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
# Contents

## Preface

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
### Plane Trusses

- **4.1 Configurations of Plane Trusses**
  - 4.1.1 Properties of a Plane Truss
  - 4.1.2 Coordinate Systems
  - 4.1.3 Plane Truss Hypothesis
  - 4.1.4 State of Displacement
  - 4.1.5 State of Strain
  - 4.1.6 Axial Bar Force

- **4.2 Analytical Governing Equations for Plane Trusses**
  - 4.2.1 System Vectors
  - 4.2.2 Governing Equations for a State of the Truss
  - 4.2.3 Increments of the State Variables
  - 4.2.4 Variation of the Incremental Variables
  - 4.2.5 Governing Equations for Displacement Increments

- **4.3 Algebraic Governing Equations for Plane Trusses**
  - 4.3.1 Interpolation of the State Variables for the Error Vector
  - 4.3.2 Interpolation of the Displacement Increments
  - 4.3.3 Error Vector in the Governing Equations
  - 4.3.4 Algebraic Incremental Equations

### Plane Frames

- **5.1 Configurations of Plane Frames**
  - 5.1.1 Properties of a Plane Frame
  - 5.1.2 Coordinate Systems
  - 5.1.3 Plane Frame Hypothesis
  - 5.1.4 State of Displacement
  - 5.1.5 State of Strain

- **5.2 Analytical Governing Equations for Plane Frames**
  - 5.2.1 System Vectors
  - 5.2.2 Governing Equations for a State of the Frame
  - 5.2.3 Increments of the State Variables
  - 5.2.4 Variation of the Incremental Variables
  - 5.2.5 Governing Equations for the Displacement Increments
  - 5.2.6 Governing Equations for a Member
  - 5.2.7 Member Loads

- **5.3 Algebraic Governing Equations for Plane Frames**
  - 5.3.1 Interpolation Functions for the Finite Element Method
  - 5.3.2 Interpolation of the Displacements and Strains
  - 5.3.3 Interpolation of the Displacement and Strain Increments
  - 5.3.4 Error Vector in the Governing Equations
  - 5.3.5 Algebraic Incremental Equations

- **5.4 Stress Resultants in a Member**
  - 5.4.1 Governing Equations
  - 5.4.2 Stress Resultants due to Node Displacements
  - 5.4.3 Displacements and Stress Resultants due to Axial Load
  - 5.4.4 Displacements and Stress Resultants due to Transverse Load

### Table

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

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1</td>
<td>Stepwise Solution Procedure</td>
<td>169</td>
</tr>
<tr>
<td>6.1.1</td>
<td>Concept of the Numerical Algorithm</td>
<td>169</td>
</tr>
<tr>
<td>6.1.2</td>
<td>Normalised Load Paths</td>
<td>170</td>
</tr>
<tr>
<td>6.1.3</td>
<td>Load Paths of a 2-Bar Truss</td>
<td>173</td>
</tr>
<tr>
<td>6.2</td>
<td>Assembly of the Incremental Governing Equations</td>
<td>177</td>
</tr>
<tr>
<td>6.3</td>
<td>Treatment of the Error Vector</td>
<td>181</td>
</tr>
<tr>
<td>6.4</td>
<td>Solution of the Incremental Governing Equations</td>
<td>183</td>
</tr>
<tr>
<td>6.5</td>
<td>Load Factor Increment</td>
<td>186</td>
</tr>
<tr>
<td>6.5.1</td>
<td>Path-Dependent Norms with Equal Signs</td>
<td>186</td>
</tr>
<tr>
<td>6.5.2</td>
<td>Change in Sign of the Path-Dependent Force Norm</td>
<td>189</td>
</tr>
<tr>
<td>6.5.3</td>
<td>Load Step for a 2-Bar Truss</td>
<td>190</td>
</tr>
<tr>
<td>6.6</td>
<td>Linear Equations with Profile Structure and Status</td>
<td>192</td>
</tr>
</tbody>
</table>

## 7 Stability Analysis

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.1</td>
<td>Stiffness of Structures</td>
<td>197</td>
</tr>
<tr>
<td>7.1.1</td>
<td>Concepts of Stiffness</td>
<td>197</td>
</tr>
<tr>
<td>7.1.2</td>
<td>Total Stiffness of a 2-Bar Truss</td>
<td>199</td>
</tr>
<tr>
<td>7.1.3</td>
<td>Tangent Stiffness of a 2-Bar Truss</td>
<td>199</td>
</tr>
<tr>
<td>7.1.4</td>
<td>Secant Stiffness of a 2-Bar Truss</td>
<td>200</td>
</tr>
<tr>
<td>7.2</td>
<td>Singular Configurations of Structures</td>
<td>202</td>
</tr>
<tr>
<td>7.2.1</td>
<td>Concept of a Singular Configuration</td>
<td>202</td>
</tr>
<tr>
<td>7.2.2</td>
<td>Singular Configurations of a 2-Bar Truss</td>
<td>203</td>
</tr>
<tr>
<td>7.3</td>
<td>Nearly Singular Configurations of Structures</td>
<td>206</td>
</tr>
<tr>
<td>7.3.1</td>
<td>Concept of a Nearly Singular Configuration</td>
<td>206</td>
</tr>
<tr>
<td>7.3.2</td>
<td>Nearly Singular Configurations of a 2-Bar Truss</td>
<td>208</td>
</tr>
<tr>
<td>7.4</td>
<td>Determination of Singular Configurations</td>
<td>209</td>
</tr>
<tr>
<td>7.4.1</td>
<td>Determination of a Limit Load Factor</td>
<td>209</td>
</tr>
<tr>
<td>7.4.2</td>
<td>Singular Points for Trusses</td>
<td>210</td>
</tr>
<tr>
<td>7.4.3</td>
<td>Singular Points for Frames</td>
<td>214</td>
</tr>
<tr>
<td>7.5</td>
<td>Bifurcation Theory</td>
<td>221</td>
</tr>
<tr>
<td>7.5.1</td>
<td>Scope of the Theory</td>
<td>221</td>
</tr>
<tr>
<td>7.5.2</td>
<td>Classification of Singular Points</td>
<td>221</td>
</tr>
<tr>
<td>7.5.3</td>
<td>Load Path at a Turning Point</td>
<td>223</td>
</tr>
<tr>
<td>7.5.4</td>
<td>Load Path at a Bifurcation Point</td>
<td>226</td>
</tr>
<tr>
<td>7.5.5</td>
<td>Bifurcation Theory for Finite Element Formulations</td>
<td>233</td>
</tr>
<tr>
<td>7.5.6</td>
<td>Classification of the Singular Points of a 2-Bar Truss</td>
<td>234</td>
</tr>
<tr>
<td>7.6</td>
<td>Continuation of Load Paths at Turning Points</td>
<td>237</td>
</tr>
<tr>
<td>7.6.1</td>
<td>Continuation Concepts</td>
<td>237</td>
</tr>
<tr>
<td>7.6.2</td>
<td>Continuation at Turning Points of a 2-Bar Truss</td>
<td>237</td>
</tr>
<tr>
<td>7.6.3</td>
<td>Nearly Singular Systems of Equations</td>
<td>239</td>
</tr>
<tr>
<td>7.6.4</td>
<td>First Cycle of Iteration in the First Step at a Turning Point</td>
<td>244</td>
</tr>
<tr>
<td>7.6.5</td>
<td>General Cycle of Iteration in the First Step at a Turning Point</td>
<td>245</td>
</tr>
<tr>
<td>7.6.6</td>
<td>General Load Step of the Continuation at a Turning Point</td>
<td>247</td>
</tr>
<tr>
<td>7.7</td>
<td>Continuation of Load Paths at Bifurcation Points</td>
<td>248</td>
</tr>
<tr>
<td>7.7.1</td>
<td>Approximation of the Governing Equations</td>
<td>248</td>
</tr>
<tr>
<td>7.7.2</td>
<td>Properties at Bifurcation Points of a 2-Bar Truss</td>
<td>248</td>
</tr>
<tr>
<td>7.7.3</td>
<td>First Step beyond the Bifurcation Point of a 2-Bar Truss</td>
<td>258</td>
</tr>
<tr>
<td>7.7.4</td>
<td>First Step beyond a Bifurcation Point</td>
<td>261</td>
</tr>
<tr>
<td>7.7.5</td>
<td>General Step of the Continuation at a Bifurcation Point</td>
<td>263</td>
</tr>
<tr>
<td>7.7.6</td>
<td>Total Stiffness Matrix of a Truss</td>
<td>264</td>
</tr>
<tr>
<td>7.7.7</td>
<td>Total Stiffness Matrix of a Frame</td>
<td>265</td>
</tr>
<tr>
<td>7.8</td>
<td>Solution of the General Eigenvalue Problem</td>
<td>268</td>
</tr>
</tbody>
</table>

## 8 Software for Nonlinear Structural Analysis

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.1</td>
<td>Scope of the Implementation</td>
<td>275</td>
</tr>
<tr>
<td>8.2</td>
<td>Object Structure of Information</td>
<td>276</td>
</tr>
<tr>
<td>8.2.1</td>
<td>Objects and Classes of Objects</td>
<td>276</td>
</tr>
<tr>
<td>8.2.2</td>
<td>Identification of Objects</td>
<td>277</td>
</tr>
<tr>
<td>8.2.3</td>
<td>Attributes</td>
<td>278</td>
</tr>
<tr>
<td>8.2.4</td>
<td>Classes of Objects for Trusses and Frames</td>
<td>279</td>
</tr>
</tbody>
</table>
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.

Vera Galishnikova             Peter Dunaiski             Peter Jan Pahl
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 nonlinear 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

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 plas-
ticity 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. Musharti 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 non-
linear 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):

<table>
<thead>
<tr>
<th>Author</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Belytschko</td>
<td>Nonlinear finite element analysis for continua and structures</td>
</tr>
<tr>
<td>Bonet, Wood</td>
<td>Nonlinear continuum mechanics for finite element analysis</td>
</tr>
<tr>
<td>Crisfield</td>
<td>Nonlinear finite element analysis of solids and structures</td>
</tr>
<tr>
<td>Kleiber</td>
<td>Incremental finite element modelling in nonlinear solid mechanics</td>
</tr>
<tr>
<td>Kyther</td>
<td>An introduction to linear and nonlinear finite element analysis</td>
</tr>
<tr>
<td>Reddy</td>
<td>An introduction to nonlinear finite element analysis</td>
</tr>
<tr>
<td>Simo, Hughes</td>
<td>Computational inelasticity</td>
</tr>
<tr>
<td>Wriggers</td>
<td>Nichtlineare Finite-Element-Methoden</td>
</tr>
<tr>
<td>Zhong</td>
<td>Finite element procedures for contact-impact problems</td>
</tr>
</tbody>
</table>
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 \( f(u) \) of a variable vector \( u \). Let the vector \( u \) be a function of a parameter \( \epsilon \) for constant vectors \( u_0 \) and \( \Delta u \):

\[
\begin{align*}
  u &= u_0 + \epsilon \Delta u \\
  f(\epsilon) &= f(u_0 + \epsilon \Delta u)
\end{align*}
\]  

(1.1) (1.2)

The Taylor series expansion of function \( f \) about the point \( \epsilon = 0 \) is:

\[
f(\epsilon) = \left[ f \right]_{\epsilon=0} + \epsilon \frac{df}{d\epsilon} \bigg|_{\epsilon=0} + \frac{1}{2} \epsilon^2 \left[ \frac{d^2f}{d\epsilon^2} \right]_{\epsilon=0} + ...
\]  

(1.3)

This series is truncated after the linear term in \( \epsilon \) and \( \epsilon \) is set equal to 1:

\[
f(u_0 + \Delta u) = f(u_0) + \frac{df(u_0 + \epsilon \Delta u)}{d\epsilon} \bigg|_{\epsilon=0}
\]  

(1.4)

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

\[
f(u_0 + \Delta u) = f(u_0) + K_T(u_0) \Delta u
\]  

(1.5)

\( K_T(u_0) \) matrix of the partial derivatives of function \( f \) with respect to \( u \), for \( \epsilon = 0 \)

**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:

\[
\begin{align*}
  f(u) &= q - u(1+u)(2+u) \\
  u &= w_2 \\
  q &= p
\end{align*}
\]

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

\[
f(\epsilon) = f(u_0 + \epsilon \Delta u) = q - (u_0 + \epsilon \Delta u)(1+u_0 + \epsilon \Delta u)(2+u_0 + \epsilon \Delta u)
\]

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

\[
f(\epsilon) = f(u_0) + \epsilon \left\{ \Delta u(1+u_0 + \epsilon \Delta u)(2+u_0 + \epsilon \Delta u) + \right.
\]

\[
\left. \Delta u(1+u_0 + \epsilon \Delta u)(2+u_0 + \epsilon \Delta u) \right\}_{\epsilon=0}
\]

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

\[
f(u_0 + \epsilon \Delta u) = f(u_0) + \Delta u(2+6u_0 + 3u_0^2)
\]  

(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 $u_{2,1}^2$ 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

\[
r^{(i)} = q - u^{(i)}(1+u^{(i)})(2+u^{(i)})
\]

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

\[
k_t = 3u^2 + 6u + 2 \quad (1.10)
\]

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

\[
u_{mn} = -0.422650 \quad q_{mn} = -0.384900
\]

The solutions in the examples presented in the following sections are obtained for the pattern load \( q_1 = q_{mn} \).

1.5.2 Direct Iteration Method

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

\[
K(u)u = q \quad (1.12)
\]

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

\[
u^{(1)} = (K(u^{(0)}))^{-1}q
\]

(1.13)
The iterated solution vector \( u^{(s)} \) is used to compute a new coefficient matrix \( K(u^{(s)}) \) and a new solution vector \( u^{(2)} \). The iterated solution \( u^{(s)} \) in step \( s \) of the iteration is obtained with equation (1.14):

\[
u^{(s)} = (K(u^{(s-1)}))^{-1} q \tag{1.14}
\]

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

\[
\frac{(u^{(s)} - u^{(s-1)})^T(u^{(s)} - u^{(s-1)})}{u^{(s)}u^{(s)}} < \delta^2
\]  

\( \delta \) 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):

\[
\text{step 0: } \quad k(u^{(0)}) = (1+u^{(0)})(2+u^{(0)}) = 2.000000
\]

\[
u^{(0)} = \frac{q_0}{k(u^{(0)})} = \frac{-0.384900}{2.000000} = -0.192450
\]

\[
q^{(1)} = u^{(1)}(1+u^{(1)})(2+u^{(1)}) = -0.280917
\]

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

\[
\text{step 1: } \quad k(u^{(1)}) = (1+u^{(1)})(2+u^{(1)}) = 1.459687
\]

\[
u^{(1)} = \frac{q_1}{k(u^{(1)})} = \frac{-0.384900}{1.459687} = -0.263687
\]

\[
q^{(2)} = u^{(2)}(1+u^{(2)})(2+u^{(2)}) = -0.337116
\]

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 $f(u)$. A solution $\Delta u$ of the equation $f(u) = 0$ is found by setting $f(u_0 + \Delta u)$ to 0:

$$K_t(u^{(0)}) \Delta u + f(u^{(0)}) = 0$$

(1.16)

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

$$f(u^{(0)}) := r^{(0)} = q_i - K(u^{(0)})u^{(0)}$$

(1.17)

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

**step 0:**

$$r^{(0)} = q_i - u^{(0)}(1+u^{(0)})(2+u^{(0)}) = -0.384900$$

$$k_t^{(0)} = 3(u^{(0)})^2 + 6u^{(0)} + 2 = 2.000000$$

$$\Delta u^{(0)} = \frac{r^{(0)}}{k_t^{(0)}} = \frac{-0.384900}{2.000000} = -0.192450$$

$$u^{(1)} = u^{(0)} + \Delta u^{(0)} = -0.192450$$

$$q^{(1)} = u^{(1)}(1+u^{(1)})(2+u^{(1)}) = -0.280917$$

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

**step 1:**

$$r^{(1)} = q_i - u^{(1)}(1+u^{(1)})(2+u^{(1)}) = -0.103983$$

$$k_t^{(1)} = 3(u^{(1)})^2 + 6u^{(1)} + 2 = 0.956411$$

$$\Delta u^{(1)} = \frac{r^{(1)}}{k_t^{(1)}} = \frac{-0.103983}{0.956411} = -0.108722$$

$$u^{(2)} = u^{(1)} + \Delta u^{(1)} = -0.301172$$

$$q^{(2)} = u^{(2)}(1+u^{(2)})(2+u^{(2)}) = -0.357548$$

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.
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 $K_i$. In the modified Newton-Raphson method, the matrix $K_i$ is kept constant for several cycles of iteration. Only the residual vector $r$ 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 $K_i$.

**Example**: Consider the linearised form (1.10) of the governing equation (1.7) for the test case. The iteration starts with the initial displacement $u_0$. The residual $r^{(0)}$ is computed with (1.9) and the displacement increment with (1.16). The tangent stiffness $k_i$ 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:

**step 1**:

\[
\begin{align*}
    r^{(1)} & = q - u^{(0)} (1 + u^{(0)}) (2 + u^{(1)}) = -0.103983 \\
    k_t^{(1)} & = 3(u^{(0)})^2 + 6u^{(0)} + 2 = 2.000000 \\
    \Delta u^{(1)} & = \frac{r}{k_t} = \frac{-0.103983}{2.000000} = -0.056991 \\
    u^{(2)} & = u^{(0)} + \Delta u^{(1)} = -0.244442 \\
    q^{(2)} & = u^{(2)} (1 + u^{(2)}) (2 + u^{(2)}) = -0.324234
\end{align*}
\]

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 1.2](image) Solution of the test case by 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 \( \Delta \lambda q_i \), where \( \Delta \lambda \) is the load factor increment in the load step. The displacement increment in the step is:

\[
\Delta u = \frac{\Delta \lambda q_i}{k_i} = \Delta \lambda u_i \quad \text{with} \quad u_i = \frac{q_i}{k_i}
\]

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

\[
\Delta u^2 + \Delta q^2 = a^2 \\
\Delta \lambda^2 (u_i^2 + q_i^2) = a^2 \\
\Delta \lambda = \frac{a}{\sqrt{u_i^2 + q_i^2}}
\]

**step 0:**

\[
k_t^{(0)} = 3(u^{(0)})^2 + 6u^{(0)} + 2 = 2.000000 \\
u_t^{(0)} = -\frac{0.384900}{2.000000} = -0.192450 \\
\Delta \lambda^{(0)} = \frac{0.100000}{\sqrt{0.192450^2 + 0.384900^2}} = 0.232379
\]

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

\[
\Delta u^{(0)} = \Delta \lambda^{(0)} u_t^{(0)} = -0.044721 \quad u^{(1)} = u^{(0)} + \Delta u^{(0)} = -0.044721 \quad \Delta q^{(0)} = \Delta \lambda^{(0)} q_t^{(0)} = -0.089443 \quad q^{(1)} = q^{(0)} + \Delta q^{(0)} = -0.089443
\]

**step 1:**

\[
k_t^{(1)} = 3(u^{(1)})^2 + 6u^{(1)} + 2 = 1.737672 \\
u_t^{(1)} = -\frac{0.384900}{1.737674} = -0.221503 \\
\Delta \lambda^{(1)} = \frac{0.100000}{\sqrt{0.221503^2 + 0.384900^2}} = 0.225182
\]

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

\[
\Delta u^{(1)} = \Delta \lambda^{(1)} u_t^{(1)} = -0.049879 \quad u^{(2)} = u^{(1)} + \Delta u^{(1)} = -0.094600 \quad \Delta q^{(1)} = \Delta \lambda^{(1)} q_t^{(1)} = -0.086673 \quad q^{(2)} = q^{(1)} + \Delta q^{(1)} = -0.176115
\]

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.
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 $\hat{u}^{(0)} = -0.044721$. 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 $k_i^{(0)} = 2.0$ at the beginning of the load step:

step 0:

\[
\begin{align*}
 r^{(0,1)} &= q^{(1)} - \hat{u}^{(10)} (1 + \hat{u}^{(10)})(2 + \hat{u}^{(10)}) = -0.005911 \\
 \Delta u^{(0,1)} &= \frac{r^{(0,1)}}{k_i^{(0)}} = \frac{-0.005911}{2.000000} = -0.002951 \\
 \hat{u}^{(1)} &= \hat{u}^{(10)} + \Delta u^{(0,1)} = -0.047672
\end{align*}
\]
The unbalanced force for iteration 2 in step 0 is computed with the corrected trial displacement \( \hat{\mathbf{u}}^{(1)} \):

**step 0:** iteration 2

\[
\begin{align*}
\mathbf{r}^{(0,2)} &= \mathbf{q}^{(1)} - \hat{\mathbf{u}}^{(1)} (1 + \hat{\mathbf{u}}^{(1)} \mathbf{k}_t^{(1)}) (2 + \hat{\mathbf{u}}^{(1)}) = -0.000809 \\
\Delta \hat{\mathbf{u}}^{(0,2)} &= \frac{\mathbf{r}}{k_t^{(0)}} = \frac{-0.000809}{2.000000} = -0.000404 \\
\hat{\mathbf{u}}^{(1,2)} &= \hat{\mathbf{u}}^{(1)} + \Delta \hat{\mathbf{u}}^{(0,2)} = -0.048077 \\
\end{align*}
\]

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

**step 1:**

\[
\begin{align*}
k_t &= 3(\mathbf{u}^{(1)})^2 + 6\mathbf{u}^{(1)} + 2 = 1.718101 \\
\mathbf{u}^{(1)} &= -\frac{0.384900}{1.718101} = -0.224026 \\
\Delta \lambda^{(1)} &= \frac{0.100000}{\sqrt{0.224026^2 + 0.384900^2}} = 0.224543 \\
\Delta \hat{\mathbf{u}}^{(1,0)} &= \Delta \lambda^{(1)} \mathbf{u}^{(1)} = -0.050304 \\
\hat{\mathbf{u}}^{(2,0)} &= \mathbf{u}^{(1)} + \Delta \hat{\mathbf{u}}^{(1,0)} = -0.098446 \\
\Delta \mathbf{q}^{(1)} &= \Delta \lambda^{(1)} \mathbf{q}_t = -0.086427 \\
\mathbf{q}^{(2)} &= \mathbf{q}^{(1)} + \Delta \mathbf{q}^{(1)} = -0.175869 \\
\end{align*}
\]

The trial displacement \( \hat{\mathbf{u}}^{(2,0)} \) 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 \(k_i^{(s)}\) and the pattern displacement \(u_{i}^{(s)}\) due to the pattern load \(q_i\) 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:

\[
\Delta \lambda^{(s,0)} = \frac{a}{\sqrt{\left(u_i^{(s)}\right)^2 + q_i^2}}
\]

\[
\Delta u^{(s,0)} = \Delta \lambda^{(s,0)} u_i^{(s)} \quad \Delta \hat{u}^{(s+1,0)} = u^{(s)} + \Delta \hat{u}^{(s,0)}
\]

\[
\Delta q^{(s,0)} = \Delta \lambda^{(s,0)} q_i \quad \Delta \hat{q}^{(s+1,0)} = q^{(s)} + \Delta \hat{q}^{(s,0)}
\]

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:

\[
(\Delta \lambda u_i + u_c)^2 + (\Delta \lambda q_i)^2 = a^2
\]

\[
(u_i^2 + q_i^2) \Delta \lambda^2 + 2 u_i u_c \Delta \lambda + (u_c^2 - a^2) = 0 \quad (1.18)
\]

The quadratic equation (1.18) is solved for \(\Delta \lambda\). The trial values \(\Delta u\) and \(\Delta q\) 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

\[
k_i^{(0)} = 3(u^{(0)})^2 + 6u^{(0)} + 2 = 2.000000
\]

\[
u_i^{(0)} = \frac{q_i}{k_i^{(0)}} = \frac{-0.384900}{2.000000} = -0.192450
\]

\[
\Delta \lambda^{(0,0)} = \frac{a}{\sqrt{(0.19245)^2 + (0.384900)^2}} = 0.232379
\]

\[
\Delta \hat{u}^{(0)} = \Delta \lambda^{(0,0)} u_i^{(0)} = -0.0447214
\]

\[
\Delta \hat{q}^{(0)} = \Delta \lambda^{(0,0)} q_i = -0.0894427
\]
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

\[
\begin{align*}
\mathbf{r} &= q_c^{(i)} - \mathbf{u}^{(i)(1.0 - \mathbf{u}^{(i)})(2 - \mathbf{u}^{(i)})} = -0.0059106 \quad \Rightarrow \quad q_c^{(i)} \\
\mathbf{u}_c^{(0,1)} &= \frac{q_c^{(i)}}{k_t^{(0)}} = -0.002955 \\
(u_i^2 + q_t^2)\Delta \lambda^2 + 2u_iu_t\Delta \lambda + (u_i^2 - a_t^2) &= 0 \\
0.1851\Delta \lambda^2 + 0.0113749\Delta \lambda - 0.00999132 &= 0 \\
\Delta \lambda^{(0,1)} &= 0.229227 \\
\mathbf{u}_c^{(0,1)} &= \mathbf{u}_c^{(0,1)} + \Delta \lambda^{(0,1)}u_t^{(0)} = -0.0470700 \\
\mathbf{q}_c^{(i)} &= \Delta \lambda^{(0,1)}q_t = -0.0882294
\end{align*}
\]

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

\[
\begin{align*}
\mathbf{r} &= \mathbf{q}_c^{(i)} - \mathbf{u}^{(i)(1.0 - \mathbf{u}^{(i)})(2 - \mathbf{u}^{(i)})} = -0.0006319 \\
q_c^{(2)} &= q_c^{(i)} + r = -0.0059106 - 0.0006319 = -0.0065425 \\
\Delta \lambda^{(0,2)} &= 0.228838
\end{align*}
\]

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:

**step 0:**

\[
\begin{align*}
u^{(i)} &= -0.0473493 \\
q^{(i)} &= -0.0880797 \\
\Delta \lambda^{(i)} &= 0.2288380
\end{align*}
\]

The tangent stiffness \( k_t^{(0)} \) and pattern displacement \( u_t^{(0)} \) for load step 1 are computed with the initial displacement \( u_t^{(0)} \) which was computed at the end of the first load step:

**step 1:** iteration 0

\[
\begin{align*}
k_t^{(1)} &= u_t^{(0)}(1+u_t^{(0)})(2+u_t^{(0)}) = 1.722630 \\
u_t^{(1)} &= \frac{q_t^{(1)}}{k_t^{(1)}} = \frac{-0.384900}{1.722630} = -0.223437 \\
q_c^{(1)} &= 0 \\
\Delta \lambda^{(1)} &= 0.224692 \\
\mathbf{u}_c^{(10)} &= \mathbf{u}_c^{(1)} + \Delta \lambda^{(1)}u_t^{(1)} = -0.0975504 \\
\mathbf{q}_c^{(1)} &= \mathbf{q}_c^{(1)} + \Delta \lambda^{(1)}q_t = -0.1745638
\end{align*}
\]

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 \( u \) is the displacement in the equilibrium state, \( \lambda \) the load factor and \( x \) an eigenvector of the tangential stiffness matrix in the equilibrium state. The function \( L \) in (1.21) sets the norm of \( x \) to 1.

\[
\text{equilibrium} \quad G(u, \lambda) = 0 \quad (1.19)
\]
\[
\text{singularity} \quad K_t(u, \lambda)x = 0 \quad (1.20)
\]
\[
\text{normalisation} \quad L(x) = 0 \quad (1.21)
\]

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

\[
\begin{array}{cccc}
K_t & 0 & -P & \Delta u \\
\frac{\partial}{\partial u}(K_t x) & K_t & \frac{\partial}{\partial \lambda}(K_t x) & \Delta x \\
0 & \frac{\partial L}{\partial x} & 0 & \Delta \lambda \\
\end{array}
= -
\begin{array}{c}
G(u, \lambda) \\
K_t(u, \lambda)x \\
L(x)
\end{array}
\]

This set of equations is solved iteratively by decomposition, starting with a chosen approximate eigenvector \( x \) 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 \( K_A \) and \( K_B \). The stiffness matrices are interpolated linearly with a parameter \( z \). It is assumed that the load factor \( \lambda \) can be interpolated with the same parameter:

\[
K_t(z) = (1-z)K_A + zK_B \quad (1.22)
\]
\[
\lambda = (1-z)\lambda_A + z\lambda_B \quad (1.23)
\]

At the limit point \( z_c \), the determinant of matrix \( K_t(z_c) \) is null and equation (1.22) has a solution \( y \) which is nontrivial:

\[
(1-z)K_A y + z K_B y = 0
\]
\[
K_A y = z(K_A - K_B)y \quad (1.24)
\]

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 $D$ in the decomposition $K_T = LDL^T$ of the tangential stiffness matrix $K_T$ 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 $D$ 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 $u_2$.

![Displacement of a 2-bar truss](image1)

**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 $e_1$ and $e_2$ are orthogonal unit vectors. Let the local coordinates of the displacement vector $v$ of node T relative to node S be $v_{1,1}, v_{2,1}$ 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 $L$.

![Reference and instant configurations of a bar](image2)

**Figure 2.2** Reference and instant configurations of a bar

The base vectors $g_1, g_2$ of the instant configuration of the bar are the derivatives of the location vector $y$ with respect to the coordinates $y_1, y_2$: 
\[ e_1 = \frac{dy}{dy_1} \]
\[ g_i = \frac{d(y + v)}{dy_1} = e_1 + v_{1,i} e_1 + v_{2,i} e_2 \]  
\[ (2.1) \]

The axial strain of the bar follows from the nonlinear theory of elasticity:
\[ \varepsilon_i = \frac{1}{2} (g_i^T g_i - e_1^T e_1) = \frac{1}{2} ((1 + v_{1,i})^2 - v_{2,i}^2 - 1) \]
\[ \varepsilon_i = \frac{1}{2} \left( \frac{L}{L_0} \right)^2 - 1 = \frac{1}{2L_0^2} (L^2 - L_0^2) \]  
\[ (2.2) \]

Expression (2.2) is used to compute the strain \( \varepsilon_A \) in bar AC of the truss:
\[ L_0^2 = a^2 + h_0^2 \]
\[ L_A^2 = (a + u_1)^2 + (h_0 + u_2)^2 \]
\[ \varepsilon_A = \frac{1}{2L_0^2} ((a + u_1)^2 + (h_0 + u_2)^2 - (a^2 + h_0^2)) \]
\[ \varepsilon_A = \frac{1}{2L_0^2} (2au_1 + u_1^2 + 2h_0u_2 + u_2^2) \]
\[ \varepsilon_A = \frac{1}{2L_0^2} \left( \frac{h_0}{L_0} \right)^2 \left( \frac{u_1}{h_0} \left( \frac{2a}{h_0} + \frac{u_1}{h_0} \right) + \frac{u_2}{h_0} \left( 2 + \frac{u_2}{h_0} \right) \right) \]  
\[ (2.3) \]

Expression (2.2) is also used to compute the strain \( \varepsilon_B \) in bar BC of the truss:
\[ L_0^2 = (a - u_1)^2 + (h_0 + u_2)^2 \]
\[ L_B^2 = (a - u_1)^2 + (h_0 + u_2)^2 \]
\[ \varepsilon_B = \frac{1}{2L_0^2} ((a - u_1)^2 + (h_0 + u_2)^2 - (a^2 + h_0^2)) \]
\[ \varepsilon_B = \frac{1}{2L_0^2} (-2au_1 + u_1^2 + 2h_0u_2 + u_2^2) \]
\[ \varepsilon_B = \frac{1}{2L_0^2} \left( \frac{h_0}{L_0} \right)^2 \left( \frac{u_1}{h_0} \left( -2a + \frac{u_1}{h_0} \right) + \frac{u_2}{h_0} \left( 2 + \frac{u_2}{h_0} \right) \right) \]  
\[ (2.4) \]

**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 \( g_i \) 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 \( g_i \):
\[ N_A = A E \frac{L_A}{L_0} \varepsilon_A \]  
\[ N_B = A E \frac{L_B}{L_0} \varepsilon_B \]  
\[ (2.5) \]

\[ (2.6) \]

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

\[(a + u_t)\varepsilon_A + (-a + u_t)\varepsilon_B = \frac{P_1 L_0}{AE}\] (2.9)
\[(h_0 + u_z)(\varepsilon_A + \varepsilon_B) = \frac{P_2 L_0}{AE}\] (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:

\[
\frac{u_i}{h_0} \left( 2 \frac{a}{h_0} + \frac{u_z}{h_0} + \frac{u_z}{h_0} \right) = \frac{P_i}{AE} \left( \frac{L_0}{h_0} \right)^3
\] (2.11)
\[
\left( 1 + \frac{u_z}{h_0} \right) \left( \frac{u_i}{h_0} \right)^2 + \frac{u_z}{h_0} \left( 2 + \frac{u_z}{h_0} \right) = \frac{P_2}{AE} \left( \frac{L_0}{h_0} \right)^3
\] (2.12)

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

- Normalised displacements: $w_i := \frac{u_i}{h_0}$ for $i \in \{1, 2\}$ (2.13)
- Normalised loads: $p_i := \frac{P_i}{AE} \left( \frac{L_0}{h_0} \right)^3$ (2.14)
- Normalised reactions: $r_i := \frac{2R_i}{AE} \left( \frac{L_0}{h_0} \right)^3$ (2.15)
- Normalised bar forces: $n_i := \frac{2N_i}{AE} \left( \frac{L_0}{h_0} \right)^3$ (2.16)
- Shape factor: $m := \frac{a}{h_0}$ (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:

\[
p_1 = w_1 (2m^2 + w_1^2 + w_z^2 (2 + w_z))
\] (2.18)
\[
p_2 = (1 + w_z) (w_1^2 + w_z^2 (2 + w_z))
\] (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):

\[
p_2 = w_2(1 + w_2)(2 + w_2)
\]

(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:

![Reference configuration, Collinear bars, Mirror image](image)

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:

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

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

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

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

\[
\varepsilon_A = \varepsilon_B = \frac{1}{2} \left( \frac{h_k}{L_0} \right)^2 w_2(1 + w_2)
\]

(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:

\[
\begin{align*}
    r_{1A} &= -m w_2(2 + w_2) \\
    r_{2A} &= -w_2(1 + w_2)(2 + w_2) \\
    r_{1B} &= m w_2(2 + w_2) \\
    r_{2B} &= -w_2(1 + w_2)(2 + w_2)
\end{align*}
\]

(2.24)

(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:

\[
\frac{dr_1}{dw_2} = -2m(1 + w_2) = 0
\]

\( w_2 = -1 \) (2.26)

\[
r_{1A} = r_{1B} = m
\]

(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:
\[
2m^2 + w_1^2 + w_2(2 + w_2) = 0 \tag{2.28}
\]
The vertical load \( p_2 \) follows from (2.19):
\[
p_2 = -2m^2(1 + w_2) \tag{2.29}
\]
The strain in bars AC and BC follows from (2.3) and (2.4):
\[
\varepsilon_A = -\left(\frac{h_0}{L_0}\right)^2 m(m - w_1) \tag{2.30}
\]
\[
\varepsilon_B = -\left(\frac{h_0}{L_0}\right)^2 m(m + w_1) \tag{2.31}
\]
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):
\[
r_{1A} = 2m(m^2 - w_1^2) \tag{2.32}
\]
\[
r_{2A} = 2m(m - w_1)(1 + w_2) \tag{2.33}
\]
\[
r_{2B} = -2m(m^2 - w_1^2) \tag{2.34}
\]
\[
r_{2B} = 2m(m + w_1)(1 + w_2) \tag{2.35}
\]
Substitution of \( w_2 \) from (2.29) into (2.28) yields a quadratic equation for \( w_1 \):
\[
w_1^2 = 1 - 2m^2 - \left(\frac{p_2}{2m^2}\right)^2 \tag{2.36}
\]
Equation (2.36) has real roots for \( w_1 \) if load \( p_2 \) satisfies condition (2.37):
\[
p_2^2 < 4m^4(1 - 2m^2) \tag{2.37}
\]
Equation (2.37) has real roots for \( p_2 \) if the shape factor \( m \) satisfies condition (2.38):
\[
m \leq m_c = \frac{1}{\sqrt{2}} = 0.707107 \tag{2.38}
\]
For shape factors \( m \leq m_c \), equation (2.37) yields two real roots \( \pm p_{2c} \) for which the truss is in equilibrium:
\[
p_{2c} = -2m^2\sqrt{1 - 2m^2} \tag{2.39}
\]
Substitution of (2.39) into (2.29) yields the critical displacement \( w_{2c} \):
\[
w_{2c} = -1 + \sqrt{1 - 2m^2} \tag{2.40}
\]
The displacement state \((w_1, w_2)\) for the load range \(-p_{2c} \leq p_2 \leq p_{2c}\) is given by (2.28):
\[
w_1^2 + (1 + w_2)^2 = 1 - 2m^2 \quad m \leq m_c \tag{2.41}
\]
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 \( m = 0.3 \).
Tangent stiffness matrix: The incremental equations for a load step of the truss are:

\[ \mathbf{K} \Delta \mathbf{u} = \Delta \mathbf{P} \] (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 \( w_m \):

\[ k_{im} = \frac{\partial P}{\partial u_m} = \frac{AE}{h_0} \left( \frac{h_0}{L_0} \right)^3 \frac{\partial p_i}{\partial w_m} \] (2.43)

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

\[ \mathbf{K} = \frac{AE}{h_0} \left( \frac{h_0}{L_0} \right)^3 \begin{bmatrix} 2m^2 + 3w_i^2 + w_j(2 + w_j) & 2w_i(1 + w_j) \\ 2w_i(1 + w_j) & w_i^2 + 3w_j^2 + 6w_j + 2 \end{bmatrix} \] (2.44)

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

\[ \mathbf{K}_{\text{linear}} = \frac{AE}{h_0} \left( \frac{h_0}{L_0} \right)^3 \begin{bmatrix} 2m^2 & 0 \\ 0 & 2 \end{bmatrix} \] (2.45)
The displacement state \((w_1, w_2)\) for which the tangent stiffness matrix \(K\) 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:

\[
\frac{AE}{h_0 L_0} \begin{bmatrix} 2m^2 + w_x(2+w_x) & 0 \\ 0 & 3w_x^2 + 6w_x + 2 \end{bmatrix} \begin{bmatrix} \Delta u_1 \\ \Delta u_2 \end{bmatrix} = \begin{bmatrix} 0 \\ \Delta P_2 \end{bmatrix}
\]  

(2.46)

**Solution 2:** \(2m^2 + w_1^2 + w_x(2+w_x) = 0\)

Expression (2.44) leads to the following equation:

\[
\frac{AE}{h_0 L_0} \begin{bmatrix} 2w_1^2 & 2w_1(1+w_1) \\ 2w_1(1+w_1) & 2(1+w_1)^2 - 2m^2 \end{bmatrix} \begin{bmatrix} \Delta u_1 \\ \Delta u_2 \end{bmatrix} = \begin{bmatrix} 0 \\ \Delta P_2 \end{bmatrix}
\]  

(2.47)

**Stability:** The truss is in a singular configuration if the tangent stiffness matrix \(K\) is singular so that a nontrivial solution \(\Delta u\) 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:

\[
\begin{align*}
\text{Case 1:} & \quad k_{11} = 2m^2 + w_x(2+w_x) = 0 \\
& \quad w_{2c} = -1 \pm \sqrt{1-2m^2} \\
& \quad k_{22} = 2(1-3m^2)
\end{align*}
\]

(2.48)

\[
\begin{align*}
\text{Case 2:} & \quad k_{11} = 3w_x^2 + 6w_x + 2 = 0 \\
& \quad w_{2c} = -1 \pm \frac{1}{\sqrt{3}} = -0.422650 \text{ or } -1.577350 \\
& \quad k_{22} = 2(m^2 - \frac{1}{3})
\end{align*}
\]

(2.49)

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

\[
\det K = -4m^2w_1^2
\]

(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):

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

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).

\[
h_0 \leq 0.100 \ L_0 \quad m > m_c \tag{2.52}
\]

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

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

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).

\[
a \leq 0.0346 \ L_0 \quad m \leq m_c \tag{2.54}
\]

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 \( M \).

![Diagram of a cantilever beam with labels and curves indicating displacements.]

**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 \( \theta \). Let the value of \( \theta \) 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:

\[
\frac{d\theta}{ds} = \frac{M}{EJ} \tag{2.55}
\]
The rotation $\theta$ at support $s = 0$ is null. The curvature (2.55) is integrated from the support to point Q:

$$\theta = \frac{M}{EJ} s$$  

(2.56)

The trigonometric functions of angle $\theta$ are determined in the triangle with base QT:

$$\sin \theta = \frac{du_i}{ds}$$  

(2.57)

$$\cos \theta = 1 + \frac{du_i}{ds}$$  

(2.58)

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

$$\alpha = \frac{ML}{EJ}$$  

(2.59)

$$v_i = \frac{u_i}{L}$$  

(2.60)

$$t = \frac{s}{L}$$  

(2.61)

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

$$\frac{dv_i}{dt} = \cos \alpha t - 1$$  

(2.62)

$$\frac{dv_i}{dt} = \sin \alpha t$$  

(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 $t = 0$:

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

(2.64)

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

(2.65)

The instant configurations of the beam for $\alpha = 0.25\pi, 0.50\pi, ..., 2.00\pi$ are shown in figure 2.7. The maximum rotation at the tip of the beam is $2\pi$. 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.

\[
q = q_0 \sin \frac{\pi x}{L}
\]

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:

\[
n \frac{d^2u_j}{dx_1^2} + q_n = 0
\]

(2.66)
The part \( q_b \) of the applied load that is carried by bending is determined by the usual beam theory:

\[
EJ \frac{d^4u_2}{dx_1^4} - q_b = 0
\]  

(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):

\[
EJ \frac{d^4u_2}{dx_1^4} - n \frac{d^2u_2}{dx_1^2} = q
\]  

(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:

\[
u_2 = u_0 \sin \left( \frac{\pi x}{L} \right)
\]  

(2.69)

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

\[
q_0 = u_0 \left( EJ \left( \frac{\pi}{L} \right)^4 + n \left( \frac{\pi}{L} \right)^2 \right)
\]  

(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:

\[
\varepsilon_n = \left( \sqrt{1 + u_{2,1}^2} - 1 \right) \frac{1}{\sqrt{1 + u_{2,1}^2}} = 1 - \frac{1}{\sqrt{1 + u_{2,1}^2}} \approx \frac{1}{2} \left( \frac{du_2}{dx_1} \right)^2
\]  

(2.71)

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

\[
\bar{\varepsilon}_n = \frac{1}{L} \int_0^L \frac{1}{2} u_0^2 \left( \frac{\pi}{L} \right)^2 \cos^2 \left( \frac{\pi x}{L} \right) dx_1 = \frac{\pi}{4} \left( \frac{u_0}{L} \right)^2
\]  

(2.72)

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

\[
n = A E \bar{\varepsilon}_n = A E \pi^2 \left( \frac{u_0}{4L} \right)^2
\]  

(2.73)

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

\[
q_0 = J E L \left( \frac{\pi}{L} \right)^4 \left( \frac{u_0}{L} \right) + A E L \left( \frac{\pi}{4L} \right)^3 \left( \frac{u_0}{L} \right)^3 = \frac{\pi^4 J E}{L^3} \left( \frac{u_0}{L} \right) \left( 1 + \frac{L^3}{\pi^4 E J} A E L \frac{\pi}{L} \frac{u_0}{L} \right)^3
\]  

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

\[
\tilde{q} = \frac{q_0}{\pi^4 E J}
\]  

(2.74)

\[
\tilde{u} = \frac{u_0}{L}
\]  

(2.75)

\[
c = \frac{L^3}{\pi^4 E J} A E L \left( \frac{\pi}{L} \right)^2 = \frac{1}{4\pi} \left( \frac{L}{r} \right)^2 \text{ with } r^2 = \frac{J}{A}
\]  

(2.76)
The normalised load on the pinned beam is thus:
\[ \bar{q} = \bar{u}(1 + c\bar{u}^2) \]  
(2.77)

The normalised stiffness \( \bar{k} \) of the beam is defined as follows:
\[ \bar{k} = \frac{\bar{q}}{\bar{u}} = 1 + c\bar{u}^2 \]  
(2.78)

The normalised axial force \( \bar{n} \) in the beam is defined as follows:
\[ \bar{n} = \frac{n}{q_o L} = \frac{c\bar{u}}{\pi^2(1 + c\bar{u}^2)} \]  
(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 $\omega$ 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 $\omega$, and in the negative direction of axis $x_2$ for negative rotation $\omega$.

![Figure 2.12 Rigid cantilever column with a rotation spring support](image)

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:

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

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

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

$$P_2 = m P_1$$ (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 \( \sin \omega \):

\[
p \left( \omega - \frac{1}{3!} \omega^3 + \frac{1}{5!} \omega^5 - \ldots \right) = \omega
\]

\[
\omega \left( p \left( 1 - \frac{1}{6} \omega^2 + \frac{1}{120} \omega^4 - \ldots \right) - 1 \right) = 0
\]

(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 \( \omega^2 \):

\[
\frac{1}{6} \omega^2 - \frac{1}{120} \omega^4 = 1 - \frac{1}{p}
\]

(2.85)

The left-hand side of (2.85) is not negative for real values of \( \omega \). 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 \( p_c \).

\[
p_c = 1.0
\]

(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):

\[
p = \frac{\omega}{\sin \omega}
\]

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

\[
\frac{dp}{d\omega} = \frac{\sin \omega - \omega \cos \omega}{\sin^2 \omega}
\]

(2.88)

\[
\frac{d^2p}{d\omega^2} = \frac{\omega \sin^3 \omega - 2 \sin \omega \cos \omega (\sin \omega - \omega \cos \omega)}{\sin^4 \omega}
\]

\[
= \frac{\omega}{\sin \omega} - \frac{2 \cos \omega}{\sin^2 \omega} + \frac{2 \omega \cos^2 \omega}{\sin^3 \omega}
\]

(2.89)
Figure 2.13 Bifurcation of a rigid cantilever column with spring support

Variation of the rotation $\omega$ with the load $p$ for load ratio $m = P_1/P_2$

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

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

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

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

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

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

$$\lim_{\omega \to 0} \frac{dp}{d\omega^2} = 1 + \frac{5}{3} = \frac{1}{3}$$

The second derivative of $p$ with respect to $\omega$ 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 \( \omega \):

\[
p = \frac{\omega}{\sin \omega + m \cos \omega}
\]  

(2.90)

The perturbation load \( P_2 \) has been specified in such a way that the sign of the variables \( \omega, \sin \omega \) 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 \( \omega \) 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 \( P_2 = 0.01 \times P_1 \) 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:

\[
u = \frac{P_a}{AE}
\]  

(2.91)

The gap closes when the load reaches the value \( P_0 \) which makes \( u = s \):

\[
P_0 = \frac{AEs}{a}
\]  

(2.92)

![Diagram](image)

**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](image)

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 $i_1, i_2, i_3$. 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.

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

\[
x = x_1 i_1 + x_2 i_2 + x_3 i_3
\]  

\[
\begin{bmatrix}
x_1 \\
x_2 \\
x_3 \\
\end{bmatrix} = \begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
x_3 \\
\end{bmatrix}
\]

$x_1, x_2, x_3$ location vector of point $P$

$x_k$ global coordinate of point $P$ for base vector $i_k$

$i_k$ base vector of the Euclidian space

The base vectors are the partial derivatives of the location vector $x$ with respect to the coordinates $x_k$:

\[
i_k = \frac{\partial x}{\partial x_k} := x_k
\]  

The entity consisting of the origin $O$ and the base vectors $i_1, i_2, i_3$ is called a global coordinate system.
3.1.2 Local Coordinate System

Figure 3.2 Cartesian local coordinate system

Let \( \mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z \) 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 \( e_k \) of the base vectors \( \mathbf{e}_k \) are specified in the global space:

\[
e_k = e_{k1}\mathbf{i}_1 + e_{k2}\mathbf{i}_2 + e_{k3}\mathbf{i}_3 \quad k \in \{1,2,3\} \tag{3.3}
\]

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 \( \mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z \), 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 \( y_1, y_2, y_3 \).

\[
\mathbf{y} = y_1\mathbf{j}_1 + y_2\mathbf{j}_2 + y_3\mathbf{j}_3 \tag{3.4}
\]

| \( y_1 \) | 1 | 0 | 0 |
| \( y_2 \) | 0 | 1 | 0 |
| \( y_3 \) | 0 | 0 | 1 |

\( \mathbf{y} \) 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 \( \mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3 \):
\[ x = x_0 + y_1 e_1 + y_2 e_2 + y_3 e_3 \]  
(3.5)

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

\[ x_{x0} \]  
global coordinates of the local origin

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

\[ x = x_0 + R y \]  
(3.6)

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

\[ x \]  
global location vector of point P

\[ y \]  
local location vector of point P

\[ x_0 \]  
global location vector of the local origin

\( R \)  
rotation matrix for the coordinate transformation

Since the base vectors \( e_k \) are orthonormal, the rotation matrix is orthonormal and its inverse is equal to its transpose:

\[ e_i^T e_m = \delta_{im} \]  
(3.7)

\[ R^T R = I \]  
(3.8)

\[ R^{-1} = R^T \]  
(3.9)

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 \( u \) if it is expressed as a linear combination of the global base vectors \( i_k \), and by \( v \) if it is expressed as a linear combination of the local base vectors \( j_k \). The coefficients of the linear combinations are called the coordinates of the vector and denoted by \( u_k \) and \( v_k \).

\[ u = u_1 i_1 + u_2 i_2 + u_3 i_3 \]  
(3.10)

\[ v = v_1 j_1 + v_2 j_2 + v_3 j_3 \]
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:

\[ \mathbf{u} = \mathbf{v}_1 \mathbf{e}_1 + \mathbf{v}_2 \mathbf{e}_2 + \mathbf{v}_3 \mathbf{e}_3 \]  
\[ \mathbf{u} = \mathbf{R} \mathbf{v} \]  
\[ \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} \mathbf{e}_1 \\ \mathbf{e}_2 \\ \mathbf{e}_3 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} \]

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 \( v_{km} \).

\[ \mathbf{U} = \sum_{k=1}^{3} \sum_{m=1}^{3} u_{km} \mathbf{i}_k \mathbf{i}_m^T \]  
\[ \mathbf{V} = \sum_{k=1}^{3} \sum_{m=1}^{3} v_{km} \mathbf{j}_k \mathbf{j}_m^T \]

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:

\[ \mathbf{U} = \sum_{k=1}^{3} \sum_{m=1}^{3} v_{km} \mathbf{e}_k \mathbf{e}_m^T \]  
\[ \mathbf{U} = \mathbf{R} \mathbf{V} \mathbf{R}^T \]  
\[ \mathbf{U} \quad \text{matrix containing the global tensor coordinates } u_{km} \]  
\[ \mathbf{V} \quad \text{matrix containing the local tensor coordinates } v_{km} \]

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

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 $\partial C$. The location vector of a material point $P$ in the reference configuration is denoted by $x$, and its coordinates in global space are denoted by $x_1, x_2, x_3$:

$$x = x_1 \mathbf{i}_1 + x_2 \mathbf{i}_2 + x_3 \mathbf{i}_3$$  \hspace{1cm} (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 $\hat{x}$:
\[ \dot{x} = \dot{x}_1 i_1 + \dot{x}_2 i_2 + \dot{x}_3 i_3 \]  
(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 \( \dot{x} \) of a material point in an instant configuration \( \hat{C} \) and its location vector \( x \) in the reference configuration \( C \) is called the displacement vector of the material point. It is denoted by \( u \). 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 \( \hat{C} \).

\[ u = \dot{x} - x \]  
(3.17)

\[ u = u_1 i_1 + u_2 i_2 + u_3 i_3 \]  
(3.18)

\[ u_k = \hat{x}_k - x_k \]  
(3.19)

\( u \) displacement vector of material point \( P \)

\( u_k \) global coordinates of the displacement vector \( u \)

#### 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 \( u \) 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 \( x_A \) and \( x_B \). The square of the distance between points \( A \) and \( B \) in the reference configuration is given by:

\[ s^2 = (x_B - x_A)^T (x_B - x_A) \]

The location vectors of points \( A \) and \( B \) in \( \hat{C} \) are \( x_A + u \) and \( x_B + u \). The square of the distance between points \( A \) and \( B \) in the instant configuration is given by:

\[ \hat{s}^2 = ((x_B + u) - (x_A + u))^T ((x_B + u) - (x_A + u)) \]

\[ = (x_B - x_A)^T (x_B - x_A) = s^2 \]  
(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 \( a \) 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 \( PM \) 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 $\beta$ to position Q. Construct the perpendicular QS to 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 $\mathbf{U}M$ and $\mathbf{UP}$, as follows:

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

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

$$\mathbf{e} = \mathbf{a} \times \mathbf{n} = \frac{1}{r} \mathbf{a} \times (\mathbf{a} \times \mathbf{p})$$

$$\mathbf{p} = \mathbf{p} + r \mathbf{e}(1 - \cos \beta) + r \mathbf{n} \sin \beta$$

$$\mathbf{p} = \mathbf{p} + (1 - \cos \beta) \mathbf{a} \times (\mathbf{a} \times \mathbf{p}) + \sin \beta \mathbf{a} \times \mathbf{p}$$  \hspace{1cm} (3.22)

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

$$\mathbf{a} \times \mathbf{p} = \mathbf{A} \mathbf{p}$$  \hspace{1cm} (3.23)

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

$$\mathbf{p} = \mathbf{R} \mathbf{p}$$  \hspace{1cm} (3.24)

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

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

$$\mathbf{a}^\top \mathbf{a} = 1 \hspace{1cm} \mathbf{A} = -\mathbf{A}^\top$$

$$\mathbf{A}^\top \mathbf{A} = \mathbf{I} - \mathbf{a} \mathbf{a}^\top \hspace{1cm} (\mathbf{A}^\top \mathbf{A})^2 = \mathbf{A}^\top \mathbf{A}$$

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

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

The rotation matrix $\mathbf{R}$ is orthonormal.
**Rotation:** The displacement vector of material point \( P \) due to rotation \( \beta \) about axis \( a \) is the difference of the location vectors of points \( P \) and \( Q \):

\[
\mathbf{u} = \hat{\mathbf{p}} - \mathbf{p} = (\mathbf{R} - \mathbf{I})\mathbf{p}
\]  
(3.27)

Let all material points of a body rotate about the same axis \( a \) through the same angle \( \beta \). 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:

\[
\hat{s}^2 = ((\mathbf{x}_B + \mathbf{u}_B) - (\mathbf{x}_A + \mathbf{u}_A))^T((\mathbf{x}_B + \mathbf{u}_B) - (\mathbf{x}_A + \mathbf{u}_A))
\]
\[
= (\mathbf{x}_B - \mathbf{x}_A)^T\mathbf{R}^T\mathbf{R}(\mathbf{x}_B - \mathbf{x}_A)
\]
\[
= (\mathbf{x}_B - \mathbf{x}_A)^T(\mathbf{x}_B - \mathbf{x}_A)
\]

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 \( \mathbf{dx} \) of the volume is the difference between the location vectors of opposite corner points of the brick shape.

\[
\mathbf{dx} = dx_1\mathbf{i}_1 + dx_2\mathbf{i}_2 + dx_3\mathbf{i}_3
\]
\[
\mathbf{dx}_k = x_{k(Q)} - x_{k(P)}
\]  
(3.28)
The material element is displaced in the instant configuration to another volume in Euclidean space that is no longer brick-shaped. The diagonal vector $d\hat{x}$ of this new volume is given by:

$$d\hat{x} = dx_1b_1 + dx_2b_2 + dx_3b_3 \quad (3.29)$$

$b_k$ instant base vector

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

$$b_k = \frac{\partial \hat{x}}{\partial x_k} = \frac{\partial (x+u)}{\partial x_k} = \frac{\partial \hat{x}}{\partial x_k} + \frac{\partial u}{\partial x_k}$$

$$b_k = i_k + \frac{\partial u_1}{\partial x_k} i_1 + \frac{\partial u_2}{\partial x_k} i_2 + \frac{\partial u_3}{\partial x_k} i_3 \quad (3.30)$$

### 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 $d\hat{x}$ of a material element of the body in the instant configuration can therefore be expressed as a function of the diagonal $dx$ in the reference configuration:

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

$$
\begin{array}{ccc}
\frac{\partial \hat{x}_1}{\partial x_1} & \frac{\partial \hat{x}_1}{\partial x_2} & \frac{\partial \hat{x}_1}{\partial x_3} \\
\frac{\partial \hat{x}_2}{\partial x_1} & \frac{\partial \hat{x}_2}{\partial x_2} & \frac{\partial \hat{x}_2}{\partial x_3} \\
\frac{\partial \hat{x}_3}{\partial x_1} & \frac{\partial \hat{x}_3}{\partial x_2} & \frac{\partial \hat{x}_3}{\partial x_3} \\
\end{array}
\quad \ast
\begin{array}{c}
dx_1 \\
dx_2 \\
dx_3 \\
\end{array}
$$

The matrix $F$ is called the material deformation gradient of the displacement field. The columns of $F$ contain the instant base vectors $b_k$ as defined in (3.30).

$$F = \begin{bmatrix} b_1 & b_2 & b_3 \end{bmatrix} \quad (3.32)$$

$F$ 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:

$$dv = (i_1dx_1) \times (i_2dx_2) \cdot (i_3dx_3) = dx_1dx_2dx_3$$
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:

\[
d\mathbf{\hat{v}} = (\mathbf{b}_1 \, dx_1) \times (\mathbf{b}_2 \, dx_2) \times (\mathbf{b}_3 \, dx_3) = \det \mathbf{F} \, dv
\]

(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 \( \mathbf{\hat{n}} \) and size \( \mathbf{\hat{d}a} \) in an instant configuration of the body. The relationship between the two area vectors is determined by forming a volume with the area vector \( da \) and an arbitrary infinitesimal material vector \( dx \):

Reference configuration: \( \mathbf{d}a = \mathbf{n} \, da \quad \mathbf{d}v = d\mathbf{x}^T da \)

Instant configuration: \( \mathbf{d}\mathbf{\hat{a}} = \mathbf{\hat{n}} \, \mathbf{\hat{d}a} \quad \mathbf{d}\mathbf{\hat{v}} = d\mathbf{\hat{x}}^T \mathbf{\hat{d}a} \)

Transformations: \( d\mathbf{\hat{x}} = \mathbf{F} \, dx \quad d\mathbf{\hat{v}} = \det \mathbf{F} \, dv \)

Substitution:

\[
\mathbf{d}\mathbf{\hat{v}} = \det \mathbf{F} \, dv = \det \mathbf{F} \, d\mathbf{x}^T da = \det \mathbf{F} \, d\mathbf{x}^T \mathbf{\hat{d}a}
\]

(3.34)

\[
\mathbf{d}\mathbf{\hat{v}} = d\mathbf{\hat{x}}^T \mathbf{\hat{d}a} = d\mathbf{x}^T \mathbf{F}^T \mathbf{\hat{d}a}
\]

(3.35)

Since (3.34) and (3.35) are valid for arbitrary vectors \( dx \), the areas in the reference and the instant configurations are related as follows:

\[
\det \mathbf{F} \, da = F^T \, d\mathbf{\hat{a}}
\]

(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 \( \mathbf{H} \):

\[
\mathbf{F} = \mathbf{I} + \mathbf{H}
\]

(3.37)

\[
\mathbf{H} =
\begin{pmatrix}
\frac{\partial u_1}{\partial x_1} & \frac{\partial u_1}{\partial x_2} & \frac{\partial u_1}{\partial x_3} \\
\frac{\partial u_2}{\partial x_1} & \frac{\partial u_2}{\partial x_2} & \frac{\partial u_2}{\partial x_3} \\
\frac{\partial u_3}{\partial x_1} & \frac{\partial u_3}{\partial x_2} & \frac{\partial u_3}{\partial x_3}
\end{pmatrix}
\]

**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:

\[
d\mathbf{\hat{s}}^2 - ds^2 = d\mathbf{x}^T \mathbf{\hat{d}x} - d\mathbf{x}^T dx
\]

\[
= d\mathbf{x}^T ((I + \mathbf{H})^T (I + \mathbf{H}) - I) dx
\]

\[
= d\mathbf{x}^T (\mathbf{H} + \mathbf{H}^T + \mathbf{H}^T \mathbf{H}) dx
\]
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:

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

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

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

$$
\mathbf{b}_i^T \mathbf{b}_i = (\mathbf{i}_i + \frac{\partial u_1}{\partial x_1} \mathbf{i}_1 + \frac{\partial u_2}{\partial x_1} \mathbf{i}_2 + \frac{\partial u_3}{\partial x_1} \mathbf{i}_3)^T (\mathbf{i}_i + \frac{\partial u_1}{\partial x_1} \mathbf{i}_1 + \frac{\partial u_2}{\partial x_1} \mathbf{i}_2 + \frac{\partial u_3}{\partial x_1} \mathbf{i}_3)
$$

$$
= 1 + 2 \left( \frac{\partial u_1}{\partial x_1} \right)^2 + \left( \frac{\partial u_2}{\partial x_1} \right)^2 + \left( \frac{\partial u_3}{\partial x_1} \right)^2 = 1 + 2e_{11}
$$

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

$$
|\mathbf{b}_i| - |\mathbf{i}_i| \approx \sqrt{1 + 2e_{11}} - 1 \approx e_{11}
$$

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

$$
\cos \theta = \frac{\mathbf{b}_1^T \mathbf{b}_2}{|\mathbf{b}_1||\mathbf{b}_2|} \approx \mathbf{b}_1^T \mathbf{b}_2
$$

$$
\mathbf{b}_1^T \mathbf{b}_2 = (\mathbf{i}_i + \frac{\partial u_1}{\partial x_2} \mathbf{i}_1 + \frac{\partial u_2}{\partial x_2} \mathbf{i}_2 + \frac{\partial u_3}{\partial x_2} \mathbf{i}_3)^T (\mathbf{i}_2 + \frac{\partial u_1}{\partial x_2} \mathbf{i}_1 + \frac{\partial u_2}{\partial x_2} \mathbf{i}_2 + \frac{\partial u_3}{\partial x_2} \mathbf{i}_3)
$$

$$
= \frac{\partial u_1}{\partial x_2} + \frac{\partial u_2}{\partial x_2} + \frac{\partial u_3}{\partial x_2} \frac{\partial u_1}{\partial x_2} + \frac{\partial u_2}{\partial x_2} \frac{\partial u_1}{\partial x_2} + \frac{\partial u_3}{\partial x_2} \frac{\partial u_1}{\partial x_2}
$$

$$
= 2e_{12}
$$

The strain $e_{12}$ is a measure of the angle change $\theta$ between vectors $\mathbf{i}_i$ and $\mathbf{i}_j$:

$$
\cos \theta = \sin(\frac{\pi}{2} - \theta) = 2e_{12}
$$

$$
\frac{\pi}{2} - \theta \approx 2e_{12}
$$

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

$$E N = N S$$

$$E n_k = s_k n_k$$

$$N = \begin{bmatrix} n_1 & n_2 & n_3 \end{bmatrix} \quad \quad S = \begin{bmatrix} s_1 & 0 & 0 \\ 0 & s_2 & 0 \\ 0 & 0 & s_3 \end{bmatrix}$$

$n_k$ unit vector in the principal direction $k$

$s_k$ principal axial strain for the direction $n_k$

The diagonal elements $s_k$ of $S$ are called principal strains of the material element; the columns of $N$ 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 $S$ 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 $\hat{d}x$ of the volume is the difference of the location vectors of opposite corner points.

$$d\hat{x} = d\hat{x}_1 i_1 + d\hat{x}_2 i_2 + d\hat{x}_3 i_3$$

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

*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 \( \mathbf{dx} \) of this original volume is given by:

\[
\mathbf{dx} = \mathbf{d}x_1 \hat{b}_1 + \mathbf{d}x_2 \hat{b}_2 + \mathbf{d}x_3 \hat{b}_3
\]  
(3.45)

\( \hat{b}_k \) reference base vector

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

\[
\hat{b}_k = \frac{\partial \hat{x}}{\partial \hat{x}_k} = \frac{\partial (\hat{x} - u)}{\partial \hat{x}_k} = \frac{\partial \hat{x}}{\partial \hat{x}_k} - \frac{\partial u}{\partial \hat{x}_k}
\]
\[
\hat{b}_k = \hat{i}_k - \frac{\partial u_1}{\partial \hat{x}_k} \hat{i}_1 - \frac{\partial u_2}{\partial \hat{x}_k} \hat{i}_2 - \frac{\partial u_3}{\partial \hat{x}_k} \hat{i}_3
\]  
(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 \( \mathbf{dx} \) of the spatial element in the reference configuration can therefore be expressed as a function of the diagonal \( \mathbf{d}x \) in the instant configuration:

\[
\mathbf{dx} = \mathbf{F} \mathbf{d}x
\]  
(3.47)

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

\[
\mathbf{F} = \begin{bmatrix}
\hat{b}_1 \\
\hat{b}_2 \\
\hat{b}_3
\end{bmatrix}
\]  
(3.48)

\( \mathbf{F} \) deformation gradient of the instant configuration

The material and spatial deformation gradients are inverse matrices:

\[
\mathbf{dx} = \mathbf{F} \mathbf{d}x = \mathbf{F} \mathbf{d}x \rightarrow \mathbf{F} \mathbf{F} = \mathbf{I}
\]  
(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):

\[
dv = \det \mathbf{F} \, d\mathbf{v} \\
\det \mathbf{F} \, da = \mathbf{F}^T da
\]

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

**Displacement gradient**: It follows from (3.46) that the deformation gradient \(\mathbf{\hat{F}}\) 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 \(\mathbf{\hat{H}}\):

\[
\mathbf{\hat{F}} = \mathbf{I} - \mathbf{\hat{H}}
\]

\[
\mathbf{\hat{H}} =
\begin{bmatrix}
\frac{\partial u_1}{\partial x_1} & \frac{\partial u_1}{\partial x_2} & \frac{\partial u_1}{\partial x_3} \\
\frac{\partial u_2}{\partial x_1} & \frac{\partial u_2}{\partial x_2} & \frac{\partial u_2}{\partial x_3} \\
\frac{\partial u_3}{\partial x_1} & \frac{\partial u_3}{\partial x_2} & \frac{\partial u_3}{\partial x_3}
\end{bmatrix}
\]

**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:

\[
ds^2 - d\mathbf{s}^2 = d\mathbf{x}^T d\mathbf{x} - d\mathbf{x}^T d\mathbf{x}
\]

\[
= d\mathbf{x}^T \left( (\mathbf{I})^T (\mathbf{I} - \mathbf{\hat{H}}^T (\mathbf{I} - \mathbf{\hat{H}}))) d\mathbf{x}
\]

\[
= d\mathbf{x}^T \left( \mathbf{\hat{H}}^T - \mathbf{\hat{H}}\right) d\mathbf{x}
\]

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

\[
\mathbf{\hat{E}} = \frac{1}{2} (\mathbf{\hat{H}} + \mathbf{\hat{H}}^T - \mathbf{\hat{H}}\mathbf{\hat{H}})
\]

\[
\hat{e}_{km} = \frac{1}{2} \left( \frac{\partial u_k}{\partial x_m} + \frac{\partial u_m}{\partial x_k} - \sum_i \frac{\partial u_k}{\partial x_i} \frac{\partial u_m}{\partial x_i} \right)
\]
**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:

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

$$\hat{\mathbf{E}} \hat{n}_k = \hat{s}_k \hat{n}_k$$

$$\hat{\mathbf{N}} = \begin{pmatrix} \hat{n}_1 & \hat{n}_2 & \hat{n}_3 \end{pmatrix} \quad \hat{\mathbf{S}} = \begin{pmatrix} \hat{s}_1 & 0 & 0 \\ 0 & \hat{s}_2 & 0 \\ 0 & 0 & \hat{s}_3 \end{pmatrix}$$

$k \in \{1, 2, 3\}$ (3.54)

$\hat{n}_k$ unit vector in the principal direction $k$

$\hat{s}_k$ principal axial strain for the direction $\hat{n}_k$

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 $\mathbf{C}$ and $\hat{\mathbf{C}}$. 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](image)

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 $\hat{a}$ when $\Delta\hat{a}$ goes to null is called the stress vector acting on the surface $\hat{n}$ at point $A$ and is denoted by $\hat{t}$:

$$\hat{t}(\hat{P}, \hat{n}) := \lim_{\Delta\hat{a} \to 0} \frac{\Delta\hat{p}}{\Delta\hat{a}}$$  \hspace{1cm} (3.55)

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

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

$$\hat{t} = \hat{t}_1 \hat{i}_1 + \hat{t}_2 \hat{i}_2 + \hat{t}_3 \hat{i}_3$$  \hspace{1cm} (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 $\hat{i}_m$ be denoted by $\hat{t}_m$:

$$\hat{t}_m = \hat{t}_{m1} \hat{i}_1 + \hat{t}_{m2} \hat{i}_2 + \hat{t}_{m3} \hat{i}_3$$  \hspace{1cm} (3.57)

- $\hat{t}_m$ stress vector on the face with outer normal $\hat{i}_m$
- $\hat{t}_{km}$ stress coordinate for base vector $\hat{i}_k$ on face $\hat{i}_m$
The matrix $\hat{T}$ whose columns contain the stress vectors acting on the three positive faces of an element is called the stress tensor of Cauchy:

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

$\hat{T}$ 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](image)

**Figure 3.8** Area vectors of a tetrahedron

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

$$\hat{n} \, d\hat{a} - i_1 \, d\hat{a}_1 - i_2 \, d\hat{a}_2 - i_3 \, d\hat{a}_3 = 0$$

$$d\hat{a}_k = \hat{n}_k \, d\hat{a}$$ \hspace{1cm} (3.59)

$d\hat{a}_k$ area of the tetrahedron face with outer normal $-i_k$

$d\hat{a}$ area of the tetrahedron face with outer normal $\hat{n}$

$\hat{n}_k$ coordinate of unit normal vector $\hat{n}$

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

$$\hat{t} \, d\hat{a} - \hat{t} \, d\hat{a}_1 - \hat{t}_2 \, d\hat{a}_2 - \hat{t}_3 \, d\hat{a}_3 = 0$$ \hspace{1cm} (3.60)

The expression for the stress vector $\hat{t}$ follows by substitution of (3.59) in (3.60):
\[ \hat{t} = \hat{T} \hat{n} \quad (3.61) \]

\[
\begin{array}{c|ccc|c|c}
\hat{t}_1 & \hat{t}_{11} & \hat{t}_{12} & \hat{t}_{13} & \hat{n}_1 \\
\hat{t}_2 & \hat{t}_{21} & \hat{t}_{22} & \hat{t}_{23} & \hat{n}_2 \\
\hat{t}_3 & \hat{t}_{31} & \hat{t}_{32} & \hat{t}_{33} & \hat{n}_3
\end{array}
\]

\[ \hat{t} \quad \text{stress vector on the plane with normal } \hat{n} \]

\[ \hat{T} \quad \text{Cauchy stress tensor} \]

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

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

\( \hat{p}_k \) principal stress acting on plane \( k \)

\( \hat{n}_k \) 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 \( \hat{n}_k \) is null.

