" type="radio"/>	show elements	
<input checked="" type="radio"/>	show loads	
<input checked="" type="radio"/>	show supports	
<input type="radio"/>	show states	
	user unit to millimeter	0.500
	x1Pivot (user space)	0.000
	x2Pivot (user sapce)	0.000

D	Analysis Editor	C
	number of steps	82
	initial load factor	0.2000
	initial slope	5.0000
	error test factor	0.0000

D	Result Editor				C
node	<input checked="" type="radio"/> displacement	<input type="radio"/> rotation	<input type="radio"/> reaction f	<input type="radio"/> reaction m	
member	<input type="radio"/> displacement	<input type="radio"/> axial force	<input type="radio"/> shear force	<input type="radio"/> bending mom.	
frame	<input type="radio"/> displacement	<input type="radio"/> axial force	<input type="radio"/> shear force	<input type="radio"/> bending mom.	
component name					
coordinate on axis					0.000
first state number					0
last state number					3
state number increment					1
scale factor					0.000

Figure 8.24 Editors of the graphical user interface

8.6.11 Method Structure of the Graphical Interface

The methods for the graphical interface are arranged in the classes, as follows. Only the main methods are listed. Methods for data management are not listed.

(a) Class GraphicPanel

double	xPixelToMilli (xPixel)	transformations
double	yPixelToMilli (yPixel)	
double	sPixelToMilli (sPixel)	
double	xPixelToUser (xPixel)	
double	yPixelToUser (yPixel)	
double	sPixelToUser (sPixel)	
int	xMilliToPixel (xMilli)	
int	yMilliToPixel (yMilli)	
int	sMilliToPixel (sMilli)	
double	xMilliToUser (xMilli)	
double	yMilliToUser (yMilli)	
double	sMilliToUser (sMilli)	
int	xUserToPixel (xUser)	
int	yUserToPixel (yUser)	
int	sUserToPixel (sUser)	
double	xUserToMilli (xUser)	
double	yUserToMilli (yUser)	
double	sUserToMilli (sUser)	
int	xPixelToPixelGrid (xPixel)	
int	yPixelToPixelGrid (yPixel)	
double	xPixelToMilliGrid (xPixel)	
double	yPixelToMilliGrid (yPixel)	
double	xPixelToUserGrid (xPixel)	
double	yPixelToUserGrid (yPixel)	
void	paintComponent (Graphics)	painting
void	drawMarkedNode (Graphics,xMark,yMark)	
void	drawMarkedElement (Graphics)	
void	drawText (Graphics,decimals,dValue,x,y,code)	
void	showMemberColors (Graphics)	
void	openResultPanel ()	components of the panel
void	openTrussPanel ()	
void	locateBarEditor ()	

ResultEditor getResultEditor ()
AnalysisEditor getAnalysisEditor ()
FormatEditor getFormatEditor ()
ModelEditor getModelEditor ()
ResultPanel getResultPanel ()
FramePanel getFramePanel ()

void mouseClicked (MouseEvent)
boolean dispatchKeyEvent (KeyEvent)

listeners

(b) **Class Node**

boolean containsPointUser (x1Test,x2Test)
void drawReferenceNode (Graphics)
void drawDisplacedNode (Graphics,state)

(c) **Class Bar**

boolean containsPointUser (x1Point,x2Point)
void drawReferenceElement (Graphics)
void drawDisplacedElement (Graphics,state)
void drawAxialForceDiagram (Graphics,state)

(d) **Class Member**

boolean containsPointUser (x1Point,x2Point)
void drawReferenceElement (Graphics)
void drawDisplacedElement (Graphics,state)

(e) **Class Load**

void drawLoad (Graphics)

(f) **Class Support**

void drawSupport (Graphics)

(g) **Class ResultPanel**

double xPixelToUser (xPixel)
double yPixelToUser (yPixel)
int xUserToPixel (xUser)
int yUserToPixel (yUser)

boolean computeXUserToMilli ()
boolean computeYUserToMilli ()
boolean setUserLimits ()
double rounded (pivotValue,increment)

void paintComponent (Graphics)
void centerString (Graphics,string,xA,xE,y)
void showComponentColors (Graphics)
void drawText (Graphics,value,x,y,code)
void mouseClicked (mouseEvent)

(h) **Class FramePanel**

void openPanel ()
void setUserToMilli (value)
void setX1MinUser (value)
void setX2MinUser (value)

double xPixelToMilli (xPixel)
double yPixelToMilli (yPixel)
double sPixelToMilli (sPixel)

double xPixelToUser (xPixel)
double yPixelToUser (yPixel)
double sPixelToUser (sPixel)

int xUserToPixel (xUser)
int yUserToPixel (yUser)
int sUserToPixel (sUser)

double xUserToMilli (xUser)
double yUserToMilli (yUser)
double sUserToMilli (sUser)

int xPixelToPixelGrid (xPixel)
int yPixelToPixelGrid (yPixel)
double xPixelToUserGrid (xPixel)
double yPixelToUserGrid (yPixel)

void paintComponent (Graphics)
void drawMarkedNode (Graphics,xMark,yMark)
void drawMarkedElement (Graphics)
void centerString (Graphics,string,xA,xE,y)
void drawText (Graphics,decimals,dValue,x,y,code)
void showElementColors (Graphics)

void mouseClicked (mouseEvent)
void computeDisplacedLocation (node)
void locateBarResult ()
void locateMemberResult ()

(i) **Class MemberResult**

void openEditor (member,state)
void closeEditor (isCancelled)
void paintComponent (Graphics)

(j) **Class Diagram**

void setCoordinates (coor[])
int userToPixel (value)
void setScale ()

void paintComponent (Graphics)
void drawGrid (Graphics)
void drawCurve (Graphics)

(k) **Class BasicEditor**

void paintComponent (Graphics)
void closeEditor (isCancelled)

double getXStartPanelMilli ()
double getYStartPanelMilli ()
double getXEndPanelMilli ()
double getYEndPanelMilli ()

void setLabel (String value)
String getLabel ()
int parseInt (text)
double parseDouble (text)

void locateAtCenter()
void locateNearPoint (xReference,yReference)
void locateNearRectangle (xReference,yReference,
wReference,hReference)

void mousePressed (MouseEvent)
void mouseDragged (MouseEvent)

JLabel makeLabel (text)
JTextField makeTextField ()
JButton makeButton (label)
JToggleButton makeToggleButton (label)
JRadioButton makeRadioButton (label)
JComboBox makeComboBox (items[])
JCheckBox makeCheckBox (label)

9 EXAMPLES OF GEOMETRICALLY NONLINEAR ANALYSIS

9.1 SCOPE AND PURPOSE OF THE INVESTIGATION

Linear structural analysis is a boundary value problem. The displacements and reactions due to specified loads and prescribed displacements are computed by solving a linear system of algebraic equations. If a direct method is used for the solution of the linear equations, the number of operations required to perform the analysis and the storage space for the data can be predicted reliably. The efficiency of the algorithms is therefore known a priori. The unbalanced forces in the loaded structure can be determined and used to compute error bounds for the behavioural variables that permit a reliable evaluation of the accuracy of the analysis.

Nonlinear structural analysis is an initial value problem. The displacements and reactions for specified loads and prescribed displacement patterns are determined stepwise. The step size is selected empirically. The solution is iterated in each step of the solution procedure. It is difficult to formulate general criteria for suitable step sizes and to predict the number of iterations required to achieve a specified level of accuracy in the behavioural variables. The suitable step size and number of iterations generally vary along the load path. Efficiency and accuracy of the algorithms are closely related. A larger step size may reduce the number of required operations and the size of the required storage space, but may at the same time increase the error in the solution.

The singular points on the load paths of the structures and the difficulties encountered in the continuation of the load path beyond singular points, particularly beyond bifurcation points, add to the complexity of the evaluation of the efficiency and accuracy of nonlinear methods of analysis. The type of the singular points, which is generally not known a priori, influences the number of operations and the storage space requirements significantly. In addition, the type of the singularity can affect the robustness of the solution procedures. To be useful in engineering practice, the algorithms must identify singular configurations of a structure reliably, even if the displacements prior to the singular configurations are small.

The quality of a linear algorithm can usually be deduced from the derivation of the theory. The same is not always true for a nonlinear algorithm. Even if the derivation of the theory is formally correct from the analytical point of view, the algorithm may yield wrong numerical results. The errors are due to unfavourable numerical properties of the governing equations for the behavioural variables, particularly near singular points and on continuations.

This chapter presents three examples of geometrically nonlinear structural analysis:

- (a) Deformation and snap-through instability of a shallow 2-bar truss
- (b) Deformation and bifurcation instability of a steep 2-bar truss
- (c) Large displacements of a cantilever beam

The examples are used to gain insight into the accuracy, efficiency and reliability of the tools for nonlinear deformation and stability analysis that are presented in this book. The exact analytical solutions are derived for all three examples. The numerical solutions that are obtained with the methods in this book are compared to the exact solutions to obtain a general impression of the quality of the new tools.

9.2 2-BAR TRUSS WITH TURNING POINT

Purpose: The geometrically nonlinear behaviour of a symmetric shallow 2-bar truss is investigated with the numerical algorithms described in chapter 8. The exact analytical solutions for the displacements at the apex and the reactions at the supports are derived in section 2.1. The numerical results are compared to the analytical results in order to determine the following effects:

- influence of the initial step size on the accuracy of the displacements and reactions
- influence of the error test factor, which controls the termination of the iteration in the load steps, on the accuracy of the displacements and reactions
- influence of the initial step size and the error test factor on the accuracy of the load and displacement at the turning point
- reliability of the continuation of the load path beyond the turning point

Structural model: The structural model for the numerical analysis of the symmetrical shallow 2-bar truss is shown in figure 9.1. The base width of the truss is 4.000 m, the height of the apex above the supports is 0.150 m. The bars can carry axial forces but not bending moments. The area of the bar cross-section is 10 cm^2 . The modulus of elasticity of the bar material is $2.0 \cdot 10^8 \text{ kN/m}^2$. The supports A and C of the truss are pinned. The truss carries a vertical pattern load $P = (0.0, -10.0) \text{ kN}$ at the apex B.

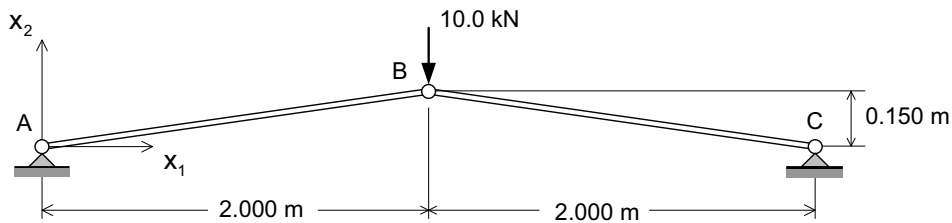


Figure 9.1 Shallow 2-bar truss

Accuracy of the numerical analysis: The step size of the constant arc increment method is determined by the initial load factor increment in step 0, as defined in chapter 6. The iteration in a load step of the deformation analysis is terminated if the norm of the unbalanced forces in the trial state at the end of the step is less than the error limit. The error limit equals the product of the linear force norm with the initial load factor and an error test factor chosen by the user.

The step sequence for the determination of a singular point is terminated if the load factor increment in the step is less than 10^{-3} . The iteration in a step towards the singular point is terminated if the change in the load factor increment in an iteration cycle is less than 10^{-4} . The error limit for the continuation of the load path at a turning point is set equal to the error limit for the deformation analysis.

The truss is analysed for initial load factors varying from 0.20 to 0.80 and for error test factors varying from 10^{-2} to 10^{-5} . Typical curves for the variation of the displacement at the apex B and the reaction at the support A with the load factor are shown in figures 9.2 and 9.3, which correspond to figures 2.4 and 2.5 in section 2.

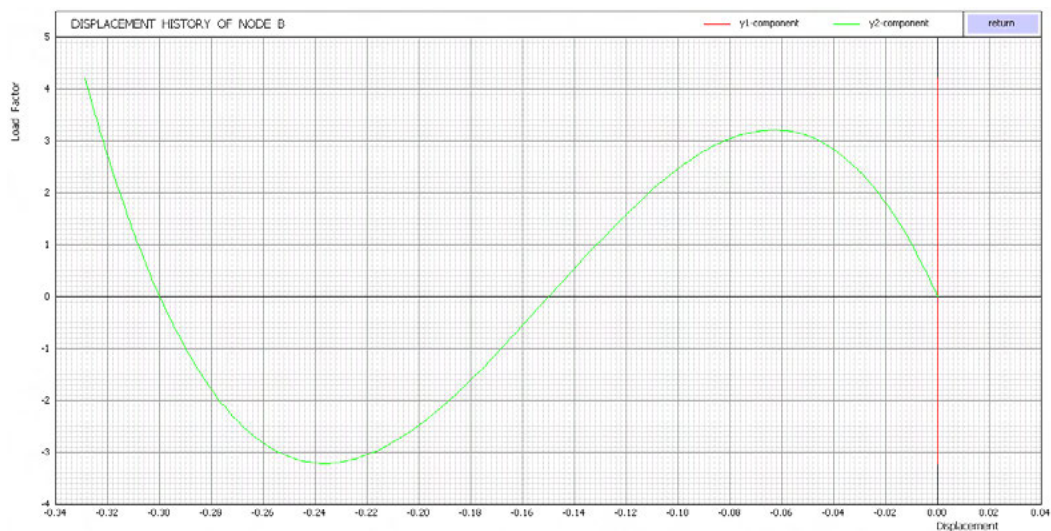


Figure 9.2 Apex displacement of the truss in figure 9.1

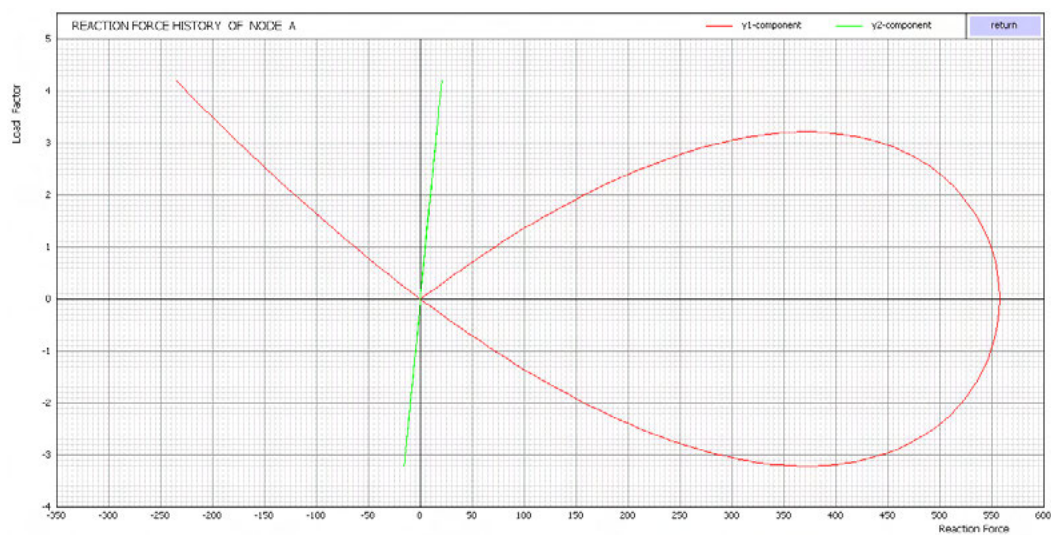


Figure 9.3 Support reaction of the truss in figure 9.1

Table 9.1 Error in the deflection – load curve for $f_e = 10^{-5}$ and LFI = 0.50

d_2	P_2	$P_{2,exact}$	d_2	P_2	$P_{2,exact}$
-0.0047006	-5.0000000	-4.9999532	-0.1935347	22.2375853	22.2376230
-0.0094016	-9.5227726	-9.5227570	-0.2055045	26.7205815	26.7206247
-0.0142628	-13.7137556	-13.7137352	-0.2157549	29.6291749	29.6291622
-0.0193008	-17.5540311	-17.5539996	-0.2248247	31.3508696	31.3509071
-0.0245358	-21.0219088	-21.0218573	-0.2290237	31.8447456	31.8446942
-0.0299917	-24.0922555	-24.0921647	-0.2369659	32.2029996	32.2029996
-0.0356956	-26.7342474	-26.7342521	-0.2366033	32.2038509	32.2038509
-0.0416839	-28.9133453	-28.9133854	-0.2366025	32.2038509	32.2038509
-0.0480034	-30.5871208	-30.5871030	-0.2438817	31.8530040	31.8530220
-0.0547099	-31.7014158	-31.7014845	-0.2506764	30.8589739	30.8590042
-0.0582342	-32.0287752	-32.0287299	-0.2570672	29.2938762	29.2939680
-0.0632523	-32.2037151	-32.2037151	-0.2631175	27.2135058	27.2135954
-0.0633973	-32.2038509	-32.2038509	-0.2688756	24.6621246	24.6621439
-0.0633975	-32.2038509	-32.2038509	-0.2743780	21.6766253	21.6766348
-0.0712745	-31.8163591	-31.8163309	-0.2796535	18.2884765	18.2885601
-0.0799137	-30.5586638	-30.5585885	-0.2847287	14.5228191	14.5228621
-0.0895659	-28.2375122	-28.2375482	-0.2896237	10.4020950	10.4021229
-0.1006482	-24.5479175	-24.5479321	-0.2943556	5.9458086	5.9458270
-0.1139180	-18.9615178	-18.9615274	-0.2989390	1.1710829	1.1710956
-0.1308569	-10.5038535	-10.5038606	-0.3005221	-0.5855415	-0.5855414
-0.1594532	5.2519267	5.2519267	-0.3049245	-5.7671632	-5.7671547
-0.1788112	15.4776120	15.4776290	-0.3092055	-11.2340669	-11.2340610

Table 9.2 Error in the deflection – load curve for $f_e = 10^{-3}$ and LFI = 0.50

d_2	P_2	$P_{2,exact}$	d_2	P_2	$P_{2,exact}$
-0.0047006	-5.000000	-4.999953	-0.2060584	26.905634	26.901391
-0.0093962	-9.517451	-9.517818	-0.2164001	29.785790	29.779542
-0.0142495	-13.702475	-13.702971	-0.2253313	31.418531	31.421104
-0.0192765	-17.535992	-17.536656	-0.2333183	32.127606	32.135259
-0.0244952	-20.996062	-20.996979	-0.2366832	32.203809	32.203809
-0.0299262	-24.057164	-24.058487	-0.2366026	32.203851	32.203851
-0.0355919	-26.689184	-26.691212	-0.2366025	32.203851	32.203851
-0.0415130	-28.855972	-28.859346	-0.2438820	31.853474	31.852986
-0.0476936	-30.513130	-30.519475	-0.2506769	30.859748	30.858890
-0.0544278	-31.670737	-31.667776	-0.2570125	29.306697	29.310079
-0.0615136	-32.177417	-32.180828	-0.2629638	27.266756	27.273935
-0.0633712	-32.203846	-32.203846	-0.2686891	24.750812	24.753820
-0.0633975	-32.203851	-32.203851	-0.2741719	21.796390	21.798426
-0.0712739	-31.817049	-31.816389	-0.2794388	18.435434	18.436772
-0.0799120	-30.560521	-30.558912	-0.2845101	14.694660	14.695600
-0.0893930	-28.280630	-28.286871	-0.2894040	10.596934	10.597620
-0.1003747	-24.648995	-24.650664	-0.2941365	6.162069	6.162585
-0.1134957	-19.154154	-19.155690	-0.2987217	1.407422	1.407821
-0.1301819	-10.860277	-10.861305	-0.3006254	-0.703711	-0.702077
-0.1597767	5.430139	5.430139	-0.3050225	-5.887733	-5.887504
-0.1792233	15.683571	15.681688	-0.3092979	-11.357139	-11.356903
-0.1940294	22.446679	22.443041	-0.3134619	-17.100113	-17.099921

Table 9.3 Error in the deflection – load curve for $f_e = 10^{-2}$ and LFI = 0.50

d_2	P_2	$P_{2,exact}$	d_2	P_2	$P_{2,exact}$
-0.004681	-5.0000	-4.9800	-0.187385	19.5483	19.5577
-0.009552	-9.6841	-9.6593	-0.199762	24.6903	24.7018
-0.014622	-14.0333	-14.0042	-0.210188	28.1531	28.1668
-0.019921	-18.0264	-17.9920	-0.219155	30.3567	30.3749
-0.025488	-21.6388	-21.5970	-0.226667	31.5642	31.5924
-0.031381	-24.8406	-24.7882	-0.230378	31.9649	31.9603
-0.037689	-27.5948	-27.5258	-0.236834	32.2035	32.2035
-0.044569	-29.8531	-29.7551	-0.236603	32.2039	32.2039
-0.051271	-31.2194	-31.2124	-0.236603	32.2039	32.2039
-0.057546	-31.9554	-31.9783	-0.243872	31.8407	31.8539
-0.061915	-32.2328	-32.1896	-0.250656	30.8312	30.8631
-0.063349	-32.2038	-32.2038	-0.256930	29.3249	29.3343
-0.063397	-32.2039	-32.2039	-0.262111	27.5046	27.6015
-0.071260	-31.8333	-31.8177	-0.267520	25.2620	25.3144
-0.079869	-30.6084	-30.5672	-0.272707	22.5925	22.6412
-0.088603	-28.4685	-28.5089	-0.277760	19.5259	19.5655
-0.099198	-25.0780	-25.0862	-0.282663	16.0829	16.1168
-0.111689	-19.9657	-19.9754	-0.287426	12.2835	12.3126
-0.127343	-12.3428	-12.3493	-0.292056	8.1454	8.1709
-0.148060	-1.0806	-1.0819	-0.296559	3.6848	3.7073
-0.150969	0.5403	0.5403	-0.301597	-1.8424	-1.8105
-0.171960	11.9841	11.9863	-0.305866	-6.9489	-6.9323

Table 9.4 Error in the deflection – load curve for $f_e = 10^{-3}$ and LFI = 0.20

d_2	P_2	$P_{2,exact}$	d_2	P_2	$P_{2,exact}$
-0.0036743	-3.951010	-3.949613	-0.1466017	-1.894516	-1.894531
-0.0074482	-7.701991	-7.700436	-0.1606496	5.910236	5.910246
-0.0113291	-11.244420	-11.242676	-0.1761411	14.138101	14.138293
-0.0153269	-14.568862	-14.566886	-0.1885667	20.089630	20.089910
-0.0194534	-17.664746	-17.662481	-0.1989321	24.388916	24.389237
-0.0237230	-20.520126	-20.517491	-0.2079111	27.486945	27.487330
-0.0281535	-23.121355	-23.118232	-0.2158945	29.661523	29.662044
-0.0327677	-25.452631	-25.448836	-0.2231089	31.091176	31.092026
-0.0375080	-27.456763	-27.456652	-0.2296362	31.897748	31.899666
-0.0423403	-29.116401	-29.116534	-0.2345360	32.174753	32.176564
-0.0473771	-30.448640	-30.448884	-0.2366025	32.203851	32.203851
-0.0526318	-31.425900	-31.426441	-0.2424661	31.975402	31.977412
-0.0580546	-32.014457	-32.016210	-0.2478657	31.350431	31.351362
-0.0633691	-32.203846	-32.203846	-0.2530972	30.339821	30.340234
-0.0664677	-32.144927	-32.143854	-0.2581128	28.976820	28.977026
-0.0728562	-31.647157	-31.648592	-0.2629259	27.288628	27.288746
-0.0796829	-30.601809	-30.602741	-0.2674723	25.332705	25.336918
-0.0872010	-28.888265	-28.888804	-0.2718143	23.132654	23.135865
-0.0955624	-26.364916	-26.365289	-0.2760297	20.669512	20.672243
-0.1050548	-22.818764	-22.819056	-0.2801265	17.956543	17.958904
-0.1161694	-17.910134	-17.910375	-0.2841119	15.005667	15.007734
-0.1297377	-11.095641	-11.095804	-0.2879928	11.827636	11.829465

Table 9.5 Error in the deflection – load curve for $f_e = 10^{-3}$ and LFI = 0.80

d_2	P_2	$P_{2,exact}$	d_2	P_2	$P_{2,exact}$
-0.0077629	-8.000000	-7.999458	-0.2218347	30.874172	30.879030
-0.0155048	-14.704563	-14.707314	-0.2288770	31.838398	31.830867
-0.0236894	-20.491710	-20.496365	-0.2369643	32.203007	32.203007
-0.0323853	-25.261892	-25.270022	-0.2366033	32.203851	32.203851
-0.0416439	-28.884407	-28.900773	-0.2366025	32.203851	32.203851
-0.0520261	-31.344658	-31.334562	-0.2480133	31.331510	31.328396
-0.0563917	-31.872545	-31.879209	-0.2583082	28.920885	28.915859
-0.0631252	-32.203373	-32.203373	-0.2676666	25.237057	25.245557
-0.0633970	-32.203851	-32.203851	-0.2763080	20.481967	20.498006
-0.0633975	-32.203851	-32.203851	-0.2844878	14.706646	14.713181
-0.0763495	-31.181705	-31.177236	-0.2922204	8.010891	8.015228
-0.0916376	-27.638804	-27.625626	-0.2995699	0.475014	0.477798
-0.1109393	-20.305816	-20.310138	-0.3002124	-0.237507	-0.237483
-0.1390535	-6.071606	-6.073320	-0.3071947	-8.615027	-8.612944
-0.1554498	3.035803	3.035803	-0.3138857	-17.709184	-17.707838
-0.1859158	18.888470	18.884835	-0.3203146	-27.475073	-27.474056
-0.2062487	26.969757	26.962906	-0.3265085	-37.873929	-37.873149

Table 9.6 Error in the displacement – reaction curve for $f_e = 10^{-3}$ and LFI = 0.50

d_2	R_1	$R_{1,exact}$	d_2	R_1	$R_{1,exact}$
-0.0047006	34.411489	34.411383	-0.2060584	479.908975	479.881782
-0.0093962	67.691615	67.692475	-0.2164001	448.519216	448.486105
-0.0142495	100.941143	100.942348	-0.2253313	417.093846	417.105700
-0.0192765	134.149094	134.150765	-0.2333183	385.660953	385.692559
-0.0244952	167.297846	167.300247	-0.2366832	371.511361	371.511361
-0.0299262	200.360567	200.364187	-0.2366026	371.857878	371.857878
-0.0355919	233.292363	233.298175	-0.2366025	371.858040	371.858040
-0.0415130	266.006466	266.016646	-0.2438820	339.289182	339.287406
-0.0476936	298.294122	298.314388	-0.2506769	306.515214	306.513958
-0.0544278	331.359497	331.349408	-0.2570125	273.883304	273.894050
-0.0615136	363.668624	363.681141	-0.2629638	241.418109	241.439668
-0.0633712	371.745224	371.745224	-0.2686891	208.551570	208.560157
-0.0633975	371.858023	371.858023	-0.2741719	175.544903	175.550449
-0.0712739	404.143072	404.140365	-0.2794388	142.432667	142.436160
-0.0799120	436.012238	436.007905	-0.2845101	109.250411	109.252769
-0.0893930	466.693689	466.726330	-0.2894040	76.019268	76.020926
-0.1003747	496.725807	496.736246	-0.2941365	42.753983	42.755189
-0.1134957	524.739619	524.752153	-0.2987217	9.465240	9.466144
-0.1301819	548.036928	548.050425	-0.3006254	-4.664720	-4.661082
-0.1597767	555.418864	555.417494	-0.3050225	-37.978886	-37.978389
-0.1792233	536.643545	536.615910	-0.3092979	-71.293969	-71.293473
-0.1940294	509.759448	509.728398	-0.3134619	-104.611436	-104.611042

Table 9.7 Iterations per load step of the analysis

Error test factor	Initial load factor	Number of steps	Total iterations	Iterations / Step
10^{-5}	0.20	110	354	3.22
	0.30	75	264	3.52
	0.40	59	236	4.00
	0.50	47	208	4.43
	0.60	41	204	4.98
	0.70	37	177	4.78
	0.80	34	176	5.18
10^{-4}	0.20	110	290	2.64
	0.30	75	207	2.76
	0.40	59	189	3.20
	0.50	47	169	3.60
	0.60	41	148	3.61
	0.70	37	144	3.89
	0.80	34	143	4.21
10^{-3}	0.20	110	172	1.56
	0.30	75	141	1.88
	0.40	59	135	2.29
	0.50	47	118	2.51
	0.60	41	108	2.63
	0.70	37	99	2.68
	0.80	34	100	2.94
10^{-2}	0.20	110	122	1.11
	0.30	75	104	1.39
	0.40	59	88	1.49
	0.50	47	83	1.77
	0.60	41	69	1.68
	0.70	37	72	1.95
	0.80	34	71	2.09

Tables 9.1 to 9.5 show the computed displacement d_2 , the computed load P_2 and the exact load $P_{2,exact}$ calculated with expression (2.20) for the value of d_2 . The initial load factor increment LFI for tables 9.1 to 9.3 is 0.50, the error test factors f_e for the tables are 10^{-5} , 10^{-3} and 10^{-2} . The displacements and loads are shown for successive states on the load curve, starting with state 1 at the top left-hand corner of the table. The tables show that the error in the computed load depends on the load test factor and is fairly independent of the magnitude of the displacement:

- The error for $f_e = 10^{-5}$ is in the 6th digit of P_2
- The error for $f_e = 10^{-3}$ is in the 5th digit of P_2
- The error for $f_e = 10^{-2}$ is in the 4th digit of P_2

In tables 9.2, 9.4 and 9.5 the error test factor is kept constant at $f_e = 10^{-3}$ and the influence of the initial load factors 0.20, 0.50 and 0.80 on the accuracy of the load is investigated. In table 9.4, only every second computed state is shown. The results show that the error is independent of the initial load factor and lies in the 5th digit for all three factors.

Table 9.6 shows the computed displacement d_2 , the computed horizontal reaction R_1 and the exact horizontal reaction $R_{1,\text{exact}}$ calculated with expression (2.24) for the value of d_2 . The initial load factor is 0.50, the error test factor is 10^{-3} . The error is in the 5th digit of R_1 , which corresponds to the error in the load.

The accuracy of the displacement and the load at the turning point is independent of the initial load factor increment and of the error test factor. The exact values are computed with expressions (2.48) and (2.20) using the normalisations (2.13) and (2.14). The first eight digits of the computed displacement and load at the turning point are equal to those of the exact solution.

$$\begin{aligned}
 w_{2c} &= -1 + \frac{1}{\sqrt{3}} &= -0.422\ 649\ 731 \\
 u_{2c} &= h_0 * w_{2c} &= -0.063\ 397\ 459 \\
 p_{2c} &= w_{2c} * (1 + w_{2c}) * (2 + w_{2c}) &= -0.384\ 900\ 179 \\
 P_{2c} &= AE \left(\frac{h_0}{L_0} \right)^3 p_{2c} &= -32.203\ 851
 \end{aligned}$$

Efficiency of the numerical analysis: Each cycle of iteration of the deformation analysis and of the continuation of the load path at the turning point, as well as each cycle of iteration of a step from the nearly-singular to the singular point, requires an assembly and a decomposition of a stiffness matrix. The total number of iterations required for the computation of the history of the frame is therefore chosen as a measure of the efficiency of a numerical analysis.