### 3.5.3 Equilibrium at a Point

The forces acting on the faces of a spatial element with normals \( \hat{i}_i \) and \( -\hat{i}_i \) 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 \( \hat{q} \) 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:

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

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

$$\hat{t}_{km} = \hat{t}_{mk}$$

(3.65)

3.5.4 Piola-Kirchhoff Stress Tensors

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

$$df = P \, d\hat{a} = \hat{T} \, d\hat{a} = \det F \, \hat{T} \hat{F}^T \, da$$

$$P = \det F \, \hat{T} \hat{F}^T$$

(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 \(i_k\) is given by:

$$p_k = p_{ik} \, i_1 + p_{2k} \, i_2 + p_{3k} \, i_3$$

(3.67)

2. Piola-Kirchhoff stress tensor: Let the force vector defined in (3.67) be referred to the base vectors \(b_1, b_2, b_3\) of the displaced material element:

$$p_k = s_{ik} \, b_1 + s_{2k} \, b_2 + s_{3k} \, b_3$$

$$k \in \{1,2,3\}$$

(3.68)

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

$$P = F \, S$$

(3.69)

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

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

$$\hat{T} = (\det \hat{F}) \, F \, SF^T$$

(3.70)

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

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

(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 \( \sigma \) and the state of strain vector \( \varepsilon \) (Voigt notation):

\[
\sigma = \begin{bmatrix}
\sigma_{11} \\
\sigma_{22} \\
\sigma_{33} \\
\sigma_{23} \\
\sigma_{31} \\
\sigma_{12}
\end{bmatrix} \quad \varepsilon = \begin{bmatrix}
\varepsilon_{11} \\
\varepsilon_{22} \\
\varepsilon_{33} \\
\varepsilon_{23} \\
\varepsilon_{31} \\
\varepsilon_{12}
\end{bmatrix}
\]  

\[(3.72)\]

- \( \sigma_{km} = s_{km} \) engineering stress
- \( \varepsilon_{km} = 2e_{km} \) engineering shear strain, \( k \neq m \)
- \( \varepsilon_{kk} = e_{kk} \) 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 $\sigma$ and the state of strain vector $\epsilon$ is described with the constitutive matrix $C$:

$$\sigma = C \epsilon$$  \hspace{1cm} (3.73)

- $\sigma$: state of stress in (3.72)
- $\epsilon$: state of strain in (3.72)
- $C$: 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 $\nu$. The strain in the direction of a normal stress $\sigma_i$ is $\varepsilon_i = \sigma_i / E$; the strain in the orthogonal direction is $-\nu \varepsilon_i$. The shear strain due to a shear stress $\sigma_k$ is $\varepsilon_k = \sigma_k / G$, where the shear modulus $G$ is a function of $E$ and $\nu$. The inverse of the constitutive matrix is thus:

$$C^{-1} = \begin{bmatrix}
\frac{1}{E} & -\frac{\nu}{E} & -\frac{\nu}{E} & 0 & 0 & 0 \\
-\frac{\nu}{E} & \frac{1}{E} & -\frac{\nu}{E} & 0 & 0 & 0 \\
-\frac{\nu}{E} & -\frac{\nu}{E} & \frac{1}{E} & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & \frac{1}{G} & 0 \\
0 & 0 & 0 & 0 & 0 & \frac{1}{G}
\end{bmatrix}$$ \hspace{1cm} (3.74)

The constitutive matrix is determined by inverting (3.74):

$$C = \begin{bmatrix}
(1-\nu)N & \nu N & \nu N & 0 & 0 & 0 \\
\nu N & (1-\nu)N & \nu N & 0 & 0 & 0 \\
\nu N & \nu N & (1-\nu)N & 0 & 0 & 0 \\
0 & 0 & 0 & G & 0 & 0 \\
0 & 0 & 0 & 0 & G & 0 \\
0 & 0 & 0 & 0 & 0 & G
\end{bmatrix}$$ \hspace{1cm} (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 $y_3$ pointing in the principal direction.

The stress $\sigma_{33}$ causes strains $\varepsilon_{33} = \sigma_{33} / E_1$ and $\varepsilon_{11} = \varepsilon_{22} = -\nu_3 \sigma_{33} / E_1$. A stress $\sigma_{kk}$ in the direction $y_1$ or $y_2$ causes a strain $\varepsilon_{kk} = \sigma_{kk} / E_1$ in that direction, a strain $-\nu_1 \varepsilon_{kk}$ normal to it in the isotropic plane and a strain $-\nu_3 \varepsilon_{kk}$ 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 $\nu_1$ in the isotropic plane, and $E_3, G_3$ and $\nu_3$ 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:

$$
C^{-1} = \begin{pmatrix}
\frac{1}{E_1} & -\frac{\nu_1}{E_1} & -\frac{\nu_3}{E_1} & 0 & 0 & 0 \\
-\frac{\nu_1}{E_1} & \frac{1}{E_1} & -\frac{\nu_3}{E_1} & 0 & 0 & 0 \\
-\frac{\nu_3}{E_1} & -\frac{\nu_3}{E_1} & \frac{1}{E_3} & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{1}{G_3} & 0 & 0 \\
0 & 0 & 0 & 0 & \frac{1}{G_3} & 0 \\
0 & 0 & 0 & 0 & 0 & \frac{1}{G_1}
\end{pmatrix}
$$

(3.76)

- $E_1$: moduli of elasticity
- $G_1$: shear moduli
- $\nu_1$: Poisson numbers

The constitutive matrix is determined by inverting (3.76):

$$
C = \begin{pmatrix}
c_{11} & c_{12} & c_{13} & 0 & 0 & 0 \\
c_{21} & c_{22} & c_{23} & 0 & 0 & 0 \\
c_{31} & c_{32} & c_{33} & 0 & 0 & 0 \\
0 & 0 & 0 & G_3 & 0 & 0 \\
0 & 0 & 0 & 0 & G_3 & 0 \\
0 & 0 & 0 & 0 & 0 & G_1
\end{pmatrix}
$$

(3.77)

$$
N = \frac{E_1}{c_0} \quad G_1 = \frac{E}{2(1+\nu_1)} \quad n_3 = \frac{E_1}{E_3} \\
c_0 = n_3(1-\nu_1^2) - 2\nu_3^2(1+\nu_1) \quad c_1 = n_3 - \nu_3^2 \\
c_2 = \nu_1 n_3 + \nu_3^2 \quad c_3 = \nu_3(1+\nu_1) \quad c_4 = 1-\nu_1^2
$$

72
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 $\sigma_{11}$ causes strains $\varepsilon_{11} = \frac{\sigma_{11}}{E_1}, \varepsilon_{22} = -\nu_1 \frac{\sigma_{11}}{E_1}$, and $\varepsilon_{33} = -\nu_2 \frac{\sigma_{11}}{E_1}$. Stress $\sigma_{22}$ causes strains $\varepsilon_{22} = \frac{\sigma_{22}}{E_2}, \varepsilon_{11} = -\nu_1 \frac{\sigma_{22}}{E_2}$, and $\varepsilon_{33} = -\nu_3 \frac{\sigma_{22}}{E_2}$. Stress $\sigma_{33}$ causes strains $\varepsilon_{33} = \frac{\sigma_{33}}{E_3}, \varepsilon_{11} = -\nu_2 \frac{\sigma_{33}}{E_3}$, and $\varepsilon_{22} = -\nu_3 \frac{\sigma_{33}}{E_3}$. 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 $\nu_i$. The inverse of the constitutive matrix is given by:

$$C^{-1} = \begin{bmatrix}
\frac{1}{E_1} & -\frac{\nu_1}{E_1} & -\frac{\nu_2}{E_1} & 0 & 0 & 0 \\
-\frac{\nu_1}{E_1} & \frac{1}{E_2} & -\frac{\nu_3}{E_2} & 0 & 0 & 0 \\
-\frac{\nu_2}{E_1} & -\frac{\nu_3}{E_1} & \frac{1}{E_3} & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{1}{G_1} & 0 & 0 \\
0 & 0 & 0 & 0 & \frac{1}{G_2} & 0 \\
0 & 0 & 0 & 0 & 0 & \frac{1}{G_3}
\end{bmatrix}$$

The constitutive matrix is determined by inverting (3.78):

$$C = \begin{bmatrix}
c_{11} & c_{12} & c_{13} & 0 & 0 & 0 \\
c_{12} & c_{22} & c_{23} & 0 & 0 & 0 \\
c_{13} & c_{23} & c_{33} & 0 & 0 & 0 \\
0 & 0 & 0 & G_1 & 0 & 0 \\
0 & 0 & 0 & 0 & G_2 & 0 \\
0 & 0 & 0 & 0 & 0 & G_3
\end{bmatrix}$$

$$c_0 = \det \begin{bmatrix} 1 & -\nu_1 & -\nu_2 \\ -\nu_1 & n_2 & -\nu_3 \\ -\nu_2 & -\nu_3 & n_3 \end{bmatrix}$$

$$N = \frac{E_1}{c_0}, \quad n_2 = \frac{E_1}{E_2}, \quad n_3 = \frac{E_1}{E_3}$$

$$c_{11} = n_2 n_3 - \nu_3^2, \quad c_{22} = n_3 - \nu_2^2, \quad c_{33} = n_2 - \nu_1^2$$

$$c_{12} = \nu_1 n_3 + \nu_2 n_3, \quad c_{13} = \nu_2 n_2 + \nu_3 n_3, \quad c_{23} = \nu_3 + \nu_1 n_2$$
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 $u$ and $u_k$. If the displacement vector is described in a local space, the vector and its coordinates are denoted by $v$ and $v_k$.

Traction: 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 $\hat{n}$ of the instant configuration, the traction and its coordinates are denoted by $t$ and $t_k$. The traction is related to the outer normal and the Cauchy stress tensor $\hat{T}$ by (3.61):

$$ t = \hat{T}\hat{n} $$  \hspace{1cm} (3.80)

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

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

$$ p = Pn $$  \hspace{1cm} (3.81)

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

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

$$ p = FSn $$  \hspace{1cm} (3.82)

$F$ material deformation gradient

$S$ 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 $e_k$ 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):

\[ \mathbf{x} = \mathbf{x}_0 + \mathbf{R} \mathbf{y} \]  
(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):

\[ \mathbf{v}_u = \mathbf{R}^T \mathbf{u}_v \]  
(3.84)

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

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

\[ \mathbf{t}^{(y)}_u = \mathbf{R}^T \mathbf{t}^{(x)}_v \]  
(3.85)

\( \mathbf{t}^{(y)}_u \), vectors containing the traction coordinates of point P in global space
\( \mathbf{t}^{(x)}_v \), 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 \( \mathbf{C}_t \) and a set of prescribed displacement coordinates \( \mathbf{C}_u \) are defined. For every local or global coordinate axis at every point on the surface, a boundary condition is contributed to either \( \mathbf{C}_t \) or \( \mathbf{C}_u \).

\[ \mathbf{C}_u := \{ \mathbf{u}_k \mid \mathbf{x} \in \partial \mathcal{C} \land \mathbf{u}_k(\mathbf{x}) \text{ is prescribed} \} \]  
(3.86)

\[ \mathbf{C}_t := \{ \mathbf{t}^{(y)}_k \mid \mathbf{x} \in \partial \mathcal{C} \land \mathbf{t}^{(y)}_k(\mathbf{x}) \text{ is prescribed} \} \]  
(3.87)

\( \mathbf{u}_k \)  prescribed global or local displacement coordinate
\( \mathbf{t}^{(y)}_k \)  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**

\[
\hat{t}_{k,1} + \hat{t}_{k,2} + \hat{t}_{k,3} + \hat{\rho} q_k = 0 \quad (3.89)
\]

- \( \hat{t}_{k,m} \) 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**

\[
\sigma = \mathbf{C} \varepsilon \quad (3.90)
\]

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

**stress vector on a plane**

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

- \( \hat{\mathbf{t}} \) stress vector on a plane of the instant configuration with normal \( \hat{\mathbf{n}} \)
- \( \mathbf{T} \) 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**

\[
x \in \partial \mathbf{C} \land \mathbf{u}_j \in \mathbf{C}_u \quad \Rightarrow \quad \mathbf{u}_j = \mathbf{u}_j \quad (3.92)
\]

\[
x \in \partial \mathbf{C} \land \hat{\mathbf{t}}_j \in \mathbf{C}_t \quad \Rightarrow \quad \hat{\mathbf{t}}_j = \hat{\mathbf{t}}_j \quad (3.93)
\]

- \( \mathbf{C}_u \) prescribed displacement coordinates, see (3.86)
- \( \mathbf{C}_t \) 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 \( u(x) \) 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 \( x \) of the instant configuration:

\[
\begin{align*}
    r_{km}(x) &= e_{km} - \frac{1}{2} (u_{k,m} + u_{m,k} + \sum_i u_{ik} u_{im}) & i,k,m \in \{1,2,3\} \\
    r_{2k}(x) &= \hat{t}_{k,11} + \hat{t}_{k,22} + \hat{t}_{k,33} + \hat{\rho} q_k & k \in \{1,2,3\} \\
    r_{ij}(x) &= \sigma - C \varepsilon 
\end{align*}
\]  

(3.94)  
(3.95)  
(3.96)

The following residual functions are defined at every point \( x \) of the surface of the instant configuration:

\[
\begin{align*}
    r_s(x) &= \hat{t} - \hat{T} \hat{n} \\
    r_{ji} &= u_j - u_i & u_i \in \mathbb{C}_u \\
    r_{sj} &= \hat{t}_j - \hat{\tau}_i & \hat{\tau}_i \in \mathbb{C}_t
\end{align*}
\]  

(3.97)  
(3.98)  
(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 \( r_{km} \) and \( r_s \) 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 \( \delta u \) of the displacements and the variations \( \delta \hat{t} \) 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 \( \delta u_k \) and the residuals \( r_3 \) with the displacement variation \( \delta u \) as weight functions. The products are integrated over the volume of the instant configuration. The residuals \( r_4 \) are multiplied with the displacement variations \( \delta u \), the residuals \( r_5 \) with the stress variations \( \delta \hat{t} \), and the residuals \( r_6 \) with the displacement variations \( \delta u \) 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:

\[
\int \sum \sum (\delta u \frac{\partial \hat{t}}{\partial x_m}) d\hat{v} = -\int \sum \delta u_i \hat{\rho} q_i d\hat{v} + \int \sum \delta u_i (\sum \hat{t}_m \hat{n}_m - \hat{t}) d\hat{a} + \int \delta u_i (\hat{t}_i - \hat{t}) d\hat{a} + \int \delta \hat{t}_i (u_i - u) d\hat{a} \quad (3.100)
\]

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

\[
\int \sum \sum (\delta u_i \frac{\partial \hat{t}}{\partial x_m}) d\hat{v} = -\int \sum \sum \hat{t}_m \delta (\frac{\partial u}{\partial x_m}) d\hat{v} + \int \sum \sum \delta u_i \hat{t}_m n_m d\hat{a} \quad (3.101)
\]

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

\[
-\int \sum \sum \hat{t}_m \delta (\frac{\partial u}{\partial x_m}) d\hat{v} = -\int \sum \delta u_i \hat{\rho} q_i d\hat{v} - \int \sum \delta u_i \hat{t}_i + \int \delta u_i (\hat{t}_i - \hat{t}) d\hat{a} + \int \delta \hat{t}_i (u_i - u) d\hat{a} \quad (3.102)
\]

The integral over \( \partial \hat{C} \) is split into integrals for the coordinates associated with \( C_u \) and with \( C_i \):

\[
\int \sum \delta u_i \hat{t}_i d\hat{a} = \int \sum \delta u_i \hat{t}_i d\hat{a} + \int \delta u_i \hat{t}_i d\hat{a} \quad (3.103)
\]

Equation (3.103) is substituted on the right-hand side of equation (3.102) and the integral containing \( u_i - u \) is set to null by choosing \( u_i = u \) on \( C_u \):

\[
\int \sum \sum \hat{t}_m \delta (\frac{\partial u}{\partial x_m}) d\hat{v} = \int \sum \delta u_i \hat{\rho} q_i d\hat{v} + \int \sum \delta u_i \hat{t}_i d\hat{a} + \int \delta u_i \hat{t}_i d\hat{a} \quad (3.104)
\]

\( u_i \in C_u \rightarrow u_i = u \)

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 $\hat{dv}$ 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 $E$ and of the 2. Piola-Kirchhoff stress tensor $\hat{T}$:

$$\sum_{i} \sum_{m} \hat{t}_{im} \delta(\frac{\partial u_i}{\partial \hat{x}_m}) \ d\hat{v} = \sum_{i} \sum_{m} \sum_{j} \sum_{k} s_{jk} \frac{\partial \hat{x}_i}{\partial x_j} \frac{\partial \hat{x}_m}{\partial x_k} \delta(\frac{\partial u_i}{\partial \hat{x}_m}) \ dv$$

$$= \sum_{i} \sum_{j} \sum_{k} s_{jk} \frac{\partial \hat{x}_i}{\partial x_j} \delta(\frac{\partial u_i}{\partial \hat{x}_m}) \ dv$$

$$= \sum_{i} \sum_{j} \sum_{k} s_{jk} (\frac{\partial \hat{x}_i}{\partial x_j} + \frac{\partial u_k}{\partial x_j}) \delta(\frac{\partial u_i}{\partial \hat{x}_m}) \ dv$$

$$= \sum_{i} \sum_{j} \sum_{k} s_{jk} [\delta(\frac{\partial u_i}{\partial \hat{x}_m}) + \frac{\partial u_k}{\partial x_j} \delta(\frac{\partial u_i}{\partial \hat{x}_m})] \ dv$$

$$= \sum_{i} \sum_{j} \sum_{k} \frac{1}{2} s_{jk} \delta(\frac{\partial u_i}{\partial \hat{x}_m}) + \frac{\partial u_k}{\partial x_j} \delta(\frac{\partial u_i}{\partial \hat{x}_m}) \ dv$$

$$= \sum_{i} \sum_{j} \sum_{k} \delta e_{jk} s_{jk} \ dv \quad (3.105)$$

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 $\hat{d}a$ is replaced by the reference vector $da$ with (3.36) and the Cauchy stress tensor $\hat{T}$ is replaced by the 2. Piola-Kirchhoff stress tensor $\hat{T}$ with (3.66):

$$\sum \delta u_i \hat{t}_i \ d\hat{a} = \delta u^T \hat{t} \ d\hat{a} = \delta u^T \hat{T} \ d\hat{a} = \delta u^T F^T d\mathbf{a} \ (\text{det}F)$$

$$= \delta u^T \mathbf{P} \ d\mathbf{a}$$

$$= \delta u^T \mathbf{p} \ da \quad (3.106)$$

Since the mass of an element does not change with the configuration of the body, the force acting on an element remains constant:

$$\rho q \ d\hat{v} = \rho q \ dv \quad (3.107)$$

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:

$$\int_{\hat{C}} \delta e^T \sigma \ dv = \int_{\hat{C}} \delta u^T q \ dv + \int_{\hat{C}} \delta u^T p \ da + \int_{\hat{C}} \delta u^T \mathbf{p} \ da \quad (3.108)$$

$u_i \text{ in } \hat{C}_u \rightarrow u_i = \hat{u}_i$

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 \( \bar{u} \) of a point P in the unknown instant state is the vector sum of the displacement u of point P in the known instant configuration and a displacement increment \( \Delta u \) from the known to the unknown instant configuration:

\[
\bar{u} = u + \Delta u
\]  

\( \bar{u} \) displacement in the unknown instant configuration \( \bar{C} \)  
\( u \) displacement in the known instant configuration \( \hat{C} \)  
\( \Delta u \) 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 \( \Delta \).
strain \[ \bar{\varepsilon}_{im} = \varepsilon_{im} + \Delta \varepsilon_{im} \] (3.110)
stress \[ \bar{s}_{im} = s_{im} + \Delta s_{im} \] (3.111)
load \[ \bar{q}_i = q_i + \Delta q_i \] (3.112)
traction \[ \bar{t}_i = \bar{t}_i + \Delta \bar{t}_i \] (3.113)

**Incremental strain-displacement relations**: Consider the coordinates of the Green strain tensor in the known and the unknown instant configuration:

\[ e_{im} = \frac{1}{2}(u_{im} + u_{mj} + \sum_k u_{k,j} u_{k,m}) \] (3.114)

\[ \bar{e}_{im} = \frac{1}{2}(\bar{u}_{im} + \bar{u}_{mj} + \sum_k \bar{u}_{k,j} \bar{u}_{k,m}) \]

\[ \bar{e}_{im} = \frac{1}{2}(u_{im} + u_{mj} + \Delta u_{im} + \Delta u_{mj} + \sum_k (u_{k,i} + \Delta u_{k,i})(u_{k,m} + \Delta u_{k,m})) \]

The difference of the two strain tensors is split into a linear and a nonlinear part:

\[ \Delta e_{im} = \Delta e_{im}^0 + \Delta e_{im}^n \] (3.115)

\[ \Delta e_{im}^0 = \frac{1}{2}(\Delta u_{im} + \Delta u_{mj} + \sum_k (u_{k,i} \Delta u_{k,m} + u_{k,m} \Delta u_{k,i})) \] (3.116)

\[ \Delta e_{im}^n = \frac{1}{2} \sum_k \Delta u_{k,i} \Delta u_{k,m} \] (3.117)

\[ \Delta e_{im}^o \] linear component of the strain increment from \( \hat{C} \) to \( \bar{C} \)

\[ \Delta e_{im}^n \] nonlinear component of the strain increment from \( \hat{C} \) to \( \bar{C} \)

**Variation of the state variables**: The variation \( \delta \bar{u} \) 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 \( \delta u \) for that state is set to null:

\[ \delta \bar{u} = \delta (u + \Delta u) = \delta u + \delta (\Delta u) = \delta (\Delta u) \] (3.118)

\[ \delta \bar{u} \] variation of the displacement in the unknown instant configuration

\[ \delta (\Delta u) \] variation of the displacement increment

The variation \( \delta \bar{e}_{im} \) 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 \( \delta e_{im} \) for that state is set to null:

\[ \delta \bar{e}_{im} = \delta (\Delta e_{im}) = \delta (\Delta e_{im}^0) + \delta (\Delta e_{im}^n) \] (3.119)

\[ \delta (\Delta e_{im}^0) = \frac{1}{2} \left( \delta (u_{im}) + \delta (u_{mj}) + \sum_k (u_{k,i} \delta (\Delta u_{k,m}) + u_{k,m} \delta (\Delta u_{k,i})) \right) \]

\[ \delta (\Delta e_{im}^n) = \frac{1}{2} \sum_k (\Delta u_{k,i} \delta (\Delta u_{k,m}) + \Delta u_{k,m} \delta (\Delta u_{k,i})) \]

\[ \delta (\Delta e_{im}^0) \] variation of the linear strain increment

\[ \delta (\Delta e_{im}^n) \] 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:

\[
\begin{align*}
\int \sum_{i} \sum_{m} \delta e^{n}_{i,m} \bar{s}_{i,m} \, dv &= \int \sum_{i} \sum_{m} \delta (\Delta e^{0}_{i,m} + \Delta e^{n}_{i,m}) (s_{i,m} + \Delta s_{i,m}) \, dv \\
&= \int \sum_{i} \sum_{m} \delta (\Delta e^{0}_{i,m}) s_{i,m} \, dv + \int \sum_{i} \sum_{m} \delta (\Delta e^{0}_{i,m}) \Delta s_{i,m} \, dv + \\
&\quad \int \sum_{i} \sum_{m} \delta (\Delta e^{0}_{i,m}) s_{i,m} \, dv + \int \sum_{i} \sum_{m} \delta (\Delta e^{0}_{i,m}) \Delta s_{i,m} \, dv \quad (3.120)
\end{align*}
\]

The displacement variation (3.118) is substituted on the right-hand side of (3.108):

\[
\int \sum_{i} \delta \bar{u}_{i} \bar{q}_{i} \, dv + \int \sum_{i} \delta \bar{u}_{i} \bar{p}_{i} \, da + \int \delta \bar{u} \bar{p} \, da = \int \delta \bar{u} \bar{p} \, da \quad (3.121)
\]

\[
\begin{align*}
&\quad \int \sum_{i} \delta (\Delta u_{i}) q_{i} \, dv + \int \delta (\Delta u_{i}) p_{i} \, da + \int \delta (\Delta u_{i}) p_{i} \, da + \\
&\quad \int \sum_{i} \delta (\Delta u_{i}) q_{i} \, dv + \int \delta (\Delta u_{i}) p_{i} \, da + \int \delta (\Delta u_{i}) p_{i} \, da
\end{align*}
\]

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 \( \delta e \):

\[
\begin{align*}
\delta e &= \int \sum_{i} \delta (\Delta u_{i}) q_{i} \, dv + \int \sum_{i} \delta (\Delta u_{i}) p_{i} \, da + \int \delta (\Delta u_{i}) q_{i} \, dv - \int \sum_{i} \sum_{m} \delta (\Delta e^{0}_{i,m}) s_{i,m} \, dv \quad (3.122)
\end{align*}
\]

The error term in equation (3.122) is null if the displacements and reactions in the computed instant configuration \( \hat{C} \) are exact. The last term in (3.120) is neglected because it is quadratic in the displacement increments \( \Delta u_{i} \). Expressions (3.120) to (3.122) are substituted into the material governing equations (3.108):

\[
\begin{align*}
\int \sum_{i} \sum_{m} \delta (\Delta e^{0}_{i,m}) \Delta s_{i,m} \, dv + \int \sum_{i} \sum_{m} \delta (\Delta e^{n}_{i,m}) s_{i,m} \, dv &= \delta e + \\
&\quad \int \sum_{i} \sum_{m} \delta (\Delta u_{i}) q_{i} \, dv + \int \delta (\Delta u_{i}) p_{i} \, da + \int \delta (\Delta u_{i}) p_{i} \, da
\end{align*}
\]

The variation of the nonlinear strain increment in the second term on the left-hand side of (3.123) is replaced by (3.119):

\[
\begin{align*}
\sum_{i} \sum_{m} \delta (\Delta e^{n}_{i,m}) s_{i,m} &= \sum_{i} \sum_{m} \frac{1}{2} s_{i,m} \sum_{k} \delta (\Delta u_{k,i}) + \Delta u_{k,m} \delta (\Delta u_{k,i}) \quad (3.124)
\end{align*}
\]

The derivatives \( \Delta u_{k,m} \) of \( \Delta u_{k} \) are collected in a vector \( h_{k} \) so that (3.124) becomes:

\[
\sum_{i} \sum_{m} \delta (\Delta e^{n}_{i,m}) s_{i,m} = \sum_{k} \delta (\Delta h_{k})^{T} S \Delta h_{k} \quad (3.125)
\]
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.

\[
\int_C \delta(\Delta \epsilon^0)^T C \Delta \epsilon^0 \, dv + \sum_i \int_C \delta(\Delta h_i)^T S \Delta h_i \, dv = \delta e + \int_C \delta(\Delta u)^T q \, dv + \int_{C_i} \delta(\Delta u)^T p \, da + \int_C \delta(\Delta u)^T p \, da
\]  \hspace{1cm} (3.126)

The error term (3.122) is also expressed in matrix form:

\[
\delta e = \int_C \delta(\Delta u)^T q \, dv + \int_{C_i} \delta(\Delta u)^T p \, da + \int_C \delta(\Delta u)^T p \, da - \int_C \delta(\Delta \epsilon^0)^T \sigma \, dv
\]  \hspace{1cm} (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 $w(x)$ 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 $w(x)$ completely, an interpolation function $w_e(x)$ is defined for each element of the body. This function yields the value of $w$ at an arbitrary point $x$ inside the element as a function of the coordinates of $x$ and selected nodal values of $w$. Outside the element, the value of the function is null. The field $w(x)$ is the sum of the element interpolation functions.

$$w(x) = \sum_{e} w_e$$  \hspace{1cm} (3.128)

$x \in E: \quad w_e(x) = f(x, w_{n0}, \ldots, w_{nk})$

$x \notin E: \quad w_e(x) = 0$

$w_{ni}$ 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 $u(x)$ of the body is expressed as a sum of element displacement functions $u_e(x)$ in analogy to (3.128). Some of the nodal displacement variables in $u(x)$ are prescribed, the others are treated as unknowns. The surface tractions $t(x)$ 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:

\[ \mathbf{u}_e = T_e \mathbf{u}_s \]  
\[ \mathbf{u}_r = T_r \mathbf{u}_s \]

\( \mathbf{u}_e \) volume element displacement vector  
\( \mathbf{u}_r \) surface element displacement vector  
\( \mathbf{u}_s \) system displacement vector  
\( T_e \) topology matrix for volume element e  
\( T_r \) 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.

\[ \mathbf{u}_s = \begin{bmatrix} \ldots \\ x \\ \ldots \\ x \\ x \\ \ldots \\ x \\ \ldots \end{bmatrix} \quad \mathbf{q}_s = \begin{bmatrix} \ldots \\ \circ \\ \ldots \\ \circ \\ x \\ \ldots \\ x \\ \ldots \end{bmatrix} \quad \mathbf{r}_s = \begin{bmatrix} \ldots \\ \circ \\ \ldots \\ \circ \\ x \\ \ldots \\ x \\ \ldots \end{bmatrix} \quad \mathbf{s}_s = \begin{bmatrix} \ldots \\ f \\ \ldots \\ t \\ t \\ \ldots \\ t \\ f \end{bmatrix} \]  
(3.131)

\( \ldots \) computed value  
\( t \) true  
\( x \) prescribed value  
\( f \) false  
\( \circ \) 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 \( s \):

\[
\begin{align*}
\mathbf{w}_k^{(s)}(\mathbf{x}) &= g_k^T(\mathbf{x}) \mathbf{w}_e^{(s)} \\
\mathbf{w}_k^{(s+1)}(\mathbf{x}) &= g_k^T(\mathbf{x}) \mathbf{w}_e^{(s+1)}
\end{align*}
\]

Since the coordinates \( \mathbf{w}_k^{(s)} \) and \( \mathbf{w}_k^{(s+1)} \) are associated with the same base vector, they can be subtracted to yield the increment \( \Delta \mathbf{w}_k^{(s)} \) in the coordinate \( w_k \) during step \( s \):

\[
\Delta \mathbf{w}_k^{(s)} = \mathbf{w}_k^{(s+1)} - \mathbf{w}_k^{(s)}
\]

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:

\[
\begin{align*}
\mathbf{w}_s &= \mathbf{w}_1^{(s)} \mathbf{f}_1^{(2)} + \mathbf{w}_2^{(s)} \mathbf{f}_2^{(s)} + \mathbf{w}_3^{(s)} \mathbf{f}_3^{(s)} \\
\mathbf{w}_{s+1} &= \mathbf{w}_1^{(s+1)} \mathbf{f}_1^{(s+1)} + \mathbf{w}_2^{(s+1)} \mathbf{f}_2^{(s+1)} + \mathbf{w}_3^{(s+1)} \mathbf{f}_3^{(s+1)}
\end{align*}
\]

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:

\[
\mathbf{u} = S_e \mathbf{u}_e = S_e \mathbf{T}_e \mathbf{u}_e \quad \mathbf{x} \in C_e \tag{3.136}
\]

\[
\mathbf{u} = S_r \mathbf{u}_r = S_r \mathbf{T}_r \mathbf{u}_s \quad \mathbf{x} \in C_r \tag{3.137}
\]

\( S_e(\mathbf{x}) \) form matrix for volume element \( e \)

\( S_r(\mathbf{x}) \) form matrix for surface element \( r \)

The interpolation of the variation of the displacement follows from (3.118):

\[
\delta (\mathbf{u}) = S_e \mathbf{T}_e \delta (\mathbf{u}_e) \quad \mathbf{x} \in C_e \tag{3.138}
\]

\[
\delta (\mathbf{u}) = S_r \mathbf{T}_r \delta (\mathbf{u}_r) \quad \mathbf{x} \in C_r \tag{3.139}
\]

The partial derivative of a displacement coordinate follows from (3.136):

\[
\frac{\partial \mathbf{u}_i}{\partial x_m} = \frac{\partial s_i^T}{\partial x_m} \mathbf{T}_e \mathbf{u}_e := g_{im}^T \mathbf{T}_e \mathbf{u}_e \tag{3.140}
\]

\( s_i^T \) \( i \)-th row of matrix \( S_e \)

\( g_{im} \) interpolation vector for displacement derivative \( \mathbf{u}_{im} \)

The three derivatives of a displacement coordinate are collected in a vector \( \mathbf{h}_k \):

\[
\mathbf{h}_k = G_k \mathbf{T}_e \mathbf{u}_e \tag{3.142}
\]

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):

\[
\Delta \varepsilon^0 = B_e \Delta \mathbf{u}_e = B_e \mathbf{T}_e \Delta \mathbf{u}_e \tag{3.143}
\]
**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:

\[ p = S_r p_r \]  
(3.144)

- \( p_r \): stress vector for element \( r \) (intensities at the nodes)
- \( S_r \): stress interpolation matrix

The stresses acting on the surface of the body are replaced by nodal forces of equal residual weight:

\[ \int_C \delta(\Delta u)^T p \, da = \delta(\Delta u_s)^T \sum_r T_r^T \int_{C_r} S_r^T S_r p_r \, da = \delta(\Delta u_s)^T (q_s + r_s) \]  
(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 \( \delta e \) for the governing equations of the configuration at the beginning of the step.

The error term \( \delta e \) 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):

\[ \int_C \delta(\Delta u)^T q \, dv = \delta(\Delta u_s)^T \sum_e T_e^T \int_{C_e} S_e^T q \, dv \]  
(3.146)

The second and third integrals in (3.127) are replaced by (3.145):

\[ \int_{C_s} \delta(\Delta u_s)^T p_r \, da + \int_{C_s} \delta(\Delta u_s)^T p \, da = \delta(\Delta u_s)^T (q_s + r_s) \]  
(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:

\[ \int_C \delta(\Delta \epsilon^0)^T \sigma \, dv = \delta(\Delta u_s)^T \sum_e T_e^T \int_{C_e} B_e^T \sigma \, dv \]  
(3.148)

Results (3.146) to (3.148) are substituted into the error term (3.127). The system error vector \( e_s \) is defined as follows:

\[ \delta e = \delta(\Delta u_s)^T e_s \]

\[ e_s = \sum_e T_e^T \int_{C_e} S_e^T q_e \, dv + q_s + r_s - \sum_e T_e^T \int_{C_e} B_e^T \sigma \, dv \]  
(3.149)

- \( e_s \): 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):

$$\int_C \delta(\Delta \mathbf{e}_0) \mathbf{C} \Delta \mathbf{e}_0 \, dv = \delta(\Delta \mathbf{u}_s)^T \left( \sum_e \mathbf{T}_e^T \mathbf{K}_{ve} \mathbf{T}_e \right) \Delta \mathbf{u}_s$$ (3.150)

$$\mathbf{K}_{ve} = \int_C \mathbf{B}_e^T \mathbf{C} \mathbf{B}_e \, dv$$

\(\mathbf{K}_{ve}\) deformation stiffness matrix of element \(e\)

The displacement derivatives (3.142) are substituted into the second integral:

$$\int_C \mathbf{K}_{ve} = \int_C \sum_i \mathbf{G}_i^T \mathbf{S} \mathbf{G}_i \, dv$$

\(\mathbf{K}_{ge}\) geometric stiffness matrix of element \(e\)

The displacement increment (3.136) is substituted into the first integral on the right:

$$\int_C \delta(\Delta \mathbf{u})^T \Delta \mathbf{p} \, dv = \delta(\Delta \mathbf{u}_s)^T \left( \sum_e \mathbf{T}_e^T \mathbf{q}_e \right)$$ (3.152)

$$\mathbf{q}_e = \int_C \mathbf{S}_e^T \Delta \mathbf{p} \, dv$$

\(\mathbf{q}_e\) volume load vector of element \(e\)

The second and third integrals on the right are replaced by (3.145):

$$\int_C \delta(\Delta \mathbf{u})^T \Delta \mathbf{p} \, da + \int_C \delta(\Delta \mathbf{u})^T \Delta \mathbf{p} \, da = \delta(\Delta \mathbf{u}_s)^T \left( \Delta \mathbf{q}_s + \Delta \mathbf{r}_s \right)$$ (3.153)

Expressions (3.149) to (3.153) are substituted into (3.126) to yield the algebraic form of the incremental governing equations:

$$\delta(\Delta \mathbf{u}_s)^T \mathbf{K}_s \Delta \mathbf{u}_s = \delta(\Delta \mathbf{u}_s)^T \left( \Delta \mathbf{q}_s + \Delta \mathbf{r}_s + \mathbf{e}_s \right)$$ (3.154)

$$\mathbf{K}_s = \sum_e \mathbf{T}_e^T \left( \mathbf{K}_{ve} + \mathbf{K}_{ge} \right) \mathbf{T}_e$$

\(\Delta \mathbf{q}_s\) 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](image)

**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

\[ x_1, x_2 \quad \text{global space with base } \mathbf{i}_1, \mathbf{i}_2 \]
\[ y_1, y_2 \quad \text{reference space with base } \mathbf{e}_1, \mathbf{e}_2 \]
\[ z_1, z_2 \quad \text{instant space with base } \mathbf{f}_1, \mathbf{f}_2 \]
\[ s_1, s_2 \quad \text{support space with base } \mathbf{h}_1, \mathbf{h}_2 \]

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 \( x_1, x_2 \):

\[
x = x_1 \mathbf{i}_1 + x_2 \mathbf{i}_2
\]

\[
\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} x_1 + \begin{bmatrix} 0 \\ 1 \end{bmatrix} x_2
\]
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:

\[
\begin{align*}
\mathbf{x}_A &= \begin{bmatrix} x_{1A} \\ x_{2A} \end{bmatrix}, & \mathbf{x}_B &= \begin{bmatrix} x_{1B} \\ x_{2B} \end{bmatrix}
\end{align*}
\]

- \( \mathbf{x}_A \) 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:

\[
L = \sqrt{(x_{1B} - x_{1A})^2 + (x_{2B} - x_{2A})^2}
\]

\[
\mathbf{e}_1 := \begin{bmatrix} e_{11} \\ e_{21} \end{bmatrix} = \frac{1}{L} \begin{bmatrix} x_{1B} - x_{1A} \\ x_{2B} - x_{2A} \end{bmatrix}
\]

The unit base vector \( \mathbf{e}_2 \) is chosen so that it is orthogonal to \( \mathbf{e}_1 \):

\[
\mathbf{e}_2 = \begin{bmatrix} e_{12} \\ e_{22} \end{bmatrix} = \begin{bmatrix} -e_{21} \\ e_{11} \end{bmatrix}
\]

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:

\[
\mathbf{x} = \mathbf{x}_A + \mathbf{R}_{xy} \mathbf{y}
\]

\[
\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_{1A} \\ x_{2A} \end{bmatrix} + \begin{bmatrix} e_{11} & e_{12} \\ e_{21} & e_{22} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}
\]

\( \mathbf{R}_{xy} \) 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:

\[
\hat{x}_A = \begin{pmatrix} \hat{x}_{1A} \\ \hat{x}_{2A} \end{pmatrix}, \quad \hat{x}_B = \begin{pmatrix} \hat{x}_{2B} \\ \hat{x}_{2B} \end{pmatrix}
\]  

\( \hat{x}_A, \hat{x}_B \) 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 \( f_1 \) for axis \( z_1 \) points from the start node \( \hat{A} \) to the end node \( \hat{B} \) of the bar. The global coordinates of \( f_1 \) are computed as follows:

\[
\hat{L} = \sqrt{(\hat{x}_{1B} - \hat{x}_{1A})^2 + (\hat{x}_{2B} - \hat{x}_{2A})^2}
\]  

\( \hat{f}_1 := \begin{pmatrix} f_{11} \\ f_{21} \end{pmatrix} = \frac{1}{\hat{L}} \begin{pmatrix} \hat{x}_{1B} - \hat{x}_{1A} \\ \hat{x}_{2B} - \hat{x}_{2A} \end{pmatrix} \)  

The unit base vector \( f_2 \) is chosen so that it is orthogonal to \( f_1 \):

\[
f_2 = \begin{pmatrix} f_{12} \\ f_{22} \end{pmatrix} = \begin{pmatrix} -f_{21} \\ f_{11} \end{pmatrix}
\]  

The columns of the rotation matrix \( R_{xz} \) for the transformation of the coordinates of a point between the global space and the reference space contain the base vectors:

\[
x = \hat{x}_A + R_{xz} z
\]  

\[
\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} x_{1A} \\ x_{2A} \end{pmatrix} + \begin{pmatrix} f_{11} & f_{12} \\ f_{21} & f_{22} \end{pmatrix} * \begin{pmatrix} z_1 \\ z_2 \end{pmatrix}
\]

\( R_{xz} \) rotation matrix global - instant space

The transformation between reference and instant space follows from (4.5) and (4.9):

\[
x_A + R_{xy} y = \hat{x}_A + R_{xz} z
\]

\[
y = y_A + R_{yz} z
\]

\[
y_A = R_{xy}^T (\hat{x}_A - x_A)
\]

\[
R_{yz} = R_{xy}^T R_{xz}
\]  

\( R_{yz} \) 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 $s_i, s_2$ 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

$\mathbf{h}_m$ coordinates of base vector $\mathbf{h}_m$

The transformation from support to reference space follows from (4.5) and (4.12):

$$\begin{bmatrix} x_A \\ y \end{bmatrix} + \mathbf{R}_{xy} \begin{bmatrix} x_N \\ y \end{bmatrix} = \begin{bmatrix} x_n \\ y \end{bmatrix} + \mathbf{R}_{xs} \mathbf{s}$$

$$\mathbf{s} = \mathbf{s}_A + \mathbf{R}_{xy} \mathbf{y} \quad (4.13)$$

$$\mathbf{s}_A = \mathbf{R}_{xs}^T (\mathbf{x}_A - \mathbf{x}_N)$$

$$\mathbf{R}_{xy} = \mathbf{R}_{xs}^T \mathbf{R}_{xy} \quad (4.14)$$

$\mathbf{R}_{xy}$ 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:

$$y_1 = \frac{1}{2} (1 + z) \hat{a} \quad -1 \leq z \leq 1$$

$$z_1 = \frac{1}{2} (1 + z) \hat{a}$$

$z$ normalised axial coordinate of the bar

$a$ length of the bar in the reference configuration

$\hat{a}$ 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 $\mathbf{i}_i$, and by $\mathbf{v}$ if it is expressed as a linear combination of the local base vectors $\mathbf{j}_j$. The coordinates of the vector transform according to (3.12):

$$\mathbf{u} = \mathbf{R} \mathbf{v}$$

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} e_{11} & e_{12} \\ e_{21} & e_{22} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$$
Consider a matrix denoted by $U$ if it is expressed as a linear combination of vector products of the global base vectors $i_k$, and by $V$ if it is expressed as a linear combination of vector products of the local base vectors $j_k$. The coordinates of the matrix transform according to (3.14):

\[
U = RVR^T
\]

(4.17)

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.

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 $n$ is parallel to the axial base vector $e_i$. The truss hypothesis postulates the following properties of the material points of the bar in the instant configuration $\hat{C}$:

(a) The axis $\hat{A}\hat{P}\hat{B}$ of the bar in the instant configuration is a straight line.
(b) The displaced cross-section $\hat{S}\hat{P}\hat{T}$ is plane.
(c) The shape of cross-section $\hat{S}\hat{P}\hat{T}$ equals the shape of cross-section $SPT$.
(d) The normal to cross-section $\hat{S}\hat{P}\hat{T}$ is parallel to the axial base vector $f_i$ 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:

\[ u := \hat{x} - x \tag{4.18} \]

\[
\begin{bmatrix}
    u_1 \\
    u_2 \\
    u_3 \\
\end{bmatrix}
= \begin{bmatrix}
      \hat{x}_1 \\
      \hat{x}_2 \\
      \hat{x}_3 \\
    \end{bmatrix} - \begin{bmatrix}
      x_1 \\
      x_2 \\
      x_3 \\
    \end{bmatrix}
\]

\[ u \] 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:

\[
\begin{bmatrix}
    u_{1n} \\
    u_{2n} \\
\end{bmatrix}
= \begin{bmatrix}
      v_{1n} \\
      v_{2n} \\
    \end{bmatrix}
= \begin{bmatrix}
      w_{1n} \\
      w_{2n} \\
    \end{bmatrix}
= \begin{bmatrix}
      d_{1n} \\
      d_{2n} \\
    \end{bmatrix}
\tag{4.19} \]

\[ u_{1n}, u_{2n}, v_{1n}, v_{2n}, w_{1n}, w_{2n}, d_{1n}, d_{2n} \] 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:

\[
\begin{align*}
    u_{1n} &= \hat{x}_{1n} - x_{1n} = x_n + R_{xy}(y_n + v_n) - (x_n + R_{xy}y_n) \\
    u_{2n} &= R_{xy}v_n \\
    R_{xy} &= \text{rotation matrix global - reference space, see (4.5)}
\end{align*}
\tag{4.20} \]

Similar transformation rules are obtained for the other spaces:

\[
\begin{align*}
    u_{1n} &= R_{xy}v_n \\
    u_{2n} &= R_{xz}w_n \\
    v_{1n} &= R_{yz}w_n \\
    d_{1n} &= R_{xy}v_n \\
\end{align*}
\tag{4.21-4.24} \]

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:
\[
\begin{align*}
\mathbf{u}_e &= \begin{bmatrix} u_{1A} \\ u_{2A} \\ u_{1B} \\ u_{2B} \end{bmatrix}, \\
\mathbf{v}_e &= \begin{bmatrix} v_{1A} \\ v_{2A} \\ v_{1B} \\ v_{2B} \end{bmatrix}, \\
\mathbf{w}_e &= \begin{bmatrix} w_{1A} \\ w_{2A} \\ w_{1B} \\ w_{2B} \end{bmatrix}, \\
\mathbf{d}_e &= \begin{bmatrix} d_{1A} \\ d_{2A} \\ d_{1B} \\ d_{2B} \end{bmatrix},
\end{align*}
\]

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):

\[
\begin{align*}
\mathbf{u}_e &= R_{exy} \mathbf{v}_e \quad (4.26) \\
\mathbf{u}_e &= R_{exz} \mathbf{w}_e \quad (4.27) \\
\mathbf{v}_e &= R_{ez} \mathbf{w}_e \quad (4.28) \\
\mathbf{d}_e &= R_{ey} \mathbf{v}_e \quad (4.29)
\end{align*}
\]

\[
R_{exy} \quad \text{element rotation matrix} \quad \text{global - reference space}
\]

\[
R_{exz} \quad \text{element rotation matrix} \quad \text{global - instant space}
\]

\[
R_{ez} \quad \text{element rotation matrix} \quad \text{reference - instant space}
\]

\[
R_{ey} \quad \text{element rotation matrix} \quad \text{support - reference space}
\]

Each of the element rotation matrices in (4.26) to (4.29) contains the coordinate rotation matrices as diagonal matrices. For example, \( R_{exy} \) has the following structure:

\[
R_{exy} = \begin{bmatrix} R_{xy} & 0 \\ 0 & R_{xy} \end{bmatrix}
\]

Since the coordinate rotation matrices are orthonormal, the element rotation matrices are also orthonormal:

\[
R_{exy}^T R_{exy} = I \quad (4.31)
\]

### 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 \( B \) 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 $\varepsilon_{11}$, which is constant over the cross-section. It is denoted by $\varepsilon$ and determined with the general formulas (3.38) of the theory of elasticity:

$$
\varepsilon = \varepsilon_{11} + \frac{1}{2}(\varepsilon_{11}^2 + \varepsilon_{22}^2)
$$

$$
\varepsilon_{11} := \frac{d\varepsilon}{dy}
$$

$\varepsilon$ axial strain in the bar
$\varepsilon_{11}$ 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 $\varepsilon$ in the bar is given by expression (4.32). The product of the strain $\varepsilon$ and the modulus of elasticity $E$ yields the 2. Piola-Kirchhoff stress $s$, which is the coefficient of the base vector $b_1$ of the instant configuration, as defined in (3.68). The axial force vector $a_1$ in the bar is therefore given by:

$$
s = E \varepsilon = E(\varepsilon_{11} + \frac{1}{2}(\varepsilon_{11}^2 + \varepsilon_{22}^2))
$$

$$
a_1 = A s b_1 = A E \varepsilon b_1
$$

$s$ axial 2. Piola-Kirchhoff stress
$\varepsilon$ axial strain of Green
$E$ modulus of elasticity
$a_1$ axial force in the bar
$A$ area of the cross-section of the bar
$b_1$ axial base vector of the instant configuration

The base vector $b_1$ 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 $\sigma$ 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 $\sigma$ in the axial direction are derived from (4.33):

$$
a_1 = A E \varepsilon \sqrt{(1 + \varepsilon_{11})^2 + \varepsilon_{22}^2} = A \sigma
$$

$$
\sigma = E(\varepsilon_{11} + \frac{1}{2}(\varepsilon_{11}^2 + \varepsilon_{22}^2)) \sqrt{(1 + \varepsilon_{11})^2 + \varepsilon_{22}^2}
$$

$a_1$ amount of the axial force
$\sigma$ axial engineering stress

The base vector $b_1$ in (4.33) points in the direction of the instant axis of the bar. The bar force $a_1$ 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.

\[
\begin{align*}
\mathbf{d}_s &= \begin{bmatrix} \ldots & x & \ldots \\ x & o & \ldots \\ \ldots & x & \ldots \\ x & o & \ldots \\ \ldots & x & \ldots 
\end{bmatrix} \\
\mathbf{q}_s &= \begin{bmatrix} \ldots \\ \ldots \end{bmatrix} \\
\mathbf{r}_s &= \begin{bmatrix} f \\ t \\ \ldots \\ t \\ f \\
\end{bmatrix} \\
\mathbf{s}_s &= \begin{bmatrix} f \\ f \\ \ldots \\ f \\ f 
\end{bmatrix}
\end{align*}
\]

... computed value
x prescribed value
o unused coefficient

\(d_s\) system displacement vector
\(q_s\) system load vector
\(r_s\) system reaction vector
\(s_s\) 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 \(d_s\). The distinction between prescribed and computed displacement coordinates is made by means of a system status vector \(s_s\), 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 \(q_s\). 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 \(r_s\). 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.

\[
\mathbf{v}_e = R_{esy}^T \mathbf{d}_e = R_{esy}^T \mathbf{T}_e \mathbf{d}_s \tag{4.37}
\]

\( v_e \) displacement vector of bar \( e \) referred to reference space
\( d_e \) element vector of bar \( e \)
\( R_{esy} \) element rotation matrix, see (4.29)
\( T_e \) 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 \( T_e(k,m) \) equals 1. All other coefficients in row \( k \) of \( T_e \) 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 \( \varepsilon \) and the axial stress \( s \) are different from null, the scalar product \( \delta \varepsilon \cdot \sigma \) is replaced by the product \( \delta \varepsilon \cdot s \). 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:

\[
\sum_{e \in C_e} \int \delta \varepsilon \cdot s \, dv = \delta \mathbf{d}_s^T (\mathbf{q}_s + \mathbf{r}_s) \tag{4.38}
\]

\( C_e \) volume of bar \( e \) in the reference configuration
\( \varepsilon \) 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)
\( d_e \) system displacement vector (coordinates in support space)
\( q_s \) system load vector (coordinates in support space)
\( r_s \) 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 \( \hat{C} \) 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 \( \hat{C} \). The following equations describe the approach if the displacement coordinates of a node are referred to the global space:

\[
\hat{x}_s = x + u
\]
\[
\hat{x}_{s+1} = \hat{x}_s + \Delta u
\]

\( x \) location vector of the node in the reference configuration
\( \hat{x}_s \) known location vector of the node in the instant configuration \( \hat{C} \)
\( \hat{x}_{s+1} \) unknown location vector of the node in the instant configuration \( \hat{C}_{s+1} \)
\( u \) known node displacement vector in the instant configuration \( \hat{C} \)
\( \Delta u \) unknown node displacement increment from configuration \( \hat{C} \) 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:

\[
\hat{y}_s = y + v
\]
\[
\hat{y}_{s+1} = \hat{y}_s + \Delta v
\]

\( y \) location vector of point \( P \) in the reference configuration
\( \hat{y}_s \) known location vector of point \( \hat{P} \) in the instant configuration \( \hat{C} \)
\( \hat{y}_{s+1} \) unknown location vector of point \( \hat{P}_{s+1} \) in the instant configuration \( \hat{C}_{s+1} \)
\( v \) known vector point displacement vector in configuration \( \hat{C} \)
\( \Delta v \) unknown point displacement increment from location \( \hat{P} \) to \( \hat{P}_{s+1} \)

Strain increment: The strain in the known instant configuration \( \hat{C} \) 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:

\[
\varepsilon + \Delta \varepsilon = (v_{1,1} + \Delta v_{1,1}) + \frac{1}{2}(v_{1,1}^2 + \Delta v_{1,1})^2 + \frac{1}{2}(v_{2,1} + \Delta v_{2,1})^2
\]

The first equation is subtracted from the second equation to obtain the increment:

\[
\Delta \varepsilon = \Delta \varepsilon^0 + \Delta \varepsilon^n
\]
\[
\Delta \varepsilon^0 = (1 + v_{1,1})\Delta v_{1,1} + v_{2,1}\Delta v_{2,1}
\]
\[
\Delta \varepsilon^n = \frac{1}{2} (\Delta v_{1,1}^2 + \Delta v_{2,1}^2)
\]

\( \Delta \varepsilon \) strain increment from configuration \( \hat{C} \) 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 \( \hat{C}_{s1} \) 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 \( \hat{C}_{s1} \). If the coordinates of the displacements of the nodes are referred to the support space, the variation equals

\[
\delta(d + \Delta d) = \delta d + \delta(\Delta d)
\]

Since the state of the truss in configuration \( \hat{C}_s \) is known, the variation \( \delta d \) of the displacement is not required and the variation of the displacement vector is therefore:

\[
\delta(d + \Delta d) = \delta(\Delta d) \quad (4.45)
\]

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):

\[
\delta(\Delta \varepsilon^0) = (1 + v_{11})\delta(\Delta \varepsilon_{11}) + v_{21}\delta(\Delta \varepsilon_{21}) \quad (4.48) \\
\delta(\Delta \varepsilon^n) = \Delta v_{11}\delta(\Delta \varepsilon_{11}) + \Delta v_{21}\delta(\Delta \varepsilon_{21}) \quad (4.49)
\]

4.2.5 Governing Equations for the Displacement Increments

The governing equation (4.38) for plane truss behaviour is applied to configuration \( \hat{C}_{s1} \) of the truss. The left-hand side of (4.38) yields:

\[
\begin{align*}
\sum_{e} \int_{C_e} \delta(\varepsilon + \Delta \varepsilon)(s + \Delta s) \, dv &= \sum_{e} \int_{C_e} \delta(\Delta \varepsilon^0 + \Delta \varepsilon^n)(s + \Delta s) \, dv \\
\sum_{e} \int_{C_e} \delta(\varepsilon + \Delta \varepsilon)(s + \Delta s) \, dv &= \sum_{e} \int_{C_e} \delta(\Delta \varepsilon^0) s \, dv + \sum_{e} \int_{C_e} \delta(\Delta \varepsilon^n) s \, dv + \sum_{e} \int_{C_e} \delta(\Delta \varepsilon) s \, dv
\end{align*}
\]

The right-hand side of (4.38) yields:

\[
\begin{align*}
\delta(d_s + \Delta d_s)^T(q_s + \Delta q_s) + \delta(d_s + \Delta d_s)^T(r_s + \Delta r_s) &= \delta(\Delta d_s)^Tq_s + \delta(\Delta d_s)^Tq_s + \delta(\Delta d_s)^Tr_s + \delta(\Delta d_s)^Tr_s \\
\delta(d_s + \Delta d_s)^T(q_s + \Delta q_s) + \delta(d_s + \Delta d_s)^T(r_s + \Delta r_s) &= \delta(\Delta d_s)^Tq_s + \delta(\Delta d_s)^Tq_s + \delta(\Delta d_s)^Tr_s + \delta(\Delta d_s)^Tr_s
\end{align*}
\]