Table 9.7 shows the total number of iterations as a function of the error test factor and the initial load factor of the analysis. The total number of load steps is chosen so that the apex has a vertical displacement of approximately -0.32 m in all cases. The total number of iterations decreases as the error test factor and the initial load factor increase. For initial load factors in excess of 0.50, the total number of iterations is fairly independent of the initial load factor. The average number of iterations per step varies in the range 1 to 5.

Conclusions: The accuracy of the numerical deformation analysis of the shallow truss depends on the error test factor and not on the initial load factor. The accuracy of the computed displacement and load at the turning point is very high. The continuation of the load path beyond the turning point is robust. The efficiency of the numerical analysis also depends on the error test factor. It increases as the error test factor increases and the accuracy of the analysis decreases. The user must balance accuracy and efficiency in the choice of the error test factor.

9.3 2-BAR TRUSS WITH BIFURCATION POINT

Purpose: The geometrically nonlinear behaviour of a steep 2-bar truss is investigated with the numerical algorithms described in chapter 8. The exact analytical solution for the displacements at the apex and the reactions at the supports is derived in section 2.1. The numerical results are compared to the analytical results in order to determine the following effects:

- influence of the initial load factor increment and the shape factor m (half base width / truss height) on the accuracy of the load and displacement at the bifurcation point
- sensitivity of the continuation of the load path beyond the bifurcation point to the initial load factor increment, the shape factor m and the point of return to the constant arc increment method

Structural model: The structural model for the numerical analysis of the symmetrical steep 2-bar truss is shown in figure 9.4. The height of the apex above the supports is 2.000 m. The truss is investigated for base widths of $2a = 0.80, 0.40$ and 0.20 m. The bars can carry axial forces but not bending moments. The area of the bar cross-section is 10 cm^2 . The modulus of elasticity of the bar material is $2.0 \cdot 10^8 \text{ kN/m}^2$. The supports A and C of the truss are pinned. The truss carries a vertical pattern load $P = (0.0, -10000.0) \text{ kN}$ at apex B.

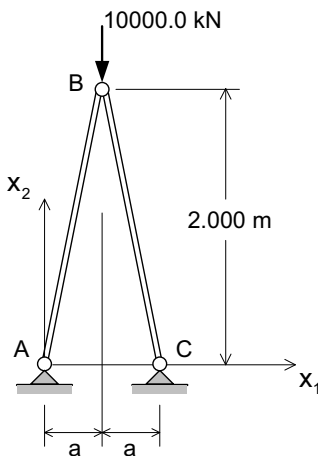


Figure 9.4 Steep 2-bar truss

Nonlinear behaviour: Figure 9.5 shows the horizontal and the vertical component of the displacement of the apex node of a steep truss with shape factor 0.100. The horizontal displacement is null until the bifurcation point is reached and varies nonlinearly on the continuation. The vertical displacement is nearly linear up to the bifurcation point, where the slope is discontinuous, and then varies linearly until the apex node reaches the level of the support nodes.

Figure 9.6 shows the horizontal and vertical components of the reaction at the left support node of the truss. Both components of the reaction vary nearly linearly until the bifurcation point is reached. At the bifurcation point, the slope of the curves is discontinuous. On the continuation, the algebraic value of the horizontal reaction decreases to reach a minimum

when the apex is at the level of the supports. The vertical reaction increases to a maximum and then decreases to null as the apex displaces to the level of the supports.

Accuracy of the bifurcation point: In order to study the accuracy of the bifurcation point, the truss is analysed for half base widths of $a = 0.40, 0.20$ and 0.10 m corresponding to the shape factors $m = 0.20, 0.10$ and 0.05 . The vertical displacement d_{2c} and the vertical load P_{2c} at the bifurcation point are computed with a range of initial load factor increments for each of the half base widths.

On the first part of the load path, where the horizontal displacement d_1 is null, the exact value $P_{2,exact}$ of the load is computed for a given value of the vertical displacement d_2 with equation (2.20) of the exact solution, together with the normalisations (2.13) and (2.14). For the continuation of the load path beyond the bifurcation point, the vertical displacement $d_{2,exact}$ for a given horizontal displacement d_1 is computed with equation (2.41) and the normalisation (2.13). The exact value $P_{2,exact}$ of the load is computed with equation (2.29) and the normalisation (2.14).

The computed values and the exact values of the vertical displacement and load at the bifurcation point are shown in table 9.8. Also shown is the number of load steps n_s from the nearly singular point to the bifurcation point. The displacements and loads at the bifurcation point are correct to at least seven digits. For most shape factors and load factor increments, two load steps are sufficient to reach the bifurcation point from a nearly singular point. If the load path from the reference configuration to the bifurcation configuration contains only a few large load steps, more than two steps are required from a nearly singular point to the bifurcation point.

Examples of the variation of the displacement and the load from a nearly singular point to the bifurcation point are shown in table 9.9 for shape factors $0.200, 0.100$ and 0.050 . The exact displacements and loads at the bifurcation points are given in table 9.8. In all three cases, a single step beyond the nearly singular state leads to a configuration whose loads and displacements differ less than 0.01 percent from the exact values.

Continuation of the load path: The continuation of the load path beyond the bifurcation point is difficult to compute. The reasons for the numerical sensitivity are treated in chapter 7. The main reason is shown in figure 7.16 for a shape factor $m = 0.20$. The figure shows that the load path is close to the curve on which the determinant of the tangent matrix is null. Successive iterated solutions tend to shift the trial state between the two sides of the curve on which the determinant is null, so that the type of the tangent matrix changes between positive and negative determinate. The sensitivity is aggravated for the smaller shape factors 0.10 and 0.05 , where the two curves are even closer to each other.

The steep 2-bar truss in figure 9.4 is used to demonstrate the influence of the shape factor m on the accuracy and the robustness of the continuation. In table 9.10, the computed horizontal displacement d_1 , vertical displacement d_2 and vertical load P_2 are shown for a truss with shape factor 0.200 and initial load factor increment 0.200 . State 7 is nearly singular, state 9 is the bifurcation point. For the first part of the load curve, where $d_1 = 0$, the exact vertical load $P_{2,exact}$ for the computed displacement d_2 is computed with equation (2.20). For the continuation of the load curve, the vertical displacement $d_{2,exact}$ and the vertical load $P_{2,exact}$ for the given displacement d_1 are computed with equations (2.29) and (2.41).

On the load path up to the bifurcation state 9, the computed displacements are correct to at least 5 digits, the computed loads to at least 7 digits. For the continuation immediately following the bifurcation, errors of 0.07 percent in the displacements and 0.08 percent in the loads are observed between states 10 and 14. These are due to the numerical difficulties that are described above. At state 16, the algorithm returns to the constant arc increment method. The displacements and loads on the path beyond state 16 are accurate to at least 5 digits. This accuracy is comparable to that obtained for continuations at turning points.

Table 9.11 shows the computed horizontal displacement d_1 , vertical displacement d_2 and vertical load for a truss with shape factor 0.100 and initial load factor increment 0.060. The truss is nearly singular in state 6. State 8 is a bifurcation point. The analysis is continued with the constant arc increment method beyond state 20. The general nature of the results is similar to that shown in table 9.10 for the truss with shape factor 0.200. The results for the continuation between states 8 and 20 are, however, more accurate for the steeper truss with shape factor 0.100. The maximum error in the displacements and loads is less than 0.01 percent. The displacements and loads on the load path beyond state 20 are exact to at least 6 digits.

Table 9.12 shows the computed horizontal displacement d_1 , vertical displacement d_2 and vertical load for a truss with shape factor 0.050 and initial load factor increment 0.020. State 6 is a bifurcation point. The analysis is continued with the constant arc increment method beyond state 25. The general nature of the results is again similar to that shown in table 9.10 for the truss with shape factor 0.200. The loss of accuracy on the continuation at the bifurcation point is even less than for the truss with shape factor 0.100 and has become negligible.

Conclusions: The computation of the displacement and the load at the bifurcation point of the steep truss in figure 9.4 is highly accurate in all the cases that were studied.

The accuracy of the numerical results for the continuation of the load path immediately following the bifurcation point depends on the shape factor of the truss. For a very steep truss with shape factor 0.05, the accuracy is high. For a truss with shape factor 0.200, errors of the order 0.1 percent are observed in the computed displacements and loads. These errors are tolerable for engineering calculations.

At a sufficient distance from the bifurcation point, the analysis of the truss is continued with the constant arc increment method. The accuracy of the displacements and loads that are computed for this part of the load path is high. It is comparable to the accuracy obtained for the continuation of load paths at turning points.

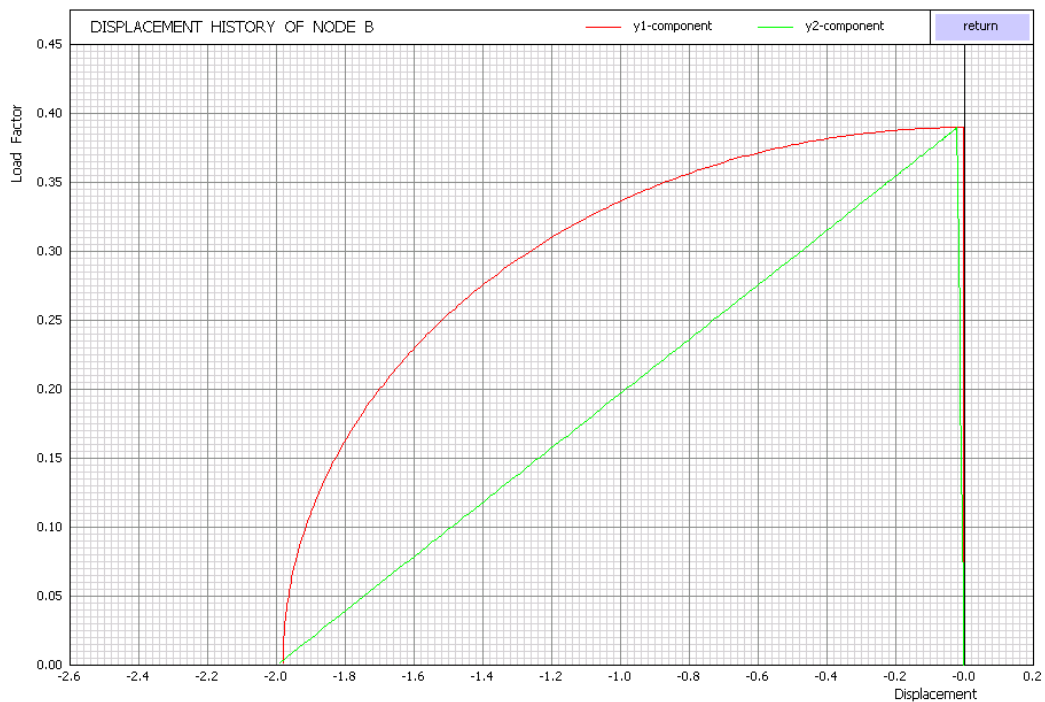


Figure 9.5 Apex displacement of the truss in figure 9.4 with shape factor 0.100

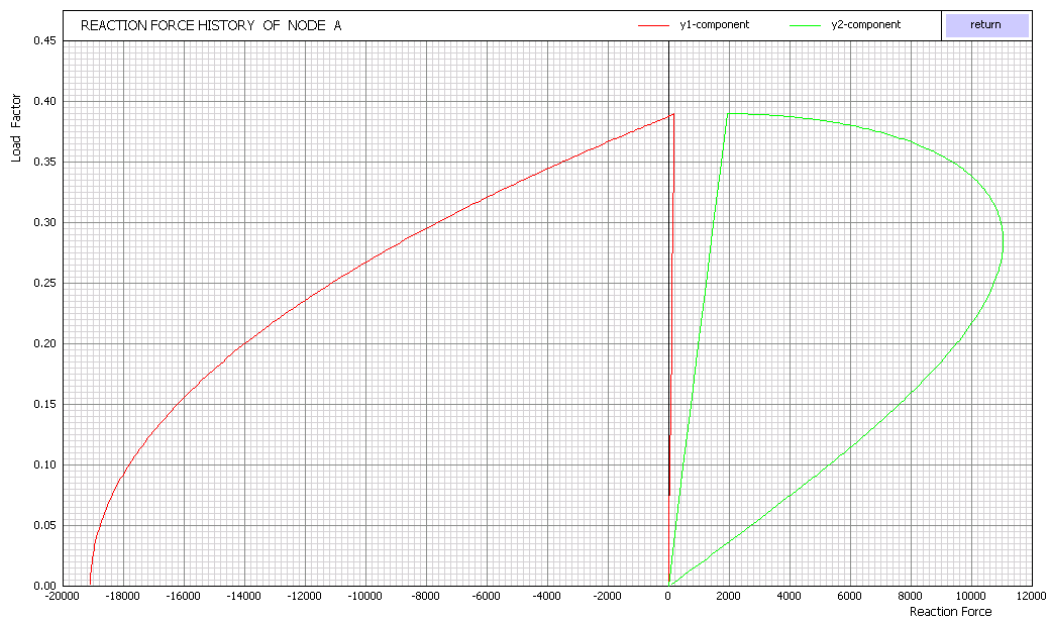


Figure 9.6 Support reactions of the truss in figure 9.4 with shape factor 0.100

Table 9.8 Displacement and load at the bifurcation point

Shape factor	LFI	n_s	d_{2c}	P_{2c}
0.200	0.500	3	-0.081 667 391	-14 469.845 279
	0.300	2	-0.081 667 391	-14 469.845 279
	0.200	2	-0.081 667 391	-14 469.845 279
	0.100	2	-0.081 667 391	-14 469.845 279
	exact		-0.081 667 391	-14 469.845 280
0.100	0.120	2	-0.020 101 013	-3 901.134 901
	0.090	2	-0.020 101 013	-3 901.134 901
	0.060	2	-0.020 101 013	-3 901.134 901
	0.030	2	-0.020 101 013	-3 901.134 901
	exact		-0.020 101 012	-3 901.134 901
0.050	0.025	2	-0.005 006 266	-993.767 909
	0.020	2	-0.005 006 266	-993.767 909
	0.015	2	-0.005 006 266	-993.767 909
	0.010	2	-0.005 006 266	-993.767 909
	exact		-0.005 006 265	-993.767 909

Table 9.9 Step sequence from a nearly-singular point to the bifurcation point

Shape factor	LFI	State	d_2	P_2
0.200	0.200	7	-0.078 629 551	-13 964.481 955
		8	-0.081 664 989	-14 469.446 760
		9	-0.081 667 391	-14 469.845 279
0.100	0.060	6	-0.018 518 353	-3 598.281 087
		7	-0.020 100 381	-3 901.014 101
		8	-0.020 101 013	-3 901.134 901
0.050	0.020	4	-0.004 026 807	-799.929 107
		5	-0.005 006 025	-993.720 384
		6	-0.005 006 266	-993.767 909

Table 9.10 Error in the displacement – load curve for $m = 0.200$ and $LFI = 0.200$