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 \( \delta e \) whose value is given by:

\[
\delta e = -\sum_{e} \int_{C_e} \delta(\Delta \varepsilon^0) s \, dv + \delta(\Delta d_s)^Tq_s + \delta(\Delta d_s)^Tr_s \quad (4.52)
\]

The variation of the strain increment is substituted from (4.48) into (4.52):

\[
\delta e = -\sum_{e} \int (1 + v_{11})\delta(\Delta \varepsilon_{11}) + v_{21}\delta(\Delta \varepsilon_{21}) s \, dv + \delta(\Delta d_s)^Tq_s + \delta(\Delta d_s)^Tr_s \quad (4.53)
\]
Since the displacements are interpolated linearly over the length of the bar, the stresses 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:

$$\delta e = - \sum_e A \delta s((1 + v_{1,1}) \delta v_{1,1} + v_{2,1} \delta v_{2,1}) + \delta (\delta d_e) \delta q_e + \delta (\delta d_e)^T r_e \quad (4.54)$$

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):

$$\sum_e \int_{c_e} \delta (\delta e_n) s dv = \sum_e \int_{c_e} [(\delta v_{1,1} \delta (\delta v_{1,1}) + \delta v_{2,1} \delta (\delta v_{2,1})) s dv \quad (4.55)$$

The third term on the right-hand side of equation (4.50) is nonlinear in the displacement increments $\delta v_k$. 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 $\Delta e$ and its variation $\delta (\Delta e)$ are expressed with (4.43), (4.44), (4.48) and (4.49):

$$\Delta e = (1 + v_{1,1} + \frac{1}{2} \delta v_{1,1}) \Delta v_{1,1} + v_{2,1} \delta v_{2,1} \quad (4.56)$$

$$\delta (\Delta e) = (1 + v_{1,1} + \Delta v_{1,1}) \delta (\Delta v_{1,1}) + (v_{2,1} + \Delta v_{2,1}) \delta (\Delta v_{2,1})$$

$v_{m,1}$ derivatives of the displacement coordinates in $C^{(i)}$ with respect to $y_i$

$\Delta v_{m,1}$ derivatives of the displacement increments from $C^{(i)}$ 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.

$$\sum_e \int_{c_e} E \delta (\Delta e) \Delta e dv = \quad (4.57)$$

$$\sum_e \int_{c_e} E ((1 + v_{1,1}) \delta (\Delta v_{1,1}) + v_{2,1} \delta (\Delta v_{2,1})) ((1 + v_{1,1}) \Delta v_{1,1} + v_{2,1} \Delta v_{2,1}) dv +$$

$$\sum_e \int_{c_e} \frac{1}{2} E ((1 + v_{1,1}) \delta (\Delta v_{1,1}) + v_{2,1} \delta (\Delta v_{2,1})) (\Delta v_{1,1} \Delta v_{1,1} + \Delta v_{2,1} \Delta v_{2,1}) dv +$$

$$\sum_e \int_{c_e} \frac{1}{2} E (\Delta v_{1,1} \delta (\Delta v_{1,1}) + \Delta v_{2,1} \delta (\Delta v_{2,1})) ((1 + v_{1,1}) \Delta v_{1,1} + v_{2,1} \Delta v_{2,1}) dv +$$

$$\sum_e \int_{c_e} \frac{1}{2} E (\Delta v_{1,1} \delta (\Delta v_{1,1}) + \Delta v_{2,1} \delta (\Delta v_{2,1})) (\Delta v_{1,1} \Delta v_{1,1} + \Delta v_{2,1} \Delta v_{2,1}) dv +$$

$$\sum_e \int_{c_e} E (\Delta v_{1,1} \delta (\Delta v_{1,1}) + \Delta v_{2,1} \delta (\Delta v_{2,1})) dv$$

$$\bar{v} = \frac{1}{2} ((1 + v_{1,1}) \Delta v_{1,1} + v_{2,1} \Delta v_{2,1}) \quad (4.58)$$
The terms of equation (4.57) are integrated analytically as in equation (4.54):

\[
\sum_{e} \int_{e} E \delta(\Delta \varepsilon) \Delta \varepsilon \, dv =
\]

\[
\sum_{e} A E a ((1 + v_{11}) \delta(\Delta v_{11}) + v_{21} \delta(\Delta v_{21}))((1 + v_{11}) \Delta v_{11} + v_{21} \Delta v_{21}) +
\]

\[
\sum_{e} \frac{1}{2} A E a ((1 + v_{11}) \delta(\Delta v_{11}) + v_{21} \delta(\Delta v_{21}))((\Delta \bar{v}_{11} \Delta v_{11} + \Delta \bar{v}_{21} \Delta v_{21}) +
\]

\[
\sum_{e} \frac{1}{2} A E a (\Delta \bar{v}_{11} \delta(\Delta v_{11}) + \Delta \bar{v}_{21} \delta(\Delta v_{21}))((1 + v_{11}) \Delta v_{11} + v_{21} \Delta v_{21}) +
\]

\[
\sum_{e} \frac{1}{2} A E a (\Delta \bar{v}_{11} \delta(\Delta v_{11}) + \Delta \bar{v}_{21} \delta(\Delta v_{21}))((\Delta \bar{v}_{11} \Delta v_{11} + \Delta \bar{v}_{21} \Delta v_{21}) +
\]

\[
\sum_{e} A E a \bar{h}(\Delta v_{11} \delta(\Delta v_{11}) + \Delta v_{21} \delta(\Delta v_{21}))
\]

Substitution of expression (4.54) to (4.59) into equation (4.50) yields the incremental governing equations for plane trusses.

\[
\sum_{e} A E a \varepsilon (\Delta v_{11} \delta(\Delta v_{11}) + \Delta v_{21} \delta(\Delta v_{21})) +
\]

\[
\sum_{e} A E a ((1 + v_{11}) \delta(\Delta v_{11}) + v_{21} \delta(\Delta v_{21}))((1 + v_{11}) \Delta v_{11} + v_{21} \Delta v_{21}) +
\]

\[
\sum_{e} \frac{1}{2} A E a ((1 + v_{11}) \delta(\Delta v_{11}) + v_{21} \delta(\Delta v_{21}))((\Delta \bar{v}_{11} \Delta v_{11} + \Delta \bar{v}_{21} \Delta v_{21}) +
\]

\[
\sum_{e} \frac{1}{2} A E a (\Delta \bar{v}_{11} \delta(\Delta v_{11}) + \Delta \bar{v}_{21} \delta(\Delta v_{21}))((1 + v_{11}) \Delta v_{11} + v_{21} \Delta v_{21}) +
\]

\[
\sum_{e} \frac{1}{2} A E a (\Delta \bar{v}_{11} \delta(\Delta v_{11}) + \Delta \bar{v}_{21} \delta(\Delta v_{21}))((\Delta \bar{v}_{11} \Delta v_{11} + \Delta \bar{v}_{21} \Delta v_{21}) +
\]

\[
\sum_{e} A E a \bar{h}(\Delta v_{11} \delta(\Delta v_{11}) + \Delta v_{21} \delta(\Delta v_{21})) = \delta \mathbf{e} + \delta(\Delta \mathbf{d}_{s})^{T} \mathbf{q}_{s} + \delta(\Delta \mathbf{d}_{s})^{T} \mathbf{r}_{s}
\]

The equations (4.60) are linear in the unknown displacement increments \(\Delta v_{1}\) and \(\Delta v_{2}\). They are approximate because the derivatives \(\Delta \bar{v}_{11}\) and \(\Delta \bar{v}_{21}\) 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.

\[ v = S v_e \]  
\[ v = \begin{pmatrix} \nu_1 \\ \nu_2 \end{pmatrix} \]

\[ S = \begin{bmatrix} 0.5(1-z) & 0 & 0.5(1+z) & 0 \\ 0 & 0.5(1-z) & 0 & 0.5(1+z) \end{bmatrix} \]

\[ v_e^T = \begin{pmatrix} \nu_{1A} \\ \nu_{2A} \\ \nu_{1B} \\ \nu_{2B} \end{pmatrix} \]

\[ v \quad \text{displacement vector of point } z \text{ from reference to instant configuration} \]

\[ S \quad \text{interpolation matrix for the displacement coordinates in instant space} \]

\[ v_e \quad \text{element displacement vector from reference to instant configuration} \]

\[ z \quad \text{normalised axial coordinate, see (4.15)} \]

The variation of displacement coordinate \( v_m \) depends on row \( m \) of the interpolation matrix \( S \), which is denoted by \( s_m^T \):

\[ v_m = s_m^T v_e \]  
\[ S = \begin{bmatrix} s_1^T \\ s_2^T \end{bmatrix} \]

\[ s_m \quad \text{interpolation vector for displacement coordinate } v_m \]

Derivatives of the displacement: The first derivatives of the displacement coordinates with respect to \( y_1 \) follow from (4.61) with the chain rule:

\[ \frac{dv}{dy_1} = \frac{dS}{dy_1} v_e := S_{,1} v_e \]  
\[ S_{,1} = \frac{dS}{dy_1} \frac{dz}{dy_1} = \frac{2}{a} \frac{dS}{dz} \]

\[ S_{,1} = \begin{bmatrix} 1 & -1 & 0 & 1 & 0 \\ a & 0 & -1 & 0 & 1 \end{bmatrix} \]

\[ a \quad \text{length of the member in the reference configuration} \]

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 \( S_{,1} \):
$v_{m,1} = \frac{1}{a} g_m^T v_e$ \hfill (4.64)

\[
\begin{bmatrix}
  -1 \\
  0 \\
  1 \\
  0 
\end{bmatrix} \quad \begin{bmatrix}
  0 \\
  -1 \\
  0 \\
  1 
\end{bmatrix}
\]

$g_m$ 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):

$$\varepsilon = g_1^T v_e + \frac{1}{2} (g_1^T v_e)^2 + (g_2^T v_e)^2$$ \hfill (4.65)

$g_m$ 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 $\Delta w_1$ and $\Delta w_2$ of the displacement coordinates of a point $P$ on the axis of the bar in instant space are interpolated linearly with the normalised coordinate $z$:

$$\Delta w = S \Delta w_e$$ \hfill (4.66)

$\Delta w = \begin{bmatrix} \Delta w_1 \\ \Delta w_2 \end{bmatrix}$

$\Delta w_e^T = \begin{bmatrix} \Delta w_{1A} & \Delta w_{2A} & \Delta w_{1B} & \Delta w_{2B} \end{bmatrix}$

107
\[ \Delta w_m = s_m^T \Delta v_e \] (4.67)

\[ \Delta w \quad \text{increment of the displacement vector of point z from configuration } \hat{C}_s \text{ to } \hat{C}_{s+1} \]
\[ \Delta w_e \quad \text{increment of the element displacement vector from configuration } \hat{C}_s \text{ to } \hat{C}_{s+1} \]

The displacement coordinates are transformed with (4.23) so that they are referred to the reference space:

\[ \Delta v = R_{yz} \Delta w \]
\[ \Delta v_A = R_{yz} \Delta w_A \quad \Delta w_A = R_{yz}^T \Delta v_A \]
\[ \Delta v_B = R_{yz} \Delta w_B \quad \Delta w_B = R_{yz}^T \Delta v_B \]

The interpolation (4.66) is multiplied from the left with the rotation matrix \( R_{yz} \). The transformations of \( \Delta w_A \) and \( \Delta w_B \) are substituted on the right-hand side:

\[ \Delta v = \begin{bmatrix} R_{yz} & 0.5(1-z)I & 0.5(1+z)I \end{bmatrix} \begin{bmatrix} R_{yz} \Delta v_A \\ \Delta v \end{bmatrix} \] (4.68)

Since the rotation matrix \( R_{yz} \) 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:

\[ \Delta v = S \Delta v_e \] (4.69)

\[ \Delta v = \begin{bmatrix} \Delta v_1 \\ \Delta v_2 \end{bmatrix} \]

\[ \Delta v_e^T = \begin{bmatrix} \Delta v_{1A} & \Delta v_{2A} & \Delta v_{1B} & \Delta v_{2B} \end{bmatrix} \]

\[ \Delta v_i = s_i^T \Delta v_e \] (4.70)

\[ \Delta v \quad \text{increment of the displacement vector of point z from configuration } \hat{C}_s \text{ to } \hat{C}_{s+1} \]
\[ \Delta v_e \quad \text{increment of the element displacement vector from configuration } \hat{C}_s \text{ to } \hat{C}_{s+1} \]
\[ s_i \quad \text{displacement interpolation vector, see (4.62)} \]

**Displacement derivatives:** The derivatives of the displacement increments with respect to \( y_1 \) follow from (4.66) with the chain rule:

\[ \frac{d(\Delta v)}{dy_1} = s_{i,1} \Delta v_e \] (4.71)

\[ \frac{d(\Delta v)}{dy_1} = \begin{bmatrix} \Delta v_{1,1} \\ \Delta v_{2,1} \end{bmatrix} \]

\[ \Delta v_{m,1} = \frac{1}{a} g_m^T \Delta v_e \] (4.72)

\[ g_m \quad \text{interpolation vector for the displacement derivative, see (4.64)} \]
Variation of the incremental variables: The variation of the displacement increments follows from (4.70):

$$
\delta(\Delta v_m) = s_m^T \delta(\Delta v_e) \quad m \in \{1, 2, 3\} \quad (4.73)
$$

The variation of the derivatives of the displacement increments follows from (4.72):

$$
\delta(\Delta v_{m,1}) = g_m^T \delta(\Delta v_e) \quad m \in \{1, 2, 3\} \quad (4.74)
$$

The variation of the strain increments follows from (4.48), (4.49) and (4.56):

$$
\delta(\Delta e^0) = (1 + v_{11}) g_1^T \delta(\Delta v_e) + v_{2,1} g_2^T \delta(\Delta v_e) \quad (4.75)
$$

$$
\delta(\Delta e^n) = \delta(\Delta v_e)^T g_1 g_1^T \Delta v_e + \delta(\Delta v_e)^T g_2 g_2^T \Delta v_e \quad (4.76)
$$

$$
\delta(\Delta e) = \frac{1}{a} \left( (1 + v_{11} + \frac{1}{2} \Delta \bar{v}_{11} ) g_1^T \delta(\Delta v_e) + (v_{2,1} + \frac{1}{2} \Delta \bar{v}_{2,1} ) g_2^T \delta(\Delta v_e) \right) + \frac{1}{2a} \delta(\Delta v_e)^T (\Delta \bar{v}_{11} g_1 + \Delta \bar{v}_{2,1} g_2) \quad (4.77)
$$

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 $\delta(\Delta v_e)$ is replaced by the variation of the increment of the system displacement vector $\delta(\Delta d_e)$ in (4.37):

$$
\delta e = \delta(\Delta d_e)^T e_s
$$

$$
e_s = \sum s_a T_a^T R_{esy} e_e + q_s + r_s \quad (4.78)
$$

$$
e_e = -AE(1 + v_{11}) g_1 + v_{2,1} g_2 \quad (4.79)
$$

$$
e_e = AE(1 + v_{11}, 1 \frac{1}{2} (v_{11}^2 + v_{2,1}^2))
$$

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):

$$
\sum \Delta \bar{v}_{11} \delta(\Delta v_{11}) + \Delta \bar{v}_{2,1} \delta(\Delta v_{2,1}) \quad (4.80)
$$

$$
\sum \delta(\Delta v_e)^T \left( AE(1 + v_{11}) g_1 + v_{2,1} g_2 \right) (1 + v_{11}) g_1^T + v_{2,1} g_2^T ) \Delta v_e \quad (4.81)
$$

$$
\sum \delta(\Delta v_e)^T \left( \frac{AE}{a} \varepsilon \left( g_1 g_1^T + g_2 g_2^T \right) \Delta v_e \quad (4.82)
$$

109
\[
\sum \frac{1}{2} \varepsilon \left((1 + v_{11}) \delta(v_{11}) + v_{21} \delta(v_{21})\right) (\Delta \vec{v}_{11} \Delta v_{11} + \Delta \vec{v}_{21} \Delta v_{21}) = \\
= \sum \delta(v_e)^T \left(\frac{AE}{2a} \left((1 + v_{11}) \text{g}_1 + v_{21} \text{g}_2\right) \left(\Delta \vec{v}_{11} \text{g}_1^T + \Delta \vec{v}_{21} \text{g}_2^T\right)\right) \Delta v_e 
\]

(4.83)

\[
\sum \frac{1}{2} \varepsilon \left(\Delta \vec{v}_{11} \delta(v_{11}) + v_{21} \delta(v_{21})\right)((1 + v_{11}) \Delta v_{11} + v_{21} \Delta v_{21}) = \\
= \sum \delta(v_e)^T \left(\frac{AE}{2a} \left(\Delta \vec{v}_{11} \text{g}_1 + \Delta \vec{v}_{21} \text{g}_2\right) \left((1 + v_{11}) \text{g}_1^T + v_{21} \text{g}_2^T\right)\right) \Delta v_e 
\]

(4.84)

\[
\sum \frac{1}{2} \varepsilon \left(\Delta \vec{v}_{11} \delta(v_{11}) + \Delta \vec{v}_{21} \delta(v_{21})\right)(\Delta \vec{v}_{11} \Delta v_{11} + \Delta \vec{v}_{21} \Delta v_{21}) = \\
= \sum \delta(v_e)^T \left(\frac{AE}{2a} \left(\Delta \vec{v}_{11} \text{g}_1 + \Delta \vec{v}_{21} \text{g}_2\right) \left(\Delta \vec{v}_{11} \text{g}_1^T + \Delta \vec{v}_{21} \text{g}_2^T\right)\right) \Delta v_e 
\]

(4.85)

\[
\sum \varepsilon \left(\Delta \vec{v}_{11} \delta(v_{11}) + \Delta \vec{v}_{21} \delta(v_{21})\right) = \\
= \sum \delta(v_e)^T \left(\frac{AE}{a} \left(\Delta \vec{v}_{11} \text{g}_1 + \Delta \vec{v}_{21} \text{g}_2\right) \right) \Delta v_e 
\]

(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):

\[
\delta(\Delta e_s)^T K_s \Delta d_s = \delta(\Delta d_s)^T (e_s + \Delta q_s + \Delta r_s) 
\]

(4.87)

\[K_s\] system stiffness matrix  
\[\Delta d_s\] system displacement increment  
\[e_s\] system error vector, see (4.79)  
\[\Delta q_s\] system load increment  
\[\Delta r_s\] system reaction increment

The system stiffness matrix is assembled from the element contributions:

\[
K_s = \sum \bar{T}_e^T R_{esy} K_e R_{esy} \bar{T}_e 
\]

(4.88)

\[
K_s = K_c + K_d + K_g + K_m + K_h + K_f 
\]

(4.89)

The vector products of the derivative vectors \(g_m\) are denoted as follows:

\[
G_{km} := g_k g_m^T \quad k, m \in \{1, 2\} 
\]

(4.90)

\[
G_{km} := g_k g_m^T + g_m g_k^T \quad k \neq m 
\]

The constant component \(K_c\) of the bar stiffness follows from (4.81):

\[
K_c = \frac{AE}{a} G_{11} 
\]

(4.91)

The deviatoric component \(K_d\) of the bar stiffness also follows from (4.81):
The geometric component $K_g$ of the bar stiffness follows from (4.82):

$$K_g = \frac{AE}{a} c_4 (G_{11} + G_{22})$$  \hspace{1cm} (4.93)

$$c_4 = v_{1,1} \left(1 + v_{1,1}\right)$$

$$c_4 = \frac{1}{2} \left(v_{1,1}^2 + v_{2,1}^2\right)$$

The corrective component $K_m$ follows from (4.83) and (4.84):

$$K_m = \frac{AE}{a} (c_5 G_{11} + c_6 G_{22} + c_7 G_{12})$$  \hspace{1cm} (4.94)

$$c_5 = \Delta \bar{v}_{1,1} \left(1 + v_{1,1}\right)$$

$$c_6 = \Delta \bar{v}_{2,1} v_{2,1}$$

$$c_7 = \frac{1}{2} \left(\Delta \bar{v}_{2,1} (1 + v_{1,1}) + \Delta \bar{v}_{1,1} v_{2,1}\right)$$

The corrective component $K_n$ follows from (4.85):

$$K_n = \frac{AE}{a} (c_8 G_{11} + c_9 G_{22} + c_{10} G_{12})$$  \hspace{1cm} (4.95)

$$c_8 = \frac{1}{2} \Delta \bar{v}_{1,1}^2$$

$$c_9 = \frac{1}{2} \Delta \bar{v}_{2,1}^2$$

$$c_{10} = \frac{1}{2} \Delta \bar{v}_{1,1} \Delta \bar{v}_{2,1}$$

The corrective component $K_r$ follows from (4.86):

$$K_r = \frac{AE}{a} (c_{11} G_{11} + c_{12} G_{22})$$  \hspace{1cm} (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 $K_s$:

$$K_s = \frac{EA}{a} \begin{bmatrix} a_1 & a_3 & -a_1 & -a_3 \\ a_3 & a_2 & -a_3 & -a_2 \\ -a_1 & -a_3 & a_1 & a_3 \\ -a_3 & -a_2 & a_3 & a_2 \end{bmatrix}$$  \hspace{1cm} (4.97)

$$a_1 = 1 + c_1 + c_4 + c_5 + c_6 + c_{11}$$

$$a_2 = c_2 + c_4 + c_6 + c_{11} + a_3 = c_3 + c_7 + c_{10}$$

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:

$$K_s \Delta d_s = e_s + \Delta q_s + \Delta r_s$$  \hspace{1cm} (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 $x_1$, $x_2$:

$$ x = x_1 \mathbf{i}_1 + x_2 \mathbf{i}_2 $$

(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 $y_i$ 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).
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:

\[ \mathbf{x}_A = \begin{bmatrix} x_{1A} \\ x_{2A} \end{bmatrix} \quad \mathbf{x}_B = \begin{bmatrix} x_{1B} \\ x_{2B} \end{bmatrix} \]

\( \mathbf{x}_A \) 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 \( \mathbf{e}_y \) for axis \( y \) points from the start node A to the end node B of the member. The global coordinates of \( \mathbf{e}_y \) are computed as follows:

\[ L = \sqrt{(x_{1B} - x_{1A})^2 + (x_{2B} - x_{2A})^2} \quad (5.3) \]

\[ \mathbf{e}_y := \frac{e_{1y}}{e_{2y}} = \frac{1}{L} \begin{bmatrix} x_{1B} - x_{1A} \\ x_{2B} - x_{2A} \end{bmatrix} \]
The unit base vector $\mathbf{e}_2$ is chosen so that it is orthogonal to $\mathbf{e}_1$:

$$e_2 = \frac{e_{12}}{e_{22}} = -\frac{e_{21}}{e_{11}}$$  \hfill (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:

$$\mathbf{x} = \mathbf{x}_A + \mathbf{R}_{xy}\mathbf{y}$$  \hfill (5.5)

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_{1A} \\ x_{2A} \end{bmatrix} + \mathbf{e}_1 \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$$

$$\mathbf{R}_{xy} = \begin{bmatrix} \mathbf{e}_1 \\ \mathbf{e}_2 \end{bmatrix}$$

$\mathbf{R}_{xy}$: 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 $\hat{A}B$ to an instant location $\hat{A}\hat{B}$. Let the global coordinates of points $\hat{A}$ and $\hat{B}$ be known:

$$\mathbf{\hat{x}}_A = \frac{\hat{x}_{1A}}{\hat{x}_{2A}} \quad \mathbf{\hat{x}}_B = \frac{\hat{x}_{1B}}{\hat{x}_{2B}}$$  \hfill (5.6)

$\hat{x}_{1A}$, $\hat{x}_{1B}$: location vectors for nodes A and B in the instant configuration

$\hat{x}_{2A}$, $\hat{x}_{2B}$: 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 $\mathbf{f}_i$ are computed as follows:

$$\mathbf{L} = \sqrt{(\hat{x}_{1B} - \hat{x}_{1A})^2 + (\hat{x}_{2B} - \hat{x}_{2A})^2}$$  \hfill (5.7)

$$\begin{bmatrix} f_{11} \\ f_{21} \end{bmatrix} = \frac{1}{\mathbf{L}} \begin{bmatrix} \hat{x}_{1B} - \hat{x}_{1A} \\ \hat{x}_{2B} - \hat{x}_{2A} \end{bmatrix}$$

The unit base vector $\mathbf{f}_2$ is chosen so that it is orthogonal to $\mathbf{f}_i$:

$$f_2 = \frac{f_{12}}{f_{22}} = -\frac{f_{21}}{f_{11}}$$  \hfill (5.8)
The columns of the rotation matrix $R_{xz}$ for the transformation of the coordinates of a point between the global and reference spaces contain the base vectors:

$$\mathbf{x} = \hat{x}_A + R_{xz} \mathbf{z}$$

$$
\begin{bmatrix}
\hat{x}_{1A} \\
\hat{x}_{2A}
\end{bmatrix}
= 
\begin{bmatrix}
f_{11} & f_{12} \\
f_{21} & f_{22}
\end{bmatrix}
\begin{bmatrix}
z_1 \\
z_2
\end{bmatrix}
$$

$$R_{xz} = \begin{bmatrix} f_1 & f_2 \end{bmatrix}$$

$R_{xz}$ 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 $h_1, h_2$.

$$\mathbf{x} = \mathbf{x}_n + R_{sx} \mathbf{s}$$

$$
\begin{bmatrix}
x_1 \\
x_2
\end{bmatrix}
= 
\begin{bmatrix}
x_{1n} \\
x_{2n}
\end{bmatrix}
+ 
\begin{bmatrix}
h_{11} & h_{12} \\
h_{21} & h_{22}
\end{bmatrix}
\begin{bmatrix}
s_1 \\
s_2
\end{bmatrix}
$$

$R_{sx}$ 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):

reference - global:  $\mathbf{y} = R_{yx}^T (\mathbf{x} - \mathbf{x}_A)$

instant - global:  $\mathbf{z} = R_{xz}^T (\mathbf{x} - \mathbf{x}_A)$

support - global:  $\mathbf{s} = R_{sx}^T (\mathbf{x} - \mathbf{x}_n)$

The transformation between reference and instant space follows from (5.5) and (5.9):

$$\mathbf{x}_A + R_{xy} \mathbf{y} = \hat{x}_A + R_{xz} \mathbf{z}$$

$$\mathbf{y} = y_A + R_{yz} \mathbf{z}$$

$$\mathbf{z} = R_{yx}^T (\mathbf{y} - y_A)$$

$$y_A = R_{yx}^T (x_A - x_A)$$

$$R_{yz} = R_{yx}^T R_{xz}$$

$$R_{yz} = \begin{bmatrix} g_{11} & g_{12} \\
g_{21} & g_{22}
\end{bmatrix}$$

$R_{yz}$ rotation matrix reference - instant space
The transformation between the support and the reference space follows from (5.5) and (5.10):

\[ x_A + R_{xy} y = x_N + R_{xz} s \]
\[ s = s_A + R_{sy} y \] \hspace{1cm} (5.13)
\[ y = R_{xy} (s - s_A) \]
\[ s_A = R_{xs}^T (x_A - x_N) \]
\[ R_{sy} = R_{xs}^T R_{xy} \]

\( R_{sy} \) rotation matrix support - reference space

The transformation between support and instant space follows from (5.9) and (5.10):

\[ \dot{x}_A + R_{xz} z = x_N + R_{xz} s \]
\[ s = \dot{s}_A + R_{sz} z \] \hspace{1cm} (5.14)
\[ z = R_{sz}^T (s - s_A) \]
\[ \dot{s}_A = R_{xs}^T (\dot{x}_A - x_N) \]
\[ R_{sz} = R_{xs}^T R_{xz} \]

\( R_{sz} \) 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 \( \mathbf{i}_i \), 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):

\[ \mathbf{u} = R \mathbf{v} \] \hspace{1cm} (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 \( \mathbf{i}_i \), 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):

\[ \mathbf{U} = R \mathbf{V} R^T \] \hspace{1cm} (5.16)
\[ \begin{bmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{bmatrix} = \begin{bmatrix} e_{11} & e_{12} \\ e_{21} & e_{22} \end{bmatrix} \begin{bmatrix} v_{11} & v_{12} \\ v_{21} & v_{22} \end{bmatrix} = \begin{bmatrix} e_{11} & e_{12} \\ e_{21} & e_{22} \end{bmatrix} \]

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 \( n \) is parallel to the axial base vector \( e_1 \). 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](image)

(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 $\beta$ 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 $\beta$ 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 $s_1, s_2$
- $\beta$: 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:

\[
\begin{align*}
R_{\text{nx}}^T R_{\text{ny}} &= R_{\text{nxz}}^T R_{\text{nz}} = R_{\text{nyz}}^T R_{\text{ny}} = R_{\text{nxz}}^T R_{\text{nxs}} = I
\end{align*}
\] (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 \( u_e, v_e, w_e \) or \( d_e \) if the displacement coordinates are referred to the global, reference, instant or support space:

\[
\begin{align*}
\begin{bmatrix}
u_{1A} \\
u_{2A} \\
u_{1B} \\
u_{2B}
\end{bmatrix}
&= \begin{bmatrix}
\beta_A \\
\beta_A \\
\beta_B \\
\beta_B
\end{bmatrix}
&= \begin{bmatrix}
\beta_A \\
\beta_B
\end{bmatrix},

\begin{bmatrix}
v_{1A} \\
v_{2A} \\
v_{1B} \\
v_{2B}
\end{bmatrix}
&= \begin{bmatrix}
\beta_A \\
\beta_B
\end{bmatrix},

\begin{bmatrix}
w_{1A} \\
w_{2A} \\
w_{1B} \\
w_{2B}
\end{bmatrix}
&= \begin{bmatrix}
\beta_A \\
\beta_B
\end{bmatrix},

\begin{bmatrix}
d_{1A} \\
d_{2A} \\
d_{1B} \\
d_{2B}
\end{bmatrix}
&= \begin{bmatrix}
\beta_A \\
\beta_B
\end{bmatrix}
\end{align*}
\] (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:

\[
\begin{align*}
u_e &= R_{\text{exy}} v_e \quad R_{\text{exy}} = \begin{bmatrix}
R_{\text{ny}} & 0 \\
0 & R_{\text{nx}}
\end{bmatrix}
\tag{5.24}

u_e &= R_{\text{exz}} w_e \quad R_{\text{exz}} = \begin{bmatrix}
R_{\text{nx}} & 0 \\
0 & R_{\text{nz}}
\end{bmatrix}
\tag{5.25}

v_e &= R_{\text{eyz}} w_e \quad R_{\text{eyz}} = \begin{bmatrix}
R_{\text{ny}} & 0 \\
0 & R_{\text{ny}}
\end{bmatrix}
\tag{5.26}

u_e &= R_{\text{exs}} d_e \quad R_{\text{exs}} = \begin{bmatrix}
R_{\text{nx}} & 0 \\
0 & R_{\text{nx}}
\end{bmatrix}
\tag{5.27}
\end{align*}
\]

- \( R_{\text{exy}} \) element rotation matrix global - reference space
- \( R_{\text{exz}} \) element rotation matrix global - instant space
- \( R_{\text{eyz}} \) element rotation matrix reference - instant space
- \( R_{\text{exs}} \) element rotation matrix global - support space

Since the node rotation matrices are orthonormal, the element rotation matrices are also orthonormal:

\[
\begin{align*}
R_{\text{exy}}^T R_{\text{exy}} &= R_{\text{exz}}^T R_{\text{exz}} = R_{\text{eyz}}^T R_{\text{eyz}} = R_{\text{exs}}^T R_{\text{exs}} = I
\end{align*}
\] (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:

\[
y_P = y_1 j_1
\]  
(5.29)

\( y_1 \) coordinate of point \( P \) in the reference configuration of member AB  
\( j_1 \) 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:

\[
v_P = v_1 j_1 + v_2 j_2
\]  
(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:

\[
y_Q = y_1 j_1 + y_2 j_2
\]  
(5.31)

\( y_1, y_2 \) coordinates of point \( Q \) in the reference configuration of member AB  
\( j_1, j_2 \) local base vectors in the reference configuration

Assume that cross-section SPT rotates as a rigid body through an angle \( \beta \) in the anti-clockwise direction about the axis through \( P \). The displacement \( v_{Q} \) of \( Q \) due to this rotation is given by (3.25) and (3.27):

\[
v_{Q} = (R - I) p = (A \sin \beta + A^2 (1 - \cos \beta)) p
\]  
(5.32)

\[
A = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}
\]

\[
a = \begin{bmatrix} 0 \\ 0 \\ y_2 \end{bmatrix}
\]

\[
p = \begin{bmatrix} 0 \\ y_2 \\ 0 \end{bmatrix}
\]

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:

\[
v_Q = v_P + y_2 \bar{v}_p
\]  
(5.33)

\[
\begin{bmatrix} v_{1Q} \\ v_{2Q} \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} + y_2 \begin{bmatrix} \bar{v}_1 \\ \bar{v}_2 \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} + y_2 \begin{bmatrix} -\sin \beta \\ -(1 - \cos \beta) \end{bmatrix}
\]

\( \beta \) 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 \( \mathbf{y}_0 + \mathbf{v}_0 \):

\[
\begin{align*}
\mathbf{c}_1 &= \frac{\partial (\mathbf{y}_0 + \mathbf{v}_0)}{\partial y_1} = \frac{\partial (\mathbf{y}_0 + \mathbf{v}_p)}{\partial y_1} + y_2 \frac{\partial \mathbf{v}_p}{\partial y_1} \\
\mathbf{c}_1 &= \begin{bmatrix} 1 + v_{11} \\ v_{21} \end{bmatrix} + y_2 \begin{bmatrix} -\sin \beta \\ -(1 - \cos \beta) \end{bmatrix} = \begin{bmatrix} 1 + v_{11} \\ v_{21} \end{bmatrix} + y_2 \begin{bmatrix} -\cos \beta \\ -\sin \beta \end{bmatrix} \frac{d\beta}{dy_1} \\
\mathbf{c}_2 &= \frac{\partial (\mathbf{y}_0 + \mathbf{v}_0)}{\partial y_2} = \frac{\partial \mathbf{v}_0}{\partial y_2} + \mathbf{v}_p \\
\mathbf{c}_2 &= \begin{bmatrix} 0 \\ 1 \end{bmatrix} + \begin{bmatrix} -\sin \beta \\ -(1 - \cos \beta) \end{bmatrix} = \begin{bmatrix} -\sin \beta \\ \cos \beta \end{bmatrix} \\
\mathbf{c}_1 &= \begin{bmatrix} \cos \beta \\ \sin \beta \end{bmatrix} (h - y_2 \frac{d\beta}{dy_1}) \\
\mathbf{c}_2 &= \begin{bmatrix} -\sin \beta \\ \cos \beta \end{bmatrix}
\end{align*}
\]

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 \( 1 + v_{11} \) to \( \cos \beta \) equals the ratio of \( v_{21} \) to \( \sin \beta \):

\[
\begin{align*}
\tan \beta &= \frac{v_{21}}{1 + v_{11}}, \\
\sin \beta &= \frac{v_{21}}{h}, \\
\cos \beta &= \frac{1 + v_{11}}{h} \\
\end{align*}
\]

\[
h = \sqrt{(1 + v_{11})^2 + (v_{21})^2}
\]

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:

\[
\begin{align*}
\mathbf{c}_1 &= \begin{bmatrix} \cos \beta \\ \sin \beta \end{bmatrix} (h - y_2 \frac{d\beta}{dy_1}) \\
\mathbf{c}_2 &= \begin{bmatrix} -\sin \beta \\ \cos \beta \end{bmatrix}
\end{align*}
\]

The derivative of \( \beta \) is determined by differentiating \( \sin \beta \) in (5.36):

\[
\begin{align*}
\frac{d}{dy_1} (\sin \beta) &= \cos \beta \frac{d\beta}{dy_1} = \frac{d}{dy_1} \left( \frac{v_{21}}{h} \right) = \frac{1}{h^2} \left( \frac{1}{h} v_{11} + \frac{1}{h} 2 v_{11} v_{21}, \frac{1}{h^2} \left( \frac{1}{h} v_{11} + \frac{1}{h} 2 v_{11} v_{21} \right) \right) \\
\cos \beta \frac{d\beta}{dy_1} &= \frac{v_{21}}{h} - \frac{\sin \beta}{h} (v_{11} \cos \beta + v_{21} \sin \beta) = \cos \beta \frac{v_{21}}{h} (v_{11} \cos \beta - v_{11} \sin \beta) \\
\frac{d\beta}{dy_1} &= \frac{1}{h} (v_{21} \cos \beta - v_{11} \sin \beta)
\end{align*}
\]

The derivative of \( \beta \) from (5.38) and the trigonometric ratios for \( \beta \) from (5.36) are substituted into (5.37):
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):

\[
\begin{align*}
\frac{\partial \mathbf{V}_Q}{\partial y_1} &= \frac{v_{1,1}}{v_{2,1}} + y_2 \frac{-\cos \beta}{-\sin \beta} \frac{d\beta}{dy_1} \\
\frac{\partial \mathbf{V}_Q}{\partial y_2} &= \frac{-\sin \beta}{-\cos \beta} \frac{d\beta}{dy_1} \\
\end{align*}
\]

(5.40)

The derivative of \( \beta \) is substituted from (5.38):

\[
\begin{align*}
\frac{\partial \mathbf{V}_Q}{\partial y_1} &= \frac{v_{1,1}}{v_{2,1}} + y_2 \frac{-\cos \beta}{-\sin \beta} \frac{d\beta}{dy_1} \\
\frac{\partial \mathbf{V}_Q}{\partial y_2} &= \frac{-\sin \beta}{-\cos \beta} \frac{d\beta}{dy_1} \\
\end{align*}
\]

(5.41)

(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 \( v_{1,1}, v_{2,1} \) 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:

\[
\begin{align*}
v_{1,1} &= v_{2,1} = 0 \\
v_{3,1} &= v_{3,2} = 0 \\
\end{align*}
\]

(5.43)

Substitution of the displacement derivatives (5.43) into expressions (3.38) shows that the following strain coordinates are null:

\[
\begin{align*}
\varepsilon_{13} = \varepsilon_{23} = \varepsilon_{33} = \varepsilon_{32} = \varepsilon_{31} = 0 \\
\end{align*}
\]

(5.44)
The remaining strains reduce to:

\[ e_{11Q} = \{v_{t1} + \frac{1}{2}(v_{1,1} + v_{2,1})\} \]

\[ e_{22Q} = \{v_{2,2} + \frac{1}{2}(v_{1,2} + v_{2,2})\} \]

\[ e_{12Q} = \frac{1}{2}\{v_{1,2} + v_{2,1} + v_{t1}v_{1,2} + v_{2,1}v_{2,2}\} \]

The partial derivatives of the displacement coordinates follow from (5.41):

\[ (v_{t1}) = v_{t1} + \frac{y_{2}t}{h} \cos \beta \]

\[ (v_{2,1}) = v_{2,1} + \frac{y_{2}t}{h} \sin \beta \]

\[ (v_{1,2}) = -\sin \beta \]

\[ (v_{2,2}) = \cos \beta - 1 \]

\[ t := v_{t1} \sin \beta - v_{2,1} \cos \beta \]

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:

\[ e_{11Q} = v_{t1} + \frac{y_{2}t}{h} \cos \beta + \frac{1}{2}\left(v_{t1} + 2\left(\frac{y_{2}t}{h}\right)v_{t1}\cos \beta + \left(\frac{y_{2}t}{h}\right)^2 \cos^2 \beta\right) + \frac{1}{2}\left(v_{2,1} + 2\left(\frac{y_{2}t}{h}\right)v_{2,1}\sin \beta + \left(\frac{y_{2}t}{h}\right)^2 \sin^2 \beta\right) \]

\[ e_{11Q} = v_{t1} + \frac{1}{2}(v_{1,1} + v_{2,1}) + \frac{y_{2}t}{h}(\cos \beta + v_{t1}\cos \beta + v_{2,1}\sin \beta) + \frac{1}{2}\left(\frac{y_{2}t}{h}\right)^2 \]

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:

\[ e_{11Q} = v_{t1} + \frac{1}{2}(v_{1,1} + v_{2,1}) + y_{2}t + \frac{1}{2}\left(\frac{y_{2}t}{h}\right)^2 \]  

\( v_{k,1} \) 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 \( \beta \) 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 \( 
\text{ASTB} \), 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 $k_1, k_2$ are constructed so that $k_{1S}$ is tangent to the instant axis of the member at $\hat{S}$, while $k_{1T}$ 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 $S$
- $\beta_S$ angle of rotation from $e_1$ to $k_{1S}$, positive anti-clockwise
- $v_{1T}, v_{2T}$ displacement coordinates of point $T$
- $\beta_T$ angle of rotation from $e_1$ to $k_{1T}$, 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 $e_1$ and $e_2$ of the reference space. The displacements $v_1, e_1$ and $v_2, e_2$ are not in the direction of the tangent $k_1$ and the normal $k_2$ 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 $k_{1S}, k_{2S}$ and $r_{1T}, r_{2T}$ of point $T$ in the local space $k_{1T}, k_{2T}$. These coordinates are related to the displacement coordinates of the points in the reference space in analogy to transformation (5.20):

$$
\begin{align*}
r_{1S} &= v_{1S} \cos \beta_S + v_{2S} \sin \beta_S \\
r_{2S} &= -v_{1S} \sin \beta_S + v_{2S} \cos \beta_S \\
r_{1T} &= v_{1T} \cos \beta_T + v_{2T} \sin \beta_T \\
r_{2T} &= -v_{1T} \sin \beta_T + v_{2T} \cos \beta_T
\end{align*}
$$

}$\frac{126}{1}$
The interpolation is simplified if the derivatives of $v_1, v_2$ and $\beta$ with respect to $y_1$ are replaced by derivatives of $r_1, r_2$ and $\beta$ with respect to $t_1$. In the case of rotations $\beta_S$ and $\beta_T$, this is possible, since the rotations refer to the base vectors $k_{S}$ and $k_{T}$, which are parallel. The displacement coordinates $r_{S}$ and $r_{T}$, however, refer to base vectors $k_{S}$ and $k_{T}$, which are not parallel. Similarly, the displacement coordinates $r_{S}$ and $r_{T}$ 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
\[ t = v_{111} \sin \beta - v_{211} \cos \beta \]
by the term $-r_{211}$ because $\beta$ 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 $v_{111}$ and $v_{211}$ of the displacement coordinates in the reference space are replaced by derivatives $w_{111}$ and $w_{211}$ 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 $w_{111}$ and $w_{211}$ 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 $\theta$ from axis $y_1$ to axis $z_1$ be

**Figure 5.5** Relative angle $\gamma = \beta - \theta$ between chord and tangent of a member

127
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:

\[
v = R_{yz}w
\]

\[
\begin{align*}
v_1 &= \cos \theta \quad w_1 \\
v_2 &= -\sin \theta \quad w_2
\end{align*}
\]

\( \theta \) angle from axis \( y_1 \) to axis \( z_1 \)

Angle \( \theta \) is independent of coordinate \( y_1 \). Denote the angle from the instant chord \( \hat{A} \hat{B} \) to the tangent \( t \) at point \( \hat{P} \) of the axis in the instant configuration by \( \gamma \). Function \( t \) in (5.47) is expressed in terms of the angle \( \gamma \):

\[
t = \frac{d^2(w_1 \cos \theta - w_2 \sin \theta)}{dy_1^2} \sin \beta - \frac{d^2(w_1 \sin \theta + w_2 \cos \theta)}{dy_1^2} \cos \beta
\]

\[
= \frac{d^2w_1}{dy_1^2}(\sin \beta \cos \theta - \cos \beta \sin \theta) - \frac{d^2w_2}{dy_1^2}(\sin \beta \sin \theta + \cos \beta \cos \theta)
\]

\[
t = w_{1,11} \sin \gamma - w_{2,11} \cos \gamma \quad (5.51)
\]

\( \gamma := \beta - \theta \)

\( \gamma \) 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 \( O[y_2 t]^3 \). 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 \( 10^{-3} \).

\[
O[y_2 h] = O[y_2 w_{2,11}] = O\left[\frac{y_2 M}{EJ}\right] = O\left[\frac{\sigma_{bending}}{E}\right] = O\left[10^{-3}\right] \quad (5.52)
\]

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 \( \varepsilon \) follows from (5.48) and (5.51):

\[
\varepsilon = v_{11} + \frac{1}{2}(v_{11}^2 + v_{21}^2) + y_2(w_{1,11} \sin \gamma - w_{2,11} \cos \gamma) \quad (5.53)
\]

\( \varepsilon \) longitudinal strain in the member

\( v_{11}, v_{21} \) displacement coordinates in reference space

\( v_{k,1} \) 1.derivative of \( v_k \) with respect to \( y_1 \)

\( w_{11}, w_{21} \) displacement coordinates in instant space

\( w_{k,11} \) 2.derivative of \( w_k \) with respect to \( y_1 \)

\( \gamma \) rotation of the axis relative to the chord

The longitudinal strain \( \varepsilon \) is decomposed into an axial strain \( \varepsilon_a \) and a bending strain derivative \( \varepsilon_b \):
\[ \varepsilon = \varepsilon_a + y_2 \varepsilon_b \]  
(5.54)  
\[ \varepsilon_a = \nu_{11} + \frac{1}{2} \left( \nu_{12}^2 + \nu_{22}^2 \right) \]  
(5.55)  
\[ \varepsilon_b = w_{11} \sin \gamma - w_{211} \cos \gamma \]  
(5.56)  
\varepsilon_a \quad \text{axial strain}  
\varepsilon_b \quad \text{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 \( z_i \) 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.

\[ w_c = w_A + \frac{z_i}{\hat{a}} (w_B - w_A) \]  
(5.57)

\[ \begin{array}{c}
w_{c1} \\
w_{c2}
\end{array} = \begin{array}{c}
w_{1A} \\
w_{2A}
\end{array} + \frac{z_i}{\hat{a}} \begin{array}{c}
w_{1B} - w_{1A} \\
w_{2B} - w_{2A}
\end{array} = \begin{array}{c}
w_{1A} \\
w_{2A}
\end{array} + \frac{z_i}{\hat{a}} \Delta w_1 \Delta w_2 \]

\( w_{c1}(z_i) \) \quad \text{chord displacement coordinate of point } z_i \text{ in instant space}  
\( w_{1A} \) \quad \text{displacement coordinate of node A in instant space}  
\( w_{1B} \) \quad \text{displacement coordinate of node B in instant space}  
\( \hat{a} \) \quad \text{length of the member in the instant configuration}  

(b) **Relative displacement**: The relative displacement \( w_r \) is a nonlinear function of the distance \( z_i \) 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.

\[ w_r = w_r(z_i) \]  
(5.58)

\[ \begin{array}{c}
w_{r1} \\
w_{r2} \\
\gamma
\end{array} = \begin{array}{c}
w_{11}(z_i) \\
w_{22}(z_i) \\
\gamma(z_i)
\end{array} \]

\( w_r(z_i) \) \quad \text{relative displacement of point } z_i \text{ in instant space}  
\( \gamma \) \quad \text{rotation of the tangent to the axis at point } z_i \text{ 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:

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):

$$\frac{dy_{1}}{dz_1} + \frac{\Delta w_{1}}{\hat{a}} \cos \theta - \frac{\Delta w_{2}}{\hat{a}} \sin \theta = \cos \theta$$

$$\frac{dw_{c1}}{dz_1} = \frac{w_{1B} - w_{1A}}{\hat{a}} = \frac{\Delta w_{1}}{\hat{a}}$$

$$\frac{dw_{c2}}{dz_1} = \frac{w_{2B} - w_{2A}}{\hat{a}} = \frac{\Delta w_{2}}{\hat{a}}$$

Figure 5.6 shows that the trigonometric ratios are given by:

$$\cos \theta = \frac{\hat{a} - \Delta w_{1}}{a} \quad \sin \theta = \frac{\Delta w_{2}}{a}$$

(5.60)
Substitution of (5.60) into (5.59) yields the derivative of $y_i$ with respect to $z_i$:
\[
\frac{dy_i}{dz_i} = \frac{a}{a} \tag{5.61}
\]

Approximation (5.61) makes it possible to express the coordinates $y_i$ and $z_i$ in terms of a common normalised coordinate $z$:
\[
y_i = \frac{1}{2}(1+z) a \quad \frac{dy_i}{dz_i} = \frac{a}{2} dz \\
z_i = \frac{1}{2}(1+z) \hat{a} \quad \frac{dz_i}{dz} = \frac{\hat{a}}{2} dz
\tag{5.62}
\]

- $z$ 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 $C = (y_i, 0)$ of the reference configuration of a member undergoing a chord displacement. Let the location of the displaced point in the instant space be $(z_i, 0)$. The displacement coordinates of the point in the reference space are:
\[
v_{c1} = v_{1A} + z_i \cos \theta - y_i \\
v_{c2} = v_{2A} + z_i \sin \theta
\]

The partial derivatives of the chord displacement with respect to $y_i$ follow from these equations by differentiation and substitution of (5.61):
\[
\frac{dv_{c1}}{dy} := v_{c1,1} = \frac{\hat{a}}{a} \cos \theta - 1 \\
\frac{dv_{c2}}{dy} := v_{c2,1} = \frac{\hat{a}}{a} \sin \theta
\]

The axial strain for the strain tensor of Green is given by (5.55):
\[
\varepsilon_a = v_{c1,1} + \frac{1}{2} v_{c1,1}^2 + \frac{1}{2} v_{c2,1}^2 
\tag{5.63}
\]

Consider the coordinates of the chord displacement referred to the instant coordinate system of the member. Their derivatives with respect to $y_i$ and the instant length of the member are given by:
\[
\frac{dw_{c1}}{dy} := w_{c1,1} = \frac{w_{1B} - w_{1A}}{a} \\
\frac{dw_{c2}}{dy} := w_{c2,1} = \frac{w_{2B} - w_{2A}}{a}
\tag{5.64}
\]

- $w_{c1}$ 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):

\[
\begin{align*}
v_{c1} &= w_{c1}\cos\theta - w_{c2}\sin\theta \\
v_{c2} &= w_{c1}\sin\theta + w_{c2}\cos\theta
\end{align*}
\]

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:

\[
\begin{align*}
\cos\theta &= 1 - w_{c1,1} \\
\sin\theta &= w_{c2,1}
\end{align*}
\]

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:

\[
\begin{align*}
v_{c1,1} &= w_{c1,1}\cos\theta - w_{c2,1}\sin\theta = w_{c1,1}^2 - w_{c2,1}^2 \\
v_{c2,1} &= w_{c1,1}^2 + w_{c2,1}^2 \\
\varepsilon_a &= w_{c1,1} - \frac{1}{2}w_{c1,1}^2 - \frac{1}{2}w_{c2,1}^2
\end{align*}
\]

(c) Strain derivative $\varepsilon_b$ 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 $w_r$ 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 $\gamma$. The bending strain derivative $\varepsilon_b$ in expression (5.56) reduces to:

\[
\varepsilon_b = -w_{r2,11}
\]

For a member that satisfies assumptions (a) to (c) for small deformations, the strain expressions (5.54) to (5.56) are replaced by:

\[
\begin{align*}
\varepsilon &= \varepsilon_a + y_2\varepsilon_b \\
\varepsilon_a &= w_{c1,1} - \frac{1}{2}w_{c1,1}^2 - \frac{1}{2}w_{c2,1}^2 \\
\varepsilon_b &= -w_{r2,11} \\
w_{c1,1} &= \text{1.derivative of the chord displacement} w_{c1} \text{ with respect to} y_1 \\
w_{c2,1} &= \text{1.derivative of the chord displacement} w_{c2} \text{ with respect to} y_1 \\
w_{r2,11} &= \text{2.derivative of the relative displacement} w_r \text{ with respect to} y_1
\end{align*}
\]

**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 $S_1, S_2, \ldots$, as shown in figure 5.7. Each finite element is a member of the frame and has its own instant coordinate space $z_1, z_2$. 

132
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 $S_1, S_2, \ldots$ leads to stretching of the elements.

5.1.7 Stress Resultants

The strain $\varepsilon$ in the member is given by expression (5.67). The product of strain $\varepsilon$ and the modulus of elasticity $E$ yields the axial 2. Piola-Kirchhoff stress $s$, which is the coefficient of base vector $c_1$ of the instant configuration:

$$ s = sc_1 = E\varepsilon c_1 = Ec_1(\varepsilon_a + y_2\varepsilon_b) \quad (5.70) $$

$s$ axial stress vector per unit area of the cross-section
$c_1$ instant base vector, see (5.39)
$s$ longitudinal 2. Piola-Kirchhoff stress
$\varepsilon_a$ axial strain, see (5.65)
$\varepsilon_b$ 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 $t = v_{111} \sin \beta - v_{211} \cos \beta = w_{111} \sin \gamma - w_{211} \cos \gamma$ in (5.39) is neglected due to (5.52), so that the expression for the base vector is:

$$\mathbf{c}_1 = \frac{1 + v_{c11}}{v_{c21}}$$

$$| \mathbf{c}_1 | = \sqrt{(1 + v_{c11})^2 + (v_{c21})^2}$$

(5.71)

The physical stress $\sigma$ 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 $\sigma_a$ and a bending stress derivative $\sigma_b$:

$$\sigma = | \mathbf{c}_1 | s = \sigma_a + y_2 \sigma_b$$

$$\sigma_a = E \varepsilon_a | \mathbf{c}_1 |$$

$$\sigma_b = E \varepsilon_b | \mathbf{c}_1 |$$

$\sigma_a$ physical axial stress

$\sigma_b$ physical longitudinal stress derivative

Stress $\sigma_a$ 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:

$$n = \int_\Lambda (\sigma_a + y_2 \sigma_b) \, da = A \sigma_a$$

(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:

$$m = \int_\Lambda (\sigma_a + y_2 \sigma_b) y_2 \, da = J \sigma_b$$

(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 $y_2$. 

134
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.

\[
\begin{align*}
\mathbf{d}_s &= \begin{bmatrix} \ldots \atop \times \atop \times \atop \times \atop \times \atop \ldots \end{bmatrix}, & \mathbf{q}_s &= \begin{bmatrix} x \atop \times \atop \times \atop \times \atop \times \atop \ldots \end{bmatrix}, & \mathbf{r}_s &= \begin{bmatrix} \circ \atop \ldots \atop \circ \atop \ldots \atop \circ \atop \ldots \end{bmatrix}, & \mathbf{s}_s &= \begin{bmatrix} f \atop \circ \atop \circ \atop \circ \atop \circ \atop \f \end{bmatrix} \\
\end{align*}
\]

(5.75)

... computed value t : prescribed displacement
x prescribed value f : free displacement
\circ unused coefficient

\[\mathbf{d}_s\] 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 $r_s$. 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 $w_e$ defined in (5.23). In order to relate the element displacement vector $w_e$ to the system displacement vector $d_s$, vector $w_e$ is related to the element displacement vector $d_s$. Element vector $d_s$ in turn is related to the system displacement vector $s$ with the element topology matrix $T_e$.

$$ w_e = R_{ez}^T d_e = R_{ez}^T T_e d_s \quad (5.76) $$

$w_e$ displacement vector of member e referred to instant space
$d_e$ system displacement vector referred to node support spaces
$R_{ez}$ element rotation matrix support - instant space
$T_e$ topology matrix of member e

If the generalised displacement with index $k$ in the element vector $w_e$ corresponds to the generalised displacement with index $m$ in the system vector $s$, the coefficient $T_e(k,m)$ equals 1. All other coefficients in row $k$ of $T_e$ 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 $\varepsilon$ and the longitudinal stress $s$ are different from null, the scalar product $\varepsilon^T \sigma$ is replaced by the product of the variation $\varepsilon$ of the longitudinal strain with the longitudinal stress $s$. The volume load $q$ is replaced by a load $p$ 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:

$$ \sum_{e} \int_{C_i} \delta \varepsilon s dV = \sum_{s} \int_{L_j} \delta \varepsilon p_a d\gamma_i + \delta d_s^T (q_s + r_s) \quad (5.77) $$

$C_e$ volume of member e in the reference configuration
$\varepsilon$ longitudinal strain in the member (strain tensor of Green)
$s$ longitudinal stress in the member (2. Piola-Kirchhoff stress tensor)
$d_s$ system displacement vector (coordinates in support space)
$q_s$ system load vector (coordinates in support space)
$r_s$ 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:

\[
\Delta \theta \approx \frac{\Delta w_{2b} - \Delta w_{2a}}{a} \quad (5.78)
\]

\[
\sin \Delta \theta \approx \Delta \theta
\]

\[
\cos \Delta \theta \approx 1 - \frac{1}{2} \Delta \theta^2
\]

The coordinates of the displacement of point \( P \) referred to the coordinate system \( t_1, t_2 \) follow from (5.15):

\[
r_1 = (w_i + \Delta w_i) \cos \Delta \theta + (w_2 + \Delta w_2) \sin \Delta \theta \quad (5.79)
\]

\[
r_2 = -(w_i + \Delta w_i) \sin \Delta \theta + (w_2 + \Delta w_2) \cos \Delta \theta
\]

\( r_1, r_2 \) 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):

\[
r_1 = (w_i + \Delta w_i)(1 - \frac{1}{2} \Delta \theta^2) + (w_2 + \Delta w_2) \Delta \theta \quad (5.80)
\]

\[
r_2 = -(w_i + \Delta w_i) \Delta \theta + (w_2 + \Delta w_2)(1 - \frac{1}{2} \Delta \theta^2)
\]

**Strain increment:** The longitudinal strain \( \varepsilon \) at point \( (y_i, y_2) \) of a member in configuration \( s \) is given by (5.67) to (5.69):

\[
\varepsilon = \varepsilon_a + y_2 \varepsilon_b
\]

\[
\varepsilon_a = \frac{w_{c11} - \frac{1}{2} w_{c21}^2 - \frac{1}{2} w_{c21}^2}{a}
\]

\[
\varepsilon_b = -w_{c21}
\]

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:

\[
\varepsilon_a + \Delta \varepsilon_a = r_{c11} - \frac{1}{2} r_{c11}^2 - \frac{1}{2} r_{c21}^2
\]

\( \Delta \varepsilon_a \) increment of the axial strain in step \( s \)

The derivatives \( r_{c11} \) and \( r_{c21} \) correspond to (5.64) and are derived with (5.80):

\[
r_{c11} = \frac{r_{b1} - r_{A1}}{a} = \left( \frac{w_{b1} - w_{A1}}{a} + \frac{\Delta w_{b1} - \Delta w_{A1}}{a} \right) \left( 1 - \frac{1}{2} \Delta \theta^2 \right) + \frac{w_{b1} - w_{A1}}{a} \Delta \theta
\]

\[
r_{c21} = \frac{r_{b2} - r_{A2}}{a} = \left( \frac{w_{b2} - w_{A2}}{a} + \frac{\Delta w_{b2} - \Delta w_{A2}}{a} \right) \left( 1 - \frac{1}{2} \Delta \theta^2 \right) - \frac{w_{b2} - w_{A2}}{a} \Delta \theta
\]
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:

$$
\Delta \theta = \Delta w_{c2,1} \tag{5.87}
$$

Expression (5.87) is substituted into (5.85) and (5.86):

$$
r_{c1,1} = (w_{c1,1} + \Delta w_{c1,1})(1 - \frac{1}{2}\Delta w_{c2,1}^2) + w_{c2,1}\Delta w_{c2,1} + \Delta w_{c2,1}^2 \tag{5.88}
$$

$$
r_{c2,1} = (w_{c2,1} + \Delta w_{c2,1})(1 - \frac{1}{2}\Delta w_{c2,1}^2) - w_{c1,1}\Delta w_{c2,1} - \Delta w_{c1,1}\Delta w_{c2,1} \tag{5.89}
$$

The increment of the axial strain follows from (5.82), (5.84), (5.88) and (5.89):

$$
\Delta \varepsilon_a = (w_{c1,1} + \Delta w_{c1,1})(1 - \frac{1}{2}\Delta w_{c2,1}^2) + w_{c2,1}\Delta w_{c2,1} + \Delta w_{c2,1}^2 - w_{c1,1} - \frac{1}{2}\left((w_{c1,1} + \Delta w_{c1,1})(1 - \frac{1}{2}\Delta w_{c2,1}^2) + w_{c2,1}\Delta w_{c2,1} + \Delta w_{c2,1}^2\right)^2 + \frac{1}{2}w_{c1,1}^2 - \frac{1}{2}\left((w_{c2,1} + \Delta w_{c2,1})(1 - \frac{1}{2}\Delta w_{c2,1}^2) - w_{c1,1}\Delta w_{c2,1} - \Delta w_{c1,1}\Delta w_{c2,1}\right)^2 + \frac{1}{2}w_{c2,1}^2\tag{5.90}
$$

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.

$$
\Delta \varepsilon_a = \Delta \varepsilon_a^0 + \Delta \varepsilon_a^n \tag{5.91}
$$

$$
\Delta \varepsilon_a^0 = (1 - w_{c1,1})\Delta w_{c1,1} \tag{5.92}
$$

$$
\Delta \varepsilon_a^n = -\frac{1}{2}\Delta w_{c1,1}^2 + \frac{1}{2}(1 - w_{c1,1})\Delta w_{c2,1}^2 \tag{5.93}
$$

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:

$$
\varepsilon_b + \Delta \varepsilon_b = -\frac{d^2}{dy_1^2}\left(-\left(w_1 + \Delta w_1\right)\Delta \theta + (w_2 + \Delta w_2)(1 - \frac{1}{2}\Delta \theta^2)\right) \tag{5.94}
$$

$\Delta \varepsilon_b$ 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.

$$
\Delta \varepsilon_b = \Delta \varepsilon_b^0 + \Delta \varepsilon_b^n \tag{5.95}
$$

$\Delta \varepsilon_b^0$ 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):

$$
\Delta \varepsilon_b = -\frac{d^2}{dy_1^2} \left[ -(w_1 + \Delta w_1) \Delta w_{c2,1} + (w_2 + \Delta w_2)(1 - \frac{1}{2} \Delta w_{c2,1}^2) \right] + w_{z,11}
$$

$$
\Delta \varepsilon_b^0 = -\Delta w_{z,11} + w_{z,11} \Delta w_{c2,1}
$$

$$
\Delta \varepsilon_b^\ast = \frac{1}{2} w_{z,11} \Delta w_{c2,1} + \Delta w_{z,11} \Delta w_{c2,1}
$$

(5.96)  

(5.97)

For small deformations, the second derivative $w_{z,11}$ of the axial displacement is negligible compared to the second derivative $w_{y,11}$ of the transverse displacement. In section 2.2, for example, $w_{z,11}$ for the cantilever beam is null for all values of the applied end moment. Expressions (5.96) and (5.97) simplify as follows:

$$
\Delta \varepsilon_b^0 = -\Delta w_{z,11}
$$

(5.98)

$$
\Delta \varepsilon_b^\ast = \frac{1}{2} w_{z,11} \Delta w_{c2,1}
$$

(5.99)

**Stress increment:** The stress in the known instant configuration $s$ is given by:

$$
s = s_a + y_2 s_b
$$

(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:

$$
s + \Delta s = s_a + \Delta s_a + y_2(s_b + \Delta s_b)
$$

(5.101)

The increment in the stress in step $s$ is thus:

$$
\Delta s = \Delta s_a + y_2 \Delta s_b
$$

(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 Variety 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 $C^{(s+1)}$:

$$
\delta(d + \Delta d) = \delta d + \delta(\Delta d)
$$

Since the state of the frame in configuration $C^{(s)}$ is known, the variation $\delta d$ of the displacement is not required and the variation of the nodal displacement is therefore:

$$
\delta(d + \Delta d) = \delta(\Delta d)
$$

(5.103)

The variation of the member displacement vector referred to reference and to instant space is determined by analogous reasoning:

$$
\delta(v_+ + \Delta v_+) = \delta(\Delta v_+)
$$

(5.104)

$$
\delta(w_+ + \Delta w_+) = \delta(\Delta w_+)
$$

(5.105)

The variation of the axial strain increment follows from (5.92) and (5.93):
\[ \delta(\Delta \varepsilon_a) = \delta(\Delta \varepsilon_a^0) + \delta(\Delta \varepsilon_a^0) \]  
\[ \delta(\Delta \varepsilon_a^0) = (1 - w_{c11}) \delta(\Delta w_{c11}) \]  
\[ \delta(\Delta \varepsilon_a^0) = -\Delta w_{c11} \delta(\Delta w_{c11}) + (1 - w_{c11}) \Delta w_{c21} \delta(\Delta w_{c21}) \]  

The variation of the increment of the bending strain derivative follows from (5.98) and (5.99):

\[ \delta(\Delta v_b) = \delta(\Delta v_b^0) + \delta(\Delta v_b^0) \]  
\[ \delta(\Delta v_b^0) = -\delta(\Delta w_{211}) \]  
\[ \delta(\Delta v_b^0) = w_{211} \Delta w_{c21} \delta(\Delta w_{c21}) \]  

5.2.5 Governing Equations for the Displacement Increments

The governing equation (5.77) for plane frame behaviour is applied to configuration \( C_{s1} \) of the frame. The left-hand side of (5.77) yields:

\[ \sum \int \delta(\varepsilon + \Delta \varepsilon)(s + \Delta s) dv = \]  
\[ \sum \int \delta(\Delta \varepsilon_a^0 + \Delta \varepsilon_a^0)(s_a + \Delta s_a) dv + \sum \int y_2^2 \delta(\Delta \varepsilon_a^0 + \Delta \varepsilon_a^0)(s_a + \Delta s_a) dv = \]  
\[ \sum \int \delta(\Delta \varepsilon_a^0) s_a dv + \sum \int \delta(\Delta \varepsilon_a^0) s_a dv + \sum \int \delta(\Delta \varepsilon_a) \Delta s_a dv + \]  
\[ \sum \int y_2^2 \delta(\Delta \varepsilon_a^0) s_a dv + \sum \int y_2^2 \delta(\Delta \varepsilon_a^0) s_a dv + \sum \int y_2^2 \delta(\Delta \varepsilon_a) \Delta s_a dv \]

The right-hand side of equation (5.77) yields:

\[ \sum \int \delta(v + \Delta v)^T (p_a + \Delta p_a) dy_1 + \delta(\Delta d_a)^T (q_a + \Delta q_a + r_a + \Delta r_a) = \]  
\[ \sum \int \delta(v + \Delta v)^T p_a dy_1 + \sum \int \delta(\Delta u)^T \Delta p_a dy_1 + \delta(\Delta d_a)^T q_a + \delta(\Delta d_a)^T r_a + \delta(\Delta d_a)^T \Delta s_a \]

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 \( \delta e \) whose value is given by:

\[ \delta e = -\sum \int \delta(\Delta \varepsilon_a^0) s_a dv - \sum \int y_2^2 \delta(\Delta \varepsilon_a^0) s_a dv + \sum \int \delta(\Delta v)^T p_a dy_1 + \delta(\Delta d_a)^T q_a + \delta(\Delta d_a)^T r_a \]  
\[ \delta(\Delta d_a)^T q_a + \delta(\Delta d_a)^T r_a \]  

Variations (5.107) and (5.110) of the strain increments are substituted into (5.114):

\[ \delta e = -\sum \int (1 - w_{c11}) \delta(\Delta w_{c11}) s_a dv + \sum \int y_2^2 \delta(\Delta w_{c21}) s_a dv + \]  
\[ \sum \int \delta(\Delta v)^T p_a dy_1 + \delta(\Delta d_a)^T q_a + \delta(\Delta d_a)^T r_a \]  

The variation \( \delta(\Delta \varepsilon_a^0) \) 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):
\[ \sum_{e \in C_e} \int \delta(\Delta \varepsilon_e^0) s_a \, dv = \sum_{e \in C_e} \int \left( -\Delta w_{c11} \delta(\Delta w_{c11}) + (1 - w_{c11}) \Delta w_{e21} \delta(\Delta w_{e21}) \right) s_a \, dv \]  

(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):

\[ \Delta \varepsilon_a = q \Delta w_{c11} - \frac{1}{2} \Delta w_{c11}^2 + \frac{1}{2} q \Delta w_{e21}^2 \]  

(5.117)

\[ \delta(\Delta \varepsilon_a) = q \delta(\Delta w_{c11}) - \Delta w_{c11} \delta(\Delta w_{c11}) + q \Delta w_{e21} \delta(\Delta w_{e21}) \]  

(5.118)

\[ q = 1 - w_{c11} \]  

(5.119)

\( w_{cm1} \) derivatives of the chord displacement coordinates in \( C^{(0)} \) with respect to \( y_i \),

\( \Delta w_{cm1} \) derivatives of the chord displacement increments from \( C^{(0)} \) to \( C^{(e+1)} \)

The product of the strain increment with its variation is expressed as follows:

\[ \Delta \varepsilon_a \delta(\Delta \varepsilon_a) = q^2 \Delta w_{c11}^2 \delta(\Delta w_{c11}) - q \Delta w_{c11} \delta(\Delta w_{c11}) + q^2 \Delta w_{c11} \Delta w_{e21} \delta(\Delta w_{e21}) \]  

\[ - \frac{1}{2} q \Delta w_{c11}^2 \delta(\Delta w_{c11}) + \frac{1}{2} q \Delta w_{e21}^2 \delta(\Delta w_{e21}) - \frac{1}{2} q \Delta w_{c11} \Delta w_{e21} \delta(\Delta w_{e21}) \]  

\[ + \frac{1}{2} q^2 \Delta w_{e21}^2 \delta(\Delta w_{e21}) - \frac{1}{2} q \Delta w_{c11} \Delta w_{e21} \delta(\Delta w_{e21}) + \frac{1}{2} q^2 \Delta w_{e21}^2 \delta(\Delta w_{e21}) \]  

(5.120)

The expressions are rearranged in such a fashion that they become symmetric in the linearised terms. Approximate displacement increments are marked with a bar:

\[ \Delta \varepsilon_a \delta(\Delta \varepsilon_a) = (q - \frac{1}{2} \Delta \bar{w}_{c11}) (q - \Delta \bar{w}_{c11}) \Delta w_{c11} \delta(\Delta w_{c11}) + \]  

\[ \frac{1}{2} q \Delta \bar{w}_{c11} (q - \Delta \bar{w}_{c11}) \left( \Delta w_{c11} \delta(\Delta w_{c11}) + \Delta w_{e21} \delta(\Delta w_{e21}) \right) + \]  

\[ \frac{1}{2} q \Delta \bar{w}_{c11} \Delta w_{e21} \delta(\Delta w_{e21}) \]  

(5.121)

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):

\[ \sum_{e \in C_e} \int \delta(\Lambda \varepsilon_e^0) s_a \, dv = \sum_{e \in C_e} \int E (q - \frac{1}{2} \Delta \bar{w}_{c11}) (q - \Delta \bar{w}_{c11}) \delta(\Delta w_{c11}) \Delta w_{c11} \, dv + \]  

(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):

$$\Delta \varepsilon_b = -\Delta w_{211} + \frac{1}{2} w_{2111} \Delta w_{c21}$$  \hspace{1cm} (5.123)$$

$$\delta(\Delta \varepsilon_b) = -\delta(\Delta w_{211}) + w_{2111} \delta(\Delta w_{c21})$$  \hspace{1cm} (5.124)$$

The terms containing third powers of the displacement increments are neglected in the product of the strain increment with its variation:

$$\Delta \varepsilon_b \delta(\Delta \varepsilon_b) = \Delta w_{211} \delta(\Delta w_{211})$$  \hspace{1cm} (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):

$$\sum_e \int y_2^2 \delta(\Delta \varepsilon_b) \Delta s_b \, dv = \sum_e \int y_2^2 E \delta(\Delta w_{211}) \Delta w_{211} \, dv$$  \hspace{1cm} (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:

$$\sum_e \int (-\Delta w_{c11} \delta(\Delta w_{c11}) + q \Delta w_{c21} \delta(\Delta w_{c21})) \Delta s_b \, dv + \sum_e \int \Delta w_{c11} \delta(\Delta w_{c11}) \Delta w_{c11} \, dv +$$

$$\sum_e \int \Delta w_{c11} \delta(\Delta w_{c11}) \Delta w_{c11} \, dv +$$

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:

\[
\sum_{e}^{a} \int_{0}^{a} E A \delta(\Delta w_{c1}) \Delta w_{c11} \, dy_{1} + \sum_{e}^{a} \int_{0}^{a} E J \delta(\Delta w_{r21}) \Delta w_{r211} \, dy_{1} = \\
\delta e + \sum_{e}^{a} \int_{0}^{a} \delta(\Delta w)^{T} \Delta p_{c} \, dy_{1} + \delta(\Delta d_{s})^{T} \Delta q_{s} + \delta(\Delta d_{s})^{T} \Delta r_{s}
\]  

(5.128)

Chord displacement: Consider a frame member with prescribed node displacement increments \(\Delta w_{1A}\) and \(\Delta w_{1B}\) that carries an incremental load \(\Delta p_{c}\) per unit length in the reference configuration. Equation (5.128) reduces to the following incremental governing equation for the member:

\[
\int_{0}^{a} E A \delta(\Delta w_{c1}) \Delta w_{c11} \, dy_{1} = \int_{0}^{a} \delta(\Delta w_{c1}) \Delta p_{c} \, dy_{1} + \delta(\Delta w_{1A}) \Delta r_{1A} + \delta(\Delta w_{1B}) \Delta r_{1B}
\]  

(5.129)

The left-hand side of the equation is integrated by parts:

\[
- \int_{0}^{a} E A \delta(\Delta w_{c1}) \Delta w_{c11} \, dy_{1} + E A \delta(\Delta w_{c1}) \Delta w_{c11} |_{A} = E A \delta(\Delta w_{c1}) \Delta w_{c11} |_{B}
\]

\[
\int_{0}^{a} \delta(\Delta w_{c1}) \Delta p_{c} \, dy_{1} + \delta(\Delta w_{1A}) \Delta r_{1A} + \delta(\Delta w_{1B}) \Delta r_{1B}
\]

This equation is satisfied for arbitrary permissible variations of the displacement if the displacement and reaction increments satisfy the following equations:

\[
E A \Delta w_{c11} + \Delta p_{c1} = 0
\]

(5.130)

\[
\Delta r_{1A} = -E A \Delta w_{c11} |_{A}
\]

\[
\Delta w_{c1} |_{A} = \text{prescribed value at node A}
\]

\[
\Delta r_{1B} = E A \Delta w_{c11} |_{B}
\]

\[
\Delta w_{c1} |_{B} = \text{prescribed value at node B}
\]

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 \(\Delta p_{c}\) per unit length in the reference configuration. Equation (5.127) reduces to the following incremental governing equation for the member:

\[
\int_{0}^{a} E J \delta(\Delta w_{r21}) \Delta w_{r211} \, dy_{1} = \int_{0}^{a} \delta(\Delta w_{r2}) \Delta p_{c} \, dy_{1} + \\
\delta(\Delta w_{2A}) \Delta r_{2A} + \delta(\Delta w_{2B}) \Delta r_{2B} + \delta(\Delta \beta_{A}) \Delta m_{A} + \delta(\Delta \beta_{B}) \Delta m_{B}
\]  

(5.131)

The left-hand side of the equation is integrated by parts:

\[
- \int_{0}^{a} E J \delta(\Delta w_{r21}) \Delta w_{r211} \, dy_{1} + E J \delta(\Delta w_{r21}) \Delta w_{r211} |_{B} = E J \delta(\Delta w_{r21}) \Delta w_{r211} |_{A}
\]

\[
\int_{0}^{a} \delta(\Delta w_{r2}) \Delta p_{c} \, dy_{1} + \delta(\Delta w_{2A}) \Delta r_{2A} + \delta(\Delta w_{2B}) \Delta r_{2B} + \delta(\Delta \beta_{A}) \Delta m_{A} + \delta(\Delta \beta_{B}) \Delta m_{B}
\]
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:

\[
\Delta m_A = -EJ \Delta w_{2,111}\big|_A \quad \Delta \beta_A = \text{prescribed value at node A} \quad (5.132)
\]

\[
\Delta m_B = EJ \Delta w_{2,111}\big|_B \quad \Delta \beta_B = \text{prescribed value at node B}
\]

The left-hand side of the equation is integrated by parts once more:

\[
\int_0^a EJ \delta(\Delta w_{r_2}) \Delta w_{r_{2,111}} \, dy_1 + EJ \delta(\Delta w_{r_2}) \Delta w_{r_{2,111}}\big|_B - EJ \delta(\Delta w_{r_2}) \Delta w_{r_{2,111}}\big|_A =
\]

\[
\int_0^a \delta(\Delta w_{r_2}) \Delta p_{e2} \, dy_1 + \delta(\Delta w_{2A}) \Delta r_{2A} + \delta(\Delta w_{2B}) \Delta r_{2B}
\]

This equation is satisfied for arbitrary permissible variations of the displacement if the displacement and reaction increments satisfy the following equations:

\[
EJ \Delta w_{r_{2,111}} - \Delta p_{e2} = 0 \quad (5.133)
\]

\[
\Delta r_{2A} = -EJ \Delta w_{r_{111}}\big|_A \quad \Delta w_{r_2}\big|_A = \text{prescribed value at node A}
\]

\[
\Delta r_{2B} = EJ \Delta w_{r_{111}}\big|_B \quad \Delta w_{r_2}\big|_B = \text{prescribed value at node B}
\]

### 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 $\Delta p_{c1}$ and fixed supports with the homogeneous solutions for unit node displacements, multiplied by the node displacement increments:

\[
\Delta w_{c1} = S_0 + S^T \Delta w_1
\]

\[
\Delta w_{c1} = S_0 + \begin{bmatrix} S_1 & S_2 & \Delta W_{1A} & \Delta W_{1B} \end{bmatrix}
\]

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:
\[
E\frac{d^2 s_k}{dy_1^2} = 0 \quad k \in \{1,2\} \tag{5.136}
\]

\[
y_1 = 0: \quad s_1 = 1 \quad s_2 = 0
\]

\[
y_1 = a: \quad s_1 = 0 \quad s_2 = 1
\]

The contribution of the member to equation (5.128) is found by substituting (5.134) into (5.128):

\[
\int s_0 (\Delta w_1^T \frac{ds}{dy_1} + 2 \Delta w_1^T \frac{ds}{dy_1}) dy_1 - \int \delta(s_0 + \Delta w_1^T s) p_{c_1} dy_1 = \int 0 dy_1
\]

\[
\int s_0 (\Delta w_1^T \frac{ds}{dy_1}) dy_1 = \int \delta(s_0 + \Delta w_1^T s) p_{c_1} dy_1
\]

The particular solution \(s_0\) with the boundary conditions (5.135) satisfies (5.129):

\[
\int s_0 (\Delta w_1^T \frac{ds}{dy_1}) dy_1 = \int \delta(s_0 + \Delta w_1^T s) p_{c_1} dy_1 = 0 \tag{5.138}
\]

The second and third terms on the right-hand side of (5.137) are integrated by parts:

\[
\int s_0 (\Delta w_1^T \frac{ds}{dy_1}) dy_1 = \left[ \frac{\delta(s_0 + \Delta w_1^T s) p_{c_1}}{dy_1} \right]_0^a
\]

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:

\[
\int s_0 (\Delta w_1^T \frac{ds}{dy_1}) dy_1 = \int \delta(s_0 + \Delta w_1^T s) p_{c_1} dy_1 \tag{5.139}
\]

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 \(\Delta p_{c_2}\) and fixed supports with the homogeneous solutions for unit node displacements, multiplied by the node displacement increments:

\[
\Delta w_{r_2} = t_0 + t^T \Delta w_2 \tag{5.140}
\]

\[
\Delta w_{r_2} = t_0 + \begin{bmatrix} t_1 & t_2 & t_3 & t_4 \end{bmatrix} \begin{bmatrix} \Delta w_{2A} \\ \Delta p_A \\ \Delta w_{2B} \\ \Delta p_B \end{bmatrix}
\]

Displacement increment \(t_0\) satisfies equation (5.133) for the fixed-end member:
\[
E J \frac{d^4t_0}{dy_1^4} - \Delta p_{c2} = 0 \quad (5.141)
\]

\[
y_1 = 0: \quad t_0 = 0 \quad \frac{dt_0}{dy_1} = 0 \quad y_1 = a: \quad t_0 = 0 \quad \frac{dt_0}{dy_1} = 0
\]

Displacement increments \( t_1, \ldots, t_4 \) satisfy the homogeneous equation (5.133) for unit values of the node displacement increments:

\[
E J \frac{d^4t_k}{dy_1^4} = 0 \quad k \in \{1,2,3,4\} \quad (5.142)
\]

\[
y_1 = 0: \quad t_1 = 1 \quad \frac{dt_1}{dy_1} = 0 \quad y_1 = a: \quad t_1 = 0 \quad \frac{dt_1}{dy_1} = 0
\]

\[
t_2 = 0 \quad \frac{dt_2}{dy_1} = 1 \quad t_2 = 0 \quad \frac{dt_2}{dy_1} = 0
\]

\[
t_3 = 0 \quad \frac{dt_3}{dy_1} = 0 \quad t_3 = 1 \quad \frac{dt_3}{dy_1} = 0
\]

\[
t_4 = 0 \quad \frac{dt_4}{dy_1} = 0 \quad t_4 = 0 \quad \frac{dt_4}{dy_1} = 1
\]

The contribution of the member to (5.128) is found by substituting (5.140) in (5.128):

\[
\int_0^a E J \delta \left( \frac{d^t_t}{dy_1^2} + \Delta w_{r2} \left( \frac{d^t_t}{dy_1^2} \right) \right) \left( \frac{d^t_t}{dy_1^2} + \Delta w_{r2} \left( \frac{d^t_t}{dy_1^2} \right) \right) dy_1 - \int_0^a \delta (t_0 + \Delta w_{r2} t) p_{c2} dy_1 =
\]

\[
\int_0^a E J \delta \left( \frac{d^t_t}{dy_1^2} \right) d^t_t dy_1 + \int_0^a E J \delta \left( \frac{d^t_t}{dy_1^2} \right) \left( \Delta w_{r2} \left( \frac{d^t_t}{dy_1^2} \right) \right) dy_1 + \int_0^a E J \delta \left( \Delta w_{r2} \right)^T d^t_t dy_1
\]

The particular solution \( t_0 \) satisfies equation (5.141) so that:

\[
\int_0^a E J \delta \left( \frac{d^t_t}{dy_1^2} \right) d^t_t dy_1 - \int_0^a \delta (t_0) \Delta p_{c2} dy_1 = 0 \quad (5.143)
\]

The second and third terms on the right-hand side are integrated by parts:

\[
\int_0^a E J \delta \left( \frac{d^t_t}{dy_1^2} \right) \left( \Delta w_{r2} \left( \frac{d^t_t}{dy_1^2} \right) \right) dy_1 = - \int_0^a E J \delta \left( \frac{d^t_t}{dy_1^2} \right) \left( \Delta w_{r2} \left( \frac{d^t_t}{dy_1^2} \right) \right) dy_1 + E J \delta \left( \Delta w_{r2} \right)^T d^t_t \left|_0^a \right.
\]

\[
\int_0^a E J \delta \left( \Delta w_{r2} \right)^T d^t_t dy_1 = - \int_0^a E J \delta \left( \Delta w_{r2} \right)^T d^t_t dy_1 + E J \delta \left( \Delta w_{r2} \right)^T d^t_t \left|_0^a \right.
\]

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:

\[
\int_0^a E J \delta \left( \frac{d^t_t}{dy_1^2} \right) \left( \Delta w_{r2} \left( \frac{d^t_t}{dy_1^2} \right) \right) dy_1 = \int_0^a \delta t_0 \left( \Delta w_{r2} \right) E J \delta \left( \frac{d^t_t}{dy_1^2} \right) dy_1 - \delta t_0 E J \Delta w_{r2} \left|_0^a \right.
\]

\[
\int_0^a E J \delta \left( \Delta w_{r2} \right)^T d^t_t dy_1 = \int_0^a \delta \left( \Delta w_{r2} \right)^T \left( E J \delta \left( \frac{d^t_t}{dy_1^2} \right) \right) dy_1 - \delta t_0 E J \Delta w_{r2} \left|_0^a \right.
\]
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:

$$
\int_0^a E J \delta(\Delta w_{12}) \tau \frac{d^2 t}{dy_1^2} \left( \frac{d^2 t}{dy_1^2} \right)^T \Delta w_{12} \, dy_1 - \int_0^a \delta(\Delta w_{12}) \tau \Delta p_{12} \, dy_1
$$

(5.144)

Since expressions (5.139) and (5.144) contain the homogeneous solutions $s$ and $t$ of the governing equations, the element stiffness matrix and the element load vector must be computed with the trial functions $s$ and $t$. 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 $\beta$ as node variable. The modified member displacement vector is denoted by $\overline{\mathbf{w}}_e$ if the displacement coordinates are referred to instant space, and by $\overline{\mathbf{v}}_e$ 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:

$$
\mathbf{w}_c = \mathbf{S}_c \mathbf{w}_e
$$

$$
\mathbf{w}_c^T = \begin{bmatrix}
    w_{c1} & w_{c2}
\end{bmatrix}
$$

$$
\mathbf{S}_c = \frac{1}{2} \begin{bmatrix}
    1 - z & 0 & 0 & 1 + z & 0 & 0 \\
    0 & 1 - z & 0 & 0 & 1 + z & 0
\end{bmatrix}
$$

$$
\overline{\mathbf{w}}_e = \begin{bmatrix}
    w_{1A} & w_{2A} & * & w_{1B} & w_{2B} & *
\end{bmatrix}
$$

$w_{ck}$ chord displacement coordinates of point $P$ in instant space, $k \in \{1,2\}$

$z$ normalised coordinate on axis $z$, 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 $\overline{\mathbf{v}}_e$ for the reference space:

$$
\mathbf{v}_c = \mathbf{R}_{nzy} \mathbf{S}_c \mathbf{R}_{eyz}^T \overline{\mathbf{v}}_e = \mathbf{S}_c \overline{\mathbf{v}}_e
$$

$$
\mathbf{v}_c^T = \begin{bmatrix}
    v_{c1} & v_{c2}
\end{bmatrix}
$$

$$
\overline{\mathbf{v}}_e^T = \begin{bmatrix}
    v_{1A} & v_{2A} & * & v_{1B} & v_{2B} & *
\end{bmatrix}
$$

$v_{ck}$ 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):
The first derivative of the chord displacement with respect to coordinate \( y \) in the reference space is determined with the chain rule using expression (5.62):

\[
\frac{\partial w_c}{\partial y} = \frac{dS_c}{dy} \frac{\partial \hat{w}_e}{\partial z} = 2 \frac{dS_c}{dz} \frac{d\hat{w}_e}{dz} \quad (5.148)
\]

Define the differentiation vectors \( g_1 \) and \( g_2 \):

\[
g_1 = \begin{bmatrix} -1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \quad g_2 = \begin{bmatrix} 0 \\ -1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (5.149)
\]

The first derivatives of the chord displacements follow from (5.148):

\[
w_{ck,1} = \frac{1}{a} g_k^T \hat{w}_e \quad k \in \{1, 2\} \quad (5.150)
\]

The second derivatives of \( \hat{w}_e \) with respect to \( y \) 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:

\[
w_r = S_r \hat{w}_e \quad (5.151)
\]

\[
\begin{bmatrix} w_{1a} \\ w_{2a} \end{bmatrix} = \frac{\hat{a}}{8} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -z + z^2 + z^3 & 0 & 0 \\ 0 & 0 & -1 - z^2 + z^3 & -1 - z^2 + z^3 \end{bmatrix} \begin{bmatrix} \hat{w}_e \end{bmatrix}
\]

\[
\overline{\hat{w}_e} = \begin{bmatrix} w_{1a} \\ w_{2a} \end{bmatrix} \begin{bmatrix} \tan(\beta_A - \theta) \\ \tan(\beta_B - \theta) \end{bmatrix}
\]

\( \hat{w}_e \) \quad element displacement vector containing the relative rotations \( \lambda = \beta - \theta \)

\( w_{rk} \) \quad relative displacement coordinates of point P in instant space

\( \hat{a} \) \quad instant length of the member

The first derivative of \( \hat{w}_e \) with respect to \( z \) is derived from (5.151):

\[
\frac{\partial w_r}{\partial z} = \frac{dS_r}{dz} \overline{\hat{w}_e} := S_{r,z} \overline{\hat{w}_e} \quad (5.152)
\]

\[
S_{r,z} = \frac{\hat{a}}{8} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -z^2 + 3z^2 & 0 & 0 \\ 0 & 0 & -1 - 2z + 3z^2 & 0 & 0 \end{bmatrix}
\]
The second derivative of \( w_r \) with respect to \( y_1 \) is derived from (5.152):

\[
\frac{d^2 w_r}{dz^2} = \frac{d^2 S_r}{dz^2} \bar{w}_e := S_{r,zz} \bar{w}_e
\]  

(5.153)

\[
S_{r,zz} = \frac{\hat{a}}{4} \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 + 3z & 0 & 0 & 1 + 3z
\end{bmatrix}
\]

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 \( \gamma \) 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:

\[
\frac{dw_r}{dy_1} = \frac{dw_r}{dz} \frac{dz}{dy_1} = \frac{2}{a} S_{r,z} \bar{w}_e
\]  

(5.154)

\[
\frac{d^2 w_r}{dy_1^2} = \frac{d^2 w_r}{dz^2} \left( \frac{dz}{dy_1} \right)^2 = \frac{4}{a^2} S_{r,zz} \bar{w}_e
\]

(5.155)

\[
w_{r,11} = \frac{\hat{a}}{a^2} \left( (-1 + 3z) \gamma_A + (1 + 3z) \gamma_B \right)
\]

\[
\gamma_A = \beta_A - \theta
\]

\[
\gamma_B = \beta_B - \theta
\]

**Strain:** The strain in the member is determined with expressions (5.67) to (5.69). Expressions (5.150) for the first derivatives \( w_{c,11} \) and \( w_{c,2,1} \) of the chord displacement are substituted into expression (5.68) for the axial strain \( \varepsilon_a \):

\[
\varepsilon_a = w_{c,11} - \frac{1}{2} w_{c,11}^2 - \frac{1}{2} w_{c,2,1}^2
\]  

(5.156)

\[
w_{c,11} = \frac{1}{a} (w_{1B} - w_{1A})
\]

\[
w_{c,2,1} = \frac{1}{a} (w_{2B} - w_{2A})
\]

Expression (5.155) for the second derivative \( w_{r,2,11} \) of the relative displacement are substituted into expression (5.69) for the bending strain derivative \( \varepsilon_b \):

\[
\varepsilon_b = -w_{r,2,11} = -\frac{\hat{a}}{a^2} \left( (-1 + 3z) \gamma_A + (1 + 3z) \gamma_B \right)
\]  

(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 \( \mathbf{w}_e + \Delta \mathbf{w}_e \):

\[
\Delta \mathbf{w}_e = \begin{bmatrix} \Delta w_{1A} & \Delta w_{2A} & \ldots & \Delta w_{1B} & \Delta w_{2B} \end{bmatrix}
\]

(5.158)

The chord displacement in configuration \( C^{(s+1)} \) follows from (5.145):

\[
\mathbf{w}_e + \Delta \mathbf{w}_e = \mathbf{S}_e (\mathbf{w}_e + \Delta \mathbf{w}_e)
\]

(5.159)

\( \Delta \mathbf{w}_c \) is the increment of the chord displacement.

The first derivative of the increment of the chord displacement with respect to the axial coordinate \( y_i \) in the reference space follows from (5.150):

\[
\Delta \mathbf{w}_{ck,1} = \frac{1}{a} \mathbf{g}_k^T \Delta \mathbf{w}_e \quad k \in \{1,2\}
\]

(5.160)

The variation of the first derivative of the increment of the chord displacement with respect to the axial coordinate \( y_i \) in the reference space follows directly from (5.160):

\[
\delta(\Delta \mathbf{w}_{ck,1}) = \frac{1}{a} \mathbf{g}_k^T \delta(\Delta \mathbf{w}_e) \quad k \in \{1,2\}
\]

(5.161)

**Relative displacement increment:** The relative displacement \( \mathbf{w}_{r2} \) 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 \( \Delta \mathbf{w}_{r2} \) in the relative displacement due to an increment \( \Delta \varphi \) in the independent variables is determined by consistent linearisation:

\[
\Delta \mathbf{w}_{r2} = \frac{d}{d\eta} \langle \mathbf{w}_{r2}(\varphi + \eta \Delta \varphi) \rangle_{\eta=0}
\]

(5.162)

The directional derivatives of the total translation coordinates \( w_i \) in the element displacement vector \( \mathbf{w}_e \) in (5.151) are:

\[
\frac{d}{d\eta} \langle w_i + \eta \Delta w_i \rangle_{\eta=0} = \Delta w_i
\]

(5.163)

The directional derivative of the coefficient \( \tan(\beta - \theta) \) in (5.151) is:

\[
\frac{d}{d\eta} \langle \tan(\beta + \eta \Delta \beta - \theta - \eta \Delta \theta) \rangle_{\eta=0} = \frac{1}{\cos^2(\beta - \theta)} (\Delta \beta - \Delta \theta)
\]

(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):

\[
\Delta \theta = \frac{\Delta W_{2B} - \Delta W_{2A}}{a}
\]

(5.165)

The increment \( \Delta \mathbf{w}_e \) of the element displacement vector follows from expressions (5.163) to (5.165):

\[
\Delta \mathbf{w}_e = \begin{bmatrix} \Delta w_{1A} & \Delta w_{2A} & \Delta \beta_A - \frac{\Delta W_{2B} - \Delta W_{2A}}{a} & \Delta w_{1B} & \Delta w_{2B} & \Delta \beta_B - \frac{\Delta W_{2B} - \Delta W_{2A}}{a} \end{bmatrix}
\]

(5.166)
The increment in the relative displacement is expressed in terms of the element displacement vector \( \mathbf{w}_e \) containing the total rotation \( \beta \) by substituting (5.166) into (5.151) and rearranging the resulting product:

\[
\Delta \mathbf{w}_{2r} = \frac{\hat{a}}{8} \begin{bmatrix}
0 & \frac{2}{a}(-z+z^3) & 1-z-z^2+z^3 & 0 & -\frac{2}{a}(-z+z^3) & -1-z+z^2+z^3
\end{bmatrix}
\]  

(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:

\[
\Delta \mathbf{w}_p = \Delta \mathbf{w}_c + \Delta \mathbf{w}_r
\]

(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:

\[
\Delta \mathbf{w}_p = \mathbf{S}_p \Delta \mathbf{w}_e
\]

(5.169)

\[
\Delta \mathbf{w}_n = \mathbf{s}_m^T \Delta \mathbf{w}_e
\]

(5.170)

\[
\mathbf{S}_p = \begin{bmatrix}
\mathbf{s}_1^T \\
\mathbf{s}_2^T
\end{bmatrix}
\]

\[
\mathbf{s}_1 = 
\begin{bmatrix}
0.50(1-z) \\
0 \\
0.50(1+z)
\end{bmatrix}
\]

\[
\mathbf{s}_2 = 
\begin{bmatrix}
0 \\
0.125\hat{a}(1-z-z^2+z^3) \\
0.50(1+z)-0.25q(-z+z^3)
\end{bmatrix}
\]

The first derivative of \( \Delta \mathbf{w}_p \) with respect to \( z \) is obtained from (5.169):

\[
\frac{d(\Delta \mathbf{w}_p)}{dz} = \frac{d\mathbf{S}_p}{dz} \Delta \mathbf{w}_e := \mathbf{S}_{p,z} \Delta \mathbf{w}_e
\]

(5.171)

\[
\mathbf{S}_{p,z} = \begin{bmatrix}
\mathbf{t}_1^T \\
\mathbf{t}_2^T
\end{bmatrix}
\]

\[
\mathbf{t}_1 = 
\begin{bmatrix}
-0.50 \\
0 \\
0.50 \\
0
\end{bmatrix}
\]

\[
\mathbf{t}_2 = 
\begin{bmatrix}
0 \\
-0.50+0.25q(-1+3z^2) \\
0 \\
0.50-0.25q(-1+3z^2)
\end{bmatrix}
\]

The second derivative of the displacement increment \( \Delta \mathbf{w}_p \) with respect to \( z \) follows by differentiation of (5.171):
\[
\frac{d^2(\Delta w_p)}{dz^2} = \frac{d^2S_p}{dz^2}\Delta w_e := S_{p,zz}\Delta w_e
\] (5.173)

\[
S_{p,zz} = \begin{bmatrix}
0 \\
0.25 h_z^T
\end{bmatrix}
\]

\[
h_z = \begin{bmatrix}
0 \\
6qz \\
\hat{a}(-1+3z) \\
0 \\
-6qz \\
\hat{a}(1+3z)
\end{bmatrix}
\]

The second derivative of \(\Delta w_2\) with respect to the axial coordinate \(y\), is determined with the chain rule from (5.173) and (5.62):

\[
\Delta w_{2,11} = \frac{1}{a^2}h_z^T\Delta w_e
\] (5.174)

The variation of the second derivative of \(\Delta w_2\) follows directly from (5.174):

\[
\delta(\Delta w_{2,11}) = \frac{1}{a^2}h_z^T\delta(\Delta w_e)
\] (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 \(\Delta w_{c,1}\) and \(w_{c,2}\) of the chord displacement increment is substituted into expressions (5.91) to (5.93) for the axial strain increment:

\[
\Delta \varepsilon_a = \Delta \varepsilon_a^0 + \Delta \varepsilon_a^n
\]

\[
\Delta \varepsilon_a^0 = (1-w_{c,1})\Delta w_{c,1}
\] (5.176)

\[
\Delta \varepsilon_a^n = -\frac{1}{2}\Delta w_{c,1}^2 + \frac{1}{2}(1-w_{c,1})\Delta w_{c,2,1}^2
\] (5.177)

\[
\Delta w_{c,1} = \frac{1}{a}g_i^T\Delta \dot{w_e}
\]

\[
\Delta w_{c,2,1} = \frac{1}{a}g_z^T\Delta \ddot{w_e}
\]

The variation of the axial strain increment follows from equations (5.106) to (5.108):

\[
\delta(\Delta \varepsilon_a) = \delta(\Delta \varepsilon_a^0) + \delta(\Delta \varepsilon_a^n)
\]

\[
\delta(\Delta \varepsilon_a^0) = (1-w_{c,1})\delta(\Delta w_{c,1})
\] (5.178)

\[
\delta(\Delta \varepsilon_a^n) = -\Delta w_{c,1}\delta(\Delta w_{c,1}) + (1-w_{c,1})\Delta w_{c,2,1}\delta(\Delta w_{c,2,1})
\] (5.179)

\[
\delta(\Delta w_{c,1}) = \frac{1}{a}g_i^T\delta(\Delta \dot{w_e})
\]

\[
\delta(\Delta w_{c,2,1}) = \frac{1}{a}g_z^T\delta(\Delta \ddot{w_e})
\]
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:

\[
\begin{align*}
\Delta \varepsilon_b & = \Delta \varepsilon_b^0 + \Delta \varepsilon_b^2 \\
\Delta \varepsilon_b^0 & = -\Delta w_{2,11} \\
\Delta \varepsilon_b^2 & = -\frac{1}{2} w_{2,11} \Delta w_{c2,1}^2 \\
\Delta w_{2,11} & = \frac{1}{a^2} h_2^\top \Delta w_e
\end{align*}
\]

(5.180) (5.181)

The variation of the increment of the bending strain derivative \( \delta(\Delta \varepsilon_b) \) follows from expressions (5.109) to (5.111):

\[
\begin{align*}
\delta(\Delta \varepsilon_b) & = \delta(\Delta \varepsilon_b^0) + \delta(\Delta \varepsilon_b^2) \\
\delta(\Delta \varepsilon_b^0) & = -\delta(\Delta w_{2,11}) \\
\delta(\Delta \varepsilon_b^2) & = -w_{2,11} \Delta w_{c2,1} \delta(\Delta w_{c2,1}) \\
\delta(\Delta \varepsilon_b) & = -\frac{1}{a^2} h_2^\top \delta(\Delta w_e) \\
\delta(\Delta \varepsilon_b^0) & = -\frac{1}{a} w_{2,11} \Delta w_{c2,1} g_2^\top \Delta w_e
\end{align*}
\]

(5.182) (5.183)

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 $\delta e$ 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):

$$
\int f(y_i)\,dv = \int_{y_1}^{y_2} b(y)\,dy = \frac{1}{2} A a \int f(z)\,dz \quad (5.184)
$$

$$
\int y_j^2 f(y_i)\,dv = \int_{y_1}^{y_2} b(y)\,dy = \frac{1}{2} J a \int f(z)\,dz \quad (5.185)
$$

$A$ area of the cross-section of the member
$J$ moment of inertia of the cross-section of the member

The variation $\delta(\Delta w_{ct1})$ of the first derivative of the chord displacement increment is substituted from (5.161) into the first term of (5.115). The strain $\varepsilon_a$ is computed with expression (5.156).

$$
\int_{-1}^{1} \frac{1}{2} A E \varepsilon_a (1-w_{ct1}) \delta(\Delta w_{ct1}) \,dz = \delta(\Delta w_a) \int_{-1}^{1} \frac{1}{2} A E \varepsilon_a (1-w_{ct1}) \,g_i \,dz \quad (5.186)
$$

The variation $\delta(\Delta w_{c21})$ 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 $\varepsilon_b$ is computed with expression (5.157).

$$
\int_{-1}^{1} \frac{1}{2} J E \varepsilon_b \delta(\Delta w_{c21}) \,dz = \delta(\Delta w_a) \int_{-1}^{1} \frac{1}{2a} EJ \varepsilon_b \,h_2 \,dz \quad (5.187)
$$

The scalar product $\delta(\Delta v) \,p_a$ in the third term on the right-hand side of (5.115) equals the scale product $\delta(\Delta w) \,p_c$:

$$
\delta(\Delta v) \,p_a = \delta(\Delta w) \,p_c \quad (5.188)
$$

$v$ 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
$p_c$ load per unit length of the chord, referred to instant space

The variation of the displacement increment $\Delta w$ is substituted from (5.170) into the third term on the right-hand side of (5.115):

$$
\int_{-1}^{1} \frac{1}{2} a \delta(\Delta w) \,p_c \,dz = \delta(\Delta w_e) \int_{-1}^{1} \frac{1}{2} (p_{c1}s_1 + p_{c2}s_2) \,dz \quad (5.189)
$$

$p_{ck}$ 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.
\[
\delta(\Delta w_s)^T = \delta(\Delta d_s)^T R_{esz} = \delta(\Delta d_s)^T T_e^T R_{esz}
\]

\[
R_{esz} = R_{esz}^TR_{esz}
\]

\[
R_{esz} \quad \text{element rotation matrix support - instant space}
\]

Expressions (5.184) to (5.190) are substituted into the error term (5.115) to obtain the system error vector \(e_s\):

\[
\delta e = \delta(\Delta d_s)^T e_s \quad (5.192)
\]

\[
e_s = \sum_s T_e^T R_{esz} (f_{pe} - f_{ae} - f_{be}) + q_s + r_s \quad (5.193)
\]

\[
f_{ae} = \int_{-1}^1 \frac{1}{2} AE \varepsilon_a (1 - w_{c1}) g_i dz \quad (5.194)
\]

\[
f_{be} = -\int_{-1}^1 \frac{1}{2a} J E \varepsilon_b h_2 dz \quad (5.195)
\]

\[
f_{pe} = \frac{1}{2} (p_{c1} s_1 + p_{c2} s_2) dz \quad (5.196)
\]

- \(f_{ae}\) member end forces due to stretching
- \(f_{be}\) member end forces due to bending
- \(f_{pe}\) member end forces due to member loads

Expressions (5.150) show that the derivatives \(w_{c1}\) and \(w_{c2}\) are independent of \(z\). It follows from (5.156) that the axial strain \(\varepsilon_a\) is also independent of \(z\). The vector \(f_{ae}\) in (5.194) is integrated analytically:

\[
\begin{array}{c}
-1 \\
0 \\
0 \\
1 \\
0 \\
0
\end{array}
\]

\[
f_{ae} = AE \varepsilon_a (1 - w_{c1}) \quad (5.197)
\]

The strain derivative \(\varepsilon_b\) is given by (5.157), and the vector \(h_2\) by (5.173). The vector \(f_{be}\) in (5.195) is integrated analytically:

\[
\begin{array}{c}
0 \\
3q \\
2\hat{a} \\
0 \\
-3q \\
\hat{a}
\end{array}
\]

\[
f_{be} = \frac{2JEq}{a^2} \begin{bmatrix}
\gamma_A \\
\gamma_B
\end{bmatrix}
\]

\[
\gamma_A = \beta_A - \theta \quad \gamma_B = \beta_B - \theta \quad q = \frac{\hat{a}}{a}
\]
Let the member load vary linearly over the length of the element:

\[ \mathbf{p}_a = \mathbf{S}_t \mathbf{p}_{ae} \]  

(5.199)

\[
\begin{pmatrix}
\mathbf{p}_{a1} \\ \mathbf{p}_{a2}
\end{pmatrix} = \begin{pmatrix}
0.5(1-z) & 0 & 0.5(1+z) & 0 \\
0 & 0.5(1-z) & 0 & 0.5(1+z)
\end{pmatrix} \begin{pmatrix}
\mathbf{p}_{e1A} \\ \mathbf{p}_{e2A} \\ \mathbf{p}_{e1B} \\ \mathbf{p}_{e2B}
\end{pmatrix}
\]

\[ \mathbf{p}_a \] load per unit length at point \( z \) referred to reference space

\[ \mathbf{p}_{ae} \] member load vector per unit length referred to reference space

\[ \mathbf{p}_{eA} \] member load per unit length at node A

\[ \mathbf{p}_{eB} \] 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:

\[ \mathbf{p}_c = \mathbf{S}_i \mathbf{p}_{ce} \]  

(5.200)

\[
\begin{pmatrix}
\mathbf{p}_{c1} \\ \mathbf{p}_{c2}
\end{pmatrix} = \begin{pmatrix}
0.5(1-z) & 0 & 0.5(1+z) & 0 \\
0 & 0.5(1-z) & 0 & 0.5(1+z)
\end{pmatrix} \begin{pmatrix}
\mathbf{p}_{e1A} \\ \mathbf{p}_{e2A} \\ \mathbf{p}_{e1B} \\ \mathbf{p}_{e2B}
\end{pmatrix}
\]

\[ \mathbf{p}_c \] load per unit length at point \( z \) referred to instant space

\[ \mathbf{p}_{ce} \] member load vector per unit length referred to instant space

\[ \mathbf{p}_{eA} \] member load per unit length at node A

\[ \mathbf{p}_{eB} \] member load per unit length at node B

The coordinates \( \mathbf{p}_{c1} \) and \( \mathbf{p}_{c2} \) in (5.196) are given by (5.200) and the vectors \( \mathbf{s}_1 \) and \( \mathbf{s}_2 \) by (5.170). The vector \( \mathbf{f}_{ce} \) in (5.196) is integrated analytically:

\[ \mathbf{f}_{ce} = \mathbf{Q}_e \mathbf{p}_{ce} \]  

(5.201)

\[
\begin{array}{cccc}
20 & 0 & 10 & 0 \\
0 & 20 + q & 0 & 10 - q \\
0 & 3\alpha & 0 & 2\alpha \\
10 & 0 & 20 & 0 \\
0 & 10 - q & 0 & 20 + q \\
0 & -2\alpha & 0 & -3\alpha
\end{array}
\]

\[ \mathbf{Q}_e = \begin{pmatrix}
\alpha \\ 60
\end{pmatrix} 
\]

\[ q = \frac{\alpha}{a} \]

Results (5.197), (5.198) and (5.201) are substituted into expression (5.193) for the system error vector \( \mathbf{e}_n \).
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:

\[ \sum_e \delta(\Delta w_e)^T K_e \Delta w_e = \sum_e \delta(\Delta w_e)^T q_e + \delta(\Delta d_z)^T \Delta q_s + \delta(\Delta d_z)^T \Delta r_s \]  \hspace{1cm} (5.202)

The element stiffness matrix \( K_e \) 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:

\[ K_e = K_{ac} + K_{ag} + K_{am} + K_{an} + K_{ar} + K_{bc} \]  \hspace{1cm} (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 \( \Delta w_{c11} \) and \( \Delta w_{c21} \) from (5.160), their variations from (5.161), the second derivative \( \Delta w_{c11}'' \) of the displacement increment \( \Delta w_c \) from (5.174) and its variation from (5.175). The products of the differentiation vectors are denoted as follows:

\[ G_{sk} = g_s g_k^T \]  \hspace{1cm} (5.204)

\[ G_{km} = g_k g_m^T + g_m g_k^T \]

The component matrices of \( K_e \) are determined by analytic integration, as follows:

- Component \( K_{ac} \) of \( K_e \) :

\[ \delta(\Delta w_e)^T K_{ac} \Delta w_e = \int_{c_s}^1 E \frac{t^2}{2a} G_{11} \, dv = \frac{AE}{a} c_0 G_{11} \]  \hspace{1cm} (5.205)

\[ c_0 = t^2 \]

\[ t = 1 - w_{c11} \]

- Component \( K_{ag} \) of \( K_e \) :

\[ \delta(\Delta w_e)^T K_{ag} \Delta w_e = \int_{c_s}^1 \frac{AE}{2a} g_s (-G_{11} + t G_{22}) \, dv = \frac{AE}{a} (c_1 G_{11} + c_2 G_{22}) \]  \hspace{1cm} (5.206)

\[ c_1 = -(w_{c11} - 0.5 w_{c11}^2 - 0.5 w_{c21}^2) \]

\[ c_2 = -t c_1 \]
- Component $K_{am}$ of $K_e$:

$$\delta(\Delta w_e)^T K_{am} \Delta w_e = \int_{C_a} \frac{1}{2} E \Delta \bar{w}_{c11} (-3t + \Delta \bar{w}_{c11}) \Delta w_{c11} \delta(\Delta w_{c11}) \, dv$$

$$K_{am} = \int_{-1}^{1} \frac{AE}{4a} \Delta \bar{w}_{c11} (-3t + \Delta \bar{w}_{c11}) G_{11} \, dz = \frac{AE}{a} c_3 G_{11}$$  \hspace{1cm} (5.207)

$$c_3 = \frac{1}{2} \Delta \bar{w}_{c11} (-3t + \Delta \bar{w}_{c11})$$

- Component $K_{an}$ of $K_e$:

$$\delta(\Delta w_e)^T K_{an} \Delta w_e = \int_{C_a} \frac{1}{2} E t (-t \Delta \bar{w}_{c11}) \Delta \bar{w}_{c21} (\Delta w_{c11} \delta(\Delta w_{c21}) + \Delta w_{c21} \delta(\Delta w_{c11})) \, dv$$

$$K_{an} = \int_{-1}^{1} \frac{AE}{4a} t (-t \Delta \bar{w}_{c11}) \Delta \bar{w}_{c21} G_{12} \, dz = \frac{AE}{a} c_4 G_{12}$$  \hspace{1cm} (5.208)

$$c_4 = \frac{1}{2} t (-t \Delta \bar{w}_{c11}) \Delta \bar{w}_{c21}$$

- Component $K_{ar}$ of $K_e$:

$$\delta(\Delta w_e)^T K_{ar} \Delta w_e = \int_{C_a} \frac{1}{2} E t^2 (\Delta \bar{w}_{c11} + \Delta \bar{w}_{c21}) \Delta w_{c21} \delta(\Delta w_{c21}) \, dv$$

$$K_{ar} = \int_{-1}^{1} \frac{AE}{4a} t^2 (\Delta \bar{w}_{c11} + \Delta \bar{w}_{c21}) G_{22} \, dz = \frac{AE}{a} c_5 G_{22}$$  \hspace{1cm} (5.209)

$$c_5 = \frac{1}{2} t^2 (\Delta \bar{w}_{c11} + \Delta \bar{w}_{c21})$$

- Component $K_{bc}$ of $K_e$:

$$\delta(\Delta w_e)^T K_{bc} \Delta w_e = \int_{C_b} y^2 E \delta(\Delta w_{211}) \Delta w_{211} \, dv$$

$$K_{bc} = \int_{-1}^{1} \frac{JE}{2a^3} h_x h_x^T \, dz = \frac{JE}{a^3}$$

$$\begin{array}{ccccccc}
0 & 0 & 0 & 0 & 0 & 0 \\
0 & c_6 & c_7 & 0 & -c_6 & c_7 \\
0 & c_7 & c_8 & 0 & -c_7 & c_9 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & -c_6 & -c_7 & 0 & c_6 & -c_7 \\
0 & c_7 & c_8 & 0 & -c_7 & c_8 \\
\end{array}$$  \hspace{1cm} (5.210)

$$c_6 = 12q^2 \hspace{1cm} c_7 = 6aq \hspace{1cm} c_8 = 4a^2 \hspace{1cm} c_9 = 2a^2 \hspace{1cm} q = \frac{\hat{a}}{a}$$
The variations \( \delta(\Delta \mathbf{w}_m) \) 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:

\[
\delta(\Delta \mathbf{w}_e)^T \Delta \mathbf{q}_e = \int_{L_e} \delta(\Delta \mathbf{w})^T \Delta \mathbf{p}_c \, dy,
\]

\[
\Delta \mathbf{q}_e = \int_{L_e} \frac{1}{2} \left( \Delta \mathbf{p}_{c1} \mathbf{s}_1 + \Delta \mathbf{p}_{c2} \mathbf{s}_2 \right) \, dz \tag{5.211}
\]

The increment of the member load, like the total member load in expression (5.200), varies linearly on the chord of the element:

\[
\Delta \mathbf{p}_e = \mathbf{S}_L \Delta \mathbf{p}_{ce} \tag{5.212}
\]

\[
\begin{pmatrix}
\Delta \mathbf{p}_{c1} \\
\Delta \mathbf{p}_{c2}
\end{pmatrix}
= \begin{pmatrix}
0.5(1-z) & 0 & 0.5(1+z) & 0 \\
0 & 0.5(1-z) & 0 & 0.5(1+z)
\end{pmatrix}
\begin{pmatrix}
\Delta \mathbf{p}_{c1A} \\
\Delta \mathbf{p}_{c2A} \\
\Delta \mathbf{p}_{c1B} \\
\Delta \mathbf{p}_{c2B}
\end{pmatrix}
\]

- \( \Delta \mathbf{p}_e \) increment of the member load per unit length referred to instant space
- \( \Delta \mathbf{p}_{eA} \) increment of the member load per unit length at node A
- \( \Delta \mathbf{p}_{eB} \) increment of the member load per unit length at node B

The element load vector increment (5.211) is obtained by analytic integration:

\[
\Delta \mathbf{q}_e = \mathbf{Q}_e \Delta \mathbf{p}_e \tag{5.213}
\]

\[
\mathbf{Q}_e = \frac{a}{60} \begin{pmatrix}
20 & 0 & 10 & 0 \\
0 & 20 + q & 0 & 10 - q \\
0 & 3a & 0 & 2a \\
10 & 0 & 20 & 0 \\
0 & 10 - q & 0 & 20 + q \\
0 & -2a & 0 & -3a
\end{pmatrix}
\]

Substitution of (5.205) to (5.213) into (5.203) leads to the following incremental element stiffness matrix:

\[
\mathbf{K}_e = \frac{A E}{a} \mathbf{M}_{e1} + \frac{J E}{a^3} \mathbf{M}_{e2} \tag{5.214}
\]

- \( A \) 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 \( \mathbf{K}_e \):
The variation $\delta(\Delta w_e)$ of the increment of the element displacement vector in the incremental governing equations (5.202) is replaced by the variation $\delta(\Delta d_s)$ of the increment of the system displacement vector, using the topology matrix $T_e$ defined in equation (5.76):

$$
\delta(\Delta d_s)^T K_s \Delta d_s = \delta(\Delta d_s)^T (e_s + \Delta q_s + \Delta r_s)
$$

(5.216)

$$
K_s = \sum_e T_e^T R_{esz} K_e R_{esz}^T T_e
$$

(5.217)

$$
\Delta q_s = \Delta q_n + \sum_e T_e^T R_{esz} \Delta q_e
$$

(5.218)

$K_s$ system stiffness matrix

$K_s$ element stiffness matrix, see (5.214)

$R_{esz}$ element rotation matrix support - instant space

$T_e$ element topology matrix, see (5.76)

$\Delta d_s$ increment of the system displacement vector

$\Delta q_n$ increment of the system vector of the node loads

$\Delta q_e$ increment of the element load vector, see (5.213)

$\Delta q_s$ increment of the system load vector

$e_s$ 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:

$$
K_s \Delta d_s = e_s + \Delta q_s + \Delta r_s
$$

(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 \( \sigma_A \) is given by expressions (5.71), (5.72) and (5.156):

\[
n = AE|c|\varepsilon_a
\]

\[
|c| = \sqrt{(1 + \nu c t_1)^2 + \nu c t_2^2} = \text{instant length / reference length}
\]

\[
\varepsilon_a = w_{c t_1} - \frac{1}{2}w_{c t_2}^2 - \frac{1}{2}w_{c t_2}^2
\]

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):

\[
m = JE|c|\varepsilon_b
\]

\[
\varepsilon_b = -\frac{d^2w_{c t_2}}{dy_1^2} = -\frac{\dot{a}}{a^1}((-1 + 3z)\gamma_A + (1 + 3z)\gamma_B)
\]

The shear force \( q \) is the derivative of the bending moment with respect to the distance \( y_o \) on the reference axis:

\[
q = \frac{dm}{dy_1}
\]

Expression (5.221) for the moment \( m \) is substituted into expression (5.223):
q = J E c \frac{d\varepsilon_b}{dy_1} \quad (5.224)

Expression (5.222) is differentiated with respect to \(y_1\):

\[
\frac{d\varepsilon_b}{dy_1} = -\frac{1}{a} \frac{d^3w_{r2}}{dy_1^3} = -\frac{6}{a^3} (\gamma_A + \gamma_B) \quad (5.225)
\]

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):

\[
z_i = \frac{1}{2}(1-z)\hat{a} \quad dz_i = \frac{1}{2}\hat{a}dz \quad -1 \leq z \leq 1 \quad (5.226)
\]

\(\hat{a}\) 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:

\[
p_i = \frac{1}{2}(1-z)p_{IA} + \frac{1}{2}(1+z)p_{IB} \quad (5.227)
\]

\(p_1\) axial load per unit length of the chord

\(p_{IA}\) axial load intensity at node A

\(p_{IB}\) axial load intensity at node B

The equilibrium condition for an infinitesimal element of the member is:

\[
\frac{dn}{dz_i} = -p_i \quad (5.228)
\]

\(n\) axial force in the member at point \(z\)

Coordinate \(z_i\) is replaced by the normalised coordinate \(z\) in (5.226):

\[
\frac{dn}{dz} = -\frac{1}{2}\hat{a}p_i = -\frac{1}{4}\hat{a}(1-z)p_{IA} - \frac{1}{4}\hat{a}(1+z)p_{IB} \quad (5.229)
\]

The axial force is determined by integration:

\[
n = \frac{1}{8}\hat{a}(1-z)^2p_{IA} - \frac{1}{8}\hat{a}(1+z)^2p_{IB} + c_1 \quad (5.230)
\]

The axial force \(n\) and the axial displacement \(w_{c1}\) are related as follows:

\[
n = E A \frac{dw_{c1}}{dz_i} = \frac{2E A}{\hat{a}} \frac{dw_{c1}}{dz} \quad (5.231)
\]

\(E\) 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:

\[
\frac{dw_{c1}}{dz} = \frac{\hat{a}}{2EA} \left( \frac{1}{8} \hat{a}(1-z)^2p_{1A} - \frac{1}{8} \hat{a}(1+z)^2p_{ib} + c_1 \right)
\]

\[
w_{c1} = \frac{\hat{a}}{2EA} \left( -\frac{1}{24} \hat{a}(1-z)^3p_{1A} - \frac{1}{24} \hat{a}(1+z)^3p_{ib} + c_1 z + c_2 \right) \tag{5.232}
\]

The displacement at both ends of the member is null:

\[z = -1: \quad w_{\infty} = -\frac{1}{3} \hat{a}p_{1A} - c_1 + c_2 = 0\]

\[z = 1: \quad w_{\infty} = -\frac{1}{3} \hat{a}p_{ib} + c_1 + c_2 = 0\]

These equations are solved for the integration constants:

\[c_1 = \frac{1}{6} \hat{a}(p_{ib} - p_{1A}) \tag{5.233}\]

\[c_2 = \frac{1}{6} \hat{a}(p_{1A} + p_{ib}) \tag{5.234}\]

The axial displacement follows from (5.232):

\[
w_{c1} = \frac{\hat{a}^2}{48EA} \left( -(1-z)^3p_{1A} - (1+z)^3p_{ib} + 4(p_{ib} - p_{1A})z + 4(p_{1A} + p_{ib}) \right) \tag{5.235}\]

The axial force follows from (5.230):

\[n = \frac{\hat{a}}{8}(1-z)^2p_{1A} - \frac{\hat{a}}{8}(1+z)^2p_{ib} + \frac{\hat{a}}{6}(p_{ib} - p_{1A}) \tag{5.236}\]

### 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:

\[p_2 = \frac{1}{2}(1-z)p_{2A} + \frac{1}{2}(1+z)p_{2B}, \quad -1 \leq z \leq 1 \tag{5.237}\]

\(p_{2A}\) 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:

\[
\frac{dq}{dz_1} = -p_2 \tag{5.238}
\]

\(q\) shear force acting on the cross-section of the member

Coordinate \(z_1\) is replaced by the normalised coordinate \(z\) in (5.226):

\[
\frac{dq}{dz} = -\frac{1}{2} \hat{a}p_2 = -\frac{1}{4}(1-z)\hat{a}p_{2A} - \frac{1}{4}(1+z)\hat{a}p_{2B} \tag{5.239}
\]

166
The shear force is determined by integration:

\[ q = \frac{1}{8}(1-z)^2 \dot{a}p_{2A} - \frac{1}{8}(1+z)^2 \dot{a}p_{2B} + c_1 \]  
(5.240)

c_1 \quad \text{integration constant}

The equilibrium equation for the bending moment \( m \) in the member is:

\[ \frac{dm}{dz} = q \]  
(5.241)

\( m \) \quad \text{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):

\[ \frac{dm}{dz} = \frac{1}{2} \dot{a}q = \frac{1}{16}(1-z)^2 \dot{a}^2 p_{2A} - \frac{1}{16}(1+z)^2 \dot{a}^2 p_{2B} + \frac{1}{2} \dot{a}c_1 \]  
(5.242)

The bending moment is determined by integration:

\[ m = -\frac{1}{48}(1-z)^3 \dot{a}^2 p_{2A} - \frac{1}{48}(1+z)^3 \dot{a}^2 p_{2B} + \frac{1}{2} \dot{a}c_1 z + c_2 \]  
(5.243)

The bending moment \( m \) and the transverse displacement \( w_{2r} \) are related, as follows:

\[ m = -EJ \frac{d^2w_{2r}}{dz^2} = -\frac{4EJ}{\dot{a}^2} \frac{d^2w_{2r}}{dz^2} \]  
(5.244)

\( J \) \quad \text{moment of inertia of the cross-section}

\( w_{2r} \) \quad \text{transverse relative displacement}

The bending moment \( m \) is substituted from (5.243) into (5.244), which is then solved for the transverse displacement:

\[ \frac{d^2w_{2r}}{dz^2} = \frac{\dot{a}^2}{4EJ} \left( \frac{1}{48}(1-z)^3 \dot{a}^2 p_{2A} + \frac{1}{48}(1+z)^3 \dot{a}^2 p_{2B} - \frac{1}{2} \dot{a}c_1 z - c_2 \right) \]

\[ \frac{dw_{2r}}{dz} = \frac{\dot{a}^2}{4EJ} \left( -\frac{1}{192}(1-z)^4 \dot{a}^2 p_{2A} + \frac{1}{192}(1+z)^4 \dot{a}^2 p_{2B} - \frac{1}{4} \dot{a}c_1 z^2 - c_2 z + c_3 \right) \]

\[ \frac{dw_{2r}}{dz_1} = \frac{\dot{a}}{2EJ} \left( -\frac{1}{192}(1-z)^4 \dot{a}^2 p_{2A} + \frac{1}{192}(1+z)^4 \dot{a}^2 p_{2B} - \frac{1}{4} \dot{a}c_1 z^2 - c_2 z + c_3 \right) \]

\[ w_{2r} = \frac{\dot{a}^2}{4EJ} \left( \frac{1}{960}(1-z)^5 \dot{a}^2 p_{2A} + \frac{1}{960}(1+z)^5 \dot{a}^2 p_{2B} - \frac{1}{12} \dot{a}c_1 z^3 - \frac{1}{2} c_2 z^2 + c_3 z + c_4 \right) \]  
(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.
\[
\begin{align*}
    z = -1: & \quad w_{r2} = \frac{\hat{a}^2}{4EJ} \left( \frac{1}{30} \hat{a}^2 p_{2A} + \frac{1}{12} \hat{a}^2 c_1 - \frac{1}{2} c_2 - c_3 + c_4 \right) = 0 \quad (5.246) \\
    z = 1: & \quad w_{r2} = \frac{\hat{a}^2}{4EJ} \left( \frac{1}{30} \hat{a}^2 p_{2B} - \frac{1}{12} \hat{a}^2 c_1 - \frac{1}{2} c_2 + c_3 + c_4 \right) = 0 \quad (5.247) \\
    z = -1: & \quad \frac{dw_{r2}}{dz} = \frac{\hat{a}^2}{4EJ} \left( -\frac{1}{12} \hat{a}^2 p_{2A} - \frac{1}{4} \hat{a} c_1 + c_2 + c_3 \right) = 0 \quad (5.248) \\
    z = 1: & \quad \frac{dw_{r2}}{dz} = \frac{\hat{a}^2}{4EJ} \left( \frac{1}{12} \hat{a}^2 p_{2B} - \frac{1}{4} \hat{a} c_1 - c_2 + c_3 \right) = 0 \quad (5.249)
\end{align*}
\]

Equations (5.246) to (5.249) are solved for the constants \(c_1\) to \(c_4\):

\[
\begin{align*}
    c_1 &= \frac{3}{20} \hat{a} (p_{2B} - p_{2A}) \\
    c_2 &= \frac{1}{24} \hat{a}^2 (p_{2A} + p_{2B}) \\
    c_3 &= -\frac{1}{240} \hat{a}^3 (p_{2B} - p_{2A}) \\
    c_4 &= \frac{1}{240} \hat{a}^3 (p_{2A} + p_{2B})
\end{align*}
\]

The relative bending displacement \(w_{r2}\) of points on the axis of the member follows by substitution of the integration constants into (5.245):

\[
\begin{align*}
    w_{r2} &= \frac{\hat{a}^4}{960EJ} \left( \frac{1}{4} (1-z)^5 p_{2A} + \frac{1}{4} (1+z)^5 p_{2B} - z(1+3z^2)(p_{2B} - p_{2A}) + (1-5z^2)(p_{2A} + p_{2B}) \right) \\
    & \quad \left(5.250\right)
\end{align*}
\]

The bending moment acting on the cross-sections of the member follows by substitution of the integration constants into (5.243):

\[
\begin{align*}
    m &= \frac{\hat{a}^2}{240} \left( -5(1-z)^2 p_{1A} - 5(1+z)^2 p_{1B} + 18 z(p_{2B} - p_{2A}) + 10(p_{2A} + p_{2B}) \right) \\
    & \quad \left(5.251\right)
\end{align*}
\]

The shear forces acting on the cross-sections of the member follow by substitution of the integration constants into (5.240):

\[
\begin{align*}
    q &= \frac{\hat{a}}{40} \left( 5(1-z)^3 p_{2A} - 5(1+z)^3 p_{2B} + 6(p_{2B} - p_{2A}) \right) \\
    & \quad \left(5.252\right)
\end{align*}
\]

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)](image)

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 $\circ$ 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 $\mathbf{p}_s$:

$$
\mathbf{p}_s = \mathbf{q}_s + \mathbf{r}_s
$$

$$
\mathbf{d}_s = \begin{bmatrix}
    \mathbf{d}_{s1} \\
    \mathbf{d}_{s2}
\end{bmatrix}, \quad \mathbf{q}_s = \begin{bmatrix}
    \mathbf{q}_{s1} \\
    \mathbf{q}_{s2}
\end{bmatrix}, \quad \mathbf{r}_s = \begin{bmatrix}
    \mathbf{0} \\
    \mathbf{r}_{s2}
\end{bmatrix}, \quad \mathbf{p}_s = \begin{bmatrix}
    \mathbf{p}_{s1} \\
    \mathbf{p}_{s2}
\end{bmatrix}
$$

$p_s$ system force vector
$q_s$ system load vector
$r_s$ 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
$\mu$ 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.

$$
a = \circ \mu \sqrt{\mathbf{d}_s' \mathbf{d}_s}
$$

$a$ general displacement norm
$\circ$ sign ($+$ or $-$ as determined by the preceding load path)
$\mu$ 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):

$$
a = \circ \mu \sqrt{\sum_i d_{i}^2 + g^2 \sum_k \beta_k^2}
$$

$d_i$ node displacement coordinate
$\beta_k$ 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:
\[ b = \circ \sqrt{\sum_{i} p_i^2} \] 

force norm

\[ \circ \] 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):

\[ b = \circ \sqrt{\sum_{i} p_i^2 + \frac{1}{g^2} \sum_{k} m_k^2} \] 

The calibration factor \( \mu \) is chosen so that the slope of the curve for the first step of the analysis has a specified value \( m \):

\[ \mu = \frac{1}{m} \sqrt{\frac{p_i}{d_i}} \] 

\( p_i \) normalised force computed by linear analysis for the load pattern \( d_i \) 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 \( d_{s1} \), as follows:

\[ \bar{a} = \circ \mu \sqrt{d_{s1} d_{s1}} \] 

\( \bar{a} \) special displacement norm \( d_{s1} \) free system displacement vector \( \circ \) sign (+ or – as determined by the preceding load path)

The special displacement norm for frames is given by:

\[ \bar{a} = \circ \mu \sqrt{\sum_{i} d_i^2 + g^2 \sum_{k} \beta_k^2} \] 

\( d_i \) node displacement coordinate in system vector \( d_{s1} \) \( \beta_k \) node rotation in system vector \( d_{s1} \)

The special force norm is defined with the system load vector \( q_s \). The special force norm for trusses is given by:

\[ \bar{b} = \circ \sqrt{q_s q_s} \] 

\( \bar{b} \) special force norm

The special displacement norm for frames is given by:

\[ \bar{\delta} = \circ \sqrt{\sum_{i} p_i^2 + \frac{1}{g^2} \sum_{k} m_k^2} \] 

\( p_i \) node load coordinate in system vector \( q_s \) \( m_k \) node rotation coordinate in system vector \( q_s \)
**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:

\[
h_s = \sqrt{(a_{s+1} - a_s)^2 + (b_{s+1} - b_s)^2}
\]

(6.11)

**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:

\[
\begin{align*}
    u_2 &= \frac{h_0}{2s} P_2 \\
    s &= AE \left( \frac{h_0}{L_0} \right)^3 \\
    R_{1A} &= -R_{1B} = \frac{m}{2} P_2 \\
    R_{2A} &= R_{2B} = \frac{1}{2} P_2
\end{align*}
\]

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):

\[
\begin{align*}
    P_2 &= s w_z (1+w_z)(2+w_z) \\
    R_{1B} &= -R_{1A} = \frac{s}{2} m w_z (2+w_z) \\
    R_{2A} &= R_{2B} = \frac{s}{2} w_z (1+w_z)(2+w_z)
\end{align*}
\]

The general displacement norm for solution 1 is computed with (6.2):

\[
a_t = \mu \sqrt{\sum d_i^2 d_i} = \mu h_0 w_z = \frac{s}{5} w_z \sqrt{2(3+m^2)}
\]

The general force norm for solution 1 is computed with (6.4):

\[
b_t = \sqrt{P_2^2 + R_{1A}^2 + R_{1B}^2 + R_{2A}^2 + R_{2B}^2} = s w_z (2+w_z) \sqrt{\frac{3}{2}(1+w_z)^2 + \frac{1}{2} m^2}
\]
Figure 6.2 shows the general load path of solution 1 for a truss with aspect ratio $m = 0.6$. The sign of the force norm changes for the displacement $w_2 = -2.00$ which corresponds to point $(a_1, b_1) = (1.037\text{s}, 0.0)$. The snap-through displacement $w_2 = -0.422650$ in (2.21) corresponds to point A with coordinates $(a_1, b_1) = (0.2191\text{s}, 0.5497\text{s})$ 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 $(a_1, b_1) = (a_2, b_2) = (0.2441\text{s}, 0.5577\text{s})$.