State	d_1	d_2	$d_{2,exact}$	P_2	$P_{2,exact}$
2	0.000 000	-0.021551	-0.021551	-3998.466	-3998.465
7	0.000 000	-0.078630	-0.078630	-13964.482	-13964.480
8	0.000 000	-0.081665	-0.081665	-14469.447	-14469.447
9	0.000 000	-0.081667	-0.081 667	-14469.845	-14469.845
10	-0.047066	-0.082237	-0.082245	-14464.115	-14465.489
12	-0.140824	-0.086940	-0.086843	-14446.713	-14430.804
14	-0.236014	-0.096307	-0.096241	-14371.149	-14359.916
16	-0.332064	-0.110627	-0.110626	-14251.587	-14251.412
18	-0.428804	-0.130199	-0.130207	-14102.873	-14103.717
20	-0.525397	-0.155013	-0.155018	-13916.015	-13916.566
22	-0.620676	-0.184848	-0.184852	-13691.152	-13691.528
24	-0.714448	-0.219670	-0.219673	-13428.605	-13428.879
26	-0.806562	-0.259464	-0.259466	-13128.515	-13128.723
28	-0.896877	-0.304235	-0.304237	-12790.855	-12791.019
30	-0.985247	-0.354007	-0.354008	-12415.463	-12415.596
32	-1.071514	-0.408817	-0.408819	-12002.056	-12002.166
34	-1.155499	-0.468717	-0.468719	-11550.252	-11550.346
36	-1.237000	-0.533769	-0.533770	-11059.587	-11059.667
38	-1.315792	-0.604044	-0.604045	-10529.519	-10529.588
40	-1.391620	-0.679622	-0.679624	-9959.444	-9959.505
42	-1.464196	-0.760592	-0.760593	-9348.702	-9348.756
44	-1.533200	-0.847046	-0.847048	-8696.589	-8696.636
46	-1.598268	-0.939084	-0.939085	-8002.365	-8002.406
48	-1.658993	-1.036804	-1.036805	-7265.272	-7265.307
50	-1.714915	-1.140308	-1.140310	-6484.553	-6484.583
52	-1.765514	-1.249692	-1.249694	-5659.48	-5659.504
54	-1.810203	-1.365043	-1.365046	-4789.396	-4789.411
56	-1.848311	-1.486434	-1.486438	-3873.759	-3873.761
58	-1.879079	-1.613910	-1.613915	-2912.225	-2912.210
60	-1.901640	-1.747476	-1.747485	-1904.751	-1904.701

Table 9.11 Error in the displacement – load curve for $m = 0.100$ and $LFI = 0.060$

State	d_1	d_2	$d_{2,exact}$	P_2	$P_{2,exact}$
2	0.000000	-0.006118	-0.006118	-1199.888	-1199.888
4	0.000000	-0.012290	-0.012290	-2399.320	-2399.320
6	0.000000	-0.018518	-0.018518	-3598.281	-3598.281
7	0.000000	-0.020100	-0.020100	-3901.014	-3901.014
8	0.000000	-0.020101	-0.020101	-3901.135	-3901.135
10	-0.029742	-0.020325	-0.020324	-3900.931	-3900.695
12	-0.059378	-0.020998	-0.020992	-3900.639	-3899.380
14	-0.089027	-0.022111	-0.022104	-3898.729	-3897.189
16	-0.118685	-0.023665	-0.023661	-3894.924	-3894.119
18	-0.148271	-0.025661	-0.025661	-3890.345	-3890.180
20	-0.177801	-0.028101	-0.028101	-3885.380	-3885.372
25	-0.251202	-0.036101	-0.036101	-3869.588	-3869.608
30	-0.324940	-0.046947	-0.046948	-3848.225	-3848.237
35	-0.398370	-0.060593	-0.060593	-3821.343	-3821.351
40	-0.471373	-0.077032	-0.077032	-3788.955	-3788.960
50	-0.615773	-0.118292	-0.118292	-3707.658	-3707.661
60	-0.757548	-0.170759	-0.170759	-3604.280	-3604.282
70	-0.896090	-0.234491	-0.234491	-3478.706	-3478.707
80	-1.030739	-0.309563	-0.309563	-3330.786	-3330.787
90	-1.160772	-0.396065	-0.396065	-3160.346	-3160.347
100	-1.285399	-0.494095	-0.494095	-2967.190	-2967.191
110	-1.403752	-0.603762	-0.603762	-2751.105	-2751.106
120	-1.514873	-0.725182	-0.725182	-2511.863	-2511.864
130	-1.617689	-0.858474	-0.858474	-2249.228	-2249.229
140	-1.710995	-1.003760	-1.003760	-1962.963	-1962.963
150	-1.793414	-1.161153	-1.161153	-1652.840	-1652.840
160	-1.863360	-1.330754	-1.330754	-1318.662	-1318.663
170	-1.918977	-1.512634	-1.512634	-960.292	-960.292
180	-1.958071	-1.706811	-1.706812	-577.690	-577.690
190	-1.977996	-1.913220	-1.913221	-170.988	-170.988

Table 9.12 Error in the displacement – load curve for $m = 0.050$ and $LFI = 0.020$

State	d_1	d_2	$d_{2,exact}$	P_2	$P_{2,exact}$
6	0.000000	-0.005006	-0.005006	-993.768	-993.768
7	-0.005059	-0.005013	-0.005013	-993.765	-993.765
10	-0.020202	-0.005109	-0.005109	-993.783	-993.717
20	-0.070476	-0.006252	-0.006251	-993.175	-993.148
40	-0.170192	-0.012279	-0.012279	-990.145	-990.145
60	-0.270184	-0.023387	-0.023387	-984.612	-984.612
80	-0.369861	-0.039591	-0.039591	-976.540	-976.540
100	-0.468935	-0.060902	-0.060902	-965.924	-965.924
120	-0.567176	-0.087329	-0.087329	-952.761	-952.761
140	-0.664365	-0.118878	-0.118878	-937.045	-937.045
160	-0.760285	-0.155558	-0.155558	-918.773	-918.773
180	-0.854720	-0.197376	-0.197376	-897.943	-897.943
200	-0.947447	-0.244339	-0.244339	-874.549	-874.549
220	-1.038239	-0.296456	-0.296456	-848.588	-848.588
240	-1.126860	-0.353736	-0.353736	-820.055	-820.055
260	-1.213067	-0.416185	-0.416185	-788.947	-788.947
280	-1.296604	-0.483815	-0.483815	-755.259	-755.259
300	-1.377205	-0.556634	-0.556634	-718.985	-718.985
320	-1.454587	-0.634651	-0.634651	-680.122	-680.122
340	-1.528450	-0.717877	-0.717877	-638.665	-638.665
360	-1.598478	-0.806321	-0.806321	-594.608	-594.608
380	-1.664328	-0.899994	-0.899994	-547.947	-547.947
400	-1.725632	-0.998904	-0.998904	-498.677	-498.677
420	-1.781993	-1.103060	-1.103060	-446.794	-446.794
440	-1.832975	-1.212470	-1.212470	-392.293	-392.293
460	-1.878100	-1.327140	-1.327140	-335.172	-335.172
480	-1.916840	-1.447075	-1.447075	-275.429	-275.429
500	-1.948603	-1.572277	-1.572277	-213.062	-213.062
520	-1.972724	-1.702743	-1.702743	-148.073	-148.073
540	-1.988443	-1.838465	-1.838465	-80.466	-80.466

9.4 CANTILEVER BEAM WITH END MOMENT

Purpose: The geometrically nonlinear behaviour of a cantilever beam that carries a moment at its tip is investigated with the numerical algorithms described in chapter 8. The exact analytical solution for the displacements at the tip and the reactions at the support is derived in section 2.2. The numerical results are compared to the analytical results in order to determine the following effects:

- accuracy of the nonlinear analysis of frames with large displacements and rotations
- modelling of large bending displacements associated with axial force null
- influence of the number of elements on the accuracy of the analysis

Structural model: The structural model for the numerical analysis of the cantilever beam is shown in figure 9.7. The length of the cantilever is 2.000 m. The member can carry axial force and bending moment. The area of the cross-section of the beam is 1.00 cm², the moment of inertia is 0.50 cm⁴. The modulus of elasticity of the beam material is 2.0 * 10⁸ kN/m². Support A is clamped. Node B carries a pattern moment M = 0.10 kNm.

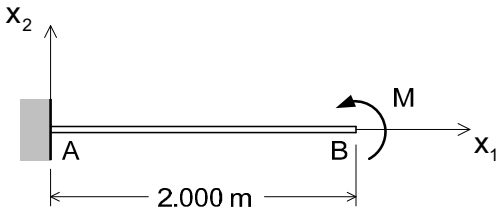


Figure 9.7 Cantilever beam with tip moment M

Nonlinear behaviour: Figure 9.8 shows the deformation of the cantilever due to a moment M = 2.120 kNm. The displacement of the tip of the cantilever from the reference configuration to the instant configuration, in which the beam has the shape of a complete circle, is shown in figure 9.9.

The exact solution shows that the bending moment is constant over the length of the beam, whereas the axial force and the shear force acting on the cross-section are null. The combination of bending moment with axial force null is a severe test for the numerical algorithm. This is due to the nonlinear expression for the axial strain:

$$\epsilon = v_{1,1} + \frac{1}{2}(v_{1,1}^2 + v_{2,1}^2) \tag{a}$$

$v_{1,1}, v_{2,1}$ derivatives for the displacement coordinates in reference space

Consider a bending displacement of 0.100 m at the tip. In linear theory, this displacement does not cause any axial strain. The nonlinear expression (a) yields the strain 0.001250, which corresponds to an axial force of 25 kN. Since the exact axial force is null, the tip of the cantilever must undergo an axial displacement so that ϵ is null in (a). In the numerical algorithm, the axial displacement is determined by iteration in the load step. If the load step is too large, this iteration diverges. The great slenderness of the beam in figure 9.7 was chosen on purpose since it can be expected to intensify the numerical sensitivity.

The accuracy of the numerical solution is affected by an additional problem. The exact solution for the displacement components is a trigonometric function. The interpolation functions for the frame members are polynomials. The behaviour of the cantilever therefore cannot be modelled exactly by a single frame member. The accuracy of the computed result will depend on the number of members used to model the cantilever. It is not obvious how the required number of members depends on the displacements of the beam.

Accuracy of the tip displacement: In order to study the accuracy that can be achieved in the computation of the tip displacement, the beam is subdivided into 20 elements and analysed with initial load factor increments of 0.100, 0.200 and 0.300. The error test factor varies from 0.010 to 0.100. The results of the computations are shown in tables 9.13 to 9.15. The following column headings are used:

- M moment that is applied at the tip
- R moment reaction at the support
- d_1 horizontal displacement coordinate of the tip
- d_2 vertical displacement coordinate of the tip
- β angle of rotation of the tangent at the tip

The exact value of the reaction is equal to the applied moment with inverted sign. The exact values of the rotation and the displacement coordinates are computed with equations (2.56), (2.64) and (2.65) using the normalisations (2.59) to (2.61).

Table 9.13 shows the results for a beam with 20 elements, an initial load factor increment of 0.100 and an error test factor of 0.010. The maximum absolute error of 1.209 mm in the horizontal displacement occurs in state 240 with a tip rotation of 277.8 degrees. This error corresponds to 0.060 % of the length of the beam. The maximum absolute error of 1.659 mm in the vertical displacement occurs in state 190 with a tip rotation of 219.4 degrees. This error corresponds to 0.083 % of the length of the beam. The absolute value of the error in the moment at the support is less than 0.000027 kNm, which may be compared to the moment of 3.14 kNm for a tip rotation of 360 degrees. In spite of the slenderness of the cantilever, the results are highly accurate.

Table 9.14 shows the results for the same beam as table 9.13, but for an initial load factor increment of 0.200 and an error test factor of 0.020. The maximum absolute error of 2.337 mm in the horizontal displacement occurs in state 80 with a tip rotation of 184.3 degrees. This error corresponds to 0.117 % of the length of the beam. The maximum absolute error of 5.371 mm in the vertical displacement occurs in state 100 with a tip rotation of 231.2 degrees. This error corresponds to 0.268 % of the length of the beam. The errors in the displacement have increased by a factor of approximately 3.2 compared to table 9.13. The absolute error in the reaction has increased to 0.002854 kNm in state 75 with a tip rotation of 231.2 degrees. This error is 0.009 % of the moment for a tip rotation of 360 degrees.

Table 9.15 shows the results for the same beam as table 9.13, but for an initial load factor increment of 0.300 and an error test factor of 0.100. The maximum absolute error of 19.614 mm in the horizontal displacement occurs in state 60 with a tip rotation of 208.0 degrees. This error corresponds to 0.981 % of the length of the beam. The maximum absolute error of 50.926 mm in the vertical displacement occurs in state 100 with a tip rotation of 263.3 degrees. This error corresponds to 2.546 % of the length of the beam. The errors in the displacement have increased by a factor of approximately 30.7 compared to table 9.13.

The absolute error in the reaction has increased to 0.047864 kNm in state 75 with a tip rotation of 263.3 degrees. This error is 1.523 % of the moment for a tip rotation of 360 degrees.

Tables 9.13 to 9.15 demonstrate the great influence of the initial load factor increment and the error test factor on the accuracy of the computations. The maximum errors in the tip displacement vary from approximately 0.09 % for the initial load factor increment 0.100 with error test factor 0.010 to approximately 2.6 % for the initial load factor increment 0.300 with the error test factor 0.100. The error in the support moment, which is nearly null for the initial load factor increment 0.001, increases to 1.523 % for the initial load factor increment 0.300. If the error test factor for the analysis with the initial load factor increment 0.300 is decreased in order to reduce the errors, the iteration diverges.

Influence of the number of elements: In order to determine the influence of the number of elements on the accuracy of the analysis, the results in table 9.14 for a cantilever with 20 elements, error test factor 0.020 and LFI = 0.200 are compared to analyses of the same cantilever with 8 and 4 elements. The results for 8 elements with error test factor 0.020 and LFI = 0.200 are shown in table 9.16. The results for 4 elements with error test factor 0.200 and LFI = 0.020 are shown in table 9.17.

The influence of the number of elements on the horizontal displacement at the tip is shown in figure 9.10. The algebraic value of the displacement error, in millimetres, is shown as a function of the angle of rotation at the tip of the cantilever. While the error increases from 2.23 to 6.28 mm as the number of elements is decreased from 20 to 8, a further decrease to 4 elements increases the error to 26.0 mm. The error depends strongly on the rotation at the tip of the cantilever. The influence of the number of elements on the vertical displacement at the tip is shown in figure 9.11. While the error increases from 5.20 to 8.89 mm as the number of elements is decreased from 20 to 8, a further decrease to 4 elements increases the error to 36.3 mm. If a single element is used to model the cantilever, the results in table 9.18 show that the error in the vertical displacement increases steadily with the tip rotation, as follows:

Rotation	$d_{2,exact}$ (m)	$d_2 - d_{2,exact}$ (m)	Relative error (%)
11.5	0.199	0.00032	0.16
39.4	0.582	0.00877	1.51
57.3	0.920	0.03940	4.28
80.2	1.186	0.10272	8.66
103.2	1.364	0.20347	14.92

For vertical tip displacements up to about 0.60 m, the error remains tolerable from an engineering point of view. For larger displacements, more than one element is required.

Conclusions: The computation of the displacements of the very flexible rod due to a tip moment is highly accurate in the entire range from 0 to 360 degrees of tip rotation if an adequate number of elements and suitable values of the initial load factor increment as well as the error test factor are used. For moderate displacements of the order of 30 % of the length of the cantilever, a single element is sufficient to yield results whose accuracy is adequate for engineering purposes. The analysis is robust for reasonable values of the initial load factor increment and the error test factor.