After the bifurcation point, the load path can follow either solution 1 or solution 2. The displacement $u_2$ for solution 2 is given by (2.41):

$$w_2 = 1 - 2m^2 - (1 + w_2)^2 \quad \text{for} \quad w_2 \leq -1 + \sqrt{1 - 2m^2} = -0.470850$$

The general displacement norm for solution 2 is computed with (6.2):

$$a_2 = \bigcirc \mu h_0 \sqrt{w_1^2 + w_2^2} = \bigcirc \mu h_0 \sqrt{-2(m^2 + w_2)} = \bigcirc \frac{2s}{5} \sqrt{-(m^2 + w_2)(m^2 + 3)}$$
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):

\[
\begin{align*}
P_2 &= -2sm^2(1+w_2) \\
R_{1A} &= -R_{1B} = sm(m+w_1)(m-w_1) \\
R_{2A} &= sm(1+w_2)(m-w_1) \\
R_{2B} &= sm(1+w_2)(m+w_1)
\end{align*}
\]

The general force norm for solution 2 is computed with (6.4) using $w_1^2$ from (2.41):

\[
b_2 = \circ \frac{\sqrt{P_2^2 + R_{1A}^2 + R_{1B}^2 + R_{2A}^2 + R_{2B}^2}}{u_2} = \circ s m \sqrt{2(1+w_2)^2(3m^2 + w_1^2) + 2(m^2 - w_1^2)^2}
\]

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):

\[
\bar{a}_1 = \circ \mu \sqrt{d_{i1}^2} = \circ \mu h_0 w_2 = \circ \frac{2}{5} s w_2
\]

The special force displacement norm is computed with (6.9):

\[
\bar{b}_1 = \circ \sqrt{q_i q_{i2}} = \circ P_2 = \circ s w_2 (1+w_2)(2+w_2)
\]

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):

\[
\bar{a}_2 = \circ \mu h_0 \sqrt{w_1^2 + w_2^2} = \circ \mu h_0 \sqrt{-2(m^2 + w_2^2)} = \circ \frac{2s}{5} \sqrt{-2(m^2 + w_2^2)}
\]

The special force norm for solution 2 is computed with (6.9):

\[
\bar{b}_2 = \circ \sqrt{P_2^2} = \circ 2s m^2 (1+w_2)
\]

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 $\Delta w_2$ 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 $\Delta w_1$ 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:

$$K_s \Delta d_s = e_s + \Delta q_s + \Delta r_s$$  \hspace{1cm} (6.12)

$K_s$ incremental system stiffness matrix
$e_s$ system error vector
$\Delta d_s$ increment of the system displacement vector
$\Delta q_s$ increment of the system load vector
$\Delta r_s$ increment of the system reaction vector

The contribution of a bar to matrix $K_s$ follows from (4.88):

$$d_e K_s = T_e^T R_{esy} K_e R_{esy}^T T_e$$  \hspace{1cm} (6.13)

$d_e K_s$ contribution of bar $e$ to the incremental system stiffness matrix
$T_e$ topology matrix for bar $e$
$R_{esy}$ bar rotation matrix from support space to reference space, see (4.29)
$K_e$ incremental bar stiffness matrix, see (4.89)

The contribution of a frame member to matrix $K_s$ follows from (5.217):

$$d_e K_s = T_e^T R_{esz} K_e R_{esz}^T T_e$$  \hspace{1cm} (6.14)

$d_e K_s$ contribution of member $e$ to the incremental system stiffness matrix
$T_e$ topology matrix for member $e$
$R_{esz}$ member rotation matrix from support space to instant space, see (5.76)
$K_e$ 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 $K_e$ are referred to reference space. In the case of members, the coefficients of $K_e$ 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):

$$K_{es} = R_e^T K_e R_e$$  \hspace{1cm} (6.15)

$K_{es}$ 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.
Addition to the system matrix: The rotated incremental stiffness matrix $K_{es}$ is used to compute the contribution of the element to the incremental system stiffness matrix by determining the product $T_e^T K_{es} T_e$ in (6.13) and (6.14). Figure 6.4 shows a bar $e$ whose end nodes have the system displacement coordinates $d_i, d_m, d_s$ and $d_k$. Variable $d_i$ is associated with row $i$ and column $i$ of the system matrix. The product $T_e^T K_{es} T_e$ is evaluated explicitly in the figure. It can be seen that this multiplication is inefficient because most of the coefficients of $T_e$ are null and the remaining coefficients are 1.

\[ (6.16) \]

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:

\[ n_0 \quad n_1 \quad n_2 \quad n_3 \]

\[ j = 0 \quad 1 \quad 2 \quad 3 \]

$n_j$ system index of the displacement coordinate with index $j$ in the element vector

Figure 6.4 Contribution of bar $e$ to the system stiffness matrix
The index table for the element in figure 6.4 is

\[
\begin{array}{|c|c|c|c|}
\hline
m & s & i & k \\
\hline
0 & 1 & 2 & 3 \\
\hline
\end{array}
\]  

(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 \( K_{es} \) be renamed \( k_{m} \) 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 \( K_{es} \) are added at the same location in the system matrix \( K_s \) as in the multiplication in figure 6.4:

\[
p: \{0,1,2,3\} \rightarrow \{n_0,n_1,n_2,n_3\}
\]

\[
j = p(i) \quad n = p(m)
\]

\[
s_{jn} \leftarrow s_{jn} + k_{im}
\]  

(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 \( K_{es} \) to system matrix \( K_s \)

**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:

\[ q_j \leftarrow q_j + \bar{q}_j \]  

(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):

\[ d_e q_e = T_e^T R_{es} q_e \]  

(6.21)

\( d_e q_e \)  

contribution of member e to the system load vector

\( T_e \)  

topology matrix for bar e

\( R_{es} \)  

element rotation matrix from support space to instant space, see (5.76)

\( q_e \)  

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):

\[ q_{es} = R_{es} q_e \]  

(6.22)

\( q_{es} \)  

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.

\[
\begin{bmatrix}
q_e \\
q_1 \\
q_2 \\
R_{a} \\
0 \\
0 \\
R_{a}
\end{bmatrix}
\begin{bmatrix}
R_{a} \\
0 \\
R_{a}
\end{bmatrix}
= 
\begin{bmatrix}
R_{a} q_e \\
R_{a} q_1 \\
R_{a} q_2
\end{bmatrix}
\]  

(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:

\[ p: \{0,1,2,3\} \rightarrow \{n_0,n_1,n_2,n_3\} \]

\[ n = p(k) \]

\[ s_n \leftarrow s_n + q_k \]  

(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):

\[
(\mathbf{K}_e^{(s)} - \mathbf{K}_e^{(s-1)}) \Delta \mathbf{d}_e^{(s)} + \mathbf{K}_e^{(s)} \Delta \mathbf{d}_e^{(s)} = \Delta \mathbf{q}_e^{(s)} + \Delta \mathbf{r}_e^{(s)} \quad (6.31)
\]

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:

\[
(\mathbf{K}_e^{(s)} - \mathbf{K}_e^{(s-1)}) \Delta \mathbf{d}_e^{(s)} = \Delta \mathbf{K}_e^{(s)} \Delta \mathbf{d}_e^{(s)} \quad (6.32)
\]

- \( \Delta \mathbf{K}_e^{(s)} \): 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:
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:

\[
K^{(s)}_s \Delta d^{(s)} = \Delta q^{(s)} + \Delta r^{(s)}
\]  

\(K^{(s)}_s\) approximate secant stiffness matrix for step \(s\)
\(\Delta d^{(s)}\) approximate displacement increment for step \(s\)
\(\Delta r^{(s)}\) approximate reaction increment for step \(s\)

Equation \((6.35)\) is solved for the approximate displacement increment \(\Delta d^{(s)}\) and the approximate reaction increment \(\Delta r^{(s)}\) 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:

\[
e^{(s)} = q^{(s)} + r^{(s)} - K^{(s)}_s d^{(s)}
\]  

\(e^{(s)}\) error vector containing the unbalanced forces in the approximate state \(s\)
\(d^{(s)}\) approximate total displacement in state \(s\)
\(r^{(s)}\) approximate total reaction in state \(s\)

Assume that the unbalanced forces are due to the use of the approximate secant matrix \(K^{(s)}_s\) instead of the exact secant matrix in equation \((6.33)\):

\[
K^{(s)}_s \Delta q^{(s)} = \Delta q^{(s)} + \Delta r^{(s)} + e^{(s)}
\]  

\(K^{(s)}_s\) exact secant stiffness matrix
\(\Delta q^{(s)}\) exact load increment for the load step \(s\)
\(\Delta r^{(s)}\) approximate reaction increment for step \(s\)
\(e^{(s)}\) error vector containing the unbalanced forces in the approximate state \(s\)

The exact load increment \(\Delta q^{(s)}\) is substituted from expression \((6.35)\) into \((6.37)\):

\[
K^{(s)}_s (\Delta d^{(s)} - \Delta d^{(s)}) = e^{(s)} + (\Delta r^{(s)} - \Delta r^{(s)})
\]  

Equation \((6.38)\) is solved for the correction \(\Delta d^{(s)} - \Delta d^{(s)}\) in the displacement and the correction \(\Delta r^{(s)} - \Delta r^{(s)}\) 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 $x^{(s,m)}$:

\[
\begin{bmatrix}
K_{11}^{(s,m)} & K_{12}^{(s,m)} & \Delta d_1^{(s,m)} \\
K_{21}^{(s,m)} & K_{22}^{(s,m)} & \Delta d_2^{(s,m)}
\end{bmatrix}
\begin{bmatrix}
\Delta q_1^{(s,m)} \\
\Delta q_2^{(s,m)}
\end{bmatrix}
= 
\begin{bmatrix}
\Delta q_1^{(s,m)} \\
\Delta q_2^{(s,m)}
\end{bmatrix}
+ 
\begin{bmatrix}
0 \\
\Delta r_2^{(s,m)}
\end{bmatrix}
+ 
\begin{bmatrix}
e_1^{(s)} \\
e_2^{(s)}
\end{bmatrix}
\]  

(6.39)

- $K_{ki}^{(s,m)}$: secant stiffness matrix for iteration cycle $m$ in step $s$
- $e_k^{(s)}$: error in state $s$
- $\Delta d_1^{(s,m)}$: free displacement increment in cycle $m$ of step $s$
- $\Delta d_2^{(s,m)}$: prescribed displacement increment in cycle $m$ of step $s$
- $\Delta q_1^{(s,m)}$: load increment in cycle $m$ of step $s$
- $\Delta q_2^{(s,m)}$: reaction increment in cycle $m$ of step $s$

Let the load pattern $q_1$ and the displacement pattern $d_2$ that act on the structure be specified. The fraction $\lambda$ 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:

\[
\Delta q_1^{(s,m)} = \Delta \lambda^{(s,m)} q_1 
\]  

(6.40)

\[
\Delta q_2^{(s,m)} = \Delta \lambda^{(s,m)} q_2 
\]  

(6.41)

\[
\Delta d_1^{(s,m)} = \Delta \lambda^{(s,m)} d_1 
\]  

(6.42)

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 $d_1^{(s,m)}$ and the reactions $r_2^{(s,m)}$ for the load pattern $q_1^{(s,m)}$ and the displacement pattern $d_2$ are computed with the secant stiffness matrix.
\[ K^{(s,m)}_s d^{(s,m)} = q^{(s,m)}_i + r^{(s,m)}_i \]  
\[ d^{(s,m)}_{1t} = (K^{(s,m)}_{11})^{-1} (q^{(s,m)}_{1t} - K_{12} d^{(s,m)}_{2t}) \] 
\[ r^{(s,m)}_{2t} = K_{21} d^{(s,m)}_{1t} + K_{22} d^{(s,m)}_{2t} - q^{(s,m)}_{2t} \]

\[
\begin{bmatrix}
K_{11} & K_{12} & d^{(s,m)}_{1t} \\
K_{21} & K_{22} & d^{(s,m)}_{2t}
\end{bmatrix}
= \begin{bmatrix}
q^{(s,m)}_{1t} \\
q^{(s,m)}_{2t}
\end{bmatrix}
+ \begin{bmatrix}
0 \\
r^{(s,m)}_{2t}
\end{bmatrix}
\]

\( K^{(s,m)} \) incremental stiffness matrix for iteration \( m \) in step \( s \)
\( d^{(s,m)}_{1t} \) free displacements due to the pattern load in iteration \( m \) of step \( s \)
\( r^{(s,m)}_{2t} \) reactions due to the pattern load in iteration \( m \) of step \( s \)

The force vector \( p^{(s,m)}_t \) for the total load is the sum of the load and reaction vectors:
\[ p^{(s,m)}_t = q^{(s,m)}_i + r^{(s,m)}_i \]  
\[ p^{(s,m)}_{1t} = \begin{bmatrix}
q^{(s,m)}_{1t} \\
q^{(s,m)}_{2t}
\end{bmatrix}
+ \begin{bmatrix}
0 \\
r^{(s,m)}_{2t}
\end{bmatrix}
\]

(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 \( \Delta \lambda \):

\[
K^{(s,m)} \Delta d^{(s,m)} = \Delta \lambda^{(s,m)} q^{(s)} + \Delta r^{(s,m)} + e^{(s,m)}
\]

\[
\begin{bmatrix}
K^{(s,m)}_{11} & K^{(s,m)}_{12} & \Delta d^{(s,m)}_{1t} \\
K^{(s,m)}_{21} & K^{(s,m)}_{22} & \Delta d^{(s,m)}_{2t}
\end{bmatrix}
= \begin{bmatrix}
\Delta \lambda^{(s,m)} q^{(s)}_{1t} \\
\Delta \lambda^{(s,m)} q^{(s)}_{2t}
\end{bmatrix}
+ \begin{bmatrix}
0 \\
0
\end{bmatrix}
+ \begin{bmatrix}
e^{(s)}_1 \\
e^{(s)}_2
\end{bmatrix}
\]

\( \Delta \lambda^{(s,m)} \) load factor increment for iteration \( m \) in step \( s \)
\( \Delta d^{(s,m)}_t \) increment of the free displacements for iteration \( m \) in step \( s \)
\( \Delta r^{(s,m)}_{2t} \) increment of the reactions for iteration \( m \) in step \( s \)
\( e^{(s)}_k \) 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:

\[
\Delta d^{(s,m)}_t = \Delta \lambda^{(s,m)} d^{(s)}_t + f^{(s,m)}_t
\]

\[
f^{(s,m)}_t = (K^{(s,m)}_{11})^{-1} e^{(s,m)}_1
\]

\[
\Delta r^{(s,m)}_{2t} = \Delta \lambda^{(s,m)} r^{(s,m)}_{2t} + t^{(s,m)}_2
\]

\[
t^{(s,m)}_2 = K_{21} f^{(s,m)}_1 - e^{(s)}_2
\]

The solution for the displacement increments in equations (6.45) is:

\[
\Delta d^{(s,m)}_t = \Delta \lambda^{(s,m)} d^{(s)}_t + f^{(s,m)}_t
\]

\[
\frac{\Delta d^{(s,m)}_t}{\Delta d^{(s,m)}_t} = \Delta \lambda^{(s,m)} d^{(s)}_t + f^{(s,m)}_t
\]  
\[
\frac{\Delta d^{(s,m)}_t}{\Delta d^{(s,m)}_t} = \begin{bmatrix}
q^{(s,m)}_{1t} \\
q^{(s,m)}_{2t}
\end{bmatrix}
+ \begin{bmatrix}
0 \\
r^{(s,m)}_{2t}
\end{bmatrix}
\]

184
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):

\[
\Delta q^{(s,m)} = \Delta \lambda^{(s,m)} q_i^{(m)} \\
\Delta q_1^{(s,m)} = \Delta \lambda^{(s,m)} q_i^{(s,m)} \\
\Delta q_2^{(s,m)} = \Delta \lambda^{(s,m)} q_i^{(s,m)} \\
\Delta r^{(s,m)} = \Delta \lambda^{(s,m)} r_i^{(s)} + t_i^{(s,m)} \\
\Delta r_1^{(s,m)} = \Delta \lambda^{(s,m)} r_i^{(s,m)} + \Delta t_1^{(s,m)} \\
\Delta r_2^{(s,m)} = \Delta \lambda^{(s,m)} r_i^{(s,m)} + \Delta t_2^{(s,m)}
\]

The increment of the force vector follows from (6.51) and (6.52):

\[
\Delta p^{(s,m)} = \Delta \lambda^{(s,m)} p_i^{(s,m)} + t_i^{(s,m)} \\
\Delta p_1^{(s,m)} = \Delta \lambda^{(s,m)} p_i^{(s,m)} + \Delta t_1^{(s,m)} \\
\Delta p_2^{(s,m)} = \Delta \lambda^{(s,m)} p_i^{(s,m)} + \Delta t_2^{(s,m)}
\]

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 \(\Delta d^{(s,m)}, \Delta q^{(s,m)}\) and \(\Delta r^{(s,m)}\). 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 \(s\):

\[
d^{(s+1,m)} = d^{(s)} + \Delta d^{(s,m)} \\
q^{(s+1,m)} = q^{(s)} + \Delta q^{(s,m)} \\
r^{(s+1,m)} = r^{(s)} + \Delta r^{(s,m)}
\]

(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:

\[
h_0^2 = \Delta \lambda_0^2 \mathbf{p}_0^T \mathbf{p}_0 + \mu^2 \Delta \lambda_0^2 \mathbf{d}_0^T \mathbf{d}_0
\]  
(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 \( \mathbf{f} \) 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:

\[
a_1 = \odot \mu \sqrt{\mathbf{d}^T \mathbf{d}} \quad a_2 = \odot \mu \sqrt{(\mathbf{d} + \Delta \mathbf{d})^T (\mathbf{d} + \Delta \mathbf{d})} \\
b_1 = \odot \sqrt{\mathbf{p}^T \mathbf{p}} \quad b_2 = \odot \sqrt{(\mathbf{p} + \Delta \mathbf{p})^T (\mathbf{p} + \Delta \mathbf{p})}
\]  
(6.58)

(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:

\[
\mathbf{d} = \mathbf{d} + \mathbf{f} + \mathbf{z} \mathbf{a}
\]  
(6.60)

\( \mathbf{d} \) 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)  
\( \mathbf{z} \) load factor increment \( \Delta \lambda \) for the step

The system force vector at the end of the step is denoted by:
\[ \mathbf{\tilde{p}} = \mathbf{p} + \mathbf{t} + z \mathbf{b} \]  
(6.61)

\[ \mathbf{\tilde{p}} \quad \text{system force vector at the end of the step} \]

\[ \mathbf{p} \quad \text{system force vector at the beginning of the step} \]

\[ \mathbf{t} \quad \text{force correction vector} \]

\[ \mathbf{b} \quad \text{force reference vector (pattern force or reaction due to eigenvector)} \]

\[ z \quad \text{load factor increment } \Delta \lambda \text{ for the step} \]

The displacement norms at the beginning and the end of the step are:

\[ a_1 = \mu \sqrt{(d + f)^T(d + f)} \]  
(6.62)

\[ a_2 = \mu \sqrt{(d + f + z \mathbf{a})^T(d + f + z \mathbf{a})} \]

\[ \mu \quad \text{displacement calibration factor, see (6.2)} \]

The force norms at the beginning and the end of the step are:

\[ b_1 = \sqrt{(\mathbf{p} + \mathbf{t})^T(\mathbf{p} + \mathbf{t})} \]  
(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 \( \mathbf{s} \) is set equal to the chord length \( h_0 \) in the first load step:

\[ (a_2 - a_1)^2 + (b_2 - b_1)^2 = h_0^2 \]  
(6.64)

\[ h_0 \quad \text{chord length of the first step} \]

The following dimensionless parameters are defined:

\[ n_0 = (d + f)^T(d + f) \]

\[ n_1 = \frac{\mathbf{a}^T(d + f)}{n_0} \]

\[ n_2 = \frac{\mathbf{a}^T\mathbf{a}}{n_0} \]  
(6.65)

\[ n_3 = (\mathbf{p} + \mathbf{t})^T(\mathbf{p} + \mathbf{t}) \]

\[ n_4 = \frac{\mathbf{b}^T(\mathbf{p} + \mathbf{t})}{n_3} \]

\[ n_5 = \frac{\mathbf{b}^T\mathbf{b}}{n_3} \]  
(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):

\[ (a_2 - a_1)^2 = \mu^2 n_0 (2 + 2n_z z + n_z^2z^2 - 2\sqrt{1 + 2n_z z + n_z^2z^2}) \]  
(6.67)

\[ (b_2 - b_1)^2 = n_2 (2 + 2n_z z + n_z^2z^2 - 2\sqrt{1 + 2n_z z + n_z^2z^2}) \]  
(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:

\[ f(z) = (a_2 - a_1)^2 + (b_2 - b_1)^2 - h_0^2 \]  
(6.69)

\[ f(z) \quad \text{residual function for Newton-Raphson iteration} \]

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 \( z = 0 \):
\[(a_2 - a_1)^2 = \mu^2 n_3 (2 + 2n_z z + n_z z^2 - 2r) \tag{6.70}\]
\[(b_2 - b_1)^2 = n_3 (2 + 2n_z z + n_z z^2 - 2s)\]
\[r = \sqrt{1 + 2n_z z + n_z z^2} \tag{6.71}\]
\[s = \sqrt{1 + 2n_z z + n_z z^2}\]

The derivatives of the residual function are given by:
\[\frac{df}{dz} = 2\mu^2 n_3 (n_4 + n_z z)(1 - \frac{1}{r}) + 2n_3 (n_4 + n_z z)(1 - \frac{1}{s}) \tag{6.72}\]
\[\frac{d^2f}{dz^2} = 2\mu^2 n_3 \left( \frac{n_z (n_1 + n_z z)}{r^3} \right) + 2n_3 \left( \frac{n_z (n_1 + n_z z)}{s^3} \right)\]
\[\frac{d^3f}{dz^3} = 6\mu^2 n_3 \left( \frac{n_z (n_1 + n_z z)}{r^5} \right) + 6n_3 \left( \frac{n_z (n_4 + n_z z)}{s^5} \right)\]
\[\frac{d^4f}{dz^4} = 6\mu^2 n_3 \left( \frac{n_z (n_1 + n_z z)}{r^7} \right) + 5(n_4 + n_z z)^4 + 6n_3 \left( \frac{n_z (n_4 + n_z z)}{s^7} \right)\]

At the origin \(z = 0\), the roots are \(r = s = 1\), so that the function and its derivatives are given by the following expressions:
\[f_0 = -h_0^2 \tag{6.73}\]
\[\frac{df_0}{dz} = 0\]
\[\frac{d^2f_0}{dz^2} = 2\mu^2 n_0 n_1^2 + 2n_3 n_4^2\]
\[\frac{d^3f_0}{dz^3} = 6\mu^2 n_0 n_1 (n_2 - n_1^2) + 6n_3 n_4 (n_5 - n_2^2)\]
\[\frac{d^4f_0}{dz^4} = 6\mu^2 n_0 (n_2 - n_1^2)(n_2 - 5n_1^2) + 6n_3 (n_5 - n_2^2)(n_5 - 5n_2^2)\]

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 \(z\):
\[f(z) = f_0 + \frac{df_0}{dz} z + \frac{1}{2} \frac{d^2f_0}{dz^2} z^2 + \frac{1}{6} \frac{d^3f_0}{dz^3} z^3 + \frac{1}{24} \frac{d^4f_0}{dz^4} z^4 \tag{6.74}\]

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):

\[
f_m + \frac{df_m}{dz}(z_{m+1} - z_m) = 0
\]

\[
z_{m+1} = z_m - \left( \frac{df_m}{dz} \right)^{-1} f_m
\]  

(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 \( e \)). Consider the load curve for solution 1 of the 2-bar truss shown in figure 2.4. The pattern load on the truss is \( p_{21} = -0.3849 \). 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 \( h_0 = 0.1 \) 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:
\[
ku = q \quad \text{with} \quad k = (1 + u)(2 + u) \quad (6.76)
\]

Let the displacement increment in iteration 0 of step \( s \) be \( \Delta u \). 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):
\[
k_s = (2 + 6u + 3u^2) + 3(1 + u)\Delta u + \Delta u^2
\]

![Figure 6.6 Load step for the 2-bar truss in figure 2.1](image)

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 $q_t = -0.384900$:

$$
k_t^{(s,0)} = (1 + u^{(s)})(2 + u^{(s)}) = 0.687500$$

$$u_t^{(s,0)} = q_t : u_t^{(s,0)} = \frac{-0.384900}{0.687500} = -0.559855$$

The load factor increment $\Delta \lambda^{(s,0)}$ is computed so that the length of the arc increment is $h = 0.1$:

$$\Delta \lambda^{(s,0)} = 0.147189$$

$$\Delta u^{(s,0)} = \Delta \lambda^{(s,0)} u_t^{(s,0)} = -0.082404$$

$$\Delta q^{(s,0)} = \Delta \lambda^{(s,0)} q_t = -0.056653$$

**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 $h = 0.1$:

$$\Delta \lambda^{(s,1)} = 0.117832$$

$$\Delta u^{(s,1)} = \Delta \lambda^{(s,1)} u_t^{(s,1)} = -0.089664$$

$$\Delta q^{(s,1)} = \Delta \lambda^{(s,1)} q_t = -0.044276$$

**Convergence:** The iteration is continued, starting with the displacement increment $\Delta u^{(s,m)} = -0.089664$. 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.

<table>
<thead>
<tr>
<th>$m$</th>
<th>$k_t$</th>
<th>$u_t$</th>
<th>$\Delta \lambda$</th>
<th>$u$</th>
<th>$q$</th>
<th>$q_{exact}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.494914</td>
<td>-0.777710</td>
<td>0.115241</td>
<td>-0.339624</td>
<td>-0.372481</td>
<td>-0.372388</td>
</tr>
<tr>
<td>1</td>
<td>0.493878</td>
<td>-0.779343</td>
<td>0.115047</td>
<td>-0.339661</td>
<td>-0.372407</td>
<td>-0.372400</td>
</tr>
<tr>
<td>2</td>
<td>0.493801</td>
<td>-0.779463</td>
<td>0.115033</td>
<td>-0.339664</td>
<td>-0.372401</td>
<td>-0.372401</td>
</tr>
</tbody>
</table>

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 $CD$, $CE$ and $CF$ equal $h$, so that the points $D$, $E$ and $F$ lie on a circle with centre $C$ and radius $h$. 

191
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 $Au = q$ with fully populated coefficient matrix $A$ and fully prescribed absolute term $q$ is considered. The unknown vector $u$ 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 $A$ and an absolute term $q$ that is prescribed. The term $u$ is to be determined.

$$Au = q \quad (6.79)$$

$A$ coefficient matrix  
$u$ unknown vector  
$q$ absolute vector

The square matrix $A$ is decomposed into the product of a left triangular matrix $S$ with unit diagonal coefficients, a diagonal matrix $D$ whose diagonal coefficients can be positive or negative but not null, and the transpose of $S$.

$$A := SDS^T \quad (6.80)$$

$S$ left triangular matrix with diagonal coefficients 1.0  
$D$ diagonal matrix with diagonal coefficients which are not null

Decomposition (6.80) is substituted into (6.79). A temporary vector $y$ is defined and substituted into the equation:

$$SDS^Tu = q \quad (6.81)$$

$$y := S^Tu$$

$$SDy = q \quad (6.82)$$
Due to the triangular population of $S$, equation (6.82) is conveniently solved for the temporary vector $y$:

$$Dy = S^{-1}q$$  \hspace{1cm} (6.83)

$$y = D^T S^{-1}q$$ \hspace{1cm} (6.84)

The temporary vector $y$ is substituted into (6.81), which is then solved for $u$:

$$S^T u = y$$

$$u = S^T y$$ \hspace{1cm} (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 $A$ is shown in figure 6.7. The known matrix $A$ 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 $S$ and the diagonal coefficients $d_0, ..., d_n$ of $D$ are known.

![Figure 6.7 Decomposition of a symmetric real matrix](image-url)
The decomposition is started by computing $d_0$ and column 0 of $S$, using the known value $s_{00} = 1$:

$$d_0 s_{00} = a_{00}; \quad d_0 = a_{00} \quad (6.86)$$

$$d_0 s_{i0} = a_{i0}; \quad s_{i0} = \frac{a_{i0}}{d_0} \quad r = 1, \ldots, n - 1 \quad (6.87)$$

Assume that the diagonals $d_0, \ldots, d_{p-1}$ of $D$ and the coefficients of columns 0 to $p-1$ of $S$ have been computed. The decomposition is continued by computing the diagonal $d_p$ and the coefficients of column $p$ of $S$ using the known value $s_{pd} = 1$:

$$\sum_{k=0}^{p-1} s_{pk} (d_k s_{pk}) + d_p = a_{pp}$$

$$d_p = a_{pp} - \sum_{k=0}^{p-1} s_{pk} (d_k s_{pk}) \quad (6.88)$$

$$\sum_{k=0}^{p-1} s_{rk} (d_k s_{pk}) + s_{rp} d_p = a_{rp}$$

$$s_{rp} = \frac{1}{d_p} \left( a_{rp} - \sum_{k=0}^{p-1} s_{rk} (d_k s_{pk}) \right) \quad (6.89)$$

**Profile matrix**: Let matrix $A$ 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 $A$ is called the row profile in row $k$. The row index of the last coefficient that is not null in column $m$ of matrix $A$ 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.

![Profile Matrix Diagram](image)

**Figure 6.8** Row and column profiles of a coefficient 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):

$$ s_{ij} = \frac{1}{d_{i}} \left( a_{ij} - \sum_{k=m}^{j-1} s_{ik} d_{jk} s_{kj} \right) $$

(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 \( y \) in (6.84) is called the forward sweep of the solution algorithm. The coefficients of \( y \) are computed in sequence from \( y_0 \) to \( y_{n-1} \):

\[
d_r \sum_{k=p[r]}^r s_k y_k = q_r \quad r = 0, \ldots, n-1
\]

\[
y_r = \frac{1}{d_r} \left( q_r - \sum_{k=p[r]}^{r-1} s_k y_k \right) \quad (6.91)
\]

Backward sweep: The computation of the solution vector \( u \) in (6.85) is called the backward sweep of the solution algorithm. The matrix \( S^T \) is not stored in the solution procedure. It is derived from matrix \( S \), which is stored row-wise. It is advantageous to arrange the operations on \( S^T \) column-wise, so that the corresponding operations on \( S \) can be performed row-wise. The coefficients of \( u \) 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 \( r-1 \):

\[
y_k \leftarrow y_k - s_{nk} u_r \quad r = n-1, \ldots, 1; \quad k = p[r], \ldots, r-1 \quad (6.92)
\]

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 \( A \) 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 \( S^{-1} \) 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 \( S \) 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 $\mathbf{p}$:

$$\mathbf{K} \mathbf{d} = \mathbf{p} \quad (7.1)$$

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 $\mathbf{K} \mathbf{d}$ 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 $\Delta p_i$ of the load coordinate $p_i$ due to an increment $\Delta d_i$ 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 $\mathbf{K}'$. The coefficients of $\mathbf{K}'$ are not the partial derivatives of the coefficients of the stiffness matrix $\mathbf{K}$ in (7.1), but rather the derivatives of the products $k_{im}d_m$.

$$\mathbf{K}' \Delta \mathbf{d} = \Delta \mathbf{p} \quad (7.3)$$

$\mathbf{K}'$ 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 $\Delta d$, which is computed for a load increment $\Delta p$ with equation (7.3), is not exact, since the variation of the tangential stiffness matrix $K'$ with the displacement is not taken into account. The total displacement $d + \Delta d$ computed for the total load $p + \Delta p$ 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 $K^s$ at point A is defined so that the solution $\Delta d^o$ of equation (7.4) leads to the total displacement of point B.

$$K^s \Delta d = \Delta p \quad (7.4)$$

- $K^s$: secant stiffness matrix from point A to point B on the load path
- $\Delta d$: displacement vector increment from point A to point B on the load path
- $\Delta p$: total force vector increment from point A to point B on the load path

In contrast with the coefficients of the tangent matrix $K'$, which are determined by differentiation, the coefficients of the secant matrix $K^s$ 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:

- equilibrium at A: $K(d) d = p$
- equilibrium at B: $K(d + \Delta d)(d + \Delta d) = p + \Delta p$

$$\begin{align*}
(K(d + \Delta d) - K(d))d + K(d + \Delta d) \Delta d &= \Delta p \\
(K(d + \Delta d) - K(d))d &= \Delta d
\end{align*} \quad (7.5)$$

The first term in equation (7.5) is transformed into the product of an auxiliary matrix $K^a$ with the displacement increment $\Delta d$:

$$K^a \Delta d = (K(d + \Delta d) - K(d))d \quad (7.6)$$

The substitution of equations (7.5) and (7.6) into equation (7.4) yields expression (7.7) for the secant matrix:

$$K^s = K(d + \Delta d) + K^a \quad (7.7)$$

Since the equilibrium equations (7.5) and (7.6) are highly nonlinear functions of the total displacement coordinates, the secant matrix $K^s$ is not unique. The decomposition $K^s \Delta d$ in equation (7.4) is made so that matrix $K^s$ 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:

\[
p_1 = w_1 (2m^2 + w^2 + w_2 (2 + w_2)) \quad \text{(7.8)}
\]

\[
p_2 = (1 + w_2) (w_1^2 + w_2 (2 + w_2)) \quad \text{(7.9)}
\]

These equations are arranged in the matrix form (7.1):

\[
\begin{bmatrix}
2m^2 + w_1^2 + w_2 & w_1 (1 + w_2) & w_1 \\
(1 + w_2) (2 + w_2) & (1 + w_2) (2 + w_2) & w_2
\end{bmatrix}
\begin{bmatrix}
w_1 \\
w_2
\end{bmatrix}
= \begin{bmatrix}
p_1 \\
p_2
\end{bmatrix}
\]

\[K\]

**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 \(K_1\) for solution 1 is obtained by substituting \(w_1 = 0\) into expression (7.10):

\[
K_1 w = p_2
\]

\[
\begin{bmatrix}
2m^2 + w_2 & 0 & w_1 \\
0 & (1 + w_2) (2 + w_2) & w_2
\end{bmatrix}
\begin{bmatrix}
w_1 \\
w_2
\end{bmatrix}
= \begin{bmatrix}
0 \\
p_2
\end{bmatrix}
\]

\[K_1\]

**total stiffness matrix for solution 1**

The total stiffness matrix \(K_2\) 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):

\[
K_2 w = p_2
\]

\[
\begin{bmatrix}
-w_2 (1 + w_2) & w_1 (1 + w_2) & w_1 \\
w_1 (1 + w_2) & (1 + w_2) (2 + w_2) & w_2
\end{bmatrix}
\begin{bmatrix}
w_1 \\
w_2
\end{bmatrix}
= \begin{bmatrix}
0 \\
p_2
\end{bmatrix}
\]

\[w_1 = \pm \sqrt{-w_2 (2 + w_2) - 2m^2}
\]

\[K_2\]

**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):

\[
K^1 \Delta w = \Delta p
\]

\[
\begin{bmatrix}
2m^2 + 3w_1^2 + w_2 (2 + w_2) & 2w_1 (1 + w_2) & \Delta w_1 \\
2w_1 (1 + w_2) & w_1^2 + 3w_2^2 + 6w_2 + 2 & \Delta w_2
\end{bmatrix}
\begin{bmatrix}
\Delta p_1 \\
\Delta p_2
\end{bmatrix}
\]

\[K^1\]

**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):
\[
\begin{align*}
K_1' \triangleq w &= \Delta p_2 \\
\begin{bmatrix} 2m^2 + w_2(2 + w_2) & 0 \\ 0 & 3w_2^2 + 6w_2 + 2 \end{bmatrix} \begin{bmatrix} \Delta w_1 \\ \Delta w_2 \end{bmatrix} &= \begin{bmatrix} 0 \\ \Delta p_2 \end{bmatrix}
\end{align*}
\]

**K**

The tangent stiffness matrix for solution 1

The tangent stiffness matrix for solution 2 in section 2.1 is obtained by substituting the condition \(2m^2 + \Delta w_1(2 + \Delta w_2) = 0\) into expression (7.13):

\[
\begin{align*}
K_2' \triangleq w &= \Delta p_2 \\
\begin{bmatrix} 2w_1^2 & 2w_1(1 + w_2) \\ 2w_1(1 + w_2) & 2(1 + w_2)^2 - 2m^2 \end{bmatrix} \begin{bmatrix} \Delta w_1 \\ \Delta w_2 \end{bmatrix} &= \begin{bmatrix} 0 \\ \Delta p_2 \end{bmatrix}
\end{align*}
\]

\[w_1 = \pm \sqrt{-w_2(2 + w_2) - 2m^2}\]

**K**

The tangent stiffness matrix for solution 2

### 7.1.4 Secant Stiffness of a 2-Bar Truss

The total stiffness matrix \(K^{(s)}\) of the truss for a displacement \((w_1, \Delta w_2)\) of the apex in step \(s\) is given in expression (7.10):

\[
K^{(s)} = p + \Delta p
\]

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 \(K^{(s+1)}\) becomes:

\[
\begin{align*}
K^{(s+1)}(w + \Delta w) &= p + \Delta p \\
\begin{bmatrix} 2m^2 + (w_1 + \Delta w_1)^2 + w_2 + \Delta w_2 \\ (w_1 + \Delta w_1)(1 + \Delta w_2) \end{bmatrix} \begin{bmatrix} w_1 + \Delta w_1 \\ w_2 + \Delta w_2 \end{bmatrix} &= \begin{bmatrix} p_1 + \Delta p_1 \\ p_2 + \Delta p_2 \end{bmatrix}
\end{align*}
\]

The secant stiffness matrix for the step from state \(s\) to state \(s + 1\) is obtained by subtracting (7.16) from (7.17):

\[
K^s = \Delta p
\]

\[
\begin{bmatrix} k_{11}^s & k_{12}^s \\ k_{21}^s & k_{22}^s \end{bmatrix} \begin{bmatrix} \Delta w_1 \\ \Delta w_2 \end{bmatrix} = \begin{bmatrix} \Delta p_1 \\ \Delta p_2 \end{bmatrix}
\]

The stiffness coefficients \(k_{im}^s\) are computed in the following steps:

\[
p_{11} = k_{11}^{(s+1)}(w_1 + \Delta w_1) - k_{11}^{(s)}w_1
\]

\[
= (2m^2 + (w_1 + \Delta w_1)^2 + w_2 + \Delta w_2)(w_1 + \Delta w_1) - (2m^2 + w_1^2 + w_2)(w_1)
\]

\[
= (2m^2 + 3w_1^2 + w_2 + 3w_1\Delta w_1 + \Delta w_1^2 + \Delta w_2)(w_1 + \Delta w_2)
\]

\[
p_{22} = k_{22}^{(s+1)}(w_2 + \Delta w_2) - k_{22}^{(s)}w_2
\]

\[
= (1 + w_2 + \Delta w_2)(2 + w_2 + \Delta w_2)(w_2 + \Delta w_2) - (1 + w_2)(2 + w_2)(w_2)
\]

\[
= (2 + 6w_2 + 3w_2^2 + 3(1 + w_2)(\Delta w_2 + \Delta w_2^2))(w_2)
\]
\[ p_{12} = k_{12}^{(s)}(w_2 + \Delta w_2) - k_{12}^{(s)} w_2 \]  
\[ = (w_1 + \Delta w_1)(1 + w_1 + \Delta w_2)(w_2 + \Delta w_2) - w_1(1 + w_2)w_2 \]  
\[ = (w_1(1 + 2w_2) + (1 + w_2)w_2 + w_i\Delta w_2)\Delta w_2 + (w_2(1 + w_2) + (w_2 + \Delta w_2)w_2)\Delta w_1 \]  

\[ p_{21} = k_{21}^{(s)}(w_1 + \Delta w_1) - k_{21}^{(s)} w_1 \]  
\[ = (w_1 + \Delta w_1)(1 + w_1 + \Delta w_2)(w_1 + \Delta w_1) - w_1(1 + w_2)w_1 \]  
\[ = (2w_1(1 + w_2) + (1 + w_2)w_1 + w_1\Delta w_1 + (w_1^2 + \Delta w_1(w_1 + \Delta w_1))\Delta w_1 \]  

Term \( k_{11}^{s} \) is formed by collecting the terms in \( p_{11} \) and \( p_{12} \) that are multiplied by \( \Delta w_1 \). Similarly, term \( k_{12}^{s} \) is formed by collecting the terms multiplied by \( \Delta w_2 \). Similarly, term \( k_{22}^{s} \) is formed by collecting the terms in \( p_{21} \) and \( p_{22} \) that are multiplied by \( \Delta w_1 \). Term \( k_{22}^{s} \) is formed by collecting the terms multiplied by \( \Delta w_2 \).

\[ k_{11}^{s} = 2m^2 + 3w_i^2 + w_i(2 + w_2) + (1 + w_2)w_i + (1 + w_2)w_i + \Delta w_i^2 + \Delta w_2^2 \]  
\[ k_{12}^{s} = k_{21}^{s} = 2w_1(1 + w_2) + (1 + w_2)w_1 + w_1\Delta w_2 \]  
\[ k_{22}^{s} = 2 + 2w_1 + 6w_2 + 3w_1^2 + w_1\Delta w_1 + 3(1 + w_2)w_2 + \Delta w_1^2 + \Delta w_2^2 \]  

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):

\[ k_{11}^{s} = 2m^2 + w_2(2 + w_2) + (1 + w_2)w_2 + \Delta w_2^2 \]  
\[ k_{12}^{s} = k_{21}^{s} = 0 \]  
\[ k_{22}^{s} = 2 + 6w_2 + 3w_1^2 + 3(1 + w_2)w_2 + \Delta w_2^2 \]  

The secant stiffness matrix for solution 2 in section 2.1 is obtained by substituting equation (2.28) into expression (7.23):

\[ k_{11}^{s} = 2w_1^2 + 2w_1 + (1 + w_2)w_1 + \Delta w_1^2 + \Delta w_1 \Delta w_2 \]  
\[ k_{12}^{s} = k_{21}^{s} = 2w_1(1 + w_2) + (1 + w_2)w_1 + w_1\Delta w_2 \]  
\[ k_{22}^{s} = 2 - 6m^2 - 2w_1^2 + w_1\Delta w_1 + 3(1 + w_2)w_2 + \Delta w_1^2 + \Delta w_2^2 \]  

Equation (2.28) is applied at the beginning and at the end of the load step and the resulting equations are subtracted:

\[ w_1^2 + w_2(2 + w_2) + 2m^2 = 0 \]  
\[ (w_1 + \Delta w_1)^2 + (w_2 + \Delta w_2)(2 + w_2 + \Delta w_2) + 2m^2 = 0 \]  
\[ \Delta w_1^2 + \Delta w_2^2 = -2w_1\Delta w_1 - 2(1 + w_2)\Delta w_2 \]  

Equation (7.26) is substituted into expressions (7.25):

\[ k_{11}^{s} = 2w_1^2 + w_1\Delta w_1 - (1 + w_2)\Delta w_2 \]  
\[ k_{12}^{s} = k_{21}^{s} = 2w_1(1 + w_2) + (1 + w_2)w_1 + w_1\Delta w_2 \]  
\[ k_{22}^{s} = 2 - 6m^2 - 2w_1^2 - w_1\Delta w_1 + (1 + w_2)\Delta w_2 \]  

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 $\mathbf{d}u$ without an infinitesimal change $\mathbf{d}q$ in the applied load. In general, the infinitesimal displacement will cause an increment $\mathbf{d}r$ 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 $\mathbf{d}u$ is null if the tangent stiffness matrix is singular.

If the second derivative of the load factor in the direction of the displacement $\mathbf{d}u$ 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 $\mathbf{d}u$ 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.

$$\det \mathbf{K}^t = 0 \quad \rightarrow \quad \text{the structure is in a singular configuration} \quad (7.28)$$

$\mathbf{K}^t$ 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):

S1: \[ w_1 = 0 \quad p_1 = 0 \] \[ p_2 = w_2(1 + w_2)(2 + w_2) \] (7.29)

S2: \[ p_1 = 0 \quad \text{for } m \leq \frac{1}{\sqrt{2}} = 0.7071 \] \[ p_2 = -2m^2(1 + w_2) \] \[ w_1^2 + w_2(2 + w_2) + 2m^2 = 0 \] \[ m \leq \frac{1}{\sqrt{2}} \] (7.30)

**Singular configurations:** The truss is in a singular configuration if the tangent stiffness matrix \( K^i \) is singular so that a nontrivial solution \( \Delta w \) exists for the homogeneous equation (7.13). Consider solution S1. Matrix \( K^i_1 \) 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:

**case 1:** \[ k^i_{22} = 3w_2^2 + 6w_2 + 2 = 0 \] \[ w_2 = -1 \pm \frac{1}{\sqrt{3}} \] \[ k^i_{11} = 2m^2 - \frac{2}{3} \] (7.31)

**case 2:** \[ k^i_1 = w_2(2 + w_2) + 2m^2 = 0 \] \[ w_2 = -1 \pm \sqrt{1 - 2m^2} \] \[ k^i_{22} = 2 - 6m^2 \] (7.32)

The value of the determinant of \( K^i_2 \) in (7.15) is \(-4w_2^2m^2\). Thus \( K^i_2 \) 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:

\[ m_s^2 = \frac{1}{3} \] \[ m_s = 0.5774 \] (7.33)

\[ w_{2s} = -0.4226 \] (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) \( m > m_c = 0.7071 \): 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 \( \Delta w_1 \) 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):

\[
\begin{align*}
    w_{2x}(1 + w_{2x})(2 + w_{2x}) &= -2m^2(1 + w_{2x}) = p_{2x} \\
    w_{2x} &= -1 \text{ or } w_{2x} = -1 + \sqrt{1 - 2m^2} \quad \text{(7.35)} \\
    p_{2x} &= 0 \text{ or } p_{2x} = -2m^2 \sqrt{1 - 2m^2} \quad \text{(7.36)}
\end{align*}
\]
The truss reaches the singular point B for S2 before it reaches the singular point C for S1.

Let \((\Delta w_1, \Delta w_2)\) 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 \(\Delta w_1\) is null and the load increment is:

\[
\Delta p_2 = (w_2 + \Delta w_2)(1 + w_2 + \Delta w_2)(2 + w_2 + \Delta w_2) - w_2(1 + w_2)(2 + w_2)
\]  

(7.37)

If the load path follows curve "b" of S2, the displacement increment \(\Delta w_1\) is chosen freely and the load increment is:

\[
\Delta p_2 = -2m^2(1 + w_2 + \Delta w_2) + 2m^2(1 + w_2) = -2m^2 \Delta w_2
\]

(7.38)

The truss can thus move to an equilibrium configuration that neighbours B either by a displacement increment \((0, \Delta w_2)\) on the curve for solution S1, or by a displacement increment \((\Delta w_1, \Delta w_2)\) 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 \((w_2^*, p_2^* )\) 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.

\(m = m_s\) : 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):

\[
(w_2^*, p_2^*) = (-0.4226, -0.3849)
\]

(7.39)

\(m_s < m < 0.7071\): 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.

\(m = 0.7071\) : 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 $K_{11}$ of the incremental governing equations (6.39). If this submatrix is singular, the pattern displacement $d_i$ in (6.43) and the corrective displacement $f_i$ in (6.47) cannot be determined. The submatrix $K_{11}$ 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 $K_{11}$ into the product of a left triangular matrix $S$ with unit diagonal coefficients, a diagonal matrix $D$ that can contain positive and negative coefficients $d_i$, and a right triangular matrix $S^T$. The determinant of $K_{11}$ is equal to the product of the coefficients of $D$:

$$K_{11} = SDS^T$$

$$\det K_{11} = \det S \det D \det S^T = d_0 \ldots d_{n-1}$$

(7.40)

In each step of the incremental solution procedure, the diagonal matrix $D$ is computed. Its diagonal coefficients contain significantly more information than the determinant of the incremental stiffness $K_{11}$, which is a single number. $D$ 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):

$$K^{s(s,\text{max})} = S^{s(s,\text{max})} D^{s(s,\text{max})} S^{s(s,\text{max}) \text{T}}$$  \hfill (7.41)

Let the diagonal coefficients of matrix $D$ be $d^{s(s,\text{max})}_k$

(b) Assume that the tangent stiffness matrix at the start of load step $s+1$ is decomposed in the form (7.42):

$$K^{l(s+1)} = S^{l(s+1)} D^{l(s+1)} S^{l(s+1) \text{T}}$$  \hfill (7.42)

Let the sign of one or several of the diagonal coefficients in $D^{s(s,\text{max})}$ and $D^{l(s)}$ 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):

$$K^{s(s+1,max)} = S^{s(s+1,max)} D^{s(s+1,max)} S^{s(s+1,max) \text{T}}$$  \hfill (7.43)

Let the sign of one or several of the diagonal coefficients in $D^{l(s+1)}$ and $D^{s(s+1,m)}$ 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 $D$ 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 $K'$ in (7.14) is decomposed into a product $SDS^T$. The diagonal coefficients of the left triangular matrix $S$ are 1; the diagonal coefficients of the diagonal matrix $D$ are denoted by $d_m$.

$K':
\begin{align*}
d_0 &= 2m^2 + w_2(2 + w_2) \\
d_i &= 3w_2^2 + 6w_2 + 2
\end{align*}$

The variation of the coefficients $d_m$ of the tangent stiffness matrix $K'$ with displacement $w_2$ is shown in figure 7.2. Curve “a” shows the variation of $d_0$ 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 $d_0 = 0$, but that the curve “b” for $m = 0.7071$ touches the axis. Curves “a” and “c” intersect at $d_0 = 0$, since the shape factor $m = 0.5774$ is the shape factor for the peak point.

![Figure 7.2](image-url)

**Figure 7.2** Variation of the diagonal coefficients of the diagonal matrix $D$ for a 2-bar truss with displacement $w_2$ and shape factor $m$

- a $d_0$ for all values of $m$
- b $d_0$ for $m = 0.7071$
- c $d_0$ for $m = 0.5774$
- d $d_0$ for $m = 0.4000$
- e $d_0$ for $m = 0.1000$
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):

\[
\Delta d_{1s} = \Delta \lambda \ d_{1t} + f_t \tag{7.44}
\]

\[
\Delta d_{2s} = \Delta \lambda \ d_{2t}
\]

\( \Delta \lambda \) load factor increment from nearly singular point A to singular point L

\( \Delta d_{1s} \) free system displacement increment from A to L

\( \Delta d_{2s} \) prescribed displacement increment from A to L

\( f_t \) displacement correction due to the error vector at point A

\( d_{1t} \) free displacement due to the load pattern

\( d_{2t} \) 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):

\[
\det(K_{s0} + \Delta \lambda K_{s1}) = 0 \tag{7.45}
\]

\[
K_{s0} \ x = \Delta \lambda (-K_{s1}) \ x \tag{7.46}
\]

\( K_{s0} \) residual stiffness matrix

\( K_{s1} \) modification stiffness matrix
Matrices $\mathbf{K}_{s0}$ and $\mathbf{K}_{s1}$ 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 $\mathbf{K}_s$ which form the submatrix $\mathbf{K}_{s1}$ 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 $\mathbf{K}_s$ 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 $\mathbf{K}_{s0}$ and $\mathbf{K}_{s1}$ 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 $\mathbf{K}_{s1}$. 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 $\delta$.

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 $\mathbf{K}_{s0}$ and the modification matrix $\mathbf{K}_{s1}$ 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):

$$
\Delta \mathbf{v}_e = \mathbf{R}_{s}^{\text{bar}} \mathbf{T}_e \Delta \mathbf{d}_e
$$

(7.47)

$\Delta \mathbf{v}_e$ increment of the element displacement vector in reference space
The derivatives of the displacement increment $\Delta v$ with respect to the axial coordinate $y_i$ are obtained by substitution of (6.46) and (7.47) into expressions (4.64):

$$\Delta v_{m1} = \frac{1}{a} g^T_m R^T_{esy} T_e (\Delta \lambda d_t + f)$$

$$\Delta v_{m1} = \Delta \lambda d_{m1} + d_{m2} \quad (7.48)$$

$$d_{m1} := \frac{1}{a} g^T_m R^T_{esy} T_e d_t$$

$$d_{m2} := \frac{1}{a} g^T_m R^T_{esy} T_e f \quad (7.50)$$

The derivatives of the node displacement of the truss at the singular point with respect to $y_i$ are denoted by $\bar{v}_{m1}$ and expressed with (7.48) as functions of the load factor increment $\Delta \lambda$:

$$\bar{v}_{m1} = v_{m1} + \Delta v_{m1} = d_{m0} + \Delta \lambda d_{m1} \quad (7.51)$$

$$d_{m0} = v_{m1} + d_{m2} \quad (7.52)$$

$$\bar{v}_{m1} \quad \text{derivative of } v_m \text{ with respect to } y_i \text{ at the singular point}$$

The components $\bar{K}_d$ and $\bar{K}_g$ of the tangential stiffness matrix $\bar{K}_s$ at the singular point are determined by substitution of (7.51) into (4.92) and (4.93):

$$\bar{K}_d = \frac{AE}{a} (c_1 G_{11} + c_2 G_{22} + c_3 G_{12}) \quad (7.53)$$

$$c_1 = (d_{10} + \Delta \lambda d_{11})(2 + d_{10} + \Delta \lambda d_{11})$$

$$c_2 = (d_{20} + \Delta \lambda d_{21})^2$$

$$c_3 = (d_{10} + \Delta \lambda d_{11})(1 + d_{10} + \Delta \lambda d_{11})$$

$$\bar{K}_g = \frac{AE}{a} c_4 (G_{11} + G_{22}) \quad (7.54)$$

$$c_4 = (d_{10} + \Delta \lambda d_{11}) + \frac{1}{2}(d_{10} + \Delta \lambda d_{11})^2 + \frac{1}{2}(d_{20} + \Delta \lambda d_{21})^2$$

The components $K_m$ and $K_n$ 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 $\Delta \lambda$:

$$\bar{K}_s = K_{e0} + \Delta \lambda K_{e1} + \Delta \lambda^2 K_{e2} \quad (7.55)$$

$$K_{e0} = K_c + K_{g0} + K_{y0}$$

$$K_{e1} = K_{g1} + K_{g1}$$

$$K_{e2} = K_{g2} + K_{g2}$$

The deviatoric stiffness matrix $\bar{K}_d$ contributes the following terms to $\bar{K}_e$:

$$K_{d0} = \frac{AE}{a} (b_1 G_{11} + b_2 G_{22} + b_3 G_{12}) \quad (7.56)$$

$$b_1 = d_{10}(2 + d_{10}) \quad b_2 = d_{20}^2 \quad b_3 = d_{20}(1 + d_{10})$$

211
The geometric stiffness matrix $\mathbf{K}_g$ contributes the following terms to $\mathbf{K}_e$:

$$
\mathbf{K}_{e0} = \frac{AE}{a} \mathbf{e}_0 \left( \mathbf{G}_{11} + \mathbf{G}_{22} \right) 
$$

$$
\mathbf{e}_0 = d_{10} + \frac{1}{2} d_{10}^2 + \frac{1}{2} d_{20}^2 
$$

$$
\mathbf{K}_{e1} = \frac{AE}{a} \mathbf{e}_1 \left( \mathbf{G}_{11} + \mathbf{G}_{22} \right) 
$$

$$
\mathbf{e}_1 = d_{11} + d_{10} d_{11} + d_{20} d_{21} 
$$

$$
\mathbf{K}_{e2} = \frac{AE}{a} \mathbf{e}_2 \left( \mathbf{G}_{11} + \mathbf{G}_{22} \right) 
$$

$$
\mathbf{e}_2 = \frac{1}{2} \left( d_{11}^2 + d_{21}^2 \right) 
$$

Expressions (4.89) and (7.53) to (7.61) lead to the components of $\mathbf{K}_e$:

\begin{align*}
\mathbf{K}_{e0} &= \begin{pmatrix}
  k_{01} & k_{03} & -k_{01} & -k_{03} \\
  k_{03} & k_{02} & -k_{03} & -k_{02} \\
 -k_{01} & -k_{03} & k_{01} & k_{03} \\
 -k_{03} & -k_{02} & k_{03} & k_{02}
\end{pmatrix} \\
&= \begin{pmatrix}
  1 + b_1 + e_0 & k_{02} & k_{03} \\
  k_{02} & b_2 + e_0 & k_{03} \\
  k_{03} & k_{03} & b_3
\end{pmatrix}
\end{align*}

\begin{align*}
\mathbf{K}_{e1} &= \begin{pmatrix}
  k_{11} & k_{13} & -k_{11} & -k_{13} \\
  k_{13} & k_{12} & -k_{13} & -k_{12} \\
 -k_{11} & -k_{13} & k_{11} & k_{13} \\
 -k_{13} & -k_{12} & k_{13} & k_{12}
\end{pmatrix} \\
&= \begin{pmatrix}
  b_4 + e_1 & k_{12} & k_{13} \\
  k_{12} & b_5 + e_1 & k_{13} \\
  k_{13} & k_{13} & b_6
\end{pmatrix}
\end{align*}

\begin{align*}
\mathbf{K}_{e2} &= \begin{pmatrix}
  k_{21} & k_{23} & -k_{21} & -k_{23} \\
  k_{23} & k_{22} & -k_{23} & -k_{22} \\
 -k_{21} & -k_{23} & k_{21} & k_{23} \\
 -k_{23} & -k_{22} & k_{23} & k_{22}
\end{pmatrix} \\
&= \begin{pmatrix}
  b_7 + e_2 & k_{22} & k_{23} \\
  k_{22} & b_8 + e_2 & k_{23} \\
  k_{23} & k_{23} & b_9
\end{pmatrix}
\end{align*}
The component system matrices $K_i$ for the incremental system stiffness matrix are assembled from the element matrices, as described in chapter 6:

$$K_i = \sum_e T_e^T R_{esy} K_{ei} R_{esy}^T T_e \quad i \in \{0,1,2\} \quad (7.65)$$

The residual and modification stiffness matrices in (7.46) are computed with the approximate load factor increment $\Delta \lambda$:

$$K_{s0} = K_0 \quad K_{s1} = K_1 + \Delta \lambda K_2 \quad (7.66)$$

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:

$$\tilde{K}^{(0)} = K_0 + \Delta \lambda^{(0)} K_1^{(0)} \quad \text{with} \quad K_1^{(0)} = K_1 \quad (7.67)$$

The eigenvalue problem (7.46) is solved for the matrix pair $(K_0, K_1^{(0)})$ to yield $\Delta \lambda^{(0)}$.

**Cycle m**: The nonlinear term $\Delta \lambda^{(m)} \Delta \lambda^{(m)}$ is approximated by $\Delta \lambda^{(m)} \Delta \lambda^{(m-1)}$:

$$\tilde{K}^{(m)} = K_0 + \Delta \lambda^{(m)} K_1^{(m)} \quad \text{with} \quad K_1^{(m)} = K_1 + \Delta \lambda^{(m-1)} K_2 \quad (7.68)$$

The eigenvalue problem (7.46) is solved for the matrix pair $(K_0, K_1^{(m)})$ 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:

$$\Delta d^{(A)} = \Delta \lambda^{(A)} d_i^{(A)}$$

$$d^{(L)} = d^{(A)} + \Delta d^{(A)}$$

$d^{(L)}$ system displacement vector at the singular point $L$

$d^{(A)}$ system displacement vector at the nearly singular point $A$

$\Delta d^{(A)}$ 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 $K_{s0}$ and modification matrix $K_{s1}$ 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):

$$\Delta w_e = R_{esz}^T T_e \Delta d_e = R_{esz}^T T_e (\Delta \lambda d_i + f) \quad (7.70)$$

$\Delta w_e$ increment of the element displacement vector in instant space
The coefficients with indices 2 and 5 of the increment \( \Delta \vec{w}_e \) 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 \( \gamma_i \) 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 \( \Delta \theta = \Delta \vec{w}_{e,2,1} \) of the rotation of the chord.

\[
\Delta w_{e,2,1} = d_{21} \Delta \lambda + d_{22} \\
d_{21} = \frac{1}{a} g^1 R^1_{esz} T_e d_1 \\
d_{22} = \frac{1}{a} g^2 R^2_{esz} T_e f
\]

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 \( \Delta \lambda : \)

\[
\vec{w}_{c,1} = w_{c,1} + \Delta w_{c,1} = d_{11} \Delta \lambda + d_{12} \\
\vec{w}_{c,2,1} = w_{c,2,1} + \Delta w_{c,2,1} = d_{21} \Delta \lambda + d_{22} \\
\gamma_A = \gamma_A + \Delta \gamma_A = d_{31} \Delta \lambda + d_{32} \\
\gamma_B = \gamma_B + \Delta \gamma_B = d_{41} \Delta \lambda + d_{42} \\
d_{10} = w_{c,1} + d_{12} \\
d_{20} = w_{c,2,1} + d_{22} \\
d_{30} = \gamma_A + d_{32} \\
d_{40} = \gamma_B + d_{42}
\]

The tangent stiffness matrix \( \bar{K}_e \) at the singular point follows from (5.203):

\[
\bar{K}_e = \bar{K}_{ac} + \bar{K}_{ab} + \bar{K}_{bc} + \bar{K}_{bg}
\]

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 \( K_{am} \) and \( K_{an} \) are not contained in the tangent stiffness matrix \( \bar{K}_e \) :

\[
\bar{K}_{ac} = \frac{AE}{a} c_0 G_{11}
\]

\[
c_0 = (1 - d_{10} - d_{11} \Delta \lambda)^2
\]
The tangent element stiffness matrix $\tilde{K}_e$ at the singular point is a polynomial in the load factor increment $\Delta \lambda$:

$$\tilde{K}_e = K_{e0} + \Delta \lambda K_{e1} + \Delta \lambda^2 K_{e2} + \Delta \lambda^3 K_{e3}$$  \hspace{1cm} (7.82)

The term $K_{bc}$ is given by (5.210). The contribution of component $\tilde{R}_{ac}$ to matrix $\tilde{K}_e$ is:

$$K_{ac0} = \frac{AE}{a} b_1 \mathbf{G}_{11} \hspace{1cm} b_1 = (1-d_{10})^2$$  \hspace{1cm} (7.83)

$$K_{ac1} = \frac{AE}{a} b_2 \mathbf{G}_{11} \hspace{1cm} b_2 = -2d_{11}(1-d_{10})$$

$$K_{ac2} = \frac{AE}{a} b_3 \mathbf{G}_{11} \hspace{1cm} b_3 = d_{11}^2$$

The contribution of component $\tilde{R}_{ag}$ to matrix $\tilde{K}_e$ is:

$$K_{ag0} = \frac{AE}{a} (b_4 \mathbf{G}_{11} + b_5 \mathbf{G}_{22})$$  \hspace{1cm} (7.84)

$$b_4 = -d_{10} + \frac{d_{10}^2}{2} + \frac{d_{20}^2}{2}$$

$$b_5 = -(1-d_{10})b_4$$

$$K_{ag1} = \frac{AE}{a} (b_6 \mathbf{G}_{11} + b_7 \mathbf{G}_{22})$$  \hspace{1cm} (7.85)

$$b_6 = -d_{11}(1-d_{10}) + d_{20}d_{21}$$

$$b_7 = d_{11}b_4 - (1-d_{10})b_6$$
\[ K_{g2} = \frac{\text{AE}}{a} (b_8 G_{11} + b_9 G_{22}) \] (7.86)

\[ b_8 = \frac{1}{2} d_{11}^2 + \frac{1}{2} d_{22}^2 \]

\[ b_9 = d_{11} b_8 - (1 - d_{10}) b_8 \]

\[ K_{g3} = \frac{\text{AE}}{a} (b_{10} G_{22}) \] (7.87)

The contribution of component \( K_{bg} \) to matrix \( K_e \) is:

\[ K_{bg0} = \frac{\text{JE}}{a^3} b_{11} G_{22} \quad b_{11} = -4 \left( \frac{\hat{a}}{a} \right)^2 (d_{30}^2 + d_{30} d_{40} + d_{40}^2) \] (7.88)

\[ K_{bg1} = \frac{\text{JE}}{a^3} b_{12} G_{22} \quad b_{12} = -4 \left( \frac{\hat{a}}{a} \right)^2 (2d_{30} d_{31} + d_{30} d_{41} + d_{31} d_{40} + 2d_{40} d_{41}) \]

\[ K_{bg2} = \frac{\text{JE}}{a^3} b_{13} G_{22} \quad b_{13} = -4 \left( \frac{\hat{a}}{a} \right)^2 (d_{31}^2 + d_{31} d_{41} + d_{41}^2) \]

Expressions (7.83) to (7.88) lead to the components of \( K_e \) in (7.82):

\[
\begin{array}{cccccccc}
  & a_0 & 0 & 0 & -a_0 & 0 & 0 & \\
 0 & a_1 & 0 & 0 & -a_1 & 0 & \\
 0 & 0 & 0 & 0 & 0 & 0 & \\
 -a_0 & 0 & 0 & a_0 & 0 & 0 & \\
 0 & -a_1 & 0 & 0 & a_1 & 0 & \\
 0 & 0 & 0 & 0 & 0 & 0 & \\
\end{array}
\]

\[ K_{s0} = K_{bc} + a_0 \]

\[ a_0 = \frac{\text{AE}}{a} (b_8 + b_4) \]

\[ a_1 = \frac{\text{AE}}{a} b_8 + \frac{\text{JE}}{a^3} b_{11} \]

\[
\begin{array}{cccccccc}
  & a_2 & 0 & 0 & -a_2 & 0 & 0 & \\
 0 & a_3 & 0 & 0 & -a_3 & 0 & \\
 0 & 0 & 0 & 0 & 0 & 0 & \\
 -a_2 & 0 & 0 & a_2 & 0 & 0 & \\
 0 & -a_3 & 0 & 0 & a_3 & 0 & \\
 0 & 0 & 0 & 0 & 0 & 0 & \\
\end{array}
\]

\[ K_{s1} = \]

\[ a_2 = \frac{\text{AE}}{a} (b_2 + b_6) \]

\[ a_3 = \frac{\text{AE}}{a} b_7 + \frac{\text{JE}}{a^3} b_{12} \]
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 $K_e$ of the element to the incremental stiffness matrix $K_s$ of the frame at the singular point is given by (5.217):

$$\begin{bmatrix}
a_4 & 0 & 0 & -a_4 & 0 & 0 \\
0 & a_6 & 0 & 0 & -a_6 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
a_4 & 0 & 0 & a_4 & 0 & 0 \\
0 & -a_6 & 0 & 0 & a_6 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix} \quad (7.91)$$

$$a_4 = \frac{AE}{a} (b_4 + b_6) \quad a_5 = \frac{AE}{a} b_5 + \frac{JE}{a^3} b_{13}$$

$$\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 \\
0 & a_6 & 0 & 0 & -a_6 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & -a_6 & 0 & 0 & a_6 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix} \quad (7.92)$$

$$a_6 = \frac{AE}{a} b_{10}$$

The element rotation matrix $R_{esz}$ for the transformation between instant and support space is derived from the rotation matrices $R_{nsz}$ at nodes A and B of the member:

$$\begin{bmatrix}
R_{nsz}^A & 0 \\
0 & R_{nsz}^B
\end{bmatrix} \quad (7.94)$$

$$\begin{bmatrix}
\bar{R}_{esz} \\
\bar{R}_{nsz}
\end{bmatrix} = \begin{bmatrix}
R_{nsz}^T & R_{nxy} & \bar{R}_{nyz}
\end{bmatrix}$$

$$\begin{bmatrix}
R_{nsz} & R_{nxy} & R_{nyz}
\end{bmatrix} \quad (7.95)$$

The transformation between global and reference space is given by (5.18):

$$u_n = R_{nxy} v_n \quad (7.96)$$
If the angle from the global axis $x_i$ to the support axis $s_i$ in figure 5.2 is denoted by $\phi$, the transformation (5.21) between global to support space is given by:

\[
\mathbf{u}_n = \mathbf{R}_{nxs} \mathbf{d}_n
\]  

(7.97)

\[
\begin{bmatrix}
\mathbf{u}_1 \\
\mathbf{u}_2 \\
\beta
\end{bmatrix} =
\begin{bmatrix}
\cos \phi & -\sin \phi & 0 \\
\sin \phi & \cos \phi & 0 \\
0 & 0 & 1
\end{bmatrix}
\]

The coordinates of the transformation matrix $\mathbf{R}_{nyz}$ between reference and instant space at the singular point are given in terms of the angle of rotation $\bar{\theta}$ by (5.50):

\[
\mathbf{v}_n = \mathbf{R}_{nyz} \mathbf{w}_n
\]  

(7.98)

\[
\begin{bmatrix}
\mathbf{v}_1 \\
\mathbf{v}_2 \\
\beta
\end{bmatrix} =
\begin{bmatrix}
\cos \bar{\theta} & -\sin \bar{\theta} & 0 \\
\sin \bar{\theta} & \cos \bar{\theta} & 0 \\
0 & 0 & 1
\end{bmatrix}
\]

$\bar{\theta}$ angle from axis $y_i$ to axis $z_i$ at the singular point

$\mathbf{w}_i$ 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 $\theta$ is approximated:

\[
\mathbf{R}_{nyz}(\theta + \Delta \theta) \approx \mathbf{R}_{nyz}(\theta) + \Delta \theta \frac{d\mathbf{R}_{nyz}(\theta)}{d\theta} + \frac{1}{2} \frac{d^2\mathbf{R}_{nyz}(\theta)}{d\theta^2} \Delta \theta^2
\]  

(7.99)

$\mathbf{R}_{nyz}(\theta + \Delta \theta) = (1-\frac{1}{2}\Delta \theta^2)\mathbf{R}_{n1} + \Delta \theta \mathbf{R}_{n2}$

$\mathbf{R}_{n1} = \begin{bmatrix}
\cos \theta & -\sin \theta & 0 \\
\sin \theta & \cos \theta & 0 \\
0 & 0 & 1
\end{bmatrix}$

$\mathbf{R}_{n2} = \begin{bmatrix}
-\sin \theta & -\cos \theta & 0 \\
\cos \theta & -\sin \theta & 0 \\
0 & 0 & 1
\end{bmatrix}$

$\theta$ angle from axis $y_i$ to axis $z_i$ 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):

\[
\mathbf{R}_{esz} = (1-\frac{1}{2}\Delta \theta^2)\mathbf{R}_{e1} + \Delta \theta \mathbf{R}_{e2}
\]  

(7.100)

$\mathbf{R}_{e1} = \begin{bmatrix}
\mathbf{R}_{nxs} & \mathbf{R}_{nys} \mathbf{R}_{nyz} \\
0 & \mathbf{R}_{nxs} & \mathbf{R}_{nys} \mathbf{R}_{nyz}
\end{bmatrix}$

$\mathbf{R}_{e2} = \begin{bmatrix}
\mathbf{R}_{nxs}^{(A)^T} \mathbf{R}_{nys} \mathbf{R}_{nyz} & 0 \\
0 & \mathbf{R}_{nxs}^{(A)^T} \mathbf{R}_{nys} & \mathbf{R}_{nys} \mathbf{R}_{nyz} \mathbf{R}_{n2}
\end{bmatrix}$

The change $\Delta \theta$ in the angle of chord rotation $\theta$ from the nearly singular point to the singular point is given by (5.78). The derivative $\Delta \mathbf{w}_{c2,1}$ is substituted from (7.72):

\[
\Delta \theta = \frac{\Delta \mathbf{w}_{c2,1} - \Delta \mathbf{w}_{c2,1}}{a} = \Delta \mathbf{w}_{c2,1}
\]  

(7.101)

\[
\Delta \theta = \Delta \mathbf{d}_{21} + \Delta \mathbf{d}_{22}
\]  

(7.102)
The increment $\Delta \theta$ is substituted from (7.102) into (7.100):

$$
\widetilde{R}_{esz} = \left(1 - \frac{1}{2} \left(2 \Delta \lambda_d + d_{zz}ight)^2\right)R_{e1} + \left(2 \Delta \lambda_d + d_{zz}\right)R_{e2}
$$

$$
\widetilde{R}_{esz} = A_0 + \Delta \lambda A_1 + \Delta \lambda^2 A_2
$$

(7.103)

$$
A_0 = \left(1 - \frac{1}{2} d_{zz}^2\right)R_{e1} + d_{zz} R_{e2}
$$

$$
A_1 = -d_{zz}^2 d_{xx} R_{e1} + d_{zz} R_{e2}
$$

$$
A_2 = -\frac{1}{2} d_{zz}^2 R_{e1}
$$

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:

$$
\widetilde{R}_{ez}^{(0)} = K_{e0} + \Delta \lambda^{(0)} K_{e1}^{(0)} \quad \text{with} \quad K_{e1}^{(0)} = K_{e1}
$$

$$
\widetilde{R}_{esz}^{(0)} = A_0 + \Delta \lambda^{(0)} A_1^{(0)} \quad \text{with} \quad A_1^{(0)} = A_1
$$

(7.104)

(7.105)

$$
\widetilde{R}_{ez}^{(0)} \left(\widetilde{R}_{esz}^{(0)}\right)^T = (A_0 + \Delta \lambda^{(0)} A_1^{(0)}) K_{e0} + \Delta \lambda^{(0)} K_{e1}^{(0)} (A_0 + \Delta \lambda^{(0)} A_1^{(0)})^T
$$

(7.106)

Only the terms linear in $\Delta \lambda$ are retained in the product (7.106):

$$
\widetilde{R}_{ez}^{(0)} \left(\widetilde{R}_{esz}^{(0)}\right)^T = H_0 + \Delta \lambda^{(0)} H_1^{(0)}
$$

(7.107)

$$
H_0^{(0)} = A_0 K_{e0} A_0^T + A_0 K_{e0} A_1^T + A_0 K_{e1} A_0^T
$$

$$
H_1^{(0)} = A_1 K_{e0} A_0^T + A_0 K_{e0} A_1^T + A_0 K_{e1} A_0^T
$$

The element matrices are assembled as described in chapter 6 to determine the system matrix $\mathbf{K}$ 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.

$$
\widetilde{R}_e^{(m)} = K_{e0} + \Delta \lambda^{(m)} K_{e1} \quad \text{with} \quad K_{e1}^{(m)} = K_{e1} + \Delta \lambda^{(m-1)} (K_{e2} + \Delta \lambda^{(m-1)} K_{e3})
$$

(7.108)

$$
\widetilde{R}_{esz}^{(m)} = A_0 + \Delta \lambda^{(m)} A_1 \quad \text{with} \quad A_1^{(m)} = A_1 + \Delta \lambda^{(m-1)} A_2
$$

(7.109)

$$
\widetilde{R}_{ez}^{(m)} \left(\widetilde{R}_{esz}^{(m)}\right)^T = (A_0 + \Delta \lambda^{(m)} A_1^{(m)}) K_{e0} + \Delta \lambda^{(m)} K_{e1}^{(m)} (A_0 + \Delta \lambda^{(m)} A_1^{(m)})^T
$$

(7.110)

Terms of the product (7.110) that are nonlinear in $\Delta \lambda^{(m)}$ are approximated in the same manner:
\begin{align*}
\mathbf{R}^{(m)}_e \mathbf{R}^{(m)}_e (\bar{\mathbf{R}}^{(m)}_e)^T &= \mathbf{H}_0 + \Delta \lambda^{(m)} \mathbf{H}_1^{(m)} \\
\mathbf{H}_1^{(m)} &= \mathbf{H}_1^{(m)} + \Delta \lambda^{(m-1)} \mathbf{H}_1^{(m)} + (\Delta \lambda^{(m-1)})^2 \mathbf{H}_2^{(m)} \\
\mathbf{H}_1^{(m)} &= \mathbf{A}_1^{(m)} K_{e0} A_0^T + \mathbf{A}_0 K_{e0} A_1^{(m)T} + \mathbf{A}_0 K_{e1} A_0^T \\
\mathbf{H}_1^{(m)} &= \mathbf{A}_0 K_{e1} A_1^{(m)T} + \mathbf{A}_1^{(m)} K_{e0} A_1^{(m)T} + \mathbf{A}_1^{(m)} K_{e1} A_0^T \\
\mathbf{H}_1^{(m)} &= \mathbf{A}_1^{(m)} K_{e1} A_1^{(m)T} 
\end{align*}

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:

$$ g(w, \lambda) = 0 $$

(7.113)

$g$ vector of smooth functions

$w$ displacement vector of the structure

$\lambda$ load factor

Assume that $w$ and $\lambda$ are smooth functions of a parameter $z$:

$$ g(z) = g((w(z), \lambda(z))) = 0 $$

(7.114)

The derivative of vector $g$ with respect to the parameter $z$ is determined by differen- tiation of equation (7.114):

$$ t := \frac{dg}{dz} = \sum_{i=1}^{m} \frac{\partial g}{\partial w_i} \frac{dw_i}{dz} + \frac{\partial g}{\partial \lambda} \frac{d\lambda}{dz} = 0 $$

(7.115)

$m$ dimension of the displacement vector $w$
**Singular point:** The derivatives of the function $g$ with respect to the displacement coordinates $w_i$ are collected in a matrix $K$. The coefficients of $K$ are functions of the displacement $w$. A point $(w, \lambda)$ that satisfies equation (7.114) is called regular if the matrix $K$ is regular for the displacement at that point. The point is called singular if the matrix $K$ is singular for the displacement $w$ at the point. Consider the special eigenvalue problem for matrix $K$:

\[
Kx = \mu x \quad \text{with} \quad x^T x = 1
\]

\[
y^T K = \mu y^T \quad \text{with} \quad y^T y = 1
\]

\[
K = \begin{bmatrix}
\frac{\partial g}{\partial w_1} & \cdots & \frac{\partial g}{\partial w_m}
\end{bmatrix}
\]

$\mu$ eigenvalue of matrix $K$

$x$ right eigenvector of matrix $K$

$y$ left eigenvector of matrix $K$

Let the value of matrix $K$ at a singular point be denoted by $K_0$. The singular point is called simple if exactly one of the eigenvalues of $K_0$ 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 $K_0$ be denoted by $x_0$ and $y_0$. Equation (7.117) has the following form for the eigenstate $(0, y_0)$:

\[
y^T_0 K_0 = 0^T
\]

\[
y^T_0 \frac{\partial g_0}{\partial w_i} = 0
\]

\[
K_0 \quad \text{value of } K \text{ at the singular point}
\]

\[
\frac{\partial g_0}{\partial w_i} \quad \text{partial derivative of } g \text{ with respect to displacement } w_i \text{ at the singular point}
\]

Let the value of vectors $w$ and $t$ at a singular point be denoted by $w_0$ and $t_0$. The scalar product of the vectors $y_0$ and $t_0$ is null since $t_0$ is null in (7.115). Substitute the expression for vector $t_0$ from (7.115) into the scalar product:

\[
y^T_0 t_0 = \sum_{i=1}^{m} \left( y^T_0 \frac{\partial g_0}{\partial w_i} \right) dw_i + \left( y^T_0 \frac{\partial g_0}{\partial \lambda} \right) d\lambda = 0
\]

\[
\frac{\partial g_0}{\partial \lambda} \quad \text{partial derivative of } g \text{ with respect to load factor } \lambda \text{ at the singular point}
\]

Substitution of the scalar product from equation (7.119) into equation (7.120) yields the following expression:

\[
\left( y^T_0 \frac{\partial g_0}{\partial \lambda} \right) \frac{d\lambda}{dz} = 0
\]
**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.

\[
\mathbf{y}_0^T \frac{\partial \mathbf{g}_0}{\partial \lambda} \neq 0 \quad \Rightarrow \quad (\mathbf{w}_0, \lambda_0) \text{ is a turning point} \quad (7.122)
\]

\[
\mathbf{y}_0^T \frac{\partial \mathbf{g}_0}{\partial \lambda} = 0 \quad \Rightarrow \quad (\mathbf{w}_0, \lambda_0) \text{ is a bifurcation point} \quad (7.123)
\]

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:

\[
(\mathbf{w}_0, \lambda_0) \text{ is a turning point} \quad \Rightarrow \quad \frac{d\lambda_0}{dz} = 0 \quad (7.124)
\]

### 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:

\[
\frac{d \mathbf{g}_0}{dz} = \sum_{j=1}^{m} \frac{\partial \mathbf{g}_0}{\partial \mathbf{w}_j} \frac{d \mathbf{w}_j}{dz} = \mathbf{0} \quad (7.125)
\]

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 \( \mu_0 = 0 \)

\( \theta \) 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:

\[
\lambda(z) = \lambda_0 + \frac{d\lambda_0}{dz} dz + \frac{1}{2!} \frac{d^2\lambda_0}{dz^2} dz^2 + \frac{1}{3!} \frac{d^3\lambda_0}{dz^3} dz^3 + \ldots \quad (7.127)
\]

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):

\[
\frac{d^2 \mathbf{g}}{dz^2} = \sum_{i=1}^{m} \sum_{k=1}^{m} \frac{\partial^2 \mathbf{g}}{\partial \mathbf{w}_i \partial \mathbf{w}_k} \frac{d \mathbf{w}_i}{dz} \frac{d \mathbf{w}_k}{dz} + 2 \sum_{i=1}^{m} \frac{\partial \mathbf{g}}{\partial \mathbf{w}_i} \frac{d \mathbf{w}_i}{dz} \frac{d \lambda}{dz} + \sum_{i=1}^{m} \frac{\partial \mathbf{g}}{\partial \mathbf{w}_i} \frac{d^2 \mathbf{w}_i}{dz^2} + \frac{\partial \mathbf{g}}{\partial \lambda} \frac{d^2 \lambda}{dz^2} + \frac{\partial \mathbf{g}}{\partial \mathbf{w}_i} \frac{d^2 \mathbf{w}_i}{dz^2} + \frac{\partial \mathbf{g}}{\partial \lambda} \frac{d^2 \lambda}{dz^2} = \mathbf{0} \quad (7.128)
\]

Property (7.124) of the turning points is substituted into equation (7.128):

\[
\sum_{i=1}^{m} \sum_{k=1}^{m} \frac{\partial^2 \mathbf{g}}{\partial \mathbf{w}_i \partial \mathbf{w}_k} \frac{d \mathbf{w}_i}{dz} \frac{d \mathbf{w}_k}{dz} + \sum_{i=1}^{m} \frac{\partial \mathbf{g}}{\partial \mathbf{w}_i} \frac{d^2 \mathbf{w}_i}{dz^2} + \frac{\partial \mathbf{g}}{\partial \lambda} \frac{d^2 \lambda}{dz^2} = \mathbf{0} \quad (7.129)
\]
Equation (7.129) is multiplied from the left with eigenvector $y_0^T$. The scalar product $y_0^T g_0$ is denoted by $a_0$.

$$\sum_{i=1}^{m} \sum_{k=1}^{n} \frac{\partial^2 a_0}{\partial w_i \partial w_k} \frac{dw_i}{dz} \frac{dw_k}{dz} + \sum_{i=1}^{m} \frac{\partial^2 a_0}{\partial \lambda_i} \frac{d^2 \lambda}{dz^2} + \frac{d a_0}{d \lambda} \frac{d^2 \lambda}{dz^2} = 0$$  \hspace{1cm} (7.130)

$$a_0 = y_0^T g_0$$  \hspace{1cm} (7.131)

The derivatives of product $a$ with respect to the displacement coordinates $w_i$ are determined with expression (7.119):

$$\frac{\partial a_0}{\partial w_i} = \frac{\partial (y_0^T g_0)}{\partial w_i} = y_0^T \frac{\partial g_0}{\partial w_i} = 0$$  \hspace{1cm} (7.132)

Substitution of the derivatives (7.132) into equation (7.130) yields:

$$\sum_{i=1}^{m} \sum_{k=1}^{n} \frac{\partial^2 a_0}{\partial w_i \partial w_k} \frac{dw_i}{dz} \frac{dw_k}{dz} + \frac{\partial^2 a_0}{\partial \lambda_i} \frac{d^2 \lambda}{dz^2} = 0$$  \hspace{1cm} (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 $c_0$:

$$\frac{d^2 \lambda_0}{dz^2} = -\theta^2 c_0 \left( \frac{\partial a_0}{\partial \lambda} \right)^{-1}$$  \hspace{1cm} (7.134)

$$c_0 = \sum_{i=1}^{m} \sum_{k=1}^{n} \frac{\partial^2 a_0}{\partial w_i \partial w_k} x_{i0} x_{k0}$$

$x_{i0}$ coordinate in row $i$ of eigenvector $x_0$

The Taylor series for the load factor follows by substitution of (7.134) into (7.127):

$$\lambda = \lambda_0 - \frac{1}{2} \theta^2 c_0 \left( \frac{\partial a_0}{\partial \lambda} \right)^{-1} dz^2 + \frac{1}{6} \frac{d^3 \lambda_0}{dz^3} dz^3$$  \hspace{1cm} (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 $x_0$:

$$\frac{\partial a_0}{\partial \lambda} \neq 0 \land \frac{d \lambda_0}{dz} = 0 \land \frac{d^2 \lambda_0}{dz^2} < 0 \Rightarrow (w_0, \lambda_0) \text{ is a maximal point}$$

$$\frac{dw_0}{dz} = \theta x_0$$  \hspace{1cm} (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 $x_0$:

$$
\frac{\partial^2 a_0}{\partial \lambda^2} \neq 0 \land \frac{d\lambda_0}{dz} = 0 \land \frac{d^2\lambda_0}{dz^2} > 0 \Rightarrow (w_0, \lambda_0) \text{ is a minimal point}
$$

$$
\frac{dw_0}{dz} = 0 x_0
$$

(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 $x_0$:

$$
\frac{\partial^3 a_0}{\partial \lambda^3} \neq 0 \land \frac{d\lambda_0}{dz} = 0 \land \frac{d^2\lambda_0}{dz^2} = 0 \land \frac{d^3\lambda_0}{dz^3} \neq 0 \Rightarrow (w_0, \lambda_0) \text{ is a saddle point}
$$

$$
\frac{dw_0}{dz} = 0 x_0
$$

(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](image)

**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 \( \lambda \) as unknowns. In order to solve these equations, the derivative of function \( g_i \) with respect to the load factor \( \lambda \) is decomposed into components parallel and normal to eigenvector \( y_0 \):

\[
\frac{\partial g_i}{\partial \lambda} = c y_0 + n_0 \quad \text{with} \quad y_0^T n_0 = 0
\]  

(7.139)

\( n_0 \) vector normal to eigenvector \( y_0 \)

Equation (7.139) is multiplied from the left with \( y_0^T \). The resulting value of coefficient \( c \) is substituted into (7.139) to obtain the normal \( n_0 \):

\[
c = y_0^T \frac{\partial g_i}{\partial \lambda}
\]  

(7.140)

\[
n_0 = (I - y_0 y_0^T) \frac{\partial g_i}{\partial \lambda}
\]  

(7.141)

The relationship between the normal \( n_0 \) and the derivatives of the displacement coordinates \( w_i \) is obtained by multiplying equation (7.115) from the left with the matrix \( I - y_0 y_0^T \):

\[
\sum_{i=1}^{m} (I - y_0 y_0^T) \frac{\partial g_i}{\partial w_i} \frac{dw_i}{dz} = -(I - y_0 y_0^T) \frac{\partial g_i}{\partial \lambda} \frac{d\lambda}{dz} = -n_0 \frac{d\lambda}{dz}
\]  

(7.142)

Expression (7.119) is substituted into equation (7.142):

\[
\sum_{i=1}^{m} \frac{\partial g_i}{\partial w_i} \frac{dw_i}{dz} = -\frac{\partial g_i}{\partial \lambda} \frac{d\lambda}{dz} = -n_0 \frac{d\lambda}{dz}
\]  

(7.143)

A linear combination of an unknown vector \( z_0 \) and the eigenvector \( x_0 \) is chosen as ansatz for the derivative of the displacement vector \( w \) at the bifurcation point:

\[
\frac{dw_0}{dz} = \theta_1 z_0 + \theta_2 x_0
\]  

(7.144)

\[
\frac{dw_i}{dz} = \theta_1 z_{io} + \theta_2 x_i
\]  

(7.145)

\( \theta_i \) coefficients \( \in \mathbb{R} \)

\( z_{io} \) coordinate in row \( i \) of vector \( z_0 \)

The ansatz (7.144) is substituted into equation (7.143):

\[
\sum_{i=1}^{m} \frac{\partial g_i}{\partial w_i} (\theta_1 z_{io} + \theta_2 x_i) = -\frac{\partial g_i}{\partial \lambda} \frac{d\lambda}{dz}
\]  

(7.146)

Due to equation (7.119), the sum of the terms containing \( x_{io} \) on the left-hand side of equation (7.146) is identically null. The coefficient \( \theta_2 \) can therefore be chosen freely. Choose coefficient \( \theta_1 \) equal to the derivative of the load factor. Equation (7.146) is satisfied if vector \( z_0 \) satisfies equation (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):

$$
\frac{d\mathbf{w}_0}{dz} = \frac{d\lambda}{dz} \mathbf{z}_0 + \theta_2 \mathbf{x}_0
$$

(7.148)

The derivatives of the displacement coordinate $w_{i0}$ at the bifurcation point are substituted from equation (7.148) into equation (7.150):

$$
\sum_{i=1}^{m} \sum_{k=1}^{m} \frac{\partial^2 a_0}{\partial w_i \partial 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 w_i \partial \lambda} z_{i0} x_{k0} \frac{d\lambda}{dz} + \theta_2^2 \sum_{i=1}^{m} \frac{\partial^2 a_0}{\partial \lambda^2} z_{i0} x_{i0} x_{k0} +

2 \theta_2 \sum_{i=1}^{m} \frac{\partial^2 a_0}{\partial w_i \partial \lambda} x_{i0} \frac{d\lambda}{dz} + \theta_2 \sum_{i=1}^{m} \frac{\partial^2 a_0}{\partial w_i \partial \lambda} x_{i0} \frac{d\lambda}{dz} + \frac{\partial^2 a_0}{\partial \lambda^2} \left( \frac{d\lambda}{dz} \right)^2 = 0
$$

(7.151)

The roots of the quadratic equation (7.152) depend on the sign of its discriminant, which is proportional to the parameter $d$:

$$
d := c_1^2 - c_0 c_2
$$

(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>
<thead>
<tr>
<th>Case</th>
<th>$c_0$</th>
<th>$c_1$</th>
<th>$c_2$</th>
<th>$d$</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>B1</td>
<td>$\neq 0$</td>
<td>$\neq 0$</td>
<td>$\neq 0$</td>
<td>$&gt; 0$</td>
<td>asymmetric</td>
</tr>
<tr>
<td>B2</td>
<td>$\neq 0$</td>
<td>$\neq 0$</td>
<td>$0$</td>
<td>$&gt; 0$</td>
<td>asymmetric</td>
</tr>
<tr>
<td>B3</td>
<td>$0$</td>
<td>$\neq 0$</td>
<td>$\neq 0$</td>
<td>$&gt; 0$</td>
<td>symmetric</td>
</tr>
<tr>
<td>B4</td>
<td>$0$</td>
<td>$\neq 0$</td>
<td>$0$</td>
<td>$&gt; 0$</td>
<td>symmetric</td>
</tr>
<tr>
<td>B5</td>
<td>$\neq 0$</td>
<td>$\neq 0$</td>
<td>$\neq 0$</td>
<td>$0$</td>
<td>cusp</td>
</tr>
</tbody>
</table>

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 $g = 0$. This solution is called an isola formation point.
Case B1: Asymmetric bifurcation point: \( c_0 \neq 0 \land c_1 \neq 0 \land c_2 \neq 0 \land d > 0 \)

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:

\[
\frac{dw_{i0}}{dz} = \frac{\theta_2}{c_2}
\left(\frac{z_i}{c_2}(-c_1 + \sqrt{d}) + x_{i0}\right) \quad i \in \{1,...,m\} \quad (7.159)
\]

The singular point \((w_0,\lambda_0)\) 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 \( \theta_2 \) 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.

\[
\frac{d\lambda_1}{dz}(2c_1 \frac{d\lambda}{dz} + \theta_2 c_0) = 0 \quad (7.161)
\]

primary branch: \( \frac{d\lambda_1}{dz} \in \mathbb{R} \)

\[
\frac{dw_{i0}}{dz} = \frac{\theta_2}{c_2} \left(\frac{z_i}{c_2}(-c_1 + \sqrt{d}) + x_{i0}\right) \quad (7.159)
\]

secondary branches:

\[
\frac{d\lambda_1}{dz} = -\frac{c_0}{2c_1} \theta_2 \quad \frac{dw_{i0}}{dz} = \frac{\theta_2}{2c_1} \left(\frac{z_i}{c_2}(-c_1 + \sqrt{d}) + x_{i0}\right) \quad (7.163)
\]

Case B3: Symmetric bifurcation point: \( c_0 = 0 \land c_1 \neq 0 \land c_2 \neq 0 \land d > 0 \)

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:

\[
\frac{d\lambda}{dz} \left(\frac{d\lambda}{dz} + 2\theta_2 c_1\right) = 0 \quad (7.164)
\]

\[
\frac{d\lambda_1}{dz} = -2\theta_2 \frac{c_1}{c_2} \quad (7.165)
\]

\[
\frac{d\lambda_2}{dz} = 0 \quad (7.166)
\]

Equation (7.144) shows that the two roots correspond to the following tangents to the branches of the load path at the bifurcation point:
primary branch: \( \frac{dw_i}{dz} = \theta_2 \left( -2zio \frac{c_1}{c_2} + x_{i0} \right) \) \( i \in \{1, \ldots, m\} \) (7.167)

secondary branches: \( \frac{dw_i}{dz} = \theta_2 x_{i0} \) (7.168)

The singular point \( (w_{i0}, \lambda_{i0}) \) 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:

\[
\frac{d\lambda_{i2}}{dz} = \frac{da_{i0}}{dw_i} = \frac{da_{i0}}{d\lambda} = 0 \tag{7.169}
\]

\[
c_0 = \sum_{i=1}^{m} \sum_{k=1}^{m} \frac{\partial^2 a_{i0}}{\partial w_i \partial w_k} x_{i0} x_{k0} = 0
\]

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:

\[
\frac{d^3 a_{i0}}{dz^3} = \sum_{i=1}^{m} \sum_{k=1}^{m} \frac{\partial^3 a_{i0}}{\partial w_i \partial w_k \partial w_n} \frac{dw_i}{dz} \frac{dw_k}{dz} \frac{dw_n}{dz} + 3 \sum_{i=1}^{m} \sum_{k=1}^{m} \frac{\partial^2 a_{i0}}{\partial w_i \partial w_k} \frac{dw_i}{dz} \frac{d^2 w_k}{dz^2} + 3 \sum_{i=1}^{m} \frac{\partial^2 a_{i0}}{\partial w_i \partial \lambda} \frac{dw_i}{dz} \frac{d^2 \lambda}{dz^2} = 0 \tag{7.170}
\]

The second derivatives of the displacement coordinates \( w_k \) in equation (7.170) are derived from equation (7.115) using properties (7.169):

\[
\sum_{i=1}^{m} \frac{\partial g_{i0}}{\partial w_i} \frac{d^2 w_i}{dz^2} = -\frac{d g_{i0}}{d \lambda} \frac{d^2 \lambda}{dz^2} - \sum_{i=1}^{m} \sum_{k=1}^{m} \frac{\partial^2 g_{i0}}{\partial w_i \partial w_k} \frac{dw_i}{dz} \frac{dw_k}{dz} \tag{7.171}
\]

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):

\[
\sum_{i=1}^{m} \frac{\partial g_{i0}}{\partial w_i} \frac{d^2 w_i}{dz^2} = -\frac{d g_{i0}}{d \lambda} \frac{d^2 \lambda}{dz^2} - 2 \sum_{i=1}^{m} \sum_{k=1}^{m} \frac{\partial^2 g_{i0}}{\partial w_i \partial w_k} x_{i0} x_{k0} + \mathbf{y}_0 \sum_{i=1}^{m} \sum_{k=1}^{m} \frac{\partial^2 a_{i0}}{\partial w_i \partial w_k} \frac{dw_i}{dz} \frac{dw_k}{dz} \tag{7.172}
\]
Since the coefficient $c_0$ is null in expressions (7.169), the last term of equation (7.172) is null. Matrix $K_0$ is substituted into equation (7.172) from equation (7.118) and vector $z_0$ is substituted from equation (7.147):

$$
\begin{align*}
K_0 \frac{d^2w_0}{dz^2} &= -\frac{dg_0}{d\lambda} \frac{d^2\lambda_0}{dz^2} - \theta_2^2 \sum_{i=1}^{m} \sum_{k=1}^{m} \frac{\partial^2g_0}{\partial w_i \partial w_k} x_{i0} x_{k0} \\
\frac{d^2w_0}{dz^2} &= z_0 \frac{d^2\lambda_0}{dz^2} + \theta_2^2 v_0 \\
v_0 &= -K_0^{-1} \sum_{i=1}^{m} \sum_{k=1}^{m} \frac{\partial^2g_0}{\partial w_i \partial w_k} x_{i0} x_{k0}
\end{align*}
$$

(7.173)

(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 $\lambda$:

$$
\frac{\partial^2\lambda_0}{\partial z^2} = -\frac{\theta_2^2}{3} \sum_{i=1}^{m} \sum_{k=1}^{m} \frac{\partial^3a_0}{\partial w_i \partial w_k \partial w_n} x_{i0} x_{k0} x_{n0} + 3 \sum_{i=1}^{m} \sum_{k=1}^{m} \frac{\partial^2a_0}{\partial w_i \partial w_k} x_{i0} x_{k0} x_{n0} + \sum_{i=1}^{m} \frac{\partial^2a_0}{\partial w_i \partial x_0} x_{i0}
$$

(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:

$$
\frac{\partial^2\lambda_0}{\partial z^2} > 0 \implies \text{stable symmetric bifurcation} \quad (7.176)
$$

$$
\frac{\partial^2\lambda_0}{\partial z^2} < 0 \implies \text{unstable symmetric bifurcation} \quad (7.177)
$$

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 $\theta_2$, 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 $x_0$ of the tangent stiffness matrix at the bifurcation point:

$$
2\theta_2 c_1 \frac{d\lambda}{dz} = 0
$$

(7.178)

**primary branch:**

$$
\frac{d\lambda_1}{dz} \in \mathbb{R} \quad \frac{dw_0}{dz} = \frac{d\lambda_1}{dz} z_0
$$

(7.179)

**secondary branches:**

$$
\frac{d\lambda_1}{dz} = 0 \quad \frac{dw_0}{dz} = \theta_2 x_0
$$

(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):

\[
\begin{align*}
0 & = \frac{d\lambda_0}{dz} \quad (7.181) \\
\frac{d}{dz} w_0 & = \frac{d\lambda_0}{dz} z_0 + \theta_2 x_0 \\ c_i^2 - c_0 c_2 & = 0 \\ (7.183)
\end{align*}
\]

The singular point \((w_0, \lambda_0)\) 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 \(x_0\):

\[
\begin{align*}
\frac{d\lambda_0}{dz} & = 0 \\
\frac{d}{dz} w_0 & = \theta_2 x_0 \\ (7.184)
\end{align*}
\]

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 \( \lambda p_i \) of the coefficients \( p_i \) of a pattern load and of the load factor \( \lambda \).
- 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 \( \lambda \) 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 \( \lambda \), are null.

\[
\frac{\partial \mathbf{g}}{\partial \lambda} = -\mathbf{p} \tag{7.185}
\]

\[
\frac{\partial^2 \mathbf{g}}{\partial \mathbf{w}_i \partial \lambda} = \frac{\partial^2 \mathbf{g}}{\partial \lambda^2} = 0 \tag{7.186}
\]

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:

\[
(7.116): \quad \mathbf{K}^i \mathbf{x} = \mu \mathbf{x} \quad \tag{7.187}
\]

\[\mathbf{K}^i \text{ tangent stiffness matrix}\]

\[
(7.118): \quad \mathbf{K}^i \mathbf{x}_0 = 0 \tag{7.188}
\]

\[\mathbf{x}_0 \text{ eigenvector corresponding to } \mu_0 = 0\]

\[
(7.122): \quad \mathbf{x}_0^T \mathbf{p} \neq 0 \rightarrow (\mathbf{w}_0, \lambda_0) \text{ is a turning point} \tag{7.189}
\]

\[
(7.123): \quad \mathbf{x}_0^T \mathbf{p} = 0 \rightarrow (\mathbf{w}_0, \lambda_0) \text{ is a bifurcation point} \tag{7.190}
\]

\[
(7.134): \quad \frac{\partial^2 \lambda}{\partial z^2} = \frac{\theta^2 c_0}{\mathbf{x}_0^T \mathbf{p}} \tag{7.191}
\]

\[
(7.153): \quad c_0 = \sum_{i=1}^{m} \sum_{k=1}^{n} \frac{\partial^2 a_0}{\partial w_i \partial w_k} x_i x_0 \tag{7.192}
\]

\[
(7.154): \quad c_1 = \sum_{i=1}^{m} \sum_{k=1}^{n} \frac{\partial^2 a_0}{\partial w_i \partial w_k} z_i x_0 \tag{7.193}
\]

\[
(7.155): \quad c_2 = \sum_{i=1}^{m} \sum_{k=1}^{n} \frac{\partial^2 a_0}{\partial w_i \partial w_k} z_i z_0 \tag{7.194}
\]
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 \( g \) that is a special case of the vector \( g \) in equation (7.113):

\[
g(w, \lambda) = \frac{w_1(2m^2 + w_1^2 + w_2(2 + w_2)) - \lambda p_1}{(1 + w_2)(w_1^2 + w_2^2(2 + w_2)) - \lambda p_2}
\]

(7.195)

The first partial derivatives of the vector \( g \) are:

\[
\frac{\partial g}{\partial w_1} = \frac{2m^2 + 3w_1^2 + w_2(2 + w_2)}{2w_1(1 + w_2)}
\]

(7.196)

\[
\frac{\partial g}{\partial w_2} = \frac{2w_1(1 + w_2)}{w_1^2 + 3w_2^2 + 6w_2 + 2}
\]

(7.197)

\[
\frac{\partial g}{\partial \lambda} = \begin{bmatrix} -p_1 \\ -p_2 \end{bmatrix}
\]

(7.198)

The second partial derivatives of the vector \( g \) are:

\[
\frac{\partial^2 g}{\partial w_1^2} = \frac{6w_1}{2(1 + w_2)}
\]

(7.199)

\[
\frac{\partial^2 g}{\partial w_1 \partial w_2} = \frac{2(1 + w_2)}{2w_1}
\]

(7.200)

\[
\frac{\partial^2 g}{\partial w_2^2} = \frac{2w_1}{6(1 + w_2)}
\]

(7.201)

**Turning point:** Consider the singular point defined by equation (2.49). The displacements and the load at this point are:

\[
w_{10} = 0
\]

\[
w_{20} = -1 + \frac{1}{\sqrt{3}}
\]

\[
p_{10} = 0
\]

\[
p_{20} = -\frac{2}{3\sqrt{3}}
\]

(7.202)

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 \( K \) in equation (7.187) contains the vectors (7.196) and (7.197), evaluated at the singular point (7.202):

\[
K_0 = \begin{bmatrix} 2m^2 - 0.6667 & 0 \\ 0 & 0 \end{bmatrix}
\]

(7.203)

The eigenstates of matrix \( K_0 \) are determined as follows:
\[ \mu_0 = 0 : \quad x_0 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \]  
\[ \mu_1 = 2m^2 - 0.6667 : \quad x_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \]  

The scalar product \( x_0^T p \) 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 \( \lambda \) with the parameter \( z \). The second derivatives (7.199) to (7.201) of \( \mathbf{g} \) are evaluated at the turning point:

\[ \frac{\partial^2 \mathbf{g}_0}{\partial w_1^2} = \begin{bmatrix} 0 \\ 2(1 + w_{20}) \end{bmatrix} \quad \frac{\partial^2 \mathbf{g}_0}{\partial w_1 \partial w_2} = \begin{bmatrix} 2(1 + w_{20}) \\ 0 \end{bmatrix} \quad \frac{\partial^2 \mathbf{g}_0}{\partial w_2^2} = \begin{bmatrix} 0 \\ 6(1 + w_{20}) \end{bmatrix} \]  

The second derivative of the load factor \( \lambda \) is determined by substituting (7.204) and (7.206) into expression (7.191):

\[ \frac{d^2 \lambda}{dz^2} = \frac{\theta^2}{p_{20}} \left( \frac{\partial^2 (x_0^T \mathbf{g}_0)}{\partial w_1^2} x_{10}^2 + 2 \frac{\partial^2 (x_0^T \mathbf{g}_0)}{\partial w_1 \partial w_2} x_{10} x_{20} + \frac{\partial^2 (x_0^T \mathbf{g}_0)}{\partial w_2^2} x_{20}^2 \right) \]

\[ \frac{d^2 \lambda}{dz^2} = \frac{6 \theta^2 (1 + w_{20})}{p_{20}} < 0 \]  

Since \( p_{20} < 0 \) and \( |w_{20}| < 1 \), the second derivative of \( \lambda \) 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):

\[ w_{10} = 0 \]  
\[ w_{20} = -1 + \sqrt{1 - 2m^2} \]  
\[ p_{10} = 0 \]  
\[ p_{20} = -2m^2 \sqrt{1 - 2m^2} \]  

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 \( \mathbf{K}_0 \) in equation (7.187) contains the vectors (7.196) and (7.197), evaluated at the singular point (7.208):

\[ \mathbf{K}_0 = \begin{bmatrix} 0 & 0 \\ 0 & 2 - 6m^2 \end{bmatrix} \]  

The eigenstates of matrix \( \mathbf{K}_0 \) are determined as follows:

\[ \mu_0 = 0 : \quad x_0 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \]  
\[ \mu_1 = 2 - 6m^2 : \quad x_1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \]  

235
The scalar product $x_0^T p$ 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:

$$\frac{\partial^2 g_0}{\partial w_1^2} = \begin{bmatrix} 0 \\ 2(1 + w_{20}) \end{bmatrix}, \quad \frac{\partial^2 g_0}{\partial w_1 \partial w_2} = \begin{bmatrix} 2(1 + w_{20}) \\ 0 \end{bmatrix}, \quad \frac{\partial^2 g_0}{\partial w_2^2} = \begin{bmatrix} 0 \\ 6(1 + w_{20}) \end{bmatrix} \quad (7.212)$$

Consider equation (7.147) for the determination of the vector $z_0$:

$$\begin{bmatrix} 0 \\ 0 \\ z_{10} \\ z_{20} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ p_{20} \\ 2 - 6m^2 \end{bmatrix} \quad (7.213)$$

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 $x_0$, the vector $z_0$ is chosen orthogonally to the eigenvector $x_0$. The solution for equation (7.213) is then:

$$z_{10} = 0,$$

$$z_{20} = \frac{p_{20}}{2 - 6m^2} \quad (7.214)$$

The coefficients $c_0$ to $c_2$ can now be evaluated:

$$c_0 = \frac{\partial^2 (x_0^T g_0)}{\partial w_1^2} x_{10} x_{10} + \frac{\partial^2 (x_0^T g_0)}{\partial w_1 \partial w_2} (x_{10} x_{20} + x_{20} x_{10}) + \frac{\partial^2 (x_0^T g_0)}{\partial w_2^2} x_{20} x_{20}$$

$$c_1 = \frac{\partial^2 (x_0^T g_0)}{\partial w_1^2} x_{10} z_{10} + \frac{\partial^2 (x_0^T g_0)}{\partial w_1 \partial w_2} (x_{10} z_{20} + x_{20} z_{10}) + \frac{\partial^2 (x_0^T g_0)}{\partial w_2^2} x_{20} z_{20}$$

$$c_2 = \frac{\partial^2 (x_0^T g_0)}{\partial w_1^2} z_{10} z_{10} + \frac{\partial^2 (x_0^T g_0)}{\partial w_1 \partial w_2} (z_{10} z_{20} + z_{20} z_{10}) + \frac{\partial^2 (x_0^T g_0)}{\partial w_2^2} z_{20} z_{20}$$

The coordinates $x_{10}, x_{20}, z_{10}$ and $z_{20}$ are substituted:

$$c_0 = c_2 = 0,$$

$$c_1 = \frac{-2m^2(1 - 2m^2)}{1 - 3m^2} \quad (7.215)$$

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 $c_2 = 0$:

$$2\theta_2 c_1 \frac{d\lambda}{dz} = 0 \quad \land \quad c_1 \neq 0 \quad \rightarrow \quad \frac{d\lambda}{dz} = 0 \quad \lor \quad \theta_2 = 0 \quad (7.216)$$

Equation (7.144) shows that the load path following the bifurcation point is either tangential to eigenvector $x_0$ if the derivative of the load factor is null, or tangential to vector $z_0$ if the parameter $\theta_2$ 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 \( (w_{c1}, w_{c2}) = (0.0, -0.422650) \) at the turning point is determined with (7.31). The tangent stiffness matrix \( \mathbf{K}_i \) at the turning point follows from (7.14) and (7.31):

\[
\begin{bmatrix}
0.953333 & 0 & \Delta w_1 \\
0 & 0 & \Delta w_2
\end{bmatrix}
\begin{bmatrix}
\Delta p_1 \\
\Delta p_2
\end{bmatrix}
= \begin{bmatrix}
0 \\
0
\end{bmatrix}
\]  

(7.217)
Consider an infinitesimal displacement step \((dw_1, dw_2)\) beyond the turning point \(C\). Equation (7.217) shows that the infinitesimal displacement \(dw\) 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 \(K^i\) 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 \((w_1, w_2)\).

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 \(w_i = \Delta w_i = 0\):

\[
\begin{align*}
2m^2 + w_2c(2 + w_2c) + (1 + w_2c)\Delta w_2c + \Delta w_2^2 &\quad 0 \\
0 &\quad 3(1 + w_2c)\Delta w_2c + \Delta w_2^2 \\
0 &\quad \Delta w_2 \\
0 &\quad \Delta p_2
\end{align*}
\]

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 \(x_0\). 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):

\[
K^{(5)} = \begin{bmatrix} 0.926965 & 0.0 \\ 0.0 & -0.084103 \end{bmatrix}
\]

Equation (7.219) 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 (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 $(\omega_i, \mathbf{x}_i)$ with $i = 0, \ldots, n-1$, where $n$ is the dimension of the matrix:

$$ \mathbf{K} = \sum_{i=0}^{n-1} \omega_i \mathbf{x}_i \mathbf{x}_i^T \quad (7.221) $$

$$ \mathbf{x}_i^T \mathbf{x}_m = \delta_{im} $$

$$ \omega_i \quad \text{eigenvalue of matrix } \mathbf{K} $$

$$ \mathbf{x}_i \quad \text{eigenvector of matrix } \mathbf{K} $$

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):

$$ \sum_{i=0}^{n-1} \omega_i \mathbf{x}_i^T \sum_{k=0}^{n-1} c_k \mathbf{x}_k = \sum_{m=0}^{n-1} \theta_m \mathbf{x}_m $$

$$ \sum_{i=0}^{n-1} \omega_i c_i \mathbf{x}_i = \sum_{m=0}^{n-1} \theta_m \mathbf{x}_m $$

Due to the orthonormality of the eigenvectors, equation (7.224) simplifies to:

$$ \sum_{i=0}^{n-1} \omega_i c_i \mathbf{x}_i = \sum_{m=0}^{n-1} \theta_m \mathbf{x}_m $$

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:

$$ \omega_i c_i = \theta_i \quad i \in \{0, \ldots, n-1\} $$

239
Let the eigenstate \((\omega_0, x_0)\) with the smallest eigenvalue \(|\omega_0|\) be known. The coefficients \(c_0\) and \(\omega_0\) for this eigenstate can be computed without knowledge of the other eigenstates of matrix \(K\). Coefficient \(\theta_0\) is obtained from equation (7.223) by multiplying with the eigenvector \(x_0^T\) from the left:

\[
\theta_0 = x_0^T \Delta p
\]  

(7.227)

The coefficient \(c_0\) follows from expression (7.226):

\[
c_0 = \frac{\theta_0}{\omega_0}
\]  

(7.228)

If the eigenstate \((\omega_0, x_0)\) 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 \(c_0 x_0\) in equation (7.232) by means of the following identity:

\[
c_0 K x_0 = c_0 \omega_0 x_0 = \theta_0 x_0
\]  

(7.229)

Equation (7.229) is subtracted from equation (7.224):

\[
\sum_{i=1}^{n-1} \omega_i x_i x_i^T \sum_{k=1}^{n-1} c_k x_k = \sum_{m=1}^{n-1} \theta_m x_m
\]  

(7.230)

The remaining sums in equation (7.230) are denoted by \(\Delta z_0\) and \(\Delta p_0\):

\[
\Delta z_0 = \sum_{k=1}^{n-1} c_k x_k = \Delta d - c_0 x_0
\]  

(7.231)

\[
\Delta p_0 = \sum_{m=1}^{n-1} \theta_m x_m = \Delta p - \theta_0 x_0
\]  

(7.232)

Expressions (7.231) and (7.232) are substituted into equation (7.230):

\[
K \Delta z_0 = \Delta p_0
\]  

(7.233)

Since matrix \(K\) is assumed to be regular, the equation can be solved for the deflated displacement increment \(\Delta z_0\). Substitution of the increment \(\Delta z_0\) into equation (7.231) yields the solution \(\Delta d\) for equation (7.220).

The theory shows that nearly singular equations, whose solution is dominated by the eigenstate \((\omega_0, x_0)\) with the smallest eigenvalue, can be solved as follows:

1. Compute the coefficient \(\theta_0\) 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 \(\Delta z_0\).

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 \(\Delta d\).

Equation (7.228) shows that the solution of equation (7.220) converges to the eigenvector \(x_0\) as the tangent matrix \(K\) 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:

\[
\begin{pmatrix}
0.147431000 & 14.469023000 \\
14.469023000 & 1327.394235000
\end{pmatrix}
\begin{pmatrix}
\Delta u_0 \\
\Delta u_1
\end{pmatrix}
= 
\begin{pmatrix}
-0.078864000 \\
-3.617205000
\end{pmatrix}
\]

The eigenstates of the coefficient matrix are:

\[
\omega_0 = -0.010284771, \quad \omega_1 = 1327.551950770,
\]

\[
x_0 = \begin{pmatrix}
-0.999940598 \\
0.010899589
\end{pmatrix}, 
\quad x_1 = \begin{pmatrix}
0.010899589 \\
0.999940598
\end{pmatrix}
\]

The singular matrix in the vicinity of the given matrix equals \( \omega_0 x_0 x_0^T \):

\[
K_0 = \begin{pmatrix}
0.157714549 & 14.468910907 \\
14.468910907 & 1327.394236222
\end{pmatrix}
\]

The parameter \( \theta_0 \) in equation (2.227) and the right-hand side of equation (7.232) are:

\[
\theta_0^{(0)} = x_0^T p = -0.0394332680,
\]

\[
\Delta p - \theta_0^{(0)} x_0 = \begin{pmatrix}
-0.039433074 \\
-3.617634806
\end{pmatrix}
\]

Solution of equation (7.233) yields an improved deflated displacement increment:

\[
\Delta d_0^{(1)} = \begin{pmatrix}
-0.000029704 \\
-0.002725042
\end{pmatrix}
\]

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):

\[
\Delta d = \begin{pmatrix}
-0.000029704 \\
-0.002725042
\end{pmatrix}
+ 3.834141567 \begin{pmatrix}
-0.999940598 \\
0.010899589
\end{pmatrix}
= \begin{pmatrix}
3.833884301 \\
-0.044515611
\end{pmatrix}
\]

The error in the governing equations is of order \( O[10^{-6}] \) relative to the load vector.

Extended deflation: Consider a system of linear equations that contains free displacements \( d_1 \), prescribed displacements \( d_2 \), loads \( q_1 \), and \( q_2 \) and reactions \( r_2 \). Let the equations be structured as shown in equation (7.234).

\[
Kd = q + r \quad (7.234)
\]

\[
\begin{pmatrix}
K_{11} & K_{12} \\
K_{21} & K_{22}
\end{pmatrix}
\begin{pmatrix}
d_1 \\
d_2
\end{pmatrix}
= 
\begin{pmatrix}
q_1 \\
q_2
\end{pmatrix}
+ 
\begin{pmatrix}
r_1 \\
r_2
\end{pmatrix}
\]

Let the submatrix \( K_{11} \) be nearly singular and let \( (\omega_0, x_0) \) be the eigenstate of \( K_{11} \) with the smallest absolute value of the eigenvalue \( \omega_0 \).
\[ \mathbf{K}_{11} \mathbf{x}_{10} = \omega_0 \mathbf{x}_{10} \quad \land \quad \omega_0 \neq 0 \quad \land \quad \mathbf{x}_{10}^\top \mathbf{x}_{10} = 1 \quad (7.235) \]

\((\omega_0, \mathbf{x}_{10})\) 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 \(\mathbf{K}\).

\[
\begin{bmatrix}
\mathbf{K}_{11} & \mathbf{K}_{12} \\
\mathbf{K}_{21} & \mathbf{K}_{22}
\end{bmatrix}
\begin{bmatrix}
\mathbf{x}_{10} \\
\mathbf{0}
\end{bmatrix}
= \omega_0
\begin{bmatrix}
\mathbf{x}_{10} \\
\mathbf{0}
\end{bmatrix}
+ \begin{bmatrix}
\mathbf{0} \\
\mathbf{r}_{20}
\end{bmatrix}
\]

\(\mathbf{r}_{20}\) reaction due to displacement \(\mathbf{x}_{10}\)
\(\mathbf{x}_0\) system eigen-displacement
\(\mathbf{r}_0\) system eigen-reaction

Let the load subvector \(\mathbf{q}\), and the displacement subvector \(\mathbf{d}\), in equation (7.234) be deflated as shown in equations (7.227) to (7.232):

\[
\begin{align*}
\mathbf{q}_i &= \mathbf{q}_i - \theta_0 \mathbf{x}_{10} \\
\mathbf{d}_i &= \mathbf{d}_i - c_0 \mathbf{x}_0 \\
\theta_0 &= \mathbf{x}_{10}^\top \mathbf{q}_i \\
\theta_0 &= c_0 \omega_0
\end{align*}
\quad (7.237)
\]

\(\theta_0\) deflation factor

The system load vector \(\mathbf{q}\) and the system displacement vector \(\mathbf{d}\) are deflated by replacing the subvectors \(\mathbf{q}_i\) and \(\mathbf{d}_i\) with their deflated counterparts:

\[
\begin{align*}
\mathbf{\bar{q}} &= \mathbf{q} - \theta_0 \mathbf{x}_0 \\
\mathbf{\bar{d}} &= \mathbf{d} - c_0 \mathbf{x}_0
\end{align*}
\quad (7.238)
\]

\(\mathbf{q}\) deflated load vector
\(\mathbf{d}\) deflated displacement vector

Equation (7.234) is deflated by multiplying equation (7.236) with \(c_0\) and subtracting it from equation (7.234):

\[
\begin{align*}
\mathbf{K} (\mathbf{d} - c_0 \mathbf{x}_0) &= (\mathbf{q} - \theta_0 \mathbf{x}_0) + (\mathbf{r} - c_0 \mathbf{r}_0) \\
\mathbf{K} \mathbf{\bar{d}} &= \mathbf{\bar{q}} + \mathbf{\bar{r}} \\
\mathbf{\bar{r}} &= \mathbf{r} - c_0 \mathbf{r}_0
\end{align*}
\quad (7.239 - 7.241)
\]

\(\mathbf{\bar{r}}\) deflated reaction vector

The total displacements and reactions follow from equations (7.238) and (7.241):

\[
\begin{align*}
\mathbf{d} &= \mathbf{\bar{d}} + c_0 \mathbf{x}_0 \\
\mathbf{r} &= \mathbf{\bar{r}} + c_0 \mathbf{r}_0
\end{align*}
\quad (7.242 - 7.243)
\]

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 $K$, displacement increment $\Delta d$ and load increment $\Delta p$:

$$K \Delta d = \Delta p$$  \hspace{1cm} (7.244)

Let the diagonal coefficient in row $i$ go to null when the matrix $K$ is decomposed in the form of expression (6.80) with diagonal matrix $D$. Let a matrix $K_s$ be called the stabilised matrix associated with matrix $K$ in equation (7.244) if their diagonal coefficients differ by a specified constant $c$:

$$K_s = K + c e_i e_i^T$$  \hspace{1cm} (7.245)

$K_s$ stabilised coefficient matrix
$e_i$ unit vector with coefficient 1 in column $i$

The term $c e_i e_i^T \Delta d$ is added on both sides of equation (7.244):

$$K_s \Delta d = \Delta p + c e_i e_i^T \Delta d$$  \hspace{1cm} (7.246)

The parameter $c$ is chosen so that matrix $K_s$ is regular. Equation (7.246) is multiplied from the left with the inverse of matrix $K_s$:

$$\Delta d = K_s^{-1} \Delta p + c (e_i^T \Delta d) K_s^{-1} e_i$$  \hspace{1cm} (7.247)

The vectors on the right-hand side of equation (7.247) are denoted by $\Delta v_1$ and $\Delta v_2$:

$$K_s \Delta v_1 = \Delta p \hspace{1cm} \Delta v_1 = K_s^{-1} \Delta p$$

$$K_s \Delta v_2 = e_i \hspace{1cm} \Delta v_2 = K_s^{-1} e_i$$

$$\Delta d = \Delta v_1 + c (e_i^T \Delta d) \Delta v_2$$  \hspace{1cm} (7.248)

Equation (7.248) is multiplied from the left with the transposed unit vector $e_i^T$ and the resulting equation is solved for the scalar product $e_i^T \Delta d$:

$$e_i^T \Delta d = \frac{e_i^T \Delta v_1}{1 - c (e_i^T \Delta v_2)}$$  \hspace{1cm} (7.249)

Substitution of expression (7.249) into equation (2.247) leads to the increment of the displacement in equation (7.244):

$$\Delta d = \Delta v_1 + \frac{c e_i^T \Delta v_1}{1 - c (e_i^T \Delta v_2)} \Delta v_2$$  \hspace{1cm} (7.250)

Equation (2.250) shows that as parameter $c$ goes to null, the stabilised matrix $K_s$ goes to the almost singular matrix $K$ and the stabilised solution $\Delta d$ goes to:

$$\Delta v_1 = K^{-1} \Delta p$$  \hspace{1cm} (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:

\[
\begin{bmatrix}
K_{11} & K_{12} & \Delta d_1 \\
K_{21} & K_{22} & \Delta d_2
\end{bmatrix}
\begin{bmatrix}
q_1 \\
q_2
\end{bmatrix}
= \Delta \lambda
\begin{bmatrix}
0 \\
\Delta r_2
\end{bmatrix}
\tag{7.252}
\]

\( K_0 \)  tangent stiffness matrix at the turning point

The stiffness submatrix \( K_{11} \) is singular and therefore has an eigenstate \((\omega_0, x_0)\) with eigenvalue \( \omega_0 = 0 \):

\[
K_{11} x_{10} = \omega_0 x_{10} \quad \wedge \quad \omega_0 = 0 \quad \wedge \quad x_{10}^T x_{10} = 1
\tag{7.253}
\]

\( \omega_0 \)  smallest eigenvalue of matrix \( K_{11} \)

\( x_{10} \)  eigenvector of matrix \( K_{11} \) corresponding to eigenvalue \( \omega_0 \)

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 \( x_{10} \). The approximate displacement increment \( \Delta d^{(s, 0)} \) for cycle 0 of the iteration in step \( s \) is therefore set equal to a fraction \( \mu \) of \( x_{10} \):

\[
\Delta d^{(s, 0)} = \mu x_{10}
\tag{7.254}
\]

\( \Delta d^{(s, 0)} \)  free displacement increment in cycle 0

\( \mu \)  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 \( \Delta d^{(s, 0)} \) is chosen so that it makes the smallest possible angle \( \theta \) with the system displacement increment \( \Delta d^{(s-1)} \) preceding the turning point. The sign of the factor \( \mu \) is chosen according to the following rule:

\[
|\Delta d^{(s, 0)}| |\Delta d^{(s-1)}| \cos \theta = \mu x_{10}^T \Delta d^{(s-1)}
\]

\[
\text{signum}(\mu) = \text{signum}(x_{10}^T \Delta d^{(s-1)})
\tag{7.255}
\]

The load factor increment \( \Delta \lambda \) in cycle 0 of the first step following the turning point is null. The displacement increment \( x_{10} \) causes a reaction increment \( r_{20} \) which is given by equation (7.236). The system eigenvector \( x_{0} \) and the system eigenreaction \( r_{0} \) 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):
\[ K_0 x_0 = r_0 \]  
\[ \begin{bmatrix} K_{11} & K_{12} & x_{10} \\ K_{21} & K_{22} & 0 \end{bmatrix} = \begin{bmatrix} 0 \\ r_{20} \end{bmatrix} \]  
\( x_0 \) system eigendisplacement  
\( r_0 \) system eigenreaction  
\( r_{20} \) reaction \( K_{20} x_{10} \) due to displacement \( x_{10} \)

The scale factor \( \mu \) 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 \( x_0 \).
- The pattern reaction vector is replaced by the eigenreaction \( r_0 \).