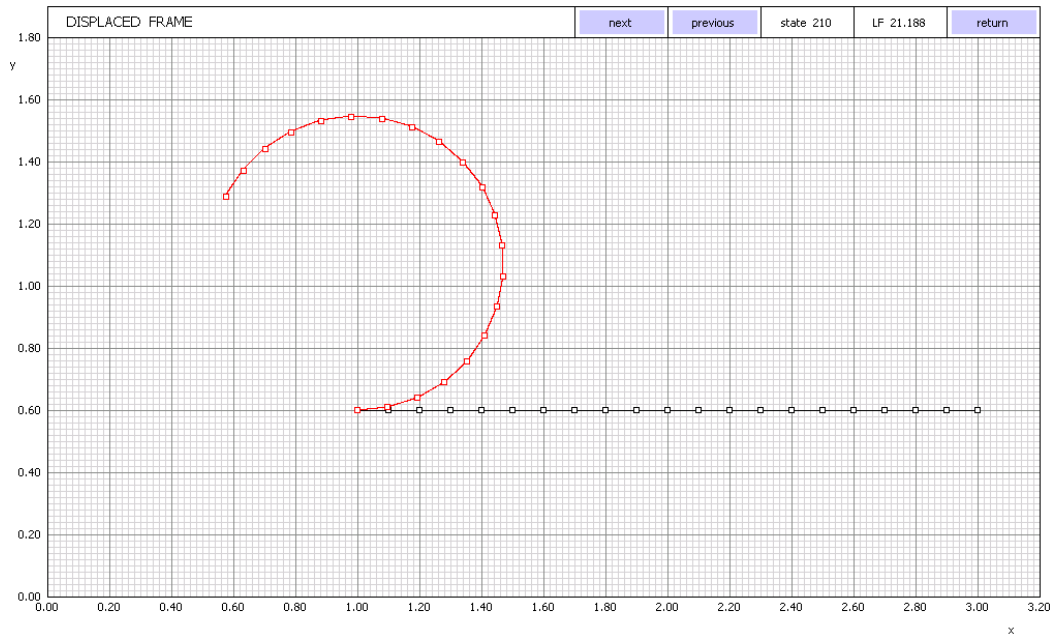


Figure 9.8 Deformation of the cantilever beam in figure 9.7 by a tip moment $M = 2.119$

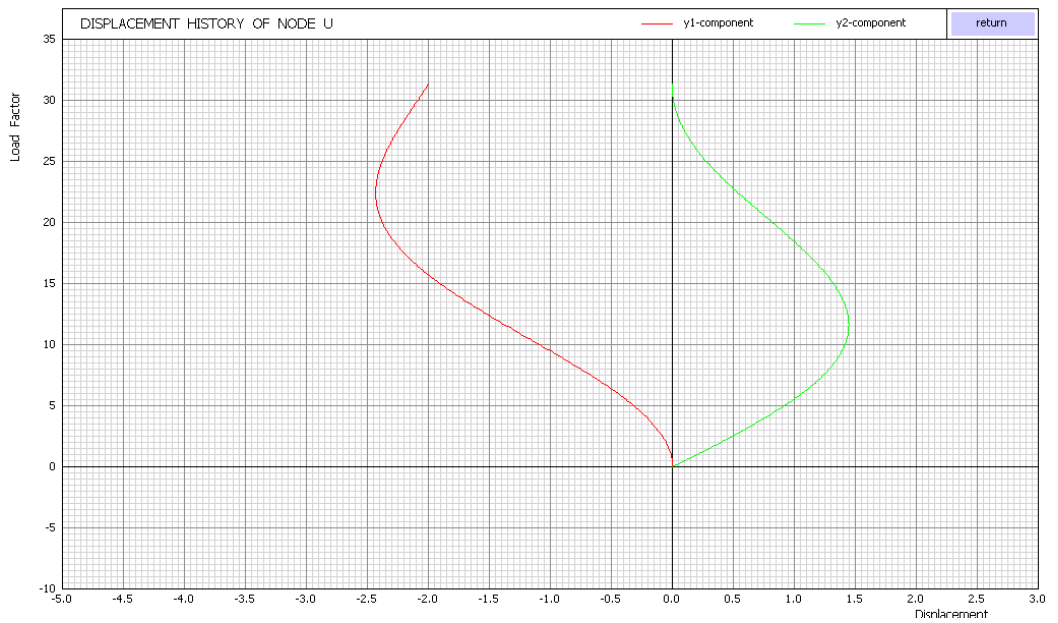


Figure 9.9 Tip displacement of the cantilever beam in figure 9.7

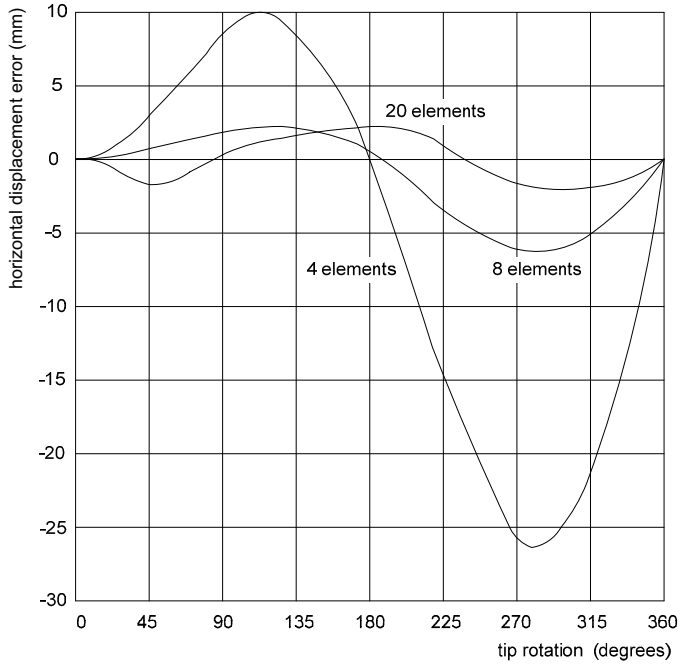


Figure 9.10 Error in the horizontal tip displacement

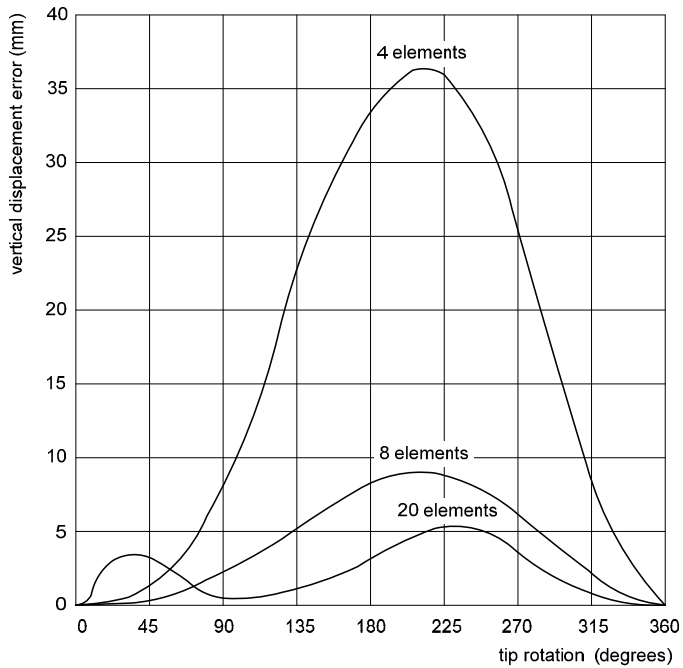


Figure 9.11 Error in the vertical tip displacement

Table 9.13 Analysis of cantilever beam with 20 elements, $f_e = 0.010$ and $LFI = 0.100$

State	M	R	d_1	$d_{1,exact}$	d_2	$d_{2,exact}$	β	β_{exact}
10	0.099 965	-0.099 992	-0.013 293	-0.013 298	0.199 302	0.199 266	0.199 946	0.199 931
20	0.199 945	-0.199 972	-0.052 851	-0.052 880	0.394 619	0.394 590	0.399 879	0.399 891
30	0.299 964	-0.299 986	-0.117 757	-0.117 831	0.582 173	0.582 149	0.599 882	0.599 928
40	0.400 040	-0.400 056	-0.206 513	-0.206 650	0.758 334	0.758 301	0.799 998	0.800 080
50	0.500 191	-0.500 202	-0.317 077	-0.317 288	0.919 752	0.919 687	1.000 264	1.000 382
60	0.600 434	-0.600 440	-0.446 910	-0.447 201	1.063 438	1.063 317	1.200 716	1.200 868
70	0.700 783	-0.700 786	-0.593 043	-0.593 410	1.186 845	1.186 639	1.401 382	1.401 567
80	0.801 251	-0.801 251	-0.752 145	-0.752 578	1.287 926	1.287 612	1.602 287	1.602 501
90	0.901 846	-0.901 846	-0.920 618	-0.921 099	1.365 200	1.364 752	1.803 448	1.803 691
100	1.002 576	-1.002 575	-1.094 680	-1.095 189	1.417 773	1.417 175	2.004 879	2.005 151
110	1.103 515	-1.103 533	-1.270 631	-1.271 111	1.445 394	1.444 629	2.206 759	2.207 029
120	1.204 524	-1.204 520	-1.444 306	-1.444 789	1.448 369	1.447 439	2.408 714	2.409 047
130	1.305 671	-1.305 665	-1.612 226	-1.612 652	1.427 706	1.426 608	2.610 975	2.611 342
140	1.406 953	-1.406 942	-1.770 903	-1.771 240	1.384 950	1.383 692	2.813 503	2.813 905
150	1.508 361	-1.508 347	-1.917 212	-1.917 430	1.322 182	1.320 780	3.016 284	3.016 723
160	1.609 888	-1.60 9869	-2.048 445	-2.048 515	1.241 946	1.240 425	3.219 301	3.219 776
170	1.711 521	-1.711 499	-2.162 383	-2.162 281	1.147 171	1.145 562	3.422 533	3.423 042
180	1.813 247	-1.813 224	-2.257 354	-2.257 065	1.041 073	1.039 417	3.625 955	3.626 495
190	1.915 054	-1.915 031	-2.332 272	-2.331 788	0.927 059	0.925 400	3.829 540	3.830 108
200	2.016 925	-2.016 904	-2.386 654	-2.385 979	0.808 615	0.806 998	4.033 259	4.033 851
210	2.118 848	-2.118 830	-2.420 623	-2.419 772	0.689 199	0.687 669	4.237 081	4.237 696
220	2.220 807	-2.220 793	-2.434 881	-2.433 880	0.572 130	0.570 729	4.440 978	4.441 613
230	2.322 789	-2.322 779	-2.430 672	-2.429 556	0.460 497	0.459 259	4.644 920	4.645 578
240	2.424 783	-2.424 776	-2.409 721	-2.408 534	0.357 062	0.356 013	4.848 879	4.849 565
250	2.526 778	-2.526 774	2.374 160	-2.372 951	0.264 187	0.263 345	5.052 850	5.053 555
260	2.628 766	-2.628 764	-2.326 445	-2.325 268	0.183 780	0.183 151	5.256 801	5.257 532
270	2.730 741	-2.730 741	-2.269 262	-2.268 172	0.117 248	0.116 827	5.460 726	5.461 482
280	2.832 699	-2.832 701	-2.205 429	-2.204 481	0.065 480	0.065 251	5.664 616	5.665 398
290	2.934 639	-2.934 643	-2.137 804	-2.137 049	0.028 840	0.028 775	5.868 472	5.869 278
300	3.036 560	-3.036 567	-2.069 187	-2.068 671	0.007 181	0.007 239	6.072 290	6.073 120
310	3.138 465	-3.138 475	-2.00 2233	-2.001 993	-0.000 125	0.000 006	6.276 078	6.276 931

Table 9.14 Analysis of cantilever beam with 20 elements, $f_e = 0.020$ and $LFI = 0.200$

State	M	R	d_1	$d_{1,exact}$	d_2	$d_{2,exact}$	β	β_{exact}
5	0.099 464	-0.100 379	-0.013 371	-0.013 165	0.199 916	0.198 273	0.200 290	0.198 928
10	0.198 837	-0.200 539	-0.053 044	-0.052 299	0.395 407	0.392 460	0.400 152	0.397 673
15	0.298 302	-0.300 393	-0.117 860	-0.116 552	0.582 530	0.579 117	0.599 545	0.596 604
20	0.397 893	-0.400 000	-0.206 138	-0.204 508	0.757 832	0.754 667	0.798 579	0.795 786
25	0.497 632	-0.499 431	-0.315 743	-0.314 211	0.918 200	0.915 774	0.997 395	0.995 264
30	0.597 534	-0.598 800	-0.444 198	-0.443 200	1.060 945	1.059 437	1.196 208	1.195 068
35	0.697 613	-0.698 291	-0.588 802	-0.588 576	1.183 836	1.183 077	1.395 354	1.395 225
40	0.797 889	-0.798 135	-0.746 658	-0.747 084	1.285 021	1.284 614	1.595 204	1.595 778
45	0.898 388	-0.898 438	-0.914 432	-0.915 195	1.362 961	1.362 508	1.795 860	1.796 775
50	0.999 126	-0.999 098	-1.088 247	-1.089 181	1.416 387	1.415 794	1.997 130	1.998 253
55	1.100 108	-1.099 985	-1.264 083	-1.265 191	1.444 922	1.444 109	2.198 836	2.200 216
60	1.201 331	-1.201 051	-1.438 022	-1.439 373	1.448 815	1.447 719	2.400 880	2.402 662
65	1.302 787	-1.302 261	-1.606 319	-1.607 981	1.428 994	1.427 516	2.603 236	2.605 573
70	1.404 463	-1.403 611	-1.765 500	-1.767 480	1.386 980	1.384 996	2.805 899	2.808 925
75	1.506 341	-1.505 100	-1.912 426	-1.914 658	1.324 830	1.322 213	3.008 869	3.012 681
80	1.608 399	-1.606 739	-2.044 377	-2.046 714	1.245 054	1.241 715	3.212 160	3.216 798
85	1.710 612	-1.708 537	-2.159 114	-2.161 347	1.150 542	1.146 465	3.415 789	3.421 225
90	1.812 953	-1.810 508	-2.254 929	-2.256 819	1.044 470	1.039 738	3.619 780	3.625 905
95	1.915 389	-1.912 669	-2.330 682	-2.332 000	0.930 213	0.925 016	3.824 153	3.830 777
100	2.017 888	-2.015 034	-2.385 815	-2.386 393	0.811 242	0.805 871	4.028 918	4.035 775
105	2.120 413	-2.117 610	-2.420 365	-2.420 135	0.691 040	0.685 847	4.234 060	4.240 827
110	2.222 928	-2.220 373	-2.434 954	-2.433 973	0.573 004	0.568 345	4.439 510	4.445 855
115	2.325 391	-2.323 271	-2.430 785	-2.429 219	0.460 366	0.456 512	4.645 117	4.650 782
120	2.427 768	-2.426 192	-2.409 615	-2.407 688	0.356 079	0.353 140	4.850 632	4.855 536
125	2.530 034	-2.528 997	-2.373 680	-2.371 602	0.262 638	0.260 583	5.055 866	5.060 068
130	2.632 179	-2.631 591	-2.325 563	-2.323 494	0.182 018	0.180 700	5.260 664	5.264 359
135	2.734 201	-2.733 946	-2.268 059	-2.266 102	0.115 560	0.114 832	5.465 023	5.468 402
140	2.836 105	-2.836 116	-2.203 998	-2.202 272	0.064 033	0.063 788	5.669 101	5.672 211
145	2.937 897	-2.938 157	-2.136 262	-2.134 864	0.027 721	0.027 857	5.872 956	5.875 794
150	3.039 586	-3.040 117	-2.067 629	-2.066 654	0.006 391	0.006 823	6.076 665	6.079 172
155	3.141 186	-3.141 999	-2.000 748	-2.000 259	-0.000 573	0.000 000	6.280 238	6.282 371

Table 9.15 Analysis of cantilever beam with 20 elements, $f_e = 0.100$ and $LFI = 0.300$

State	M	R	d_1	$d_{1,exact}$	d_2	$d_{2,exact}$	β	β_{exact}
5	0.145 523	-0.153 716	-0.031 043	-0.028 117	0.303 913	0.288 998	0.304 001	0.291 047
10	0.290 556	-0.306 627	-0.121 641	-0.110 679	0.591 991	0.564 942	0.605 780	0.581 113
15	0.437 281	-0.456 482	-0.263 659	-0.245 378	0.848 573	0.820 221	0.902 745	0.874 562
20	0.585 981	-0.602 752	-0.446 346	-0.427 399	1.064 109	1.043 815	1.194 444	1.171 962
25	0.736 129	-0.746 104	-0.659 507	-0.648 132	1.233 890	1.224 814	1.482 868	1.472 258
30	0.886 804	-0.890 032	-0.898 037	-0.895 467	1.357 046	1.354 780	1.774 876	1.773 608
35	1.038 255	-1.038 336	-1.155 903	-1.157 405	1.430 522	1.429 737	2.074 034	2.076 510
40	1.191 111	-1.188 947	-1.416 932	-1.421 997	1.449 794	1.448 454	2.375 668	2.382 221
45	1.345 371	-1.339 840	-1.666 175	-1.676 125	1.416 149	1.412 307	2.677 798	2.690 742
50	1.500 889	-1.490 629	-1.892 013	-1.907 149	1.335 043	1.326 042	2.980 196	3.001 779
55	1.657 520	-1.641 415	-2.085 257	-2.104 119	1.214 371	1.197 569	3.283 164	3.315 041
60	1.815 170	-1.792 401	-2.239 051	-2.258 665	1.063 614	1.037 325	3.587 242	3.630 340
65	1.973 896	-1.943 858	-2.348 995	-2.365 603	0.893 261	0.857 313	3.893 245	3.947 792
70	2.134 134	-2.096 113	-2.413 119	-2.423 117	0.714 261	0.669 904	4.202 440	4.268 268
75	2.297 395	-2.249 531	-2.431 654	-2.432 269	0.537 272	0.486 346	4.516 981	4.594 789
80	2.434 796	-2.421 866	-2.409 919	-2.405 648	0.360 266	0.346 413	4.845 382	4.869 591
85	2.590 524	-2.577 395	-2.350 971	-2.344 399	0.22 2918	0.211 660	5.156 729	5.181 048
90	2.745 091	-2.734 422	-2.266 779	-2.259 541	0.115 563	0.108 663	5.469 016	5.490 183
95	2.897 956	-2.891 646	-2.167 435	-2.161 568	0.042 726	0.040 162	5.780 225	5.795 912
100	3.049 183	-3.048 429	-2.062 999	-2.060 269	0.005 068	0.005 585	6.090 111	6.098 365
103	3.139 310	-3.142 078	-2.001 892	-2.001 454	-0.001 249	0.000 003	6.275 738	6.278 620

Table 9.16 Analysis of cantilever beam with 8 eight elements, $f_e = 0.020$ and $LFI = 0.200$

state	M	R	d_1	$d_{1,exact}$	d_2	$d_{2,exact}$	β	β_{exact}
5	0.099 598	-0.099 615	-0.013 153	-0.013 200	0.198 575	0.198 539	0.199 223	0.199 196
10	0.199 127	-0.199 142	-0.052 257	-0.052 451	0.393 085	0.393 018	0.398 277	0.398 254
15	0.298 722	-0.298 734	-0.116 445	-0.116 874	0.580 038	0.579 883	0.597 462	0.597 444
20	0.398 416	-0.398 424	-0.204 293	-0.205 029	0.755 877	0.755 553	0.796 843	0.796 832
25	0.498 239	-0.498 244	-0.313 855	-0.314 940	0.917 303	0.916 704	0.996 484	0.996 479
30	0.598 221	-0.598 658	-0.442 955	-0.444 147	1.061 694	1.060 358	1.196 426	1.196 442
35	0.698 294	-0.698 739	-0.588 043	-0.589 614	1.185 632	1.183 845	1.396 420	1.396 589
40	0.798 586	-0.798 797	-0.746 001	-0.748 221	1.287 414	1.285 237	1.596 559	1.597 171
45	0.899 108	-0.899 157	-0.913 735	-0.916 425	1.365 816	1.362 978	1.797 296	1.798 217
50	0.999 870	-0.999 849	-1.087 556	-1.090 477	1.419 794	1.416 095	1.998 628	1.999 741
55	1.100 884	-1.100 884	-1.264 220	-1.266 539	1.448 798	1.444 230	2.201 768	2.201 768
60	1.202 052	-1.202 051	-1.438 490	-1.440 597	1.453 119	1.447 657	2.404 105	2.404 104
65	1.303 411	-1.303 410	-1.607 261	-1.608 992	1.433 653	1.427 321	2.606 825	2.606 822
70	1.404 946	-1.404 945	-1.767 019	-1.768 211	1.391 885	1.384 743	2.809 896	2.809 892
75	1.506 643	-1.506 642	-1.914 575	-1.915 073	1.329 841	1.321 999	3.013 293	3.013 286
80	1.608 484	-1.608 484	-2.047 140	-2.046 817	1.250 040	1.241 641	3.216 977	3.216 968
85	1.710 451	-1.710 451	-2.162 421	-2.161 181	1.155 400	1.146 625	3.420 910	3.420 902
90	1.812 522	-1.812 522	-2.258 670	-2.256 459	1.049 153	1.040 206	3.625 049	3.625 044
95	1.914 676	-1.914 676	-2.334 739	-2.331 548	0.934 722	0.925 833	3.829 360	3.829 352
100	2.016 889	-2.016 889	-2.390 085	-2.385 964	0.815 645	0.807 041	4.033 784	4.033 778
105	2.119 138	-2.119 137	-2.424 792	-2.419 839	0.695 428	0.687 332	4.238 276	4.238 275
110	2.221397	-2.221396	-2.439535	-2.433907	0.577450	0.570065	4.442788	4.442794
115	2.323 643	-2.323 642	-2.435 548	-2.429 447	0.464 860	0.458 357	4.647 271	4.647 286
120	2.425 853	-2.425 852	-2.414 560	-2.408 232	0.360 477	0.354 982	4.851 679	4.851 706
125	2.528 010	-2.528 009	-2.378 717	-2.372 442	0.266 712	0.262 298	5.056 009	5.056 021
130	2.630 096	-2.630 095	-2.330 508	-2.324 578	0.185 513	0.182 194	5.260 145	5.260 191
135	2.732 114	-2.732 115	-2.272 624	-2.267 351	0.118 313	0.116 033	5.464 191	5.464 229
140	2.834 062	-2.834 063	-2.207 918	-2.203 598	0.066 029	0.064 664	5.668 102	5.668 124
145	2.935 935	-2.935 940	-2.139 280	-2.136 180	0.029 046	0.028 408	5.871 856	5.871 871
150	3.037 730	-3.037 735	-2.069 538	-2.067 891	0.007 244	0.007 077	6.075 464	6.075 460
155	3.139 457	-3.139 464	-2.001 380	-2.001 361	0.000 000	0.000 003	6.278 929	6.278 913

Table 9.17 Analysis of cantilever beam with 4 four elements, $f_e = 0.020$ and $LFI = 0.200$

State	M	R	d_1	$d_{1,exact}$	d_2	$d_{2,exact}$	β	β_{exact}
5	0.099 620	-0.099 634	-0.013 004	-0.013 206	0.198 628	0.198 581	0.199 262	0.199 240
10	0.199 175	-0.199 188	-0.051 677	-0.052 476	0.393 296	0.393 111	0.398 370	0.398 351
15	0.298 796	-0.298 806	-0.115 186	-0.116 931	0.580 575	0.580 019	0.597 607	0.597 592
20	0.398 514	-0.398 521	-0.202 160	-0.205 126	0.756 981	0.755 719	0.797 038	0.797 028
25	0.498 359	-0.498 364	-0.310 722	-0.315 084	0.919 270	0.916 888	0.996 724	0.996 719
30	0.598 359	-0.598 361	-0.438 522	-0.444 337	1.064 511	1.060 542	1.196 718	1.196 717
35	0.698 534	-0.698 535	-0.582 787	-0.589 979	1.190 155	1.184 114	1.397 065	1.397 067
40	0.798 883	-0.799 051	-0.740 149	-0.748 707	1.294 083	1.285 502	1.597 113	1.597 765
45	0.899 409	-0.899 463	-0.907 351	-0.916 938	1.374 747	1.363 174	1.797 931	1.798 817
50	1.000 174	-1.000 164	-1.080 919	-1.091 005	1.431 074	1.416 217	1.999 266	2.000 348
55	1.101 166	-1.101 166	-1.257 689	-1.267 030	1.462 679	1.444 274	2.202 326	2.202 333
60	1.202 346	-1.202 346	-1.432 592	-1.441 097	1.469 664	1.447 632	2.404 690	2.404 693
65	1.303 716	-1.303 715	-1.602 491	-1.609 487	1.452 809	1.427 225	2.607 436	2.607 433
70	1.405 262	-1.405 261	-1.763 865	-1.768 688	1.413 473	1.384 579	2.810 527	2.810 524
75	1.506 968	-1.506 966	-1.913 491	-1.915 518	1.353 558	1.321 769	3.013 941	3.013 935
80	1.608 816	-1.608 815	-2.048 522	-2.047 219	1.275 460	1.241 354	3.217 640	3.217 633
85	1.710 789	-1.710 789	-2.166 566	-2.161 529	1.181 991	1.146 289	3.421 587	3.421 579
90	1.812 866	-1.812 866	-2.265 756	-2.256 746	1.076 294	1.039 832	3.625 741	3.625 732
95	1.915 025	-1.915 024	-2.344 794	-2.331 769	0.961 747	0.925 433	3.830 057	3.830 049
100	2.017 243	-2.017 242	-2.402 984	-2.386 116	0.841 852	0.806 627	4.034 492	4.034 485
105	2.119 498	-2.119 497	-2.440 240	-2.419 923	0.720 129	0.686 912	4.239 004	4.238 996
110	2.221 768	-2.221 767	-2.457 067	-2.433 923	0.600 022	0.569 648	4.443 535	4.443 536
115	2.324 032	-2.324 031	-2.454 544	-2.429 397	0.484 758	0.457 946	4.648 056	4.648 064
120	2.426 270	-2.426 268	-2.434 262	-2.408 114	0.377 292	0.354 581	4.852 519	4.852 539
125	2.528 460	-2.528 459	-2.398 262	-2.372 256	0.280 207	0.261 916	5.056 882	5.056 921
130	2.630 584	-2.630 583	-2.348 954	-2.324 324	0.195 644	0.181 843	5.261 110	5.261 168
135	2.732646	-2.732646	-2.289018	2.267033	0.125241	0.115726	5.465202	5.465293
140	2.834617	-2.834617	-2.221316	-2.203238	0.070145	0.064426	5.669173	5.669234
145	2.936524	-2.936526	-2.148801	-2.135785	0.030939	0.028242	5.873008	5.873048
150	3.038365	-3.038368	-2.074418	-2.067468	0.007701	0.007701	6.076711	6.076731
155	3.140141	-3.140147	-2.001018	-2.000924	-0.000005	0.000001	6.280280	6.280282

Table 9.18 Analysis of cantilever beam with 1 element , $f_e = 0.100$ and $LFI = 0.100$

State	M	R	d_1	$d_{1,exact}$	d_2	$d_{2,exact}$	β	β_{exact}
5	0.049 997	0.050 000	-0.002 499	-0.003 331	0.099 947	0.099 911	0.099 987	0.099 994
10	0.099 994	-0.099 997	-0.009 989	-0.013 305	0.199 640	0.199 323	0.199 972	0.199 988
15	0.149 993	-0.149 996	-0.022 452	-0.029 863	0.298 837	0.297 743	0.299 959	0.299 986
20	0.199 995	-0.199 998	-0.039 858	-0.05 2905	0.397 293	0.39 4684	0.399 952	0.399 989
25	0.250 000	-0.250 003	-0.062 164	-0.082 298	0.494 763	0.489 669	0.499 953	0.499 999
30	0.300 010	-0.300 013	-0.089 316	-0.117 866	0.591 005	0.582 232	0.599 962	0.600 019
35	0.350 025	-0.350 028	-0.121 249	-0.159 400	0.685 780	0.671 923	0.699 983	0.700 050
40	0.400 047	-0.400 050	-0.157 885	-0.206 657	0.778 853	0.758 312	0.800 016	0.800 094
45	0.450 076	-0.450 079	-0.199 134	-0.259 358	0.869 990	0.840 990	0.900 065	0.900 152
50	0.500 114	-0.500 116	-0.244 897	-0.317 195	0.958 966	0.919 569	1.000 130	1.000 227
55	0.550 161	-0.550 163	-0.295 063	-0.379 831	1.045 557	0.993 692	1.100 213	1.100 321
60	0.600 217	-0.600 219	-0.349 508	-0.446 902	1.129 547	1.063 027	1.200 317	1.200 435
65	0.650 285	-0.650 287	-0.408 100	-0.518 018	1.210 725	1.127 276	1.300 442	1.300 570
70	0.700 365	-0.700 366	-0.470 696	-0.592 771	1.288 887	1.186 170	1.400 590	1.400 729
75	0.750 457	-0.750 458	-0.537 144	-0.670 730	1.363 837	1.239 477	1.500 764	1.500 913
80	0.800 562	-0.800 563	-0.607 279	-0.751 452	1.435 384	1.287 000	1.600 964	1.601 124
85	0.850 681	-0.850 682	-0.680 930	-0.834 477	1.503 348	1.328 577	1.701 193	1.701 362
90	0.900 815	-0.900 816	-0.757 917	-0.919 339	1.567 555	1.364 086	1.801 450	1.801 630
95	0.950 965	-0.950 966	-0.838 049	-1.005 563	1.627 842	1.393 443	1.901 739	1.901 929
100	1.001 130	-1.001 131	-0.921 129	-1.092 671	1.684 054	1.416 600	2.002 060	2.002 260
105	1.051 312	-1.051 313	-1.006 952	-1.180 185	1.736 046	1.433 551	2.102 414	2.102 625
110	1.101 512	-1.101 512	-1.095 306	-1.267 631	1.783 684	1.44 432	2.202 802	2.203 024
115	1.151 729	-1.151 730	-1.185 971	-1.354 539	1.826 843	1.448 996	2.303 226	2.303 458
120	1.201 965	-1.201 965	-1.278 722	-1.440 449	1.865 411	1.447 665	2.403 687	2.403 930
125	1.252 219	-1.252 219	-1.373 329	-1.524 915	1.899 285	1.440 476	2.504 185	2.504 438
130	1.302 493	-1.302 493	-1.469 554	-1.607 504	1.928 374	1.427 607	2.604 722	2.604 985
135	1.352 786	-1.352 786	-1.567 157	-1.687 803	1.952 600	1.409 269	2.705 298	2.705 572
140	1.403 099	-1.403 099	-1.665 895	-1.765 418	1.971 896	1.385 705	2.805 914	2.806 198
145	1.453 432	-1.453 432	-1.765 518	-1.839 979	1.986 207	1.357 186	2.906 570	2.906 864
150	1.503 786	-1.503 786	-1.865 775	-1.911 144	1.995 491	1.324 013	3.007 267	3.007 572

9.5 GENERAL CONCLUSIONS

The examples in sections 9.2 to 9.4 show that the deformation analysis with the tools presented in this book is highly accurate even if the translations and rotations of the elements of the structure are very large. Additional investigations that are not presented in this book have shown that the tools are readily extended to space trusses and yield equally accurate results for such structures.