The first trial state following the turning point is given by:

\[ d^{(s+1,0)} = d^{(s)} + \mu x_0 \]  
(7.257)

\[ r^{(s+1,0)} = r^{(s)} + \mu r_0 \]  
(7.258)

\[ q^{(s+1,0)} = q^{(s)} \]  
(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 \( K^{(s,m)} \) 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:

\[ K \Delta d = \Delta \lambda q_i + \Delta r + e \]  
(7.260)

\[ \begin{bmatrix} K_{11} & K_{12} & \Delta d_1 \\ K_{21} & K_{22} & \Delta d_{21} \end{bmatrix} = \Delta \lambda \begin{bmatrix} q_{11} \\ q_{21} \end{bmatrix} + \begin{bmatrix} 0 \\ \Delta r_2 \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} \]

\( e_1 \) error vector for the free nodes at the start of the step  
\( e_2 \) error vector for the supports at the start of the step

Submatrix \( K_{11} \) of the secant matrix \( K \) 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 \( \omega_0 \) 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 \( q \) in equation (7.234) with the load increment \( \Delta \lambda q_i + e \) in equation (7.260). Since the load is now a function of the load factor increment \( \Delta \lambda \), the deflation coefficient \( \theta_0 \) in equation (7.237) is also a function of the load factor increment.
The deflated load increments are derived from equation (7.238):
\[
\delta q = \lambda(\mathbf{q}_t - \mathbf{x}_t^0) + (\mathbf{e} - \theta_20 \mathbf{x}_0)
\]  

(7.262)

The deflated governing equations follow from equations (7.242), (7.260) and (7.262):
\[
\mathbf{K}\delta\mathbf{d} = \delta\mathbf{q} + \delta\mathbf{r}
\]  

(7.263)

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:
\[
\mathbf{K}u_t = \bar{\mathbf{q}}_t + \bar{w}_t
\]  

(7.264)

Since submatrix \(K_{11}\) is assumed to be regular, the first equation can be solved for the deflated pattern displacement \(\bar{\mathbf{v}}_t\) and the second for the deflated pattern reaction \(\bar{w}_{2t}\):
\[
\bar{\mathbf{v}}_t = K_{11}^{-1} (\mathbf{q}_t - K_{12} \mathbf{d}_{2t} - \theta_{10} \mathbf{x}_{10})
\]  

(7.265)

\[
\bar{w}_{2t} = K_{22} u_t + K_{21} \mathbf{d}_{2t} - \mathbf{q}_{2t}
\]  

(7.266)

The deflated error state is defined as follows:
\[
\mathbf{K}e_t = \bar{\mathbf{e}} + \bar{t}
\]  

(7.267)

The first of equations (7.267) is solved for the deflated correction displacement \(\bar{t}_1\), the second for the deflated reaction correction \(\bar{t}_2\):
\[
\bar{t}_1 = K_{11}^{-1} (\mathbf{e}_t - \theta_{20} \mathbf{x}_{20})
\]  

(7.268)

\[
\bar{t}_2 = K_{21} \bar{t}_1 - \mathbf{e}_2
\]  

(7.269)

The solution to equation (7.263) is obtained by multiplying the solution to equation (7.264) by the load factor increment \(\lambda\) and adding the solution to equation (7.267):
\[
\mathbf{d}_{1t} = \lambda \bar{\mathbf{v}}_t + \bar{t}_1
\]  

(7.270)

\[
\bar{r}_{2t} = \lambda \bar{w}_{2t} + \bar{t}_2
\]
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):

\[
\Delta d_i = \Delta \lambda \bar{d}_i + c_0 \bar{x}_0 \\
\Delta r_2 = \Delta \lambda \bar{r}_2 + c_0 \bar{r}_0 \\
c_0 = \frac{1}{\omega_0} (\Delta \lambda (\theta_{10} + \theta_{20})
\]

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:

\[
|m(a_2 - a_1)| \leq |\eta (b_2 - b_1)|
\]

$a_1, a_2$ 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
$\eta$ 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 $p_0$ 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):

$$
\begin{bmatrix}
0 & 0 \\
0 & 1.0400
\end{bmatrix}
\begin{bmatrix}
\Delta w_1 \\
\Delta w_2
\end{bmatrix}
= 
\begin{bmatrix}
0 \\
\Delta p_2
\end{bmatrix}
$$

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 \(K_1^{\prime}\) 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 \((w_{c_1}, w_{c_2}) = (0.0, -0.175379)\):

\[
\begin{pmatrix}
0.144621 & 0 & 0 & 0 \\
0 & 1.504621 & -0.175379 & 0 \\
\end{pmatrix}
= \begin{pmatrix}
0 \\
-0.263879 \\
\end{pmatrix}
\] (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:

\[
\begin{pmatrix}
0.142500 & -0.041000 & 0.050000 & 0.014505 \\
-0.041000 & 1.492400 & -0.180000 & -0.270682 \\
\end{pmatrix}
= \begin{pmatrix}
0.014505 \\
-0.270682 \\
\end{pmatrix}
\] (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 \((w_{c_1}, w_{c_2}) = (0.0, -0.040833)\) 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 isolines for \(\Delta p_{1}\) and \(\Delta p_{2}\) 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 \(\Delta p_{1}\) and \(\Delta p_{2}\) 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:

\[
f(w_{1} + \Delta w_{1}, w_{2} + \Delta w_{2}) = f(w_{1}, w_{2}) + f_{m} \Delta w_{m} + f_{2} \Delta w_{2} +
\]

\[
\frac{1}{2} (f_{11} \Delta w_{1}^2 + 2 f_{12} \Delta w_{1} \Delta w_{2} + f_{22} \Delta w_{2}^2)
\]

\[
\frac{1}{6} (f_{111} \Delta w_{1}^3 + 3 f_{112} \Delta w_{1}^2 \Delta w_{2} + f_{122} \Delta w_{1} \Delta w_{2}^2 + f_{222} \Delta w_{2}^3)
\]

\(w_{1}, w_{2}\) pivot of the Taylor series

\(\Delta w_{1}\) increment of \(w_{1}\)

\(f_{m}\) 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:

\[
\frac{\partial^3 p_1}{\partial w_1^3} = 6 \quad \frac{\partial^3 p_2}{\partial w_1^3} = 2 \quad (7.282)
\]

\[
\frac{\partial^3 p_1}{\partial w_1 \partial w_2^2} = 2 \quad \frac{\partial^3 p_2}{\partial w_1 \partial w_2^2} = 6
\]

\[
\frac{\partial^3 p_1}{\partial w_2^3} = 2
\]

Figure 7.7 shows the approximation of the isolines for the load increments \(\Delta p_1\) and \(\Delta p_2\), which is obtained if the Taylor series is written at the bifurcation point and truncated after the linear term. The load increment \(\Delta p_1\) is null in the plane, and the isolines for \(\Delta p_2\) 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 \(\Delta p_2\) 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 \(\Delta p_1\) and \(\Delta p_2\), which is obtained if the quadratic terms are retained in the Taylor series at the bifurcation point. While the approximation of the isolines for \(\Delta p_2\) 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 \(\Delta p_1\) and \(\Delta p_2\), 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

a: \( \Delta p_1 = 0.0001 \)
b: \( \Delta p_1 = 0.0000 \)
c: \( \Delta p_1 = -0.0001 \)
d: \( \Delta p_2 = 0.0020 \)
e: \( \Delta p_2 = 0.0000 \)
f: \( \Delta p_2 = -0.0020 \)

at all points:
\( \Delta p_1 = 0.0000 \)
d: \( \Delta p_2 = 0.0020 \)
e: \( \Delta p_2 = 0.0000 \)
f: \( \Delta p_2 = -0.0020 \)

exact isoline for \( \Delta p_1 = 0 \)
Figure 7.8: Isolines of the approximate load increments for the quadratic Taylor series at bifurcation point S

- a: $\Delta p_1 = 0.0001$
- b: $\Delta p_1 = 0.0000$
- c: $\Delta p_1 = -0.0001$
- d: $\Delta p_2 = 0.0020$
- e: $\Delta p_2 = 0.0000$
- f: $\Delta p_2 = -0.0020$

-- exact isoline for $\Delta p_1 = 0$

Figure 7.9: Isolines of the approximate load increments for the cubic Taylor series at the bifurcation point S

- a: $\Delta p_1 = 0.0001$
- b: $\Delta p_1 = 0.0000$
- c: $\Delta p_1 = -0.0001$
- d: $\Delta p_2 = 0.0020$
- e: $\Delta p_2 = 0.0000$
- f: $\Delta p_2 = -0.0020$
In order to investigate the approximation in the second step of the continuation, load increments $\Delta p_1$ and $\Delta p_2$ 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 $\Delta p_1$ and $\Delta p_2$ which is obtained if the Taylor series is written at the pivot $P$ and truncated after the linear term. The isolines for $\Delta p_1$ 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 $\Delta p_1$ and $\Delta p_2$ 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 $\Delta p_1$ and $\Delta p_2$ which is obtained if the cubic terms are retained in the Taylor series at pivot $P$. While the isolines for $\Delta p_2$ do not change significantly relative to figure 7.11, the isolines for $\Delta p_1$ differ significantly at distances $\Delta w_1 > 0.03$.

**Figure 7.10**: Isolines of the approximate increments for the linear Taylor series at pivot $P = (0.03, -0.041303)$

- a: $\Delta p_1 = 0.0001$
- b: $\Delta p_1 = 0.0000$
- c: $\Delta p_1 = -0.0001$
- d: $\Delta p_2 = 0.0020$
- e: $\Delta p_2 = 0.0000$
- f: $\Delta p_2 = -0.0020$

---

Exact isoline for $\Delta p_1 = 0$
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 $\Delta p_1$ and $\Delta p_2$ 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 $\Delta p_1$ and $\Delta p_2$ which is obtained if the Taylor series is written at the pivot $Q$ and truncated after the linear term. The isolines for $\Delta p_1$ 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 $\Delta p_1$ and $\Delta p_2$ 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 $\Delta p_1$ and $\Delta p_2$ 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 $Q = (0.10, -0.046061)$](image)

\[
\begin{align*}
a: \Delta p_1 &= 0.0001 \\
b: \Delta p_1 &= 0.0000 \\
c: \Delta p_1 &= -0.0001 \\
d: \Delta p_2 &= 0.0020 \\
e: \Delta p_2 &= 0.0000 \\
f: \Delta p_2 &= -0.0020 \\
\text{--- exact isoline}
\end{align*}
\text{for } \Delta p_1 = 0
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](image)

**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

<table>
<thead>
<tr>
<th>( w_1 )</th>
<th>( w_2 ) for load path</th>
<th>( w_2 ) for null line</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.000</td>
<td>-0.0408337</td>
<td>-0.0408337</td>
</tr>
<tr>
<td>0.010</td>
<td>-0.0408858</td>
<td>-0.0408811</td>
</tr>
<tr>
<td>0.020</td>
<td>-0.0410422</td>
<td>-0.0410233</td>
</tr>
<tr>
<td>0.030</td>
<td>-0.0413030</td>
<td>-0.0412603</td>
</tr>
<tr>
<td>0.040</td>
<td>-0.0416681</td>
<td>-0.0415922</td>
</tr>
<tr>
<td>0.050</td>
<td>-0.0421378</td>
<td>-0.0420192</td>
</tr>
<tr>
<td>0.060</td>
<td>-0.0427122</td>
<td>-0.0425413</td>
</tr>
</tbody>
</table>
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 \( p_i \) may differ from null. Therefore the general equilibrium equations (2.18) and (2.19) must be considered:

\[
p_i = w_i(2m^2 + w_i^2 + w_2(2 + w_2)) = w_i(w_i^2 + x^2 + 2m^2 - 1)
\]

\[
p_2 = (1 + w_2)(w_i^2 + w_2(2 + w_2)) = x(w_i^2 + x^2 - 1)
\]

\[
w := 1 + w_2
\]

These equations contain the variables \( w_i, w_2, p_i, p_2 \). 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.

(a) \( w_i, w_2 \):

\[
p_i = w_i(w_i^2 + x^2 + 2m^2 - 1) \quad (7.284)
\]

\[
p_2 = x(w_i^2 + x^2 - 1)
\]

(b) \( w_i, p_i \):

\[
x^2 = 1 - 2m^2 - w_i^2 + \frac{p_i}{w_i} \quad (7.284)
\]

\[
p_2 = x(w_i^2 + x^2 - 1)
\]

(c) \( w_2, p_2 \):

\[
w_i^2 = 1 - x^2 + \frac{p_2}{x} \quad (7.285)
\]

\[
p_1 = w_i(w_i^2 + x^2 + 2m^2 - 1)
\]

(d) \( w_i, p_2 \):

\[
x^3 + (w_i^2 - 1)x - p_2 = 0 \quad (7.286)
\]

\[
p_i = w_i(w_i^2 + x^2 + 2m^2 - 1)
\]

(e) \( w_2, p_1 \):

\[
w_i^2 + (x^2 + 2m^2 - 1)w_i - p_1 = 0 \quad (7.287)
\]

\[
p_2 = x(w_i^2 + x^2 - 1)
\]

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 \( w_i, w_2, p_1, p_2 \) 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 \( w_i \) 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 \( x_0 \) corresponding to eigenvalue 0 of the tangent stiffness matrix at the bifurcation point, and a component \( z_0 \) that satisfies the governing equations at the bifurcation point. The eigenvector \( x_0 \) 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 \( x_0 \). The component \( z_0 \) 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:

\[
K_w p_s = 0.039166304 0 0 0 0 1.879166306 -0.040833695 0.076733303
\]

\[
K_s \quad \text{total stiffness matrix at the singular point} \\
\mathbf{w}_s \quad \text{total displacement at the singular point} \\
p_s \quad \text{total load at the singular point}
\]

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 \( \mathbf{w}_a = (w_{1a}, w_{2a}) =\)
The total stiffness matrix \( K_a \) is computed for displacement state \( w_a \). Load \( p_a \) is set equal to the total load \( p_s \) at the beginning of the cycle. The equations are solved for the total displacement \( w_b \):

\[
K_a w_b = p_a
\]

\[
\begin{bmatrix}
0.040066305 & 0.028774989 & 0.029652215 \\
0.028774989 & 1.879166305 & -0.041287749
\end{bmatrix}
\begin{bmatrix}
w_b \\
p_a
\end{bmatrix}
= \begin{bmatrix} 0 \\ -0.076733304 \end{bmatrix}
\]

- \( K_a(w_a) \) approximate total stiffness matrix in the trial state
- \( p_a \) total load at the start of the cycle
- \( w_b \) approximate total displacement at the end of the cycle

(b) The total stiffness matrix \( K_b \) is computed for the displacement state \( w_a \) determined in operation (a). Matrix \( K_b \) is multiplied with displacement \( w_b \) to determine the load \( p_b \) for which the frame is in equilibrium:

\[
K_b w_b = p_b
\]

\[
\begin{bmatrix}
0.039591505 & 0.028427938 & 0.029652211 \\
0.028427938 & 1.877841432 & -0.041287749
\end{bmatrix}
\begin{bmatrix}
w_b \\
p_b
\end{bmatrix}
= \begin{bmatrix} 0.000000250 \\ -0.076688894 \end{bmatrix}
\]

- \( K_b(w_b) \) total stiffness matrix at the end of operation (a)
- \( p_b \) total load at the end of operation (b)
- \( w_b \) total displacement at the end of operation (a)

(c) The component \( p_{b1} \) of load \( p_b \) contradicts the specified load \( p_i = 0 \). In order to correct the load, the displacement \( \Delta w_c \) due to the load \( \Delta p_c = (p_{b1}, 0) \) is computed with the total stiffness matrix \( K_b \):

\[
K_b \Delta w_c = \Delta p_c
\]

\[
\begin{bmatrix}
0.039591505 & 0.028427938 & 0.000006386 \\
0.028427938 & 1.877841432 & -0.041287749
\end{bmatrix}
\begin{bmatrix}
\Delta w_c \\
\Delta p_c
\end{bmatrix}
= \begin{bmatrix} 0.000000250 \\ 0.000000000 \end{bmatrix}
\]

- \( \Delta p_b \) total load correction
- \( \Delta w_b \) total displacement correction

The corrected displacements and loads at the end of operation (c) are:

\[
w_c = w_b - \Delta w_c = \\
\begin{bmatrix}
0.029652211 & -0.041287749 \\
0.028427938 & -0.000000097
\end{bmatrix}
\begin{bmatrix}
\Delta w_c \\
\Delta p_c
\end{bmatrix}
= \begin{bmatrix} 0.029645825 \\ -0.041287652 \end{bmatrix}
\]

\[
p_c = p_b - \Delta p_c = \\
\begin{bmatrix}
0.00000250 & -0.076688894 \\
0.00000250 & 0.000000000
\end{bmatrix}
\begin{bmatrix}
\Delta w_c \\
\Delta p_c
\end{bmatrix}
= \begin{bmatrix} 0.000000000 \\ -0.076688894 \end{bmatrix}
\]

(d) Since the total stiffness matrix was computed for displacement \( w_b \), the truss is not in equilibrium for load \( p_c \) and displacement \( w_c \). The unbalanced forces \( \Delta p_e \) are computed with the equilibrium equations (7.278) and (7.279). The displacement \( \Delta w_e \) due to the unbalanced forces is determined with the total stiffness \( K_b \)
The corrected displacements are:

\[
\mathbf{w}_c = \mathbf{w}_c - \Delta \mathbf{w}_e = \begin{bmatrix}
0.029645825 \\
-0.041287652
\end{bmatrix} - \begin{bmatrix}
-0.000006101 \\
0.000004452
\end{bmatrix} = \begin{bmatrix}
0.029651926 \\
-0.041292104
\end{bmatrix}
\]

The total stiffness matrix is recomputed and multiplied with the computed displacement. The resulting load is nearly proportional to the pattern load:

\[
\mathbf{K}_s \mathbf{w}_s = \mathbf{p}_s
\]

\[
\begin{bmatrix}
0.039587133 & 0.028427536 & 0.029651926 & 0.000000002 \\
0.028427536 & 1.877828726 & 0.041292104 & 0.076696568
\end{bmatrix}
\]

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 \( \lambda \mathbf{p}_2 \). The load \( p_1 \) that is null for the 2-bar truss is replaced by the load component normal to \( \mathbf{q}_1 \) 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 \( \mathbf{q} \) 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 \( \mu \) 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:

\[
\begin{align*}
\mathbf{d}^{(s+1,0)} &= \mathbf{d}^{(s)} + \mu \mathbf{x}_0 \\
\mathbf{r}^{(s+1,0)} &= \mathbf{r}^{(s)} + \mu \mathbf{r}_0 \\
\mathbf{q}^{(s+1,0)} &= \mathbf{q}^{(s)}
\end{align*}
\]
\[ K_s d_c = q_e + r_e \]  
(7.294)

\[
\begin{bmatrix}
K_{a11} & K_{a12} & d_{b1} \\
K_{a21} & K_{a22} & d_{b2}
\end{bmatrix}
\begin{bmatrix}
q_{a1} \\
q_{a2}
\end{bmatrix}
= 
\begin{bmatrix}
d_{b1} \\
d_{b2}
\end{bmatrix}
+ 
\begin{bmatrix}
0 \\
0
\end{bmatrix}
\]

\[ K_a = K(d_b) \]
\[ q_a = q^{(s-1/m-1)} \]

(b) The total stiffness matrix \( K_b \) is computed for the displacement state \( d_b \) determined in operation (a). Matrix \( K_b \) is multiplied with displacement \( d_b \) to determine the total equilibrium load \( q_b \) for which the frame is in equilibrium:

\[
q_b = K_b d_b - r_b
\]  
(7.295)

\[
\begin{bmatrix}
q_{b1} \\
q_{b2}
\end{bmatrix}
\begin{bmatrix}
K_{b11} & K_{b12} & d_{b1} \\
K_{b21} & K_{b22} & d_{b2}
\end{bmatrix}
\begin{bmatrix}
q_{b1} \\
q_{b2}
\end{bmatrix}
= 
\begin{bmatrix}
d_{b1} \\
d_{b2}
\end{bmatrix}
+ 
\begin{bmatrix}
0 \\
0
\end{bmatrix}
\]

\[ K_b = K(d_b) \]

(c) The load \( q_b \) in (7.295) is not proportional to the pattern load \( q \). It is decomposed into a component \( \theta q_l \) and a component \( \Delta q_b \) which is normal to the pattern load \( q \):

\[ q_b = \theta q_l + \Delta q_b \]  
(7.296)

\[ q_l \Delta q_b = 0 \quad \rightarrow \quad \theta = \frac{q_l^T q_b}{q_l^T q_l} \]

The displacement correction \( \Delta d_b \) and the reaction correction \( \Delta r_b \) due to the load correction \( \Delta q_b \) are determined with the total stiffness matrix \( K_b \):

\[
\begin{bmatrix}
K_{b11} & K_{b12} & \Delta d_{b1} \\
K_{b21} & K_{b22} & \Delta d_{b2}
\end{bmatrix}
\begin{bmatrix}
\Delta q_{b1} \\
\Delta q_{b2}
\end{bmatrix}
= 
\begin{bmatrix}
0 \\
0
\end{bmatrix}
\]

\[ K_b (d_b - \Delta d_b) = (q_b - \Delta q_b) + (r_b - \Delta r_b) \]  
(7.297)

The corrected displacements, reactions and loads are computed by subtracting (7.297) from (7.295):

\[
K_b (d_b - \Delta d_b) = (q_b - \Delta q_b) + (r_b - \Delta r_b)
\]

\[ K_b d_c = q_c + r_c \]  
(7.298)

\[ d_c = d_b - \Delta d_b \]  
(7.299)

\[ r_c = r_b - \Delta r_b \]  
(7.300)

\[ q_c = q_b - \Delta q_b \]  
(7.301)

(d) Let the exact total stiffness matrix, displacements and reactions for load \( q_e \) be \( K_e, d_e \) and \( r_e \):

\[ K_e d_e = q_e + r_e \]  
(7.302)

The unbalanced forces \( e \) in state \( (d_e, r_e) \) are computed with the methods in sections 4.3.3 and 5.3.4:

\[ e = q_e + r_e - K_e d_c \]  
(7.303)
It is assumed that the unbalanced forces are due to the use of the approximate total stiffness matrix $K_a$ instead of the exact total stiffness matrix $K_e$ in equation (7.302):

$$K_b \Delta d_c = q_c + r_e + e$$  \hspace{1cm} (7.304)

Equation (7.298) is subtracted from (7.304) to obtain the correction $\Delta d_c$ of the total displacement and $\Delta r_c$ of the total reaction for the unbalanced forces $e$:

$$K_b \Delta d_c = \Delta r_c + e$$  \hspace{1cm} (7.305)

$$\Delta d_c = d_e - d_c$$

$$\Delta r_c = r_e - r_c$$

The increments (7.305) are used to compute the trial state $m$ for the load step:

$$d^{(s+1,m)} = d_c + \Delta d_c$$  \hspace{1cm} (7.306)

$$r^{(s+1,m)} = r_c + \Delta r_c$$

$$q^{(s+1,m)} = q_e$$

**Termination of the iteration:** State (7.306) of the frame is not an exact equilibrium state, since the displacement correction $\Delta d_c$ and the reaction correction $\Delta r_c$ are computed with the total stiffness matrix $K_a$ that corresponds to the displacement state $d_e$. If the error norm $|e|$ 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 $K^{(s+1)}$ 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 $x_0^{(s+1)}$ of matrix $K^{(s+1)}$ as well as the associated eigenreaction $r_0^{(s+1)}$ are computed. The vectors $x_0^{(s+1)}$ and $r_0^{(s+1)}$ replace the vectors $x_0$ and $r_0$ used in the first step of the continuation.

$$K^{(s+1)}x_0^{(s+1)} = \omega_0^{(s+1)}x_0^{(s+1)} + r_0^{(s+1)}$$  \hspace{1cm} (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 $x_0^{(s+1)}$. 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:

$$|e| \geq \text{limit } |e| \Rightarrow \text{return to the constant arc increment method}$$  \hspace{1cm} (7.308)

where $|e|$ is the norm of the unbalanced forces at the end of the load step, and the limit $|e|$ is the 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 $K_s$ of a truss is derived from the governing equation (4.52) for the error term $\delta e = 0$:

$$\sum_c \int_c \delta \epsilon^0_s \, sdv = \delta d_s^T K_s d_s$$  \hspace{1cm} (7.309)

$K_s$  total stiffness matrix of the truss  
$d_s$  system displacement vector

The axial strain $\varepsilon$ is substituted from expression (4.32), the variation $\delta \varepsilon^0$ from (4.48):

$$\varepsilon = v_{t1} + \frac{1}{2}(v_{t1}^2 + v_{t2}^2)$$  \hspace{1cm} (7.310)

$$\delta \varepsilon^0 = \delta v_{t1} + v_{t1} \delta v_{t1} + v_{t2} \delta v_{t2}$$

$v_m$  displacement coordinates in reference space  
$v_{m,1}$  derivative of $v_i$ with respect to reference coordinate $y_i$

The terms of the product $\varepsilon \delta \varepsilon$ are arranged so that the product leads to a symmetric total stiffness matrix:

$$\varepsilon^0 \delta \varepsilon = c_1 v_{t1} \delta v_{t1} + c_2 (v_{t1} \delta v_{t1} + v_{t2} \delta v_{t2}) + c_3 v_{t2} \delta v_{t2}$$  \hspace{1cm} (7.311)

$$c_1 = 1 + \frac{3}{2} v_{t1} + \frac{1}{2} v_{t2}^2$$

$$c_2 = \frac{1}{2} v_{t2}$$

$$c_3 = \frac{1}{2} v_{t1} + \frac{1}{2} v_{t1}^2 + \frac{1}{2} v_{t2}^2$$

The derivatives of the bar displacements are interpolated with expression (4.64):

$$v_{m,1} = \frac{1}{a} g_{m}^T v_e$$  \hspace{1cm} (7.312)

$a$  length of the bar in the reference configuration

$g_m$  displacement derivative interpolation vector

$v_e$  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:

$$\int_c \delta \epsilon^0_s \, sdv = \delta v_e^T K_e v_e$$  \hspace{1cm} (7.313)

$K_e = \frac{EA}{a} \left( c_1 g_1 g_1^T + c_2 (g_1 g_1^T + g_2 g_2^T) + c_3 g_2 g_2^T \right)$

The differentiation vectors $g_m$ are substituted from (4.64):

$$K_s = \frac{EA}{a}$$  \hspace{1cm} (7.314)
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 \( \mathbf{K}_s \):

\[
\delta \mathbf{d}_s^T \mathbf{K}_s \mathbf{d}_s = \sum_{e} \delta \mathbf{v}_e^T \mathbf{K}_e \mathbf{v}_e
\]

\[
\mathbf{K}_s = \sum_{e} \mathbf{T}_e^T \mathbf{R}_{esy}^T \mathbf{K}_e \mathbf{R}_{esy}^T \mathbf{T}_e
\]  

(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:

\[
\sum_{e} \int_{\mathcal{C}_e} \delta \varepsilon^0 \mathbf{sdv} = \delta \mathbf{d}_s^T \mathbf{K}_e \mathbf{d}_s
\]  

(7.315)

\( \mathbf{K}_s \) global stiffness matrix of the frame
\( \mathbf{d}_s \) system displacement vector

The strain \( \varepsilon \) is substituted from expression (5.81):

\[
\varepsilon = \varepsilon_a + y_2 \varepsilon_b
\]  

(7.316)

\( \varepsilon_a \) axial strain
\( \varepsilon_b \) 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):

\[
\delta \varepsilon_a^0 = (1 - w_{c1}) \delta w_{c1}
\]  

(7.317)

\[
\delta \varepsilon_b^0 = -\delta w_{2,11}
\]  

(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 \( \varepsilon_a \delta \varepsilon_b^0 \) is not symmetrical in the indices 1 and 2 and therefore does not lead to a symmetric stiffness matrix.
\[ \varepsilon_a \delta \varepsilon_a^0 = c_1 w_{c1,1} \delta w_{c1,1} + c_2 w_{c2,1} \delta w_{c2,1} \] (7.319)

\[ c_1 = (1 - w_{c1,1})(1 - \frac{1}{2} w_{c1,1}) \quad c_2 = -\frac{1}{2} (1 - w_{c1,1}) w_{c2,1} \]

The derivatives of the chord displacements are interpolated with expression (5.150):

\[ w_{c1,1} = \frac{1}{a} g_m^T w_e \] (7.320)

\( a \) length of the member in the reference configuration
\( g_m \) displacement derivative interpolation vector
\( w_e \) 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:

\[ \int_{c_s} \delta \varepsilon_a^0 s_a \, dv = \delta w_e^T K_s w_e \] (7.321)

\[ K_s = \frac{EA}{a} \left( c_1 g_1 g_1^T + c_2 g_2 g_2^T \right) \]

The differentiation vectors \( g_m \) are substituted from (5.149):

\[ \begin{array}{cccccc}
  c_1 & c_2 & 0 & -c_1 & -c_2 & 0 \\
  0 & 0 & 0 & 0 & 0 & 0 \\
  0 & 0 & 0 & 0 & 0 & 0 \\
 -c_1 & -c_2 & 0 & c_1 & c_2 & 0 \\
  0 & 0 & 0 & 0 & 0 & 0 \\
  0 & 0 & 0 & 0 & 0 & 0 \\
\end{array} \] (7.322)

The contribution of the bending strains to the product \( \varepsilon \delta \varepsilon^0 \) is:

\[ \varepsilon_b \delta \varepsilon_b^0 = y^2 \varepsilon_{w_{2,11}} \delta w_{2,11} \] (7.323)

The 2. derivative of the relative displacement is interpolated with expression (5.155):

\[ w_{2,11} = \frac{\hat{a}}{a^2} \left( (-1 + 3z) \gamma_A + (1 + 3z) \gamma_B \right) \] (7.324)

\[ \gamma_A = \beta_A - \theta \] (7.325)

\[ \gamma_B = \beta_B - \theta \] (7.326)

\( \beta_A, \beta_B \) rotation of nodes A and B of the member
\( \theta \) 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):

\[ w_{2,11} = \frac{\hat{a}}{a^2} \left( (-1 + 3z) \beta_A + (1 + 3z) \beta_B - 6 \theta z \right) \] (7.327)

The chord rotation \( \theta \) 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):

\[ w_{2,11} = k^T w_e \quad (7.330) \]

\[ k^T = \frac{\hat{a}}{a^2} \begin{bmatrix} 0 & 6r/z & -1+3z & 0 & -6r/z & 1+3z \end{bmatrix} \]

\[ w_e^T = \begin{bmatrix} w_{1a} & w_{2a} & \beta_A & w_{1b} & w_{2b} & \beta_B \end{bmatrix} \]

The variation of the second derivative of the relative displacement \( w \) with respect to the reference coordinate \( y \), follows from (7.330):

\[ \delta w_{2,11} = k^T \delta w_e \quad (7.331) \]

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:

\[ \int \delta \varepsilon_b s_b \, dv = \delta w_e^T K_b w_e \quad (7.332) \]

\[ K_b = \frac{EJ}{a^3} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & c_4 & c_5 & 0 & -c_4 & c_5 \\ 0 & c_5 & c_6 & 0 & -c_5 & c_7 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -c_4 & -c_5 & 0 & c_4 & -c_5 \\ 0 & c_5 & c_7 & 0 & -c_5 & c_6 \end{bmatrix} \]

\[ c_4 = 12 \left( \frac{\hat{a}^2}{a} \right) \]

\[ c_5 = 6 \hat{a} \left( \frac{\hat{a}^3}{a} \right) \]

\[ c_6 = 4 \hat{a}^2 \]

\[ c_7 = 2 \hat{a}^2 \]

The member displacement vector \( w_e \) is expressed in terms of the system displacement vector \( d_e \) with equation (5.76) and substituted into (7.314). The contributions of the members are added to obtain the total system stiffness matrix \( K_e \):
\[ \delta d_s^T K_s d_s = \sum_e \delta w_e^T K_e w_e \]
\[ K_s = \sum_e T_e^T R_{esz} K_e R_{esz}^T T_e \]
\[ K_e = K_a + K_b \]
\[ T_e \quad \text{topology matrix of bar e} \]
\[ R_{esz} \quad \text{element rotation matrix support-instant space} \]

7.8 SOLUTION OF THE GENERAL EIGENVALUE PROBLEM

Concept: Consider the general eigenvalue problem (7.46):
\[ A x_i = \lambda_i B x_i \quad i = 1, ..., n \] (7.334)

\[ A \quad \text{real regular symmetric coefficient matrix of dimension n} \]
\[ B \quad \text{real regular symmetric coefficient matrix of dimension n} \]
\[ \lambda_i \quad \text{eigenvalue of the i-th eigenstate} \]
\[ x_i \quad \text{eigenvector of the i-th eigenstate, } x_i^T B x_i = \delta_{ik} \]

Assume that the eigenvalue \( \lambda_1 \) with the smallest absolute value is to be determined by vector iteration. The start vector \( u_0 \) of the iteration is chosen freely. In step \( s \) of the iteration, the vector \( u_{s-1} \) is used in equation (7.335) to compute an auxiliary vector \( y_s \) which is then normalised with respect to \( B \) in (7.336) to yield the iterated vector \( u_s \). The iteration converges to the eigenstate \( (\lambda_1, x_1) \).

Compute \( y_s \):
\[ A y_s = B u_{s-1} \] (7.335)

Scale with \( m_s \):
\[ u_s = m_s y_s \quad \text{so that} \quad u_s^T B u_s = \delta_s \in \{-1, 1\} \] (7.336)

Convergence to the eigenvector: The start vector \( u_0 \) of the iteration is a linear combination \( \sum c_k x_k \) of the eigenvectors. It is modified as follows by the iteration, since none of the eigenvalues equals null:
The product $m_1 \ldots m_s$ is eliminated between the last two equations to yield:

$$u_s = \frac{\sum_{k=1}^{n} c_k x_k}{\sqrt{\sum_{k=1}^{n} \left( \frac{c_k}{\lambda_k^s} \right)^2 \frac{\delta_k}{\delta_s}}}$$  \hspace{1cm} (7.337)

Since $\lambda_1$ 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):

$$u_s = \frac{c_1 x_1 + \sum_{k=1}^{n} \left( \frac{\lambda_1}{\lambda_k^s} \right)^s c_k x_k}{\sqrt{c_1^2 + \sum_{k=2}^{n} \left( \frac{\lambda_1}{\lambda_k^s} \right)^{2s} c_k^2 \frac{\delta_k}{\delta_s}}} \hspace{1cm} (7.338)$$

As the power $s$ increases, the root in (7.338) tends towards $c_1$. If $c_1$ differs from null, the limit value of the iterated vector $u_s$ is the eigenvalue $x_1$:

$$\lim_{s \to \infty} (u_s) = x_1 \hspace{1cm} (7.339)$$

**Rate of convergence to the eigenvector:** The rate of convergence $k_u$ of the eigenvector $x_1$ is defined as the limit of the ratio of the $B$-quadratic form of the error in the iterated vector for steps $s$ and $s-1$:

$$k_u = \lim_{s \to \infty} \frac{(u_s - x_1)^T B (u_s - x_1)}{(u_{s-1} - x_1)^T B (u_{s-1} - x_1)}$$  \hspace{1cm} (7.340)

Substitution of (7.337) into (7.340) leads to:
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:

\[ k_u = \frac{\lambda_1}{\lambda_2} \]  

(7.342)

**Special eigenvalues:** All eigenvalues of (7.334) differ from null. If one of the eigenvalues were null, the determinant of \( A \) would be null. This contradicts the assumption that \( A \) is regular. If the smallest eigenvalue \( \lambda_m \) is multiple, equation (7.338) is replaced by:

\[
\begin{align*}
\mathbf{u}_s &= \frac{c_1\mathbf{x}_1 + c_2\mathbf{x}_2 + \sum_{k=3}^{n} \left( \frac{\lambda_m}{\lambda_k} \right)^s \frac{\delta_k}{\delta_s} c_k \mathbf{x}_k}{\sqrt{c_1^2 + c_2^2 + \sum_{k=3}^{n} \left( \frac{\lambda_m}{\lambda_k} \right)^{2s} \frac{\delta^2_k}{\delta^2_s} c_k^2}} \\
\lim_{s \to \infty}(\mathbf{u}_s) &= \frac{c_1\mathbf{x}_1 + c_2\mathbf{x}_2}{c_1^2 + c_2^2} 
\end{align*}
\]  

(7.343)

The iterated vector now converges to:

\[
\lim_{s \to \infty}(\mathbf{u}_s) = \frac{c_1\mathbf{x}_1 + c_2\mathbf{x}_2}{c_1^2 + c_2^2} 
\]  

(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 \( \lambda_m \) 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:

\[
\begin{align*}
\mathbf{u}_s &= \frac{c_1\mathbf{x}_1 + (-1)^s c_2\mathbf{x}_2 + \sum_{k=3}^{n} \left( \frac{\lambda_1}{\lambda_k} \right)^s \frac{\delta_k}{\delta_s} c_k \mathbf{x}_k}{\sqrt{c_1^2 + c_2^2 + \sum_{k=3}^{n} \left( \frac{\lambda_1}{\lambda_k} \right)^{2s} \frac{\delta^2_k}{\delta^2_s} c_k^2}} \\
\lim_{s \to \infty}(\mathbf{u}_s) &= \frac{c_1\mathbf{x}_1 + (-1)^s c_2\mathbf{x}_2}{c_1^2 + c_2^2} 
\end{align*}
\]  

(7.345)

The iterated vector now converges to:

\[
\lim_{s \to \infty}(\mathbf{u}_s) = \frac{c_1\mathbf{x}_1 + (-1)^s c_2\mathbf{x}_2}{c_1^2 + c_2^2} 
\]  

(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:

\[
r(\mathbf{u}_s) = \frac{\left(\sum_{k=1}^{n} \frac{c_k}{\lambda_k} \mathbf{x}_k^T\right) \mathbf{A} \left(\sum_{m=1}^{n} \frac{c_m}{\lambda_m} \mathbf{x}_m\right)}{\left(\sum_{k=1}^{n} \frac{c_k}{\lambda_k} \mathbf{x}_k^T\right) \mathbf{B} \left(\sum_{m=1}^{n} \frac{c_m}{\lambda_m} \mathbf{x}_m\right)} = \frac{\sum_{k=1}^{n} \frac{c_k^2}{\lambda_k^2} \delta_k}{\sum_{k=1}^{n} \frac{c_k^2}{\lambda_k^2} \delta_k} \quad (7.348)
\]

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 \( \lambda_1 \):

\[
\lim_{s \to \infty} r(\mathbf{u}_s) = \lambda_1 \quad (7.349)
\]

**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:

\[
k_r = \lim_{s \to \infty} \frac{r(\mathbf{u}_s) - \lambda_1}{r(\mathbf{u}_{s-1}) - \lambda_1} \quad (7.350)
\]

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:
\[
\begin{align*}
    r(u_s) - \lambda_1 &= \sum_{k=1}^{n} \frac{c_k^2}{\lambda_k^{2s-1}} \delta_k - \lambda_1 \sum_{k=1}^{n} \frac{c_k^2}{\lambda_k^{2s}} \delta_k = \sum_{k=2}^{n} \frac{c_k^2}{\lambda_k^{2s-1}} (1 - \frac{\lambda_1}{\lambda_k}) \delta_k - \sum_{k=1}^{n} \frac{c_k^2}{\lambda_k^{2s}} \delta_k \\
    &= \frac{\lambda_1^{2s}}{\lambda_2^{2s-1}} \left( c_2^2 (1 - \frac{\lambda_1}{\lambda_k}) \delta_2 + \sum_{k=3}^{n} c_k^2 (1 - \frac{\lambda_1}{\lambda_k}) \delta_k \left( \frac{\lambda_2}{\lambda_k} \right)^{2s-1} \right) \\
    &= \frac{\lambda_1^{2s}}{\lambda_2^{2s-1}} \left( c_2^2 (1 - \frac{\lambda_1}{\lambda_k}) \delta_2 + \sum_{k=3}^{n} c_k^2 (1 - \frac{\lambda_1}{\lambda_k}) \delta_k \left( \frac{\lambda_2}{\lambda_k} \right)^{2s-1} \right) \\
    &\quad - \frac{\lambda_2^{2s-1}}{\lambda_1^{2s-1}} \left( c_1^2 \delta_1 + \sum_{k=2}^{n} c_k^2 \left( \frac{\lambda_1}{\lambda_k} \right)^{2s} \delta_k \right) \\
    \lim_{s \to \infty} (r(u_s) - \lambda_1) &= \frac{\lambda_1^{2s}}{\lambda_2^{2s-1}} \left( c_2^2 (1 - \frac{\lambda_1}{\lambda_k}) \delta_2 \right) \quad \text{(7.351)}
\end{align*}
\]

An analogous formula is derived for \( r(u_{s-1}) - \lambda_1 \):

\[
\lim_{s \to \infty} (r(u_{s-1}) - \lambda_1) = \frac{\lambda_1^{2s-2}}{\lambda_2^{2s-3}} \left( c_2^2 (1 - \frac{\lambda_1}{\lambda_k}) \delta_2 \right) \quad \text{(7.352)}
\]

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:

\[
    k_r = \lim_{s \to \infty} \frac{r(u_s) - \lambda_1}{r(u_{s-1}) - \lambda_1} = \lim_{s \to \infty} \frac{r(u_s) - \lambda_1}{r(u_{s-1}) - \lambda_1} = \left( \frac{\lambda_1}{\lambda_2} \right)^2 \quad \text{(7.353)}
\]

**Neighbouring eigenstates:** Assume that the eigenstate \( (\lambda_1, x_1) \) of (7.334) with the smallest absolute value of the eigenvalue is known. The eigenstate \( (\lambda_2, x_2) \) with the second smallest absolute value of the eigenvalue is to be determined by vector iteration. This is achieved by making the start vector \( u_0 \) B-orthogonal to \( x_1 \). Let \( \hat{u}_0 \) be an arbitrary vector that is not B-orthogonal to \( x_1 \) and let the component of \( \hat{u}_0 \) that is B-orthogonal to \( x_1 \) be \( u_0 \):

\[
\begin{align*}
    \hat{u}_0 &= c_1 x_1 + \ldots + c_n x_n \\
    u_0 &= \hat{u}_0 - c_1 x_1 = c_2 x_2 + \ldots + c_n x_n \\
    u_0 \cdot B x_1 &= 0 \quad \land \quad x_1 \cdot B x_1 = \delta_1 \quad \Rightarrow \quad c_i = \delta_i \hat{u}_0 \cdot B x_i \\
    \delta_i &\in \{-1, 1\} \quad \text{(7.354)}
\end{align*}
\]

The first step of the iteration now yields:

\[
    A y_1 = B u_0 = \sum_{k=2}^{n} c_k B x_k \quad \Rightarrow \quad y_1 = \sum_{k=2}^{n} \frac{c_k}{\lambda_k} x_k
\]

Since \( u_0 \) does not contain the eigenvector \( x_1 \), the iteration will converge to the eigenvector \( x_1 \) corresponding to the remaining eigenvalue with the smallest absolute value. Due to the round-off in the numerical solution, the iterated vector \( u_s \) must periodically be made B-orthogonal to the eigenvector \( x_1 \). 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:

$$\hat{u}_0 = c_i x_i + \ldots + c_n x_n$$

$$u_0 = \hat{u}_0 - \sum_{i=1}^{m} c_i x_i = c_{m+1} x_{m+1} + \ldots + c_n x_n$$

$$i \in \{1, \ldots, m\}: \quad u_i^T B x_i = 0 \quad \land \quad x_i^T B x_i = \delta_{ij} \quad \Rightarrow \quad c_i = \delta_{ij} u_i^T B x_i$$  \hspace{1cm} (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:

$$u_s = m_s y_s$$  \hspace{1cm} (7.356)

$$z_s = B y_s$$  \hspace{1cm} (7.357)

$$w_s = B u_s$$  \hspace{1cm} (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 $w_0$ and set $s = 1$.
2. Decompose $A$: $A = LDL^T$  \hspace{1cm} (7.359)
3. Compute $y_s$:
   $$y_s = B u_{s-1}$$
   $$y_s = L^T D^T L^T w_{s-1}$$  \hspace{1cm} (7.360)
4. Compute $z_s$:
   $$z_s = B y_s$$  \hspace{1cm} (7.361)
5. Compute $a_s$:
   $$a_s = y_s^T B y_s = y_s^T z_s$$  \hspace{1cm} (7.362)
6. Compute $r_s$:
   $$r_s = \begin{array}{l}
u_i^T A u_s \\
   u_i^T B u_s \\
   y_s^T A y_s \\
   y_s^T B y_s \\
   a_s \\
   \end{array}$$
   $$r_s = \begin{array}{l}
u_i^T A u_s \\
   u_i^T B u_s \\
   y_s^T A y_s \\
   y_s^T B y_s \\
   a_s \\
   \end{array}$$  \hspace{1cm} (7.363)
7. Compute $w_s$:
   $$w_s = m_s z_s \quad \text{with} \quad m_s = \frac{1}{\sqrt{|a_s|}}$$  \hspace{1cm} (7.364)
8. Repeat from step 3 if $|r_s - r_{s-1}| > \varepsilon$, otherwise set $u_s = m_s y_s$ and terminate.

For the determination of the second eigenstate, an initial value $\hat{w}_0 = B \hat{u}_0$ is chosen for cycle 1. The $B$-orthogonalisation of $\hat{u}_0$ is replaced by corresponding operations on $\hat{w}_0$:

$$\hat{w}_0 = B \hat{u}_0 = B (c_i x_i + \ldots + c_n x_n)$$

$$w_0 = B u_0 = B (\hat{u}_0 - c_i x_i) = \hat{w}_0 - c_i p_i$$

$$p_i := B x_i$$

$$u_i^T B x_i = w_i^T x_i = 0 \quad \land \quad x_i^T B x_i = p_i^T x_i = \delta_i \quad \Rightarrow \quad c_i = \delta_i \hat{w}_i^T x_i$$

$$w_0 = \hat{w}_0 - c_i p_i$$  \hspace{1cm} (7.365)

Due to round-off, the value of $w_{s-1}$ computed in cycle $s - 1$ is no longer $B$-orthogonal to $x_i$ and is therefore denoted by $\hat{w}_{s-1}$. The $B$-orthogonalisation of $u_{s-1}$ is replaced by corresponding operations on $\hat{w}_{s-1}$.
\( \mathbf{\hat{w}}_{s-1} = \mathbf{B}_{s-1} = \mathbf{B}(a, x_1 + \ldots + a, x_n) \)

\( \mathbf{w}_{s-1} = \mathbf{B}_{s-1} = \mathbf{B}(\mathbf{u}_{s-1} - a, x_1) = \mathbf{\hat{w}}_{s-1} - a, p_1 \)

\[
\mathbf{u}_{s-1}^T \mathbf{B} \mathbf{x}_1 = \mathbf{w}_{s-1}^T \mathbf{x}_1 = 0 \quad \land \quad \mathbf{x}_1^T \mathbf{B} \mathbf{x}_1 = p, \mathbf{x}_1 = \delta, \Rightarrow \quad a_i = \delta, \mathbf{\hat{w}}_{s-1}^T \mathbf{x}_1
\]

\( \mathbf{w}_{s-1} = \mathbf{\hat{w}}_{s-1} - a, p_1 \) \hspace{1cm} (7.366)

Step 7 is extended to include the \( \mathbf{B} \)-orthogonalisation of \( \mathbf{w}_{s-1} \) according to (7.366). If more than two eigenstates are determined, the \( \mathbf{B} \)-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 database 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:

$$\text{id} : S \rightarrow C \quad \text{with} \quad \text{id}(s) = c \quad (8.1)$$

- $\text{id}$ 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](image-url)
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.

- **nodeName**: String - persistent identifier of the node
- **nodeNumber**: int - sequence number of the node in the system
- **xUser1, xUser2**: double - node coordinates in the user coordinate system
- **theta**: double - angle from axis x1 to axis s1 (support space)
- **loadNameS1, loadNameS2, loadNameRot**: String - prescribed force in the direction s1, or null, prescribed force in the direction s2, or null, prescribed moment about axis s3, or null
- **supportNameS1, supportNameS2, supportNameRot**: String - prescribed displacement in direction s1, or null, prescribed displacement in direction s2, or null, 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.

- **barName**: String - persistent identifier of the bar
- **startNodeName**: String - persistent identifier of the start node of the bar
- **endNodeName**: String - persistent identifier of the end node of the bar
- **barArea**: double - area of the cross-section of the bar in user units
- **modulusOfElasticity**: double - 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.

- **memberName**: String - persistent identifier of the member
- **startNodeName**: String - persistent identifier of the start node of the member
- **endNodeName**: String - 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 ∈ A and b ∈ B, are called the Cartesian product of the sets A and B. The Cartesian product is denoted by A × B.

Relation: A subset of the Cartesian product A × 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 is reflexive := \( \forall a \in S: aRa \)
R is symmetric := \( \forall a \in S, \forall b \in S: aRb \Rightarrow bRa \)
R is transitive := \( \forall a \in S, \forall b \in S, \forall c \in S: aRb \land bRc \Rightarrow aRc \)
R is identitive := \( \forall a \in S, \forall b \in S: aRb \land bRa \Rightarrow a = b \)
R is linear := \( \forall a \in S, \forall b \in S: aRb \lor bRa \)

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 is reflexive \( a \sim a \) \quad (8.3)
E is symmetric \( a \sim b \Rightarrow b \sim a \)
E is transitive \( a \sim b \land b \sim c \Rightarrow a \sim c \)

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 \( \leq \):

\[ \leq \text{ is reflexive} \quad a \in S \quad \Rightarrow \quad a \leq a \quad (8.4) \]
\[ \leq \text{ is identitive} \quad a \leq b \land b \leq a \quad \Rightarrow \quad a = b \]
\[ \leq \text{ is transitive} \quad a \leq b \land b \leq c \quad \Rightarrow \quad a \leq c \]

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.

\[ \leq \text{ is linear} : \quad a,b \in S \quad \Rightarrow \quad a \leq b \lor b \leq a \quad (8.5) \]

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 < :

\[ < \text{ is antireflexive} : \quad a \in S \quad \Rightarrow \quad \neg(a < a) \quad (8.6) \]
\[ < \text{ is asymmetric} : \quad a < b \quad \Rightarrow \quad \neg(b < a) \]
\[ < \text{ is transitive} : \quad a < b \land b < c \quad \Rightarrow \quad a < c \]
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.

\[
\text{\textless \text{ is total : } a, b \in S \Rightarrow (a \neq b \Rightarrow a \text{ \textless } b \lor b \text{ \textless } a)}
\]

(8.7)

**Mapping:** A relation \( f \subseteq A \times Z \) is called a mapping if it is left total and right unique:

\[
f : A \rightarrow B \quad \text{with} \quad f(a) = b
\]

(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 \( \forall (a \in A) \forall (b \in B) \forall (c \in C): f(a) = b \land f(a) = c \Rightarrow b = c \)

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.

```java
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.

```java
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:

```java
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 nextIndex() 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 runtime 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.

![Object table for mapping of identifiers](image)

**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.

![Diagram](image)

**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:

```java
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:

```java
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:

```java
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:

```java
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](image)

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
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:
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

trial displacement at end of step


displacement increment in step

static double uSupportIncrement[4] : support space coordinates
static double uGlobalIncrement[4] : global space coordinates

displacement correction

static double bsSupport[4] : support space coordinates
static double bsGlobal[4] : global 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 kInstant[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

**State displacement at start of step**

- `static double uSupport[6]` : support space coordinates
- `static double uGlobal[6]` : global space coordinates
- `static double wlInstant[6]` : instant space coordinates

**Trial displacement at end of step**


---

297
dispersion 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

dispersion 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: \( x = a + R_{xy} y \)
- double referenceLength
- double cosInstantState: instant configuration at start of step
- double sinInstantState: \( x = b + R_{xz} z \)
- double instantLengthState
- double cosLocalState: increment of the configuration
- double sinLocalState: \( y = c + R_{xz} z \)
- 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 : \( x = b + R_{xz} z \)
double instantLengthTrial
double cosLocalTrial : increment of the configuration
double sinLocalTrial : \( y = c + R_{yz} z \)
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

299
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

![Class diagram]

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**

```java
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**

```java
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)
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

```c
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.
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 $R_{esc}$ 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 $R_{esc}$ 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, $\varepsilon$ small):

- The member was stiff in the iteration cycle:
  \[ s < \varepsilon \land t \geq -\varepsilon : \text{ condition has changed} \]
  \[ s \leq \varepsilon \land t < -\varepsilon : \text{ no change} \]
  \[ s \geq \varepsilon \land t \geq \varepsilon : \text{ no change} \]
  \[ s \geq \varepsilon \land t < \varepsilon : \text{ compute new segment length} \]

- The member was flexible in the iteration cycle:
  \[ s < \varepsilon \land t \leq -\varepsilon : \text{ no change} \]
  \[ s < \varepsilon \land t > -\varepsilon : \text{ compute new segment length} \]
  \[ s \geq \varepsilon \land t \leq \varepsilon : \text{ no change} \]
  \[ s \geq \varepsilon \land t > \varepsilon : \text{ condition has changed} \]

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 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:
  \[ s \leq \varepsilon \land t \geq -\varepsilon : \text{ stiff} \]
  \[ s \leq \varepsilon \land t < -\varepsilon : \text{ flexible} \]
  \[ s > \varepsilon \land t \geq \varepsilon : \text{ stiff} \]
  \[ s > \varepsilon \land t < \varepsilon : \text{ flexible} \]

- The member was flexible in the iteration cycle:
  \[ s \leq -\varepsilon \land t > -\varepsilon : \text{ stiff} \]
  \[ s \leq -\varepsilon \land t \leq -\varepsilon : \text{ flexible} \]
  \[ s > -\varepsilon \land t > \varepsilon : \text{ stiff} \]
  \[ s > -\varepsilon \land t \leq \varepsilon : \text{ flexible} \]

(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.

![Class structure of the graphical user interface](image)

**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:

\[
\begin{align*}
x_{\text{Pixel}} &= x_{\text{OriginPixel}} + x_{\text{Milli}} \times \text{milliToPixel} \\
y_{\text{Pixel}} &= y_{\text{OriginPixel}} - y_{\text{Milli}} \times \text{milliToPixel}
\end{align*}
\]  \hspace{1cm} (8.13)\hspace{1cm} (8.14)

The inverse transformation is given by:

\[
\begin{align*}
x_{\text{Milli}} &= (x_{\text{Pixel}} - x_{\text{OriginPixel}}) \div \text{milliToPixel} \\
y_{\text{Milli}} &= (-y_{\text{Pixel}} + y_{\text{OriginPixel}}) \div \text{milliToPixel}
\end{align*}
\]  \hspace{1cm} (8.15)\hspace{1cm} (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 \((x_{\text{User}}, y_{\text{User}})\) of point \(P\) in the user coordinate system to its coordinates \((x_{\text{Milli}}, y_{\text{Milli}})\) in the millimetre coordinate system is:

\[
x_{\text{Milli}} = x_{\text{User}} \times \text{userToMilli} \quad (8.17)
\]
\[
y_{\text{Milli}} = y_{\text{User}} \times \text{userToMilli} \quad (8.18)
\]

The transformation from user to pixel coordinates is obtained by substitution of (8.17) and (8.18) into (8.13) and (8.14):

\[
x_{\text{Pixel}} = x_{\text{OriginPixel}} + x_{\text{User}} \times \text{userToPixel} \quad (8.19)
\]
\[
y_{\text{Pixel}} = y_{\text{OriginPixel}} - y_{\text{User}} \times \text{userToPixel} \quad (8.20)
\]
\[
\text{userToPixel} = \text{userToMilli} \times \text{MilliToPixel} \quad (8.21)
\]

The inverse transformation is given by:

\[
x_{\text{User}} = \left( x_{\text{Pixel}} - x_{\text{OriginPixel}} \right) / \text{userToPixel} \quad (8.22)
\]
\[
y_{\text{User}} = \left( -y_{\text{Pixel}} + y_{\text{OriginPixel}} \right) / \text{userToPixel} \quad (8.23)
\]

### 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 \(\text{userToPixel}\) in (8.21).

The conventions are implemented with symbolic constants \(_{\text{MESHES\_PER\_BLOCK}}, _{\text{MARGIN\_MILLI}}\) and \(_{\text{RIM\_MILLI}}\). The parameters of the grid are:

\[
\text{widthPixel} = \text{width of panel} \quad (8.24)
\]
\[
\text{heightPixel} = \text{height of panel}
\]
\[
\text{blockSizePixel} = \text{meshesPerBlock} \times \text{milliToPixel}
\]
\[
\text{blockSizeUser} = \text{blockSizePixel} \times \text{pixelToUser}
\]
\[
\text{marginPixel} = _{\text{MARGIN\_MILLI}} \times \text{milliToPixel}
\]
\[
\text{rimPixel} = _{\text{RIM\_MILLI}} \times \text{milliToPixel}
\]
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.