The tools also yield highly plausible results for structures with a large number of nodes and members. Unfortunately, exact solutions are not available for these structures, so that explicit conclusions such as those in sections 9.2 to 9.4 cannot be reached for these large structures.

The examples in sections 9.2 and 9.3 show that the tools for the identification of nearly singular configurations of structures are highly reliable. Similar results have been obtained for space trusses with large numbers of nodes and bars.

The examples in sections 9.2 and 9.3 also show that the method to determine singular configurations is efficient and highly accurate. Analyses of axially loaded columns with various support conditions, which are not presented here, show that the bifurcation loads are computed accurately if the columns are subdivided into a suitable number of members.

The continuation of load paths beyond turning points is robust and accurate. In the vicinity of bifurcation points, the continuation method is also robust and accurate. After the load path has left the vicinity of a bifurcation point, the analysis must be continued with the constant arc increment method.

Further research is required to investigate the error bounds for the termination of the iteration systematically. This applies to both the deformation analysis and the continuation of load paths beyond singular points.

The algorithm for the continuation of load paths beyond bifurcation points should be developed further so that it becomes less dependent on the state of the structure in which the analysis is continued with the constant arc increment method.

The tools should be extended to other types of structures, particularly to space frames, plates and folded plates. The favourable results obtained for plane trusses and frames make it likely that equally good results can be obtained for the other types of structures.

References

- [1] Ananyan, V.V. 1987. "Analysis of geometrically and physically nonlinear rod systems by finite element method", *Research on the Analysis of Elements of Space Systems*, pp. 116–122. Moscow: Univ. Druzhby Narodov.
- [2] Argyris, J.H., Dunne, P.C. & Scharpf, D.W. 1978. "Large displacement-small strain analysis of structures with rotational degrees of freedom", *Computer Methods in Applied Mechanics and Engineering*, 14, pp. 401–451.
- [3] Argyris, J.H. & Mjelnik, H.P. 1987. *Die Methode der Finiten Elemente. Band 2: Kraft- und gemischte Methoden, Nichtlinearitäten*. Wiesbaden: Vieweg. ISBN 3-528-08920-0.
- [4] Bathe, K.-J. & Bolourchi, S. 1979. "Large displacement analysis of three-dimensional beam structures", *International Journal for Numerical Methods in Engineering*, 14, pp. 961-986. New York: Wiley.
- [5] Belytschko, T., Liu, W. & Moran, B. 2000. *Nonlinear finite elements for continua and structures*. New York: Wiley. ISBN 0-387-16226-7.
- [6] Bergan, D.G., Bathe, K.-J. & Wunderlich, W. (Eds.) 1985. *Finite element methods for nonlinear problems*. Proceedings of the Europe-USA Workshop, Trondheim. Berlin: Springer. ISBN 0-387-16226-7.
- [7] Bilchuk, Irina. 2005. *Generalisierte Informationsstrukturen für Anwendungen im Bauwesen*. Dissertation Dr.-Ing., Technische Universität Berlin.
- [8] Bonet, J. & Wood, R. 1997. *Nonlinear continuum mechanics for finite element analysis*. Cambridge: Cambridge University Press. ISBN 0-521-57272-X.
- [9] Crisfield, M.A. 1997a. *Non-linear finite element analysis of solids and structures*. Volume 1. New York: Wiley. ISBN 0-471-97059.
- [10] Crisfield, M.A. 1997b. *Non-linear finite element analysis of solids and structures*. Volume 2. New York: Wiley. ISBN 0-471-95649.
- [11] Darkov, A.V. & Shaposhnikov, N.N. 1986. *Structural mechanics*. Moscow: Vyshaya Shkola.
- [12] Davidenko, D.F. 1953. "On the new method for numerical solution of systems of nonlinear equations", *Dokl. Akad. Nauka SSSR*, 88(4), pp. 601–602.
- [13] Davidenko, D.F. 1955. "On the approximate solution of systems of nonlinear equations", *Ukrainian Math. Journal*, 5(2), pp. 196–206.
- [14] Feodosiev, V.I. 1966. *Geometrically nonlinear problems of the theory of plates and shells*, Moscow: Nauka.
- [15] Gemmerling, A.V. 1974. *Analysis of rod systems*. Moscow: Strojizdat.
- [16] Geniev, G.A. 1956. *Selected problems of analysis of bars with generally nonlinear dependence between strains and stresses*. Moscow: Gosstrojizdat.
- [17] Goldenblatt, I.I. 1969. *Nonlinear problems of the theory of elasticity*. Moscow: Nauka.
- [18] Gorodetsky, A.S. & Evzerov, I.D. 2005. *Computer models of structures*. Kiev: FACT.

- [19] Grigolyuk, E.I. & Shalashilin, V.I. 1988. *Problems of nonlinear behavior: parameter continuation method for nonlinear problems of mechanics of deformable solids*. Moscow: Nauka.
- [20] Hechumov, R.A. & Pokrovsky, A.A. 1991. "Mixed form of FEM for the analysis of rod structures with respect to physical and geometrical nonlinearity", *Stroit. Meh. i Raschet Sooruzhenii*, 2, pp. 5–11.
- [21] Huddleston, J.V. 1968. "Finite deflection and snap through of high circular arches", *J. Appl. Mech.*, 35 (December), pp. 763–769.
- [22] Ilyushin, A.A. 1948. *Plasticity*. Moscow: Gostehizdat.
- [23] Ilyushin, A.A. 1954. "On the relation between stresses and small strains in continua mechanics", *Prikl. Matem. i Meh.*, XVIII.
- [24] Jepson, A.D. & Spence, A. 1985. "Folds in solutions of two parameter systems and their calculation", *SIAM J. Num. Anal.*, 22, pp. 347–369.
- [25] Kleiber, M. 1989. *Incremental finite element modeling in nonlinear solid mechanics*. New York: Wiley. ISBN 0470-20832-5.
- [26] Kmiecik, M., Pfau, H., Wiebeck, E. & Wismur, M. 1993. *Nichtlineare Berechnung Ebener Flächentragwerke*. Berlin: Verlag für Bauwesen Berlin. ISBN 3-345-00267-1.
- [27] Kolar, R. & Kamel, H.A. 1986. *On some efficient solution algorithms for nonlinear finite element analysis*. Proceedings, Europe-US Symposium on Finite Element Methods for Nonlinear Problems, Trondheim. Berlin: Springer.
- [28] Kornishin, M.S. 1964. *Nonlinear problems of theory of plates and shallow shells and methods for their solutions*. Moscow: Nauka.
- [29] Krätzig, W.B. & Qian, Y. 1991. "On stability conditions of nonlinear static and dynamic buckling responses of arbitrary structures". In P. Wriggers & W. Wagner (eds), *Nonlinear computational mechanics*, Berlin: Springer. ISBN 3-540-54254-X
- [30] Krishnam Raju, N. & NagaBhushanam, J. 2000. "Nonlinear structural analysis using integrated force method", *Sadhana*, 25(4), pp. 353–365.
- [31] Kryloff, N. & Bogoliuboff, N. 1949. *Introduction to non-linear mechanics*. Princeton, NJ: Princeton University Press. ISBN 13-978-0-691-07985-1.
- [32] Kutznetsov, Y.A. 2004. *Elements of applied bifurcation theory*. Third Edition. Berlin: Springer.
- [33] Kyther, P. & Wie, D. 2003. *An introduction to linear and nonlinear finite element analysis*. Basel: Birkhäuser. ISBN 0-817-643087.
- [34] Lee, S.L., Manuel, F.S. & Russow, E.C. 1968. "Large deflections and stability of elastic frames", *J. Engng. Mech. Div. ASCE*, 94(EM2), pp. 521–547.
- [35] Lukash P.A. 1978. *Basics of nonlinear structural mechanics*. Moscow: Strojizdat.
- [36] Lurie, A.I. 1970. *Theory of elasticity*. Moscow: Nauka.

- [37] Mattiasson, K., Bengtsson, A. & Samuelsson, A. 1986. *On the accuracy and efficiency of numerical algorithms for geometrically nonlinear structural analysis*. Proceedings, Europe-US Symposium on Finite Element Methods for Nonlinear Problems, Trondheim. Berlin: Springer.
- [38] Moore, M. 1983. *The numerical buckling of a visco-elastic rod*. Proceedings, Conference on Numerical Methods for Bifurcation Problems, Dortmund. Basel: Birkhäuser.
- [39] Mushtari, Kh.M. & Galimov, K.Z. 1957. *Nonlinear theory of elasticity of shells*. Kazan: Tatknigoizdat
- [40] Novozhilov, V.V. 1948. *Basics of the nonlinear theory of elasticity*. Leningrad-Moscow: Gostehizdat.
- [41] Novozhilov V.V. 1958. *Theory of elasticity*. (Translated from the Russian in 1961.) Washington: US Department of Commerce, Office of Technical Services.
- [42] Oden, J.T. 1972. *Finite elements of nonlinear continua*. New York: McGraw Hill. ISBN 07-047604-7.
- [43] Ogibalov, P.M. & Koltunov, M.A. 1969. *Shells and plates*. Moscow: Izd. MGU (publishing office of Moscow State University).
- [44] Pahl, P.J. & Damrath, R. 2001. *Mathematical foundations of computational engineering*. Berlin: Springer. ISBN 3-540-67995-2.
- [45] Perelmuter, A.V. & Slivker, V.I. 2005. *Models of structures and possibilities for their analysis*. Kiev: KOMPAS.
- [46] Perelmuter, A.V. & Slivker, V.I. 2007. *Ustoichivost ravnovesoya konstruksij I rod-vennie* (stability of the equilibrium of structures and related problems), Tom 1. Moscow: SKAD SOFT.
- [47] Petrov, V.V. 1959. "On the analysis of shallow shells with finite deflections", *Nauchn. Dokl. Vyshey Shkoly. Stroitelstvo*, 1, pp. 27–35.
- [48] Petrov, V.V. 1974. "On the analysis of plates and shallow shells with respect to physical and geometrical nonlinearity", *Mechanics of Deformable Continua*, 1, Saratov University, pp. 123–130.
- [49] Petrov, V.V. 1975. "A consistent load step method in the nonlinear theory of plates and shells", Saratov University.
- [50] Petrov, V.V., Ovtchinnikov, I.G. & Yaroslavsky, V.I. 1975. "Analysis of plates and shells made of nonlinearly-elastic material", Saratov University.
- [51] Pokrovsky, A.A. 2000. *Mixed form of FEM for the analysis of rod structures and continua*. DSc Dissertation, Penza University of Architecture and Civil Engineering.
- [52] Rank, A., Niggel, A. & Düster, A. 2001. *A high-order finite element approach to non-linear thin-walled solids*. Munich: Lehrstuhl für Bauinformatik, Technische Universität München.
- [53] Reddy, J.N. 2004. *An introduction to nonlinear finite element analysis*. Oxford: Oxford University Press. ISBN 0-19-852529-X.

- [54] Rekach, F.V. 1988. *A method for the analysis of space rod structures with respect to physical and geometrical nonlinearity*. Moscow: Univ. Druzhby Narodov.
- [55] Shalashilin, V.I. & Kuznetsov, E.B. 1999. *Parameter continuation method and the best parameterization*. Kiev: Editorial URSS.
- [56] Simo, J.C. & Hughes, T.J.R. 1998. *Computational inelasticity*. Heidelberg: Springer. ISBN 0-13-978-354-09752-05.
- [57] Spence, A. & Jepson, A.D. 1984. "The numerical calculation of cusps, bifurcation points and isola formation points". In T Küpper, H.D. Mittelmann & H Weber (eds), *Numerical methods for bifurcation problems*, Basel: Birkhäuser, pp. 502–514.
- [58] Stein, E. (Ed.) 1989. *Nichtlineare Berechnungen im Konstruktiven Ingenieurbau*. Berlin : Schlußkolloquium des gleichnamigen DFG-Schwerpunktprogramms. ISBN 3-540-50850-3.
- [59] Steinrück, H., Troger, H. & Weiss R. 1983. *Mode jumping of imperfect buckled rectangular plates*. Proceedings, Conference on Numerical Methods for Bifurcation Problems, Dortmund. Basel: Birkhäuser.
- [60] Tolokonnikov, A.A. 1956. "On the relation between stresses and strains in nonlinear theory of elasticity", *Prikl. Mat. i Meh.*, 20(3).
- [61] Topping, B. & Papadrakakis, M. (Eds.) 1994. *Advances in nonlinear finite element methods*. Edinburgh: Saxe-Coburg. ISBN 0-948-749-26-1.
- [62] Troger, H. 1983. *Application of bifurcation theory to the solution of nonlinear stability problems in mechanical engineering*. Proceedings, Conference on Numerical Methods for Bifurcation Problems, Dortmund. Basel: Birkhäuser.
- [63] Turner, M.R., Clough, R., Martin, H. & Topp, L. 1956. "Stiffness and deflection analysis of complex structures", *J.Aero.Sci.*, 23(9), pp. 805–823.
- [64] Vlasov, V.Z. 1949. *General theory of shells*. Moscow-Leningrad: State publishers for technical-theoretical literature.
- [65] Volmir, A.S. 1956. *Flexible plates and shells*. Moscow: Gostehizdat.
- [66] Wagner, W. 1991. "Zur Behandlung von Stabilitätsproblemen der Elastostatik mit der Methode der Finiten Elemente". *Forschungs- und Seminarberichte aus dem Bereich der Mechanik der Universität Hannover*, F91/1.
- [67] Wagner, W. & Wriggers, P. 1988. "A simple method for the calculation of secondary branches", *Engineering Computations*, 5, pp. 103–109.
- [68] Wagner, W. & Wriggers, P. 1991. "Calculation of bifurcation points via fold curves". In P. Wriggers & W. Wagner (eds), *Nonlinear computational mechanics*, Berlin: Springer. ISBN 3-540-54254-X
- [69] Williams, F.W. 1964. "An approach to the nonlinear behaviour of a rigid-jointed plane framework with finite deflection", *Quart. J. Mech. Appl. Maths.* 17(4), pp. 451–469.
- [70] Wood, R.D. & Zienkiewicz, O.C. 1977. "Geometrically nonlinear finite element analysis of beams, frames, arches and axisymmetric shells", *Computers & Structures*, 7, pp. 725–735.

- [71] Wriggers, P. 2001. *Nichtlineare Finite-Element-Methoden*. Berlin: Springer. ISBN 3-540-67747-X.
- [72] Wriggers, P. & Wagner, W. (Eds.) 1991. *Nonlinear computational mechanics*. Berlin: Springer. ISBN 3-540-54254-X.
- [73] Wriggers, P., Wagner, W. & Mische, W. 1988. "A quadratically convergent procedure for the calculation of stability points in finite element analysis", *Comp. Meth. Appl. Mech. Engng.*, 70, pp. 329–347.
- [74] Wunderlich, W. & Beverungen, G. 1977. "Geometrisch Nichtlineare Theorie und Berechnung eben gekrümmter Stäbe", *Bauingenieur*, 52, pp. 225–237.
- [75] Wunderlich, W., Stein, E. & Bathe, K.J. (Eds.) 1980. *Nonlinear finite element analysis in structural mechanics*. Proceedings, Europe-US Workshop, Ruhr-Universität Bochum. ISBN 3-540-10582-4.
- [76] Zhong, Z.H. 1993. *Finite element procedures for contact-impact problems*. Oxford: Oxford University Press. ISBN 0-19-856383-3.

Subject Index

- accuracy 2, 4, 23, 28
- aggregation of objects 280
- algorithm, numerical 171
- analysis, nonlinear structural 4, 5, 7, 8, 11, 15, 25, 26, 27, 29
- a-priori conditions 77
- arc increment method, basic 16, 17
- arc increment method, constant 20, 21, 29
- arc increment method, refined 17, 18
- arc increments 169
- attributes of objects 276-280, 286-290, 316, 320, 321, 328, 331, 332
- axial force 116, 134, 165, 166
- axial load 165
- bar 91, 92, 99, 101, 279, 280, 284, 286, 288, 292, 296, 314, 316-318, 322, 329
- bar displacement vector 97, 98, 103
- base vector, instant 59, 61, 63
- base vector, reference 63
- beam, cantilever 40
- beam, pinned 43-45
- beam, stretching 40-45
- behaviour, nonlinear structural 2, 3, 5, 7
- bending displacement, relative 168
- bending moment 114, 128, 134, 164, 167, 168
- bending strain derivative 128, 132, 139, 141, 142, 152, 156, 157
- bifurcation 1, 2, 5, 6, 22, 23, 26, 29, 173-176, 188
- bifurcation point 202, 205, 209, 221, 223, 226-232, 236, 237, 240, 248-250, 255, 257, 258, 261, 263
- bifurcation point, accuracy of 353
- bifurcation point, classification of 221
- bifurcation point, column 39, 46
- bifurcation point, cusp 232
- bifurcation point, symmetric 230-232
- bifurcation theory 221, 233, 237
- body, rigid 54, 56, 58, 74
- body, solid 54
- boundary condition 74-76
- boundary conditions, mixed 74, 75
- calibration factor 171-173, 175
- Cauchy, stress tensor of 67
- change of volume 64
- chord length 187
- clarity 3
- class Bar 286, 287, 316
- class Base 287, 289
- class BasicEditor 331
- class Deformation 301, 316, 317
- class Diagram 321, 322
- class Frame 293, 330
- class FramePanel 330
- class GraphicPanel 321, 327
- class Member 317
- class Node 286, 287
- class Path 302
- class Stability 302, 313
- classification of singular point 221
- collection 280, 329
- column, instability 45-48
- column, perturbed 48
- column, rigid 45
- column, unperturbed 46
- complexity 276
- component, constant 110
- component, corrective 111
- component, deviatoric 110
- component, geometric 111
- components, structural 1, 3-6, 10, 25-30
- configuration, critical 38
- configuration, instant 56, 58-60, 62-64, 66-69, 74, 76-78, 81-83
- configuration, nearly singular 23, 28, 29
- configuration, reference 55, 56, 58-64, 69, 74, 78, 79, 81, 85, 87
- configuration, singular 1, 8, 23, 24
- consistency 29
- consistent linearisation 153, 248
- constitutive equation 77, 89
- constitutive equation linear 70
- constitutive matrix 71-73
- continuation 276, 291, 299, 306, 308-310, 312, 313
- continuation at a bifurcation point 310, 313
- continuation at a turning point 308, 312, 313
- continuation of load path 6, 8, 23, 24, 26, 28, 29, 341, 342, 351-353
- continuum 54
- convergence 169, 191
- coordinate grid 325
- coordinate system 92-95, 114-118, 125, 126, 131, 137, 138, 279, 280, 314, 317, 322-324, 328-330
- coordinate system, Cartesian 52, 72, 73
- coordinate system, global 51, 52, 54
- coordinate system, local 52, 74, 75
- coordinates, location 114, 117, 121
- coordinates, tensor 118
- cross-section, rigid 119, 122, 124
- curvature 40-42
- data bases 287, 288
- data types, primitive 278, 331

decomposition 192-196
 deflation of linear equations 239, 241, 245, 247
 deformation gradient, material 59, 60, 63, 64, 69
 deformation gradient, spatial 63, 64
 deformation of a member, small 130
 deformation, large 1, 126, 127, 149
 deformation, small 127, 129, 132, 133, 140, 149, 152, 159, 164
 deformations analysis, methods for 304
 direct iteration method 13
 displacement, axial 140, 165, 166
 displacement, chord 129-133, 139, 144, 145, 149-155, 157, 160, 164
 displacement field 59, 63, 85, 88, 89
 displacement gradient, material 60, 64
 displacement gradient, spatial 60, 64
 displacement, increment 102, 105, 108, 109
 displacement norm, general 171, 173, 174
 displacement of cross-section 113, 114, 119-132, 135-140, 142-151, 153-157, 160, 162, 163, 166, 167
 displacement, relative 129, 131-133, 144, 146, 149-154, 164
 displacement, state of 56, 58-60, 63, 64, 70, 74-78, 80-90, 126, 141, 152
 displacement, 2-bar truss 31, 33-40, 42-44, 48, 49
 displacement vector, member 121, 140, 150
 displacement vector, node 120, 121
 displacement vector, system 86, 100, 101, 109-111
 displacements 91, 95, 103, 104, 111
 displacements, node 145, 146, 148, 149, 164
 displacements, prescribed 55, 76
 efficiency 4, 28
 eigenstates, neighbouring 270
 eigenvector, convergence to the 269
 element displacement vector 86
 element on graphics panel 275, 276, 279, 280, 283, 291, 297, 298, 314-317, 322, 328-330
 element vector 87
 elements 58, 62, 65, 85, 86, 88-90
 equation, equilibrium 68, 70
 equation, homogeneous 38
 equation, strain-displacement 70
 equilibrium equation, exact 190
 equilibrium, 2-bar truss 33, 34, 36, 38, 40, 42, 45, 46, 48
 error term 83, 84, 89, 157, 158
 error vector 101, 137, 149-151, 157-181, 185, 190, 302, 306, 312, 313, 317
 error vector for governing equations 89
 error vector, system 185
 Euclidian space 51, 52, 54, 55, 59
 Euler, strain tensor of 64
 evaluation of data base 275
 finite element method 145, 149
 finite elements 85, 127, 132, 133, 149, 164
 force, axial 116, 134, 165, 166
 force norm, general 170-173, 175
 force vector, system 171, 172, 186
 forces, 2-bar-truss 32-34, 36
 frame 113-120, 124, 127, 129, 130, 132-137, 140, 141, 143, 144, 148, 149, 154, 163, 164
 frame models, structure of 293
 functions, residual 77
 functions, trial 77
 governing equation 127, 136, 137, 140, 141, 143-146, 148, 149, 156, 157, 160, 163, 164
 governing equation, algebraic 111, 163
 governing equation, analytical 100, 135
 governing equation, approximation of 258
 governing equation, incremental 80, 81, 84, 87, 89, 90, 101, 102, 105, 109, 170, 177, 181, 183, 192
 governing equations, for displacement increments 143
 governing equations, for trusses 101, 103, 109-111
 graphical user interface 276, 290, 292, 301, 321-323, 335
 Green, strain tensor of 61
 implementation of serialisation 275-277, 279, 285, 287, 288, 293, 315, 316, 322, 331
 increment of chord displacement 154, 155, 157, 160
 increment of relative displacement 154
 incremental equation, algebraic 90, 109
 incremental stiffness, secant 190, 191
 increments, displacement 139, 142-149, 162
 initial value problem 3, 15, 169, 173
 integral form, material 79
 interface 279, 283, 284, 288, 290, 291, 293, 302, 319-322, 336
 interface, functionality of the 319
 interpolation function 85, 87, 126, 145, 149
 isotropic elasticity 71, 72
 iteration, first cycle at turning point 237, 238, 244
 Java classes for aggregation 283
 Java interfaces for aggregation 283

Kirchhoff hypothesis 40
 layout of editors 331
 linearisation, consistent 8, 29
 linearisation of governing equations 80, 89
 literature 4, 7, 10, 23, 27, 29
 load, axial 165
 load factor increment 170, 183-189, 191
 load factor increment, prescribed 169
 load path 169-171, 173-177, 186, 189
 load, perturbation 45, 47, 48
 load, transverse 164, 166, 168
 load, 2-bar truss 34, 36
 load vector 297, 313, 317
 load vector, system 86, 100, 171, 172, 179, 180
 load-on graphics panel 275, 279, 280, 291, 293-299, 302, 304, 306, 312, 313, 317, 318, 320, 328, 329
 loads 113, 114, 135, 136, 148
 loads, member 144, 148
 location of editors 278, 282, 284-286, 322, 326, 328, 330-332
 location vector 51, 52, 55, 56, 58, 59, 62, 63, 85
 location, identification by 278, 282, 284-286, 322, 326, 328, 330, 331, 332
 mapping of object names 276-278, 282, 283, 285, 288, 290, 291, 317, 323
 material elements 78
 member 113, 114, 124, 126, 127, 132, 136, 148, 149, 152, 164, 279-284, 292, 297, 314-318, 320, 322, 329, 330, 339
 member load, increment of the 145, 162
 model 277, 282, 289-293, 300, 301, 315, 316, 320, 321
 modulus of elasticity 71
 modulus, shear 71-73
 moment of inertia 114
 name of objects 278, 283, 285, 286, 287, 328
 nearly singular configurations 205, 206, 208
 Newton-Raphson iteration 13, 14, 189
 Newton-Raphson iteration, modified 15
 node displacement vector 97, 98
 node on graphic panel 279, 280, 284, 287, 291, 292, 295, 296, 298-300, 314-317, 320, 322, 328, 329, 330, 338
 nodes 85, 89, 91, 97, 100, 101, 103, 106, 108, 111, 113, 114, 116, 127, 129, 132, 133, 135, 136, 146, 148-150, 163, 164
 nonlinear behaviour 31, 39
 nonlinear behaviour of examples 342, 352, 360
 nonlinearity, geometric 28
 nonlinearity, material 1, 28
 normal force 164
 normal of cross-section 113, 114, 118-120, 123, 126, 127, 134, 135, 150, 164
 normalised variables 33, 37, 43
 norms 169-171, 173, 175, 176, 186, 187, 189
 number of elements, influence of the 360, 362
 numerical analysis, efficiency of 351
 numerical properties near bifurcation points 237, 248
 numerical solution techniques 11
 object structure 277, 290
 objects 276-291, 299, 320, 321
 objects, identification of 277, 278
 objects, named 285, 286, 287, 321
 ordered pair 276, 278, 281, 283, 286
 orthotropic elasticity 73
 painting result panel 320, 322, 329, 336
 path, general 175
 path, special 175
 Piola-Kirchhoff stress tensor, 1. 69, 74
 Piola-Kirchhoff stress tensor, 2. 69, 74, 79
 plane 113, 114, 118-120, 122-124, 132, 134-136, 141, 143, 149, 163, 164
 plane frame 113, 114, 119, 122-124, 132, 135, 136, 141, 143, 149, 163, 164
 plane frame hypothesis 119, 122, 123, 124
 plane truss 91, 92, 96, 100-103, 105, 111
 plane truss hypothesis 96
 Poisson ratio 71
 profile matrix 194, 195
 purpose of examples 360
 Rayleigh quotient, convergence of 271
 reaction, 2-bar truss 34, 35
 reaction vector, system 86, 100, 171
 relation 281, 282
 relation, equivalence 276, 281, 290
 relation, order 281
 relation, strict order 281, 282
 reliability 4, 28
 research, needs 25
 research, nonlinear structural analysis 11
 research, topics 28, 30
 residual method, weighted 77
 residuals, weighted 77
 robustness 3, 4, 28, 30
 rotation 53, 54, 56-58, 62, 65
 rotation matrix 53, 57, 62, 65, 93, 94, 108, 116, 117
 rotation, rigid body 74
 rotations, large 41
 serialisation of objects 285, 288
 set 276-278, 280-284, 287-290, 302, 312, 316, 320-322, 329, 331, 332

set object 280, 282-284
 shape 85
 shape factor 35, 36-39
 shear force 143, 164-168
 singular configuration 1, 8, 23, 24, 198,
 202-205, 208, 209, 221
 singular point 6, 22-24, 26, 28-30, 198,
 202-207, 209-211, 213-215, 217-219,
 222, 223, 229, 230, 232, 234-238, 243
 singular points for trusses 207
 snap-through point 202, 204, 209, 223
 software 3, 4, 6, 7, 10, 25, 27, 28, 30
 software packages 7, 27
 space frame 28, 30
 space, global 92, 102
 space, instant 94, 103, 107
 space, reference 93-95, 97, 102, 108
 space, support 95, 100, 103
 space, truss 28, 30
 spatial elements 78
 stabilisation of linear equations 239, 243
 stability 1, 2, 5, 6, 8, 22, 23, 25-28
 stability analysis 210, 213, 237
 state of displacement 99, 103
 state of strain 98, 124
 state of strain vector 70, 71
 state of stress vector 70, 71
 state of the art 4, 6, 7, 21, 23, 25, 29
 state of the art, numerical 4, 6, 7, 21, 23,
 25, 29
 state of the art stability analysis 4, 6, 7, 21,
 23, 25, 29
 status vector, system 100, 135
 stiffness, concepts of 197
 stiffness matrix, element 145, 148, 160,
 162
 stiffness matrix, incremental system 178
 stiffness matrix, of element 111
 stiffness matrix, secant 181, 183, 192
 stiffness matrix, system 110, 178
 stiffness matrix, tangent 37, 38
 stiffness matrix, total 197, 199, 200, 248,
 249, 258-264, 266
 stiffness, secant 197, 198, 200, 201, 207,
 210, 238, 247, 248, 263
 stiffness, tangent 197-200, 202, 203, 206-
 210, 214, 231, 237, 239, 242, 244,
 247, 248, 257, 258, 263
 stiffness, total 197-200, 248, 249, 258-264,
 266
 strain 124-126, 128, 131-133, 136, 138-
 143, 152, 155-158
 strain coordinates, interpolation of 61, 70,
 80, 82, 88
 strain increment 103, 104, 109, 141, 156
 strain, principal 62, 65
 strain, state of 62, 65, 70, 71
 strain, 2-bar truss 32
 strain, yield 39
 stress resultants 148, 164
 stress tensor 67, 69, 70, 74, 79, 83, 89
 stress vector 66-69, 74, 79
 stress, principal 68
 stress, state of 68, 70, 71, 76, 79
 stress-strain relationship 70
 structural model of examples 342, 352,
 360
 subset of objects 280, 281, 289
 support on graphics panel 275, 279, 280,
 295-298, 314, 316, 317, 328
 support, spring 45, 47
 supports 113, 114, 145, 146
 sweep, backward 196
 sweep, forward 196
 system indices of node displacements 314
 tangent matrix 198, 234, 235, 239, 240,
 243, 248, 257
 tangent matrix, determinant of 257
 Taylor series approximation 249, 250
 tension-only analysis 318
 tension-only member 49
 theory of elasticity, nonlinear 3, 5, 8, 10,
 27, 28, 76, 85
 topology matrix 101, 110, 136, 157, 163
 total displacement increment 154
 traction 74, 75, 85
 transformation 114, 116-118, 121, 126
 translation 54, 56
 translation, rigid body 74
 transverse isotropic elasticity 72
 traversal of sets 282-284, 318
 trial function, consistent 28
 truss 91-94, 96-98, 100-103, 106, 111
 turning point 202, 223-225, 235, 237-239,
 243-245, 247
 turning point, simple cubic 225
 unified approach 25, 27, 28, 30
 unified theory 28
 vector, local base 31
 Voigt notation 70, 79, 84
 weight functions 77, 78
 workflow in editors 276, 302, 306, 331
 x coordinates on result panel 329
 y coordinates on result panel 329