![Flow diagram of the paint algorithm for the graphics panel](image)

**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.

start

mouse click outside grid

yes

remove marking of components

no

register mouse coordinates

switch on component type

Node

existing node picked

yes

mark node, open node editor

no

define new node at location

mark node, open node editor

Element: first operation of the transaction

existing element picked

yes

mark element, open element editor

no

existing node picked

yes

warning

no

existing node picked

mark the node and use it as start node of a new element

Element: second operation of the transaction

existing node picked

no

warning

yes

mark the node and use it as end node of a new element

define a new element

mark element, open element editor
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 und 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](image)

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:

![Node Editor]

<table>
<thead>
<tr>
<th>D</th>
<th>Node Editor</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>node name</td>
<td>B</td>
<td></td>
</tr>
<tr>
<td>number</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>x1User</td>
<td>74,000</td>
<td></td>
</tr>
<tr>
<td>x2User</td>
<td>52,000</td>
<td></td>
</tr>
<tr>
<td>theta</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>dimension</td>
<td>y1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>y2</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>rot</td>
<td>0</td>
</tr>
</tbody>
</table>

![Load Editor]

<table>
<thead>
<tr>
<th>D</th>
<th>Load Editor</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>node name</td>
<td>B</td>
<td></td>
</tr>
<tr>
<td>force</td>
<td>y1</td>
<td></td>
</tr>
<tr>
<td>force</td>
<td>y2</td>
<td>-100,000</td>
</tr>
<tr>
<td>moment</td>
<td>m</td>
<td></td>
</tr>
<tr>
<td>local index</td>
<td>y1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>y2</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>m</td>
<td>0</td>
</tr>
</tbody>
</table>

![Element Editor]

<table>
<thead>
<tr>
<th>D</th>
<th>Element Editor</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>member</td>
<td>bar</td>
<td></td>
</tr>
<tr>
<td></td>
<td>area</td>
<td>6.00</td>
</tr>
<tr>
<td></td>
<td>moment of inertia</td>
<td>4.58</td>
</tr>
<tr>
<td></td>
<td>E-modulus</td>
<td>200000.0</td>
</tr>
<tr>
<td>start load</td>
<td>p1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>p2</td>
<td></td>
</tr>
<tr>
<td>end load</td>
<td>p1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>p2</td>
<td></td>
</tr>
<tr>
<td>start node</td>
<td>B</td>
<td></td>
</tr>
<tr>
<td>local index</td>
<td>y1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>y2</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>rot</td>
<td>0</td>
</tr>
</tbody>
</table>

![Support Editor]

<table>
<thead>
<tr>
<th>D</th>
<th>Support Editor</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>node name</td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>displacement</td>
<td>y1</td>
<td>0.000</td>
</tr>
<tr>
<td>displacement</td>
<td>y2</td>
<td>0</td>
</tr>
<tr>
<td>rotation</td>
<td>a</td>
<td></td>
</tr>
<tr>
<td>local index</td>
<td>y1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>y2</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>a</td>
<td>0</td>
</tr>
</tbody>
</table>
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**

```java
double xPixelToMilli(double xPixel)
double yPixelToMilli(double yPixel)
double sPixelToMilli(double sPixel)
double xPixelToUser(double xPixel)
double yPixelToUser(double yPixel)
double sPixelToUser(double sPixel)
int xMilliToPixel(int xMilli)
int yMilliToPixel(int yMilli)
int sMilliToPixel(int sMilli)
double xMilliToUser(double xMilli)
double yMilliToUser(double yMilli)
double sMilliToUser(double sMilli)
int xUserToPixel(int xUser)
int yUserToPixel(int yUser)
int sUserToPixel(int sUser)
double xUserToMilli(double xUser)
double yUserToMilli(double yUser)
double sUserToMilli(double sUser)
int xPixelToPixelGrid(int xPixel)
int yPixelToPixelGrid(int yPixel)
double xPixelToMilliGrid(double xPixel)
double yPixelToMilliGrid(double yPixel)
double xPixelToUserGrid(double xPixel)
double yPixelToUserGrid(double 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)

(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 cm². The modulus of elasticity of the bar material is 2.0 × 10⁸ kN/m². The supports A and C of the truss are pinned. The truss carries a vertical pattern load P = (0.0, –10.0) kN at the apex B.

![Figure 9.1 Shallow 2-bar truss](image)

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 form 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 $f_s = 10^{-5}$ and LFI = 0.50

<table>
<thead>
<tr>
<th>$d_2$</th>
<th>$P_2$</th>
<th>$P_{2,exact}$</th>
<th>$d_2$</th>
<th>$P_2$</th>
<th>$P_{2,exact}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>-0.0047006</td>
<td>-5.0000000</td>
<td>-4.9999532</td>
<td>-0.0193547</td>
<td>22.2375853</td>
<td>22.2376230</td>
</tr>
<tr>
<td>-0.0094016</td>
<td>-9.5227726</td>
<td>-9.5227570</td>
<td>-0.2055045</td>
<td>26.7205815</td>
<td>26.7206247</td>
</tr>
<tr>
<td>-0.0142628</td>
<td>-13.7137556</td>
<td>-13.7137352</td>
<td>-0.2157549</td>
<td>29.6291749</td>
<td>29.6291622</td>
</tr>
<tr>
<td>-0.0193008</td>
<td>-17.5540311</td>
<td>-17.5539996</td>
<td>-0.2248247</td>
<td>31.3508696</td>
<td>31.3509071</td>
</tr>
<tr>
<td>-0.0245358</td>
<td>-21.0219088</td>
<td>-21.0218573</td>
<td>-0.2290237</td>
<td>31.8447456</td>
<td>31.8446942</td>
</tr>
<tr>
<td>-0.0299917</td>
<td>-24.0922555</td>
<td>-24.0921647</td>
<td>-0.2369659</td>
<td>32.2029996</td>
<td>32.2029996</td>
</tr>
<tr>
<td>-0.0356956</td>
<td>-26.7342474</td>
<td>-26.7342521</td>
<td>-0.2366033</td>
<td>32.2038509</td>
<td>32.2038509</td>
</tr>
<tr>
<td>-0.0416839</td>
<td>-28.9133453</td>
<td>-28.9133854</td>
<td>-0.2366025</td>
<td>32.2038509</td>
<td>32.2038509</td>
</tr>
<tr>
<td>-0.0480034</td>
<td>-30.5871208</td>
<td>-30.5871030</td>
<td>-0.2438817</td>
<td>31.8530040</td>
<td>31.8530220</td>
</tr>
<tr>
<td>-0.0547099</td>
<td>-31.7014158</td>
<td>-31.7014845</td>
<td>-0.2506764</td>
<td>30.8589739</td>
<td>30.8590042</td>
</tr>
<tr>
<td>-0.0582342</td>
<td>-32.0287752</td>
<td>-32.0287299</td>
<td>-0.2570672</td>
<td>29.2938762</td>
<td>29.2939680</td>
</tr>
<tr>
<td>-0.0632523</td>
<td>-32.2037151</td>
<td>-32.2037151</td>
<td>-0.2631175</td>
<td>27.2135058</td>
<td>27.2135954</td>
</tr>
<tr>
<td>-0.0633973</td>
<td>-32.2038509</td>
<td>-32.2038509</td>
<td>-0.2688756</td>
<td>24.6621246</td>
<td>24.6621439</td>
</tr>
<tr>
<td>-0.0633975</td>
<td>-32.2038509</td>
<td>-32.2038509</td>
<td>-0.2743780</td>
<td>21.6766253</td>
<td>21.6766348</td>
</tr>
<tr>
<td>-0.0712745</td>
<td>-31.8163591</td>
<td>-31.8163309</td>
<td>-0.2796535</td>
<td>18.2884765</td>
<td>18.2885601</td>
</tr>
<tr>
<td>-0.0799137</td>
<td>-30.5586638</td>
<td>-30.5585885</td>
<td>-0.2847287</td>
<td>14.5228191</td>
<td>14.5228621</td>
</tr>
<tr>
<td>-0.0895659</td>
<td>-28.2375122</td>
<td>-28.2375482</td>
<td>-0.2896237</td>
<td>10.4020950</td>
<td>10.4021229</td>
</tr>
<tr>
<td>-0.1006482</td>
<td>-24.5479175</td>
<td>-24.5479321</td>
<td>-0.2943556</td>
<td>5.9458086</td>
<td>5.9458270</td>
</tr>
<tr>
<td>-0.1139180</td>
<td>-18.9615178</td>
<td>-18.9615274</td>
<td>-0.2989390</td>
<td>1.1710829</td>
<td>1.1710956</td>
</tr>
<tr>
<td>-0.1308569</td>
<td>-10.5038535</td>
<td>-10.5038606</td>
<td>-0.3005221</td>
<td>-0.5855415</td>
<td>-0.5855414</td>
</tr>
<tr>
<td>-0.1594532</td>
<td>5.2519267</td>
<td>5.2519267</td>
<td>-0.3049245</td>
<td>-5.7671632</td>
<td>-5.7671547</td>
</tr>
<tr>
<td>-0.1788112</td>
<td>15.4776120</td>
<td>15.4776290</td>
<td>-0.3092055</td>
<td>-11.2340669</td>
<td>-11.2340610</td>
</tr>
</tbody>
</table>
Table 9.2  Error in the deflection – load curve for $f_1 = 10^{-3}$ and LFI = 0.50

<table>
<thead>
<tr>
<th>$d_2$</th>
<th>$P_2$</th>
<th>$P_{2,\text{exact}}$</th>
<th>$d_2$</th>
<th>$P_2$</th>
<th>$P_{2,\text{exact}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>-0.0047006</td>
<td>-5.000000</td>
<td>-4.999953</td>
<td>-0.2060584</td>
<td>26.905634</td>
<td>26.901391</td>
</tr>
<tr>
<td>-0.0093962</td>
<td>-9.517451</td>
<td>-9.517818</td>
<td>-0.2164001</td>
<td>29.785790</td>
<td>29.779542</td>
</tr>
<tr>
<td>-0.0142495</td>
<td>-13.702475</td>
<td>-13.702971</td>
<td>-0.2253313</td>
<td>31.418531</td>
<td>31.421104</td>
</tr>
<tr>
<td>-0.0192765</td>
<td>-17.535992</td>
<td>-17.536656</td>
<td>-0.2333183</td>
<td>32.127606</td>
<td>32.135259</td>
</tr>
<tr>
<td>-0.0244952</td>
<td>-20.996062</td>
<td>-20.996979</td>
<td>-0.2368832</td>
<td>32.203809</td>
<td>32.203809</td>
</tr>
<tr>
<td>-0.0299262</td>
<td>-24.057164</td>
<td>-24.058487</td>
<td>-0.2366026</td>
<td>32.203851</td>
<td>32.203851</td>
</tr>
<tr>
<td>-0.0355919</td>
<td>-26.689184</td>
<td>-26.691212</td>
<td>-0.2366025</td>
<td>32.203851</td>
<td>32.203851</td>
</tr>
<tr>
<td>-0.0415130</td>
<td>-28.855972</td>
<td>-28.859346</td>
<td>-0.2438820</td>
<td>31.853474</td>
<td>31.852986</td>
</tr>
<tr>
<td>-0.0476936</td>
<td>-30.513130</td>
<td>-30.519475</td>
<td>-0.2506769</td>
<td>30.859748</td>
<td>30.858890</td>
</tr>
<tr>
<td>-0.0544278</td>
<td>-31.670737</td>
<td>-31.667776</td>
<td>-0.2570125</td>
<td>29.306697</td>
<td>29.310179</td>
</tr>
<tr>
<td>-0.0615136</td>
<td>-32.177417</td>
<td>-32.180828</td>
<td>-0.2629638</td>
<td>27.266756</td>
<td>27.273935</td>
</tr>
<tr>
<td>-0.0633712</td>
<td>-32.203846</td>
<td>-32.203846</td>
<td>-0.2686891</td>
<td>24.750812</td>
<td>24.753820</td>
</tr>
<tr>
<td>-0.0633975</td>
<td>-32.203851</td>
<td>-32.203851</td>
<td>-0.2741719</td>
<td>21.796390</td>
<td>21.798426</td>
</tr>
<tr>
<td>-0.0712739</td>
<td>-31.817049</td>
<td>-31.816389</td>
<td>-0.2794388</td>
<td>18.435434</td>
<td>18.436772</td>
</tr>
<tr>
<td>-0.0799120</td>
<td>-30.560521</td>
<td>-30.558912</td>
<td>-0.2845101</td>
<td>14.694660</td>
<td>14.695600</td>
</tr>
<tr>
<td>-0.0893930</td>
<td>-28.280630</td>
<td>-28.286871</td>
<td>-0.2894040</td>
<td>10.596934</td>
<td>10.597620</td>
</tr>
<tr>
<td>-0.1003747</td>
<td>-24.648995</td>
<td>-24.650664</td>
<td>-0.2941365</td>
<td>6.162069</td>
<td>6.162585</td>
</tr>
<tr>
<td>-0.1134957</td>
<td>-19.154154</td>
<td>-19.155690</td>
<td>-0.2987217</td>
<td>1.407422</td>
<td>1.407821</td>
</tr>
<tr>
<td>-0.1301819</td>
<td>-10.860277</td>
<td>-10.861305</td>
<td>-0.3006254</td>
<td>-0.703711</td>
<td>-0.702077</td>
</tr>
<tr>
<td>-0.1597767</td>
<td>5.430139</td>
<td>5.430139</td>
<td>-0.3050225</td>
<td>-5.887733</td>
<td>-5.887504</td>
</tr>
<tr>
<td>-0.1792233</td>
<td>15.683571</td>
<td>15.681688</td>
<td>-0.3092979</td>
<td>-11.357139</td>
<td>-11.356903</td>
</tr>
<tr>
<td>-0.1940294</td>
<td>22.446679</td>
<td>22.443041</td>
<td>-0.3134619</td>
<td>-17.100113</td>
<td>-17.099921</td>
</tr>
</tbody>
</table>
Table 9.3  Error in the deflection – load curve for $f_e = 10^{-2}$ and LFI = 0.50

<table>
<thead>
<tr>
<th>$d_2$</th>
<th>$P_2$</th>
<th>$P_2,\text{exact}$</th>
<th>$d_2$</th>
<th>$P_2$</th>
<th>$P_2,\text{exact}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>-0.004681</td>
<td>-5.0000</td>
<td>-4.9800</td>
<td>-0.187385</td>
<td>19.5483</td>
<td>19.5577</td>
</tr>
<tr>
<td>-0.009552</td>
<td>-9.6841</td>
<td>-9.6593</td>
<td>-0.199762</td>
<td>24.6903</td>
<td>24.7018</td>
</tr>
<tr>
<td>-0.014622</td>
<td>-14.0333</td>
<td>-14.0042</td>
<td>-0.210188</td>
<td>28.1531</td>
<td>28.1668</td>
</tr>
<tr>
<td>-0.019921</td>
<td>-18.0264</td>
<td>-17.9920</td>
<td>-0.219155</td>
<td>30.3567</td>
<td>30.3749</td>
</tr>
<tr>
<td>-0.025488</td>
<td>-21.6388</td>
<td>-21.5970</td>
<td>-0.226667</td>
<td>31.5642</td>
<td>31.5924</td>
</tr>
<tr>
<td>-0.031381</td>
<td>-24.8406</td>
<td>-24.7882</td>
<td>-0.230378</td>
<td>31.9649</td>
<td>31.9603</td>
</tr>
<tr>
<td>-0.037689</td>
<td>-27.5948</td>
<td>-27.5258</td>
<td>-0.236834</td>
<td>32.2035</td>
<td>32.2035</td>
</tr>
<tr>
<td>-0.044569</td>
<td>-29.8531</td>
<td>-29.7551</td>
<td>-0.236603</td>
<td>32.2039</td>
<td>32.2039</td>
</tr>
<tr>
<td>-0.051271</td>
<td>-31.2194</td>
<td>-31.2124</td>
<td>-0.236603</td>
<td>32.2039</td>
<td>32.2039</td>
</tr>
<tr>
<td>-0.057546</td>
<td>-31.9554</td>
<td>-31.9783</td>
<td>-0.243872</td>
<td>31.8407</td>
<td>31.8539</td>
</tr>
<tr>
<td>-0.061915</td>
<td>-32.2328</td>
<td>-32.1896</td>
<td>-0.250656</td>
<td>30.8312</td>
<td>30.8631</td>
</tr>
<tr>
<td>-0.063349</td>
<td>-32.2038</td>
<td>-32.2038</td>
<td>-0.256930</td>
<td>29.3249</td>
<td>29.3343</td>
</tr>
<tr>
<td>-0.063397</td>
<td>-32.2039</td>
<td>-32.2039</td>
<td>-0.262111</td>
<td>27.5046</td>
<td>27.6015</td>
</tr>
<tr>
<td>-0.071260</td>
<td>-31.8333</td>
<td>-31.8177</td>
<td>-0.267520</td>
<td>25.2620</td>
<td>25.3144</td>
</tr>
<tr>
<td>-0.079869</td>
<td>-30.6084</td>
<td>-30.5672</td>
<td>-0.272707</td>
<td>22.5925</td>
<td>22.6412</td>
</tr>
<tr>
<td>-0.088603</td>
<td>-28.4685</td>
<td>-28.5089</td>
<td>-0.277760</td>
<td>19.5259</td>
<td>19.5655</td>
</tr>
<tr>
<td>-0.099198</td>
<td>-25.0780</td>
<td>-25.0862</td>
<td>-0.282663</td>
<td>16.0829</td>
<td>16.1168</td>
</tr>
<tr>
<td>-0.111689</td>
<td>-19.9657</td>
<td>-19.9754</td>
<td>-0.287426</td>
<td>12.2835</td>
<td>12.3126</td>
</tr>
<tr>
<td>-0.127343</td>
<td>-12.3428</td>
<td>-12.3493</td>
<td>-0.292056</td>
<td>8.1454</td>
<td>8.1709</td>
</tr>
<tr>
<td>-0.148060</td>
<td>-1.0806</td>
<td>-1.0819</td>
<td>-0.296559</td>
<td>3.6848</td>
<td>3.7073</td>
</tr>
<tr>
<td>-0.150969</td>
<td>0.5403</td>
<td>0.5403</td>
<td>-0.301597</td>
<td>-1.8424</td>
<td>-1.8105</td>
</tr>
<tr>
<td>-0.171960</td>
<td>11.9841</td>
<td>11.9863</td>
<td>-0.305866</td>
<td>-6.9489</td>
<td>-6.9323</td>
</tr>
</tbody>
</table>
Table 9.4  Error in the deflection – load curve for $f_e = 10^{-3}$ and LFI = 0.20

<table>
<thead>
<tr>
<th>$d_2$</th>
<th>$P_2$</th>
<th>$P_{2,exact}$</th>
<th>$d_2$</th>
<th>$P_2$</th>
<th>$P_{2,exact}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>-0.0036743</td>
<td>-3.951010</td>
<td>-3.949613</td>
<td>-0.1466017</td>
<td>-1.894516</td>
<td>-1.894531</td>
</tr>
<tr>
<td>-0.0074482</td>
<td>-7.701991</td>
<td>-7.700436</td>
<td>-0.1606496</td>
<td>5.910236</td>
<td>5.910246</td>
</tr>
<tr>
<td>-0.0113291</td>
<td>-11.244420</td>
<td>-11.242676</td>
<td>-0.1761411</td>
<td>14.138101</td>
<td>14.138293</td>
</tr>
<tr>
<td>-0.0153269</td>
<td>-14.568862</td>
<td>-14.566886</td>
<td>-0.1885667</td>
<td>20.089630</td>
<td>20.089910</td>
</tr>
<tr>
<td>-0.0194534</td>
<td>-17.664746</td>
<td>-17.662481</td>
<td>-0.1989321</td>
<td>24.388916</td>
<td>24.389237</td>
</tr>
<tr>
<td>-0.0237230</td>
<td>-20.520126</td>
<td>-20.517491</td>
<td>-0.2079111</td>
<td>27.486945</td>
<td>27.487330</td>
</tr>
<tr>
<td>-0.0281535</td>
<td>-23.121355</td>
<td>-23.118232</td>
<td>-0.2158945</td>
<td>29.661523</td>
<td>29.662044</td>
</tr>
<tr>
<td>-0.0327677</td>
<td>-25.452631</td>
<td>-25.448836</td>
<td>-0.2231089</td>
<td>31.091176</td>
<td>31.092026</td>
</tr>
<tr>
<td>-0.0375080</td>
<td>-27.456763</td>
<td>-27.456652</td>
<td>-0.2296362</td>
<td>31.897748</td>
<td>31.899666</td>
</tr>
<tr>
<td>-0.0423403</td>
<td>-29.116401</td>
<td>-29.116534</td>
<td>-0.2345360</td>
<td>32.174753</td>
<td>32.176564</td>
</tr>
<tr>
<td>-0.0477711</td>
<td>-30.448640</td>
<td>-30.448884</td>
<td>-0.2366025</td>
<td>32.203851</td>
<td>32.203851</td>
</tr>
<tr>
<td>-0.0526318</td>
<td>-31.425900</td>
<td>-31.426441</td>
<td>-0.2424661</td>
<td>31.975402</td>
<td>31.977412</td>
</tr>
<tr>
<td>-0.0580546</td>
<td>-32.014457</td>
<td>-32.016210</td>
<td>-0.2478657</td>
<td>31.350431</td>
<td>31.351362</td>
</tr>
<tr>
<td>-0.0633691</td>
<td>-32.203846</td>
<td>-32.203846</td>
<td>-0.2530972</td>
<td>30.339821</td>
<td>30.340234</td>
</tr>
<tr>
<td>-0.0664677</td>
<td>-32.144927</td>
<td>-32.143854</td>
<td>-0.2581128</td>
<td>28.976820</td>
<td>28.977026</td>
</tr>
<tr>
<td>-0.0728562</td>
<td>-31.647157</td>
<td>-31.648592</td>
<td>-0.2629259</td>
<td>27.288628</td>
<td>27.288746</td>
</tr>
<tr>
<td>-0.0796829</td>
<td>-30.601809</td>
<td>-30.602741</td>
<td>-0.2674723</td>
<td>25.332705</td>
<td>25.336918</td>
</tr>
<tr>
<td>-0.0872010</td>
<td>-28.888265</td>
<td>-28.888804</td>
<td>-0.2718143</td>
<td>23.132654</td>
<td>23.135865</td>
</tr>
<tr>
<td>-0.0955624</td>
<td>-26.364916</td>
<td>-26.365289</td>
<td>-0.2760297</td>
<td>20.669512</td>
<td>20.672243</td>
</tr>
<tr>
<td>-0.1050548</td>
<td>-22.818764</td>
<td>-22.819056</td>
<td>-0.2801265</td>
<td>17.956543</td>
<td>17.958904</td>
</tr>
<tr>
<td>-0.1161694</td>
<td>-17.910134</td>
<td>-17.910375</td>
<td>-0.2841119</td>
<td>15.006667</td>
<td>15.007734</td>
</tr>
<tr>
<td>-0.1297377</td>
<td>-11.095641</td>
<td>-11.095804</td>
<td>-0.2879928</td>
<td>11.827636</td>
<td>11.829465</td>
</tr>
</tbody>
</table>
Table 9.5  Error in the deflection – load curve for $f_e = 10^{-3}$ and LFI = 0.80

<table>
<thead>
<tr>
<th>$d_2$</th>
<th>$P_2$</th>
<th>$P_{2,exact}$</th>
<th>$d_2$</th>
<th>$P_2$</th>
<th>$P_{2,exact}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>-0.0077629</td>
<td>-8.000000</td>
<td>-7.999458</td>
<td>-0.2218347</td>
<td>30.874172</td>
<td>30.879030</td>
</tr>
<tr>
<td>-0.0155048</td>
<td>-14.704563</td>
<td>-14.707314</td>
<td>-0.2288770</td>
<td>31.838398</td>
<td>31.830867</td>
</tr>
<tr>
<td>-0.0236894</td>
<td>-20.491710</td>
<td>-20.496365</td>
<td>-0.2369643</td>
<td>32.203007</td>
<td>32.203007</td>
</tr>
<tr>
<td>-0.0323853</td>
<td>-25.261892</td>
<td>-25.270022</td>
<td>-0.2366033</td>
<td>32.203851</td>
<td>32.203851</td>
</tr>
<tr>
<td>-0.0416439</td>
<td>-28.884407</td>
<td>-28.900773</td>
<td>-0.2366025</td>
<td>32.203851</td>
<td>32.203851</td>
</tr>
<tr>
<td>-0.0520261</td>
<td>-31.344658</td>
<td>-31.334562</td>
<td>-0.2480133</td>
<td>31.331510</td>
<td>31.328396</td>
</tr>
<tr>
<td>-0.0563917</td>
<td>-31.872545</td>
<td>-31.879209</td>
<td>-0.2583082</td>
<td>28.920885</td>
<td>28.915859</td>
</tr>
<tr>
<td>-0.0631252</td>
<td>-32.203373</td>
<td>-32.203373</td>
<td>-0.2676666</td>
<td>25.237057</td>
<td>25.245557</td>
</tr>
<tr>
<td>-0.063970</td>
<td>-32.203851</td>
<td>-32.203851</td>
<td>-0.2763080</td>
<td>20.481967</td>
<td>20.498006</td>
</tr>
<tr>
<td>-0.0633975</td>
<td>-32.203851</td>
<td>-32.203851</td>
<td>-0.2844878</td>
<td>14.706646</td>
<td>14.713181</td>
</tr>
<tr>
<td>-0.0763495</td>
<td>-31.181705</td>
<td>-31.177236</td>
<td>-0.2922204</td>
<td>8.010891</td>
<td>8.015228</td>
</tr>
<tr>
<td>-0.0916376</td>
<td>-27.638804</td>
<td>-27.625626</td>
<td>-0.2995699</td>
<td>0.475014</td>
<td>0.477798</td>
</tr>
<tr>
<td>-0.1109393</td>
<td>-20.305816</td>
<td>-20.310138</td>
<td>-0.3002124</td>
<td>-0.237507</td>
<td>-0.237483</td>
</tr>
<tr>
<td>-0.1390535</td>
<td>-6.071606</td>
<td>-6.073320</td>
<td>-0.3071947</td>
<td>-8.615027</td>
<td>-8.612944</td>
</tr>
<tr>
<td>-0.1554498</td>
<td>3.035803</td>
<td>3.035803</td>
<td>-0.3138857</td>
<td>-17.709184</td>
<td>-17.707838</td>
</tr>
<tr>
<td>-0.1859158</td>
<td>18.88470</td>
<td>18.884835</td>
<td>-0.3203146</td>
<td>-27.475073</td>
<td>-27.474056</td>
</tr>
<tr>
<td>-0.2062487</td>
<td>26.969757</td>
<td>26.962906</td>
<td>-0.3265085</td>
<td>-37.873929</td>
<td>-37.873149</td>
</tr>
</tbody>
</table>
Table 9.6  Error in the displacement – reaction curve for $f_e = 10^{-3}$ and LFI = 0.50

<table>
<thead>
<tr>
<th>$d_2$</th>
<th>$R_t$</th>
<th>$R_{t,exact}$</th>
<th>$d_2$</th>
<th>$R_t$</th>
<th>$R_{t,exact}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>-0.0047006</td>
<td>34.411489</td>
<td>34.411383</td>
<td>-0.02060584</td>
<td>479.908975</td>
<td>479.881782</td>
</tr>
<tr>
<td>-0.0093962</td>
<td>67.691615</td>
<td>67.692475</td>
<td>-0.2144001</td>
<td>448.519216</td>
<td>448.486105</td>
</tr>
<tr>
<td>-0.0142495</td>
<td>100.941143</td>
<td>100.942348</td>
<td>-0.2253313</td>
<td>417.093846</td>
<td>417.105700</td>
</tr>
<tr>
<td>-0.0192765</td>
<td>134.149094</td>
<td>134.150765</td>
<td>-0.233183</td>
<td>385.660953</td>
<td>385.692559</td>
</tr>
<tr>
<td>-0.0244952</td>
<td>167.297846</td>
<td>167.300247</td>
<td>-0.236632</td>
<td>371.511361</td>
<td>371.511361</td>
</tr>
<tr>
<td>-0.0299262</td>
<td>200.360567</td>
<td>200.364187</td>
<td>-0.2366026</td>
<td>371.857878</td>
<td>371.857878</td>
</tr>
<tr>
<td>-0.0355919</td>
<td>233.292363</td>
<td>233.298175</td>
<td>-0.2366025</td>
<td>371.858040</td>
<td>371.858040</td>
</tr>
<tr>
<td>-0.0415130</td>
<td>266.006466</td>
<td>266.016646</td>
<td>-0.2438820</td>
<td>339.289182</td>
<td>339.287406</td>
</tr>
<tr>
<td>-0.0476936</td>
<td>298.294122</td>
<td>298.314388</td>
<td>-0.2506769</td>
<td>306.515214</td>
<td>306.513958</td>
</tr>
<tr>
<td>-0.0544278</td>
<td>331.359497</td>
<td>331.349408</td>
<td>-0.2570125</td>
<td>273.883304</td>
<td>273.894050</td>
</tr>
<tr>
<td>-0.0615136</td>
<td>363.668624</td>
<td>363.681141</td>
<td>-0.2629638</td>
<td>241.418109</td>
<td>241.439668</td>
</tr>
<tr>
<td>-0.0633712</td>
<td>371.745224</td>
<td>371.745224</td>
<td>-0.2686891</td>
<td>208.551570</td>
<td>208.560157</td>
</tr>
<tr>
<td>-0.0633975</td>
<td>371.858023</td>
<td>371.858023</td>
<td>-0.2741719</td>
<td>175.544903</td>
<td>175.550449</td>
</tr>
<tr>
<td>-0.0712739</td>
<td>404.143072</td>
<td>404.140365</td>
<td>-0.2794388</td>
<td>142.432667</td>
<td>142.436160</td>
</tr>
<tr>
<td>-0.0799120</td>
<td>436.012238</td>
<td>436.007905</td>
<td>-0.2845101</td>
<td>109.250411</td>
<td>109.252769</td>
</tr>
<tr>
<td>-0.0893930</td>
<td>466.693689</td>
<td>466.726330</td>
<td>-0.2894040</td>
<td>76.019268</td>
<td>76.020926</td>
</tr>
<tr>
<td>-0.1003747</td>
<td>496.725807</td>
<td>496.736246</td>
<td>-0.2941365</td>
<td>42.753983</td>
<td>42.755189</td>
</tr>
<tr>
<td>-0.1134957</td>
<td>524.739619</td>
<td>524.752153</td>
<td>-0.2987217</td>
<td>9.465240</td>
<td>9.466144</td>
</tr>
<tr>
<td>-0.1301819</td>
<td>548.036928</td>
<td>548.050425</td>
<td>-0.3006254</td>
<td>-4.664720</td>
<td>-4.661082</td>
</tr>
<tr>
<td>-0.1597767</td>
<td>555.418864</td>
<td>555.417494</td>
<td>-0.3050225</td>
<td>-37.978886</td>
<td>-37.978389</td>
</tr>
<tr>
<td>-0.1792233</td>
<td>536.643545</td>
<td>536.615910</td>
<td>-0.3092979</td>
<td>-71.293969</td>
<td>-71.293473</td>
</tr>
<tr>
<td>-0.1940294</td>
<td>509.759448</td>
<td>509.728398</td>
<td>-0.3134619</td>
<td>-104.611436</td>
<td>-104.611042</td>
</tr>
</tbody>
</table>
Table 9.7  Iterations per load step of the analysis

<table>
<thead>
<tr>
<th>Error test factor</th>
<th>Initial load factor</th>
<th>Number of steps</th>
<th>Total iterations</th>
<th>Iterations / Step</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^{-6}$</td>
<td>0.20</td>
<td>110</td>
<td>354</td>
<td>3.22</td>
</tr>
<tr>
<td></td>
<td>0.30</td>
<td>75</td>
<td>264</td>
<td>3.52</td>
</tr>
<tr>
<td></td>
<td>0.40</td>
<td>59</td>
<td>236</td>
<td>4.00</td>
</tr>
<tr>
<td></td>
<td>0.50</td>
<td>47</td>
<td>208</td>
<td>4.43</td>
</tr>
<tr>
<td></td>
<td>0.60</td>
<td>41</td>
<td>204</td>
<td>4.98</td>
</tr>
<tr>
<td></td>
<td>0.70</td>
<td>37</td>
<td>177</td>
<td>4.78</td>
</tr>
<tr>
<td></td>
<td>0.80</td>
<td>34</td>
<td>176</td>
<td>5.18</td>
</tr>
<tr>
<td>$10^{-4}$</td>
<td>0.20</td>
<td>110</td>
<td>290</td>
<td>2.64</td>
</tr>
<tr>
<td></td>
<td>0.30</td>
<td>75</td>
<td>207</td>
<td>2.76</td>
</tr>
<tr>
<td></td>
<td>0.40</td>
<td>59</td>
<td>189</td>
<td>3.20</td>
</tr>
<tr>
<td></td>
<td>0.50</td>
<td>47</td>
<td>169</td>
<td>3.60</td>
</tr>
<tr>
<td></td>
<td>0.60</td>
<td>41</td>
<td>148</td>
<td>3.61</td>
</tr>
<tr>
<td></td>
<td>0.70</td>
<td>37</td>
<td>144</td>
<td>3.89</td>
</tr>
<tr>
<td></td>
<td>0.80</td>
<td>34</td>
<td>143</td>
<td>4.21</td>
</tr>
<tr>
<td>$10^{-3}$</td>
<td>0.20</td>
<td>110</td>
<td>172</td>
<td>1.56</td>
</tr>
<tr>
<td></td>
<td>0.30</td>
<td>75</td>
<td>141</td>
<td>1.88</td>
</tr>
<tr>
<td></td>
<td>0.40</td>
<td>59</td>
<td>135</td>
<td>2.29</td>
</tr>
<tr>
<td></td>
<td>0.50</td>
<td>47</td>
<td>118</td>
<td>2.51</td>
</tr>
<tr>
<td></td>
<td>0.60</td>
<td>41</td>
<td>108</td>
<td>2.63</td>
</tr>
<tr>
<td></td>
<td>0.70</td>
<td>37</td>
<td>99</td>
<td>2.68</td>
</tr>
<tr>
<td></td>
<td>0.80</td>
<td>34</td>
<td>100</td>
<td>2.94</td>
</tr>
<tr>
<td>$10^{-2}$</td>
<td>0.20</td>
<td>110</td>
<td>122</td>
<td>1.11</td>
</tr>
<tr>
<td></td>
<td>0.30</td>
<td>75</td>
<td>104</td>
<td>1.39</td>
</tr>
<tr>
<td></td>
<td>0.40</td>
<td>59</td>
<td>88</td>
<td>1.49</td>
</tr>
<tr>
<td></td>
<td>0.50</td>
<td>47</td>
<td>83</td>
<td>1.77</td>
</tr>
<tr>
<td></td>
<td>0.60</td>
<td>41</td>
<td>69</td>
<td>1.68</td>
</tr>
<tr>
<td></td>
<td>0.70</td>
<td>37</td>
<td>72</td>
<td>1.95</td>
</tr>
<tr>
<td></td>
<td>0.80</td>
<td>34</td>
<td>71</td>
<td>2.09</td>
</tr>
</tbody>
</table>

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.

$$w_{2c} = -1 + \frac{1}{\sqrt{3}} = -0.422649731$$

$$u_{2c} = h_0 \cdot w_{2c} = -0.063397459$$

$$p_{2c} = w_{2c} \cdot (1 + w_{2c}) \cdot (2 + w_{2c}) = -0.384900179$$

$$P_{2c} = A E \left(\frac{h_0}{L_0}\right)^3 p_c = -32.203851$$

**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 \( 2.0 \times 10^8 \) kN/m\(^2\). The supports A and C of the truss are pinned. The truss carries a vertical pattern load \( P = (0.0, -10000.0) \) 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 \( d_{zc} \) and the vertical load \( P_{zc} \) 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 \( P_{2,\text{exact}} \) 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 \( d_{2,\text{exact}} \) for a given horizontal displacement \( d_1 \) is computed with equation (2.41) and the normalisation (2.13). The exact value \( P_{2,\text{exact}} \) 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 \( n_1 \) 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 \( P_{2,\text{exact}} \) for the computed displacement \( d_2 \) is computed with equation (2.20). For the continuation of the load curve, the vertical displacement \( d_{2,\text{exact}} \) and the vertical load \( P_{2,\text{exact}} \) 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

<table>
<thead>
<tr>
<th>Shape factor</th>
<th>LFI</th>
<th>$n_1$</th>
<th>$d_{sc}$</th>
<th>$P_{sc}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.200</td>
<td>0.500</td>
<td>3</td>
<td>-0.081 667 391</td>
<td>-14 469.845 279</td>
</tr>
<tr>
<td>0.300</td>
<td>2</td>
<td></td>
<td>-0.081 667 391</td>
<td>-14 469.845 279</td>
</tr>
<tr>
<td>0.200</td>
<td>2</td>
<td></td>
<td>-0.081 667 391</td>
<td>-14 469.845 279</td>
</tr>
<tr>
<td>0.100</td>
<td>2</td>
<td></td>
<td>-0.081 667 391</td>
<td>-14 469.845 279</td>
</tr>
<tr>
<td>exact</td>
<td></td>
<td></td>
<td>-0.081 667 391</td>
<td>-14 469.845 280</td>
</tr>
<tr>
<td>0.100</td>
<td>0.120</td>
<td>2</td>
<td>-0.020 101 013</td>
<td>-3 901.134 901</td>
</tr>
<tr>
<td>0.090</td>
<td>2</td>
<td></td>
<td>-0.020 101 013</td>
<td>-3 901.134 901</td>
</tr>
<tr>
<td>0.060</td>
<td>2</td>
<td></td>
<td>-0.020 101 013</td>
<td>-3 901.134 901</td>
</tr>
<tr>
<td>0.030</td>
<td>2</td>
<td></td>
<td>-0.020 101 013</td>
<td>-3 901.134 901</td>
</tr>
<tr>
<td>exact</td>
<td></td>
<td></td>
<td>-0.020 101 012</td>
<td>-3 901.134 901</td>
</tr>
<tr>
<td>0.050</td>
<td>0.025</td>
<td>2</td>
<td>-0.005 006 266</td>
<td>-993.767 909</td>
</tr>
<tr>
<td>0.020</td>
<td>2</td>
<td></td>
<td>-0.005 006 266</td>
<td>-993.767 909</td>
</tr>
<tr>
<td>0.015</td>
<td>2</td>
<td></td>
<td>-0.005 006 266</td>
<td>-993.767 909</td>
</tr>
<tr>
<td>0.010</td>
<td>2</td>
<td></td>
<td>-0.005 006 265</td>
<td>-993.767 909</td>
</tr>
<tr>
<td>exact</td>
<td></td>
<td></td>
<td>-0.005 006 265</td>
<td>-993.767 909</td>
</tr>
</tbody>
</table>

### Table 9.9  Step sequence from a nearly-singular point to the bifurcation point

<table>
<thead>
<tr>
<th>Shape factor</th>
<th>LFI</th>
<th>State</th>
<th>$d_s$</th>
<th>$P_s$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.200</td>
<td>0.200</td>
<td>7</td>
<td>-0.078 629 551</td>
<td>-13 964.481 955</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8</td>
<td>-0.081 664 989</td>
<td>-14 469.446 760</td>
</tr>
<tr>
<td></td>
<td></td>
<td>9</td>
<td>-0.081 667 391</td>
<td>-14 469.845 279</td>
</tr>
<tr>
<td>0.100</td>
<td>0.060</td>
<td>6</td>
<td>-0.018 518 353</td>
<td>-3 598.281 087</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7</td>
<td>-0.020 100 381</td>
<td>-3 901.014 101</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8</td>
<td>-0.020 101 013</td>
<td>-3 901.134 901</td>
</tr>
<tr>
<td>0.050</td>
<td>0.020</td>
<td>4</td>
<td>-0.004 026 807</td>
<td>-799.929 107</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5</td>
<td>-0.005 006 025</td>
<td>-993.720 384</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
<td>-0.005 006 266</td>
<td>-993.767 909</td>
</tr>
</tbody>
</table>
Table 9.10  Error in the displacement – load curve for m = 0.200 and LFI = 0.200

<table>
<thead>
<tr>
<th>State</th>
<th>$d_1$</th>
<th>$d_2$</th>
<th>$d_{2,\text{exact}}$</th>
<th>$P_2$</th>
<th>$P_{2,\text{exact}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0.000 000</td>
<td>-0.021551</td>
<td>-0.021551</td>
<td>-3998.466</td>
<td>-3998.465</td>
</tr>
<tr>
<td>7</td>
<td>0.000 000</td>
<td>-0.078630</td>
<td>-0.078630</td>
<td>-13964.482</td>
<td>-13964.480</td>
</tr>
<tr>
<td>8</td>
<td>0.000 000</td>
<td>-0.081665</td>
<td>-0.081665</td>
<td>-14469.447</td>
<td>-14469.447</td>
</tr>
<tr>
<td>9</td>
<td>0.000 000</td>
<td>-0.081667</td>
<td>-0.081667</td>
<td>-14469.845</td>
<td>-14469.845</td>
</tr>
<tr>
<td>10</td>
<td>-0.047066</td>
<td>-0.082237</td>
<td>-0.082245</td>
<td>-14464.115</td>
<td>-14465.489</td>
</tr>
<tr>
<td>12</td>
<td>-0.140824</td>
<td>-0.086940</td>
<td>-0.086843</td>
<td>-14446.713</td>
<td>-14430.804</td>
</tr>
<tr>
<td>14</td>
<td>-0.236014</td>
<td>-0.096307</td>
<td>-0.096241</td>
<td>-14371.149</td>
<td>-14359.916</td>
</tr>
<tr>
<td>16</td>
<td>-0.332064</td>
<td>-0.110627</td>
<td>-0.110626</td>
<td>-14251.587</td>
<td>-14251.412</td>
</tr>
<tr>
<td>18</td>
<td>-0.428804</td>
<td>-0.130199</td>
<td>-0.130207</td>
<td>-14102.873</td>
<td>-14103.717</td>
</tr>
<tr>
<td>20</td>
<td>-0.525397</td>
<td>-0.155013</td>
<td>-0.155018</td>
<td>-13916.015</td>
<td>-13916.566</td>
</tr>
<tr>
<td>22</td>
<td>-0.620676</td>
<td>-0.184848</td>
<td>-0.184852</td>
<td>-13691.152</td>
<td>-13691.528</td>
</tr>
<tr>
<td>24</td>
<td>-0.714448</td>
<td>-0.219670</td>
<td>-0.219673</td>
<td>-13428.605</td>
<td>-13428.879</td>
</tr>
<tr>
<td>26</td>
<td>-0.806562</td>
<td>-0.259464</td>
<td>-0.259466</td>
<td>-13128.515</td>
<td>-13128.723</td>
</tr>
<tr>
<td>28</td>
<td>-0.896877</td>
<td>-0.304235</td>
<td>-0.304237</td>
<td>-12790.855</td>
<td>-12791.019</td>
</tr>
<tr>
<td>30</td>
<td>-0.985247</td>
<td>-0.354007</td>
<td>-0.354008</td>
<td>-12415.463</td>
<td>-12415.596</td>
</tr>
<tr>
<td>32</td>
<td>-1.071514</td>
<td>-0.408817</td>
<td>-0.408819</td>
<td>-12002.056</td>
<td>-12002.166</td>
</tr>
<tr>
<td>34</td>
<td>-1.155499</td>
<td>-0.468717</td>
<td>-0.468719</td>
<td>-11550.252</td>
<td>-11550.346</td>
</tr>
<tr>
<td>36</td>
<td>-1.237000</td>
<td>-0.533769</td>
<td>-0.533770</td>
<td>-11059.587</td>
<td>-11059.667</td>
</tr>
<tr>
<td>38</td>
<td>-1.315792</td>
<td>-0.604044</td>
<td>-0.604045</td>
<td>-10529.519</td>
<td>-10529.588</td>
</tr>
<tr>
<td>40</td>
<td>-1.391620</td>
<td>-0.679622</td>
<td>-0.679624</td>
<td>-9959.444</td>
<td>-9959.505</td>
</tr>
<tr>
<td>42</td>
<td>-1.464196</td>
<td>-0.760592</td>
<td>-0.760593</td>
<td>-9348.702</td>
<td>-9348.756</td>
</tr>
<tr>
<td>44</td>
<td>-1.533200</td>
<td>-0.847046</td>
<td>-0.847048</td>
<td>-8696.589</td>
<td>-8696.636</td>
</tr>
<tr>
<td>46</td>
<td>-1.598268</td>
<td>-0.939084</td>
<td>-0.939085</td>
<td>-8002.365</td>
<td>-8002.406</td>
</tr>
<tr>
<td>48</td>
<td>-1.658993</td>
<td>-1.036804</td>
<td>-1.036805</td>
<td>-7265.272</td>
<td>-7265.307</td>
</tr>
<tr>
<td>50</td>
<td>-1.714915</td>
<td>-1.140308</td>
<td>-1.140310</td>
<td>-6484.553</td>
<td>-6484.583</td>
</tr>
<tr>
<td>52</td>
<td>-1.765514</td>
<td>-1.249692</td>
<td>-1.249694</td>
<td>-5659.48</td>
<td>-5659.504</td>
</tr>
<tr>
<td>54</td>
<td>-1.810203</td>
<td>-1.365043</td>
<td>-1.365046</td>
<td>-4789.396</td>
<td>-4789.411</td>
</tr>
<tr>
<td>56</td>
<td>-1.848311</td>
<td>-1.486434</td>
<td>-1.486438</td>
<td>-3873.759</td>
<td>-3873.761</td>
</tr>
<tr>
<td>58</td>
<td>-1.879079</td>
<td>-1.613910</td>
<td>-1.613915</td>
<td>-2912.225</td>
<td>-2912.210</td>
</tr>
<tr>
<td>60</td>
<td>-1.901640</td>
<td>-1.747476</td>
<td>-1.747485</td>
<td>-1904.751</td>
<td>-1904.701</td>
</tr>
</tbody>
</table>
Table 9.11  Error in the displacement – load curve for m = 0.100 and LFI = 0.060

<table>
<thead>
<tr>
<th>State</th>
<th>$d_1$</th>
<th>$d_2$</th>
<th>$d_{2,\text{exact}}$</th>
<th>$P_2$</th>
<th>$P_{2,\text{exact}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0.000000</td>
<td>-0.006118</td>
<td>-0.006118</td>
<td>-1199.888</td>
<td>-1199.888</td>
</tr>
<tr>
<td>4</td>
<td>0.000000</td>
<td>-0.012290</td>
<td>-0.012290</td>
<td>-2399.320</td>
<td>-2399.320</td>
</tr>
<tr>
<td>6</td>
<td>0.000000</td>
<td>-0.018518</td>
<td>-0.018518</td>
<td>-3598.281</td>
<td>-3598.281</td>
</tr>
<tr>
<td>7</td>
<td>0.000000</td>
<td>-0.020100</td>
<td>-0.020100</td>
<td>-3901.014</td>
<td>-3901.014</td>
</tr>
<tr>
<td>8</td>
<td>0.000000</td>
<td>-0.020101</td>
<td>-0.020101</td>
<td>-3901.135</td>
<td>-3901.135</td>
</tr>
<tr>
<td>10</td>
<td>-0.029742</td>
<td>-0.020325</td>
<td>-0.020324</td>
<td>-3900.931</td>
<td>-3900.695</td>
</tr>
<tr>
<td>12</td>
<td>-0.059378</td>
<td>-0.020998</td>
<td>-0.020992</td>
<td>-3900.639</td>
<td>-3899.380</td>
</tr>
<tr>
<td>14</td>
<td>-0.089027</td>
<td>-0.022111</td>
<td>-0.022104</td>
<td>-3898.729</td>
<td>-3897.189</td>
</tr>
<tr>
<td>16</td>
<td>-0.118685</td>
<td>-0.023665</td>
<td>-0.023661</td>
<td>-3894.924</td>
<td>-3894.119</td>
</tr>
<tr>
<td>18</td>
<td>-0.148271</td>
<td>-0.025661</td>
<td>-0.025661</td>
<td>-3890.345</td>
<td>-3890.180</td>
</tr>
<tr>
<td>20</td>
<td>-0.177801</td>
<td>-0.028101</td>
<td>-0.028101</td>
<td>-3885.380</td>
<td>-3885.372</td>
</tr>
<tr>
<td>25</td>
<td>-0.251202</td>
<td>-0.036101</td>
<td>-0.036101</td>
<td>-3869.588</td>
<td>-3869.608</td>
</tr>
<tr>
<td>30</td>
<td>-0.324940</td>
<td>-0.046947</td>
<td>-0.046948</td>
<td>-3848.225</td>
<td>-3848.237</td>
</tr>
<tr>
<td>35</td>
<td>-0.398370</td>
<td>-0.060593</td>
<td>-0.060593</td>
<td>-3821.343</td>
<td>-3821.351</td>
</tr>
<tr>
<td>40</td>
<td>-0.471373</td>
<td>-0.077032</td>
<td>-0.077032</td>
<td>-3788.955</td>
<td>-3788.960</td>
</tr>
<tr>
<td>50</td>
<td>-0.615773</td>
<td>-0.118292</td>
<td>-0.118292</td>
<td>-3707.658</td>
<td>-3707.661</td>
</tr>
<tr>
<td>60</td>
<td>-0.757548</td>
<td>-0.170759</td>
<td>-0.170759</td>
<td>-3604.280</td>
<td>-3604.282</td>
</tr>
<tr>
<td>70</td>
<td>-0.896090</td>
<td>-0.234491</td>
<td>-0.234491</td>
<td>-3478.706</td>
<td>-3478.707</td>
</tr>
<tr>
<td>80</td>
<td>-1.030739</td>
<td>-0.309563</td>
<td>-0.309563</td>
<td>-3330.786</td>
<td>-3330.787</td>
</tr>
<tr>
<td>90</td>
<td>-1.160772</td>
<td>-0.396065</td>
<td>-0.396065</td>
<td>-3160.346</td>
<td>-3160.347</td>
</tr>
<tr>
<td>100</td>
<td>-1.285399</td>
<td>-0.494095</td>
<td>-0.494095</td>
<td>-2967.190</td>
<td>-2967.191</td>
</tr>
<tr>
<td>110</td>
<td>-1.403752</td>
<td>-0.603762</td>
<td>-0.603762</td>
<td>-2751.105</td>
<td>-2751.106</td>
</tr>
<tr>
<td>120</td>
<td>-1.514873</td>
<td>-0.725182</td>
<td>-0.725182</td>
<td>-2511.863</td>
<td>-2511.864</td>
</tr>
<tr>
<td>130</td>
<td>-1.617689</td>
<td>-0.858474</td>
<td>-0.858474</td>
<td>-2249.228</td>
<td>-2249.229</td>
</tr>
<tr>
<td>140</td>
<td>-1.710995</td>
<td>-1.003760</td>
<td>-1.003760</td>
<td>-1962.963</td>
<td>-1962.963</td>
</tr>
<tr>
<td>150</td>
<td>-1.793414</td>
<td>-1.161153</td>
<td>-1.161153</td>
<td>-1652.840</td>
<td>-1652.840</td>
</tr>
<tr>
<td>160</td>
<td>-1.863360</td>
<td>-1.330754</td>
<td>-1.330754</td>
<td>-1318.662</td>
<td>-1318.663</td>
</tr>
<tr>
<td>170</td>
<td>-1.918977</td>
<td>-1.512634</td>
<td>-1.512634</td>
<td>-960.292</td>
<td>-960.292</td>
</tr>
<tr>
<td>180</td>
<td>-1.958071</td>
<td>-1.706811</td>
<td>-1.706812</td>
<td>-577.690</td>
<td>-577.690</td>
</tr>
<tr>
<td>190</td>
<td>-1.977996</td>
<td>-1.913220</td>
<td>-1.913221</td>
<td>-170.988</td>
<td>-170.988</td>
</tr>
</tbody>
</table>
Table 9.12 Error in the displacement – load curve for \( m = 0.050 \) and LFI = 0.020

<table>
<thead>
<tr>
<th>State</th>
<th>( d_1 )</th>
<th>( d_2 )</th>
<th>( d_{2,exact} )</th>
<th>( P_2 )</th>
<th>( P_{2,exact} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>0.000000</td>
<td>-0.005006</td>
<td>-0.005006</td>
<td>-993.768</td>
<td>-993.768</td>
</tr>
<tr>
<td>7</td>
<td>-0.005059</td>
<td>-0.005013</td>
<td>-0.005013</td>
<td>-993.765</td>
<td>-993.765</td>
</tr>
<tr>
<td>10</td>
<td>-0.020202</td>
<td>-0.005109</td>
<td>-0.005109</td>
<td>-993.783</td>
<td>-993.717</td>
</tr>
<tr>
<td>20</td>
<td>-0.070476</td>
<td>-0.006252</td>
<td>-0.006251</td>
<td>-993.175</td>
<td>-993.148</td>
</tr>
<tr>
<td>40</td>
<td>-0.170192</td>
<td>-0.012279</td>
<td>-0.012279</td>
<td>-990.145</td>
<td>-990.145</td>
</tr>
<tr>
<td>60</td>
<td>-0.270184</td>
<td>-0.023387</td>
<td>-0.023387</td>
<td>-984.612</td>
<td>-984.612</td>
</tr>
<tr>
<td>80</td>
<td>-0.369861</td>
<td>-0.039591</td>
<td>-0.039591</td>
<td>-976.540</td>
<td>-976.540</td>
</tr>
<tr>
<td>100</td>
<td>-0.468935</td>
<td>-0.060902</td>
<td>-0.060902</td>
<td>-965.924</td>
<td>-965.924</td>
</tr>
<tr>
<td>120</td>
<td>-0.567176</td>
<td>-0.087329</td>
<td>-0.087329</td>
<td>-952.761</td>
<td>-952.761</td>
</tr>
<tr>
<td>140</td>
<td>-0.664365</td>
<td>-0.118878</td>
<td>-0.118878</td>
<td>-937.045</td>
<td>-937.045</td>
</tr>
<tr>
<td>160</td>
<td>-0.760285</td>
<td>-0.155558</td>
<td>-0.155558</td>
<td>-918.773</td>
<td>-918.773</td>
</tr>
<tr>
<td>180</td>
<td>-0.854720</td>
<td>-0.197376</td>
<td>-0.197376</td>
<td>-897.943</td>
<td>-897.943</td>
</tr>
<tr>
<td>200</td>
<td>-0.947447</td>
<td>-0.244339</td>
<td>-0.244339</td>
<td>-874.549</td>
<td>-874.549</td>
</tr>
<tr>
<td>220</td>
<td>-1.038239</td>
<td>-0.296456</td>
<td>-0.296456</td>
<td>-848.588</td>
<td>-848.588</td>
</tr>
<tr>
<td>240</td>
<td>-1.126860</td>
<td>-0.353736</td>
<td>-0.353736</td>
<td>-820.055</td>
<td>-820.055</td>
</tr>
<tr>
<td>260</td>
<td>-1.213067</td>
<td>-0.416185</td>
<td>-0.416185</td>
<td>-788.947</td>
<td>-788.947</td>
</tr>
<tr>
<td>280</td>
<td>-1.296604</td>
<td>-0.483815</td>
<td>-0.483815</td>
<td>-755.259</td>
<td>-755.259</td>
</tr>
<tr>
<td>300</td>
<td>-1.377205</td>
<td>-0.556634</td>
<td>-0.556634</td>
<td>-718.985</td>
<td>-718.985</td>
</tr>
<tr>
<td>320</td>
<td>-1.454587</td>
<td>-0.634651</td>
<td>-0.634651</td>
<td>-680.122</td>
<td>-680.122</td>
</tr>
<tr>
<td>340</td>
<td>-1.528450</td>
<td>-0.717877</td>
<td>-0.717877</td>
<td>-638.665</td>
<td>-638.665</td>
</tr>
<tr>
<td>360</td>
<td>-1.598478</td>
<td>-0.806321</td>
<td>-0.806321</td>
<td>-594.608</td>
<td>-594.608</td>
</tr>
<tr>
<td>380</td>
<td>-1.664328</td>
<td>-0.899994</td>
<td>-0.899994</td>
<td>-547.947</td>
<td>-547.947</td>
</tr>
<tr>
<td>400</td>
<td>-1.725632</td>
<td>-0.998904</td>
<td>-0.998904</td>
<td>-498.677</td>
<td>-498.677</td>
</tr>
<tr>
<td>420</td>
<td>-1.781993</td>
<td>-1.103060</td>
<td>-1.103060</td>
<td>-446.794</td>
<td>-446.794</td>
</tr>
<tr>
<td>440</td>
<td>-1.832975</td>
<td>-1.212470</td>
<td>-1.212470</td>
<td>-392.293</td>
<td>-392.293</td>
</tr>
<tr>
<td>460</td>
<td>-1.878100</td>
<td>-1.327140</td>
<td>-1.327140</td>
<td>-335.172</td>
<td>-335.172</td>
</tr>
<tr>
<td>480</td>
<td>-1.916840</td>
<td>-1.447075</td>
<td>-1.447075</td>
<td>-275.429</td>
<td>-275.429</td>
</tr>
<tr>
<td>500</td>
<td>-1.948603</td>
<td>-1.572277</td>
<td>-1.572277</td>
<td>-213.062</td>
<td>-213.062</td>
</tr>
<tr>
<td>520</td>
<td>-1.972724</td>
<td>-1.702743</td>
<td>-1.702743</td>
<td>-148.073</td>
<td>-148.073</td>
</tr>
<tr>
<td>540</td>
<td>-1.988443</td>
<td>-1.838465</td>
<td>-1.838465</td>
<td>-80.466</td>
<td>-80.466</td>
</tr>
</tbody>
</table>
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 \(2.0 \times 10^8\) 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](image)

**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:

\[
\varepsilon = v_{11} + \frac{1}{2}(v_{11}^2 + v_{21}^2)
\]

\(v_{11}, v_{21}\) 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 \(\varepsilon\) 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
- d₁: horizontal displacement coordinate of the tip
- d₂: 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:

<table>
<thead>
<tr>
<th>Rotation</th>
<th>(d_{2,\text{exact}}) (m)</th>
<th>(d_2 - d_{2,\text{exact}}) (m)</th>
<th>Relative error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>11.5</td>
<td>0.199</td>
<td>0.00032</td>
<td>0.16</td>
</tr>
<tr>
<td>39.4</td>
<td>0.582</td>
<td>0.00877</td>
<td>1.51</td>
</tr>
<tr>
<td>57.3</td>
<td>0.920</td>
<td>0.03940</td>
<td>4.28</td>
</tr>
<tr>
<td>80.2</td>
<td>1.186</td>
<td>0.10272</td>
<td>8.66</td>
</tr>
<tr>
<td>103.2</td>
<td>1.364</td>
<td>0.20347</td>
<td>14.92</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>State</th>
<th>$M$</th>
<th>$R$</th>
<th>$d_1$</th>
<th>$d_{exact}$</th>
<th>$d_2$</th>
<th>$d_{exact}$</th>
<th>$\beta$</th>
<th>$\beta_{exact}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0.0999965</td>
<td>-0.099992</td>
<td>-0.013293</td>
<td>-0.013298</td>
<td>0.199302</td>
<td>0.199268</td>
<td>0.199946</td>
<td>0.199931</td>
</tr>
<tr>
<td>20</td>
<td>0.1999945</td>
<td>-0.199972</td>
<td>-0.052851</td>
<td>-0.052880</td>
<td>0.394619</td>
<td>0.394590</td>
<td>0.399879</td>
<td>0.399891</td>
</tr>
<tr>
<td>30</td>
<td>0.2999964</td>
<td>-0.299986</td>
<td>-0.117757</td>
<td>-0.117831</td>
<td>0.582173</td>
<td>0.582149</td>
<td>0.599882</td>
<td>0.599928</td>
</tr>
<tr>
<td>40</td>
<td>0.4000040</td>
<td>-0.400056</td>
<td>-0.206513</td>
<td>-0.206650</td>
<td>0.758334</td>
<td>0.758301</td>
<td>0.799998</td>
<td>0.800080</td>
</tr>
<tr>
<td>50</td>
<td>0.500191</td>
<td>-0.500202</td>
<td>-0.317077</td>
<td>-0.317288</td>
<td>0.919752</td>
<td>0.919687</td>
<td>1.000264</td>
<td>1.000382</td>
</tr>
<tr>
<td>60</td>
<td>0.600434</td>
<td>-0.600440</td>
<td>-0.446910</td>
<td>-0.447201</td>
<td>1.063438</td>
<td>1.063317</td>
<td>1.200716</td>
<td>1.200868</td>
</tr>
<tr>
<td>70</td>
<td>0.700783</td>
<td>-0.700786</td>
<td>-0.593043</td>
<td>-0.593410</td>
<td>1.186845</td>
<td>1.186639</td>
<td>1.401382</td>
<td>1.401567</td>
</tr>
<tr>
<td>80</td>
<td>0.801251</td>
<td>-0.801251</td>
<td>-0.752145</td>
<td>-0.752578</td>
<td>1.287926</td>
<td>1.287612</td>
<td>1.602287</td>
<td>1.602501</td>
</tr>
<tr>
<td>90</td>
<td>0.901846</td>
<td>-0.901846</td>
<td>-0.920618</td>
<td>-0.921099</td>
<td>1.365200</td>
<td>1.364752</td>
<td>1.803448</td>
<td>1.803691</td>
</tr>
<tr>
<td>100</td>
<td>1.002576</td>
<td>-1.002575</td>
<td>-1.094680</td>
<td>-1.095189</td>
<td>1.417773</td>
<td>1.417175</td>
<td>2.004879</td>
<td>2.005151</td>
</tr>
<tr>
<td>110</td>
<td>1.103515</td>
<td>-1.103533</td>
<td>-1.270631</td>
<td>-1.271111</td>
<td>1.445394</td>
<td>1.444629</td>
<td>2.206759</td>
<td>2.207029</td>
</tr>
<tr>
<td>120</td>
<td>1.204524</td>
<td>-1.204520</td>
<td>-1.444306</td>
<td>-1.444789</td>
<td>1.448369</td>
<td>1.447439</td>
<td>2.408714</td>
<td>2.409047</td>
</tr>
<tr>
<td>130</td>
<td>1.305671</td>
<td>-1.305665</td>
<td>-1.612226</td>
<td>-1.612652</td>
<td>1.427706</td>
<td>1.426708</td>
<td>2.610975</td>
<td>2.611342</td>
</tr>
<tr>
<td>140</td>
<td>1.406953</td>
<td>-1.406942</td>
<td>-1.770903</td>
<td>-1.771240</td>
<td>1.384950</td>
<td>1.383692</td>
<td>2.813503</td>
<td>2.813905</td>
</tr>
<tr>
<td>150</td>
<td>1.508361</td>
<td>-1.508347</td>
<td>-1.917212</td>
<td>-1.917430</td>
<td>1.322182</td>
<td>1.320780</td>
<td>3.016284</td>
<td>3.016723</td>
</tr>
<tr>
<td>160</td>
<td>1.609888</td>
<td>-1.609869</td>
<td>-2.048445</td>
<td>-2.048515</td>
<td>1.241946</td>
<td>1.240425</td>
<td>3.219301</td>
<td>3.219776</td>
</tr>
<tr>
<td>170</td>
<td>1.711521</td>
<td>-1.711499</td>
<td>-2.162383</td>
<td>-2.162821</td>
<td>1.147171</td>
<td>1.146562</td>
<td>3.422533</td>
<td>3.423042</td>
</tr>
<tr>
<td>180</td>
<td>1.813247</td>
<td>-1.813224</td>
<td>-2.257354</td>
<td>-2.257065</td>
<td>1.041073</td>
<td>1.039417</td>
<td>3.625955</td>
<td>3.626495</td>
</tr>
<tr>
<td>190</td>
<td>1.915054</td>
<td>-1.915031</td>
<td>-2.332272</td>
<td>-2.331788</td>
<td>0.927059</td>
<td>0.926400</td>
<td>3.829540</td>
<td>3.830108</td>
</tr>
<tr>
<td>200</td>
<td>2.016925</td>
<td>-2.016904</td>
<td>-2.386654</td>
<td>-2.385979</td>
<td>0.808615</td>
<td>0.807998</td>
<td>4.034259</td>
<td>4.034851</td>
</tr>
<tr>
<td>210</td>
<td>2.118848</td>
<td>-2.118830</td>
<td>-2.420623</td>
<td>-2.419772</td>
<td>0.689199</td>
<td>0.687669</td>
<td>4.237081</td>
<td>4.237696</td>
</tr>
<tr>
<td>220</td>
<td>2.220807</td>
<td>-2.220793</td>
<td>-2.434881</td>
<td>-2.433880</td>
<td>0.572130</td>
<td>0.570729</td>
<td>4.440978</td>
<td>4.441613</td>
</tr>
<tr>
<td>230</td>
<td>2.322789</td>
<td>-2.322779</td>
<td>-2.430672</td>
<td>-2.429556</td>
<td>0.460497</td>
<td>0.459259</td>
<td>4.644920</td>
<td>4.645578</td>
</tr>
<tr>
<td>240</td>
<td>2.424783</td>
<td>-2.424776</td>
<td>-2.409721</td>
<td>-2.408534</td>
<td>0.357062</td>
<td>0.356013</td>
<td>4.848879</td>
<td>4.849565</td>
</tr>
<tr>
<td>250</td>
<td>2.526778</td>
<td>-2.526774</td>
<td>-2.374160</td>
<td>-2.372951</td>
<td>0.264187</td>
<td>0.263345</td>
<td>5.052850</td>
<td>5.053555</td>
</tr>
<tr>
<td>260</td>
<td>2.628766</td>
<td>-2.628764</td>
<td>-2.326445</td>
<td>-2.325268</td>
<td>0.183780</td>
<td>0.183151</td>
<td>5.256801</td>
<td>5.257532</td>
</tr>
<tr>
<td>270</td>
<td>2.730741</td>
<td>-2.730741</td>
<td>-2.269262</td>
<td>-2.268717</td>
<td>0.117248</td>
<td>0.116827</td>
<td>5.460726</td>
<td>5.461482</td>
</tr>
<tr>
<td>280</td>
<td>2.832699</td>
<td>-2.832701</td>
<td>-2.205429</td>
<td>-2.204481</td>
<td>0.065480</td>
<td>0.065251</td>
<td>5.664616</td>
<td>5.665398</td>
</tr>
<tr>
<td>290</td>
<td>2.934639</td>
<td>-2.934643</td>
<td>-2.137804</td>
<td>-2.137049</td>
<td>0.028840</td>
<td>0.028775</td>
<td>5.868472</td>
<td>5.869278</td>
</tr>
<tr>
<td>300</td>
<td>3.036560</td>
<td>-3.036567</td>
<td>-2.069187</td>
<td>-2.068671</td>
<td>0.007181</td>
<td>0.007239</td>
<td>6.072290</td>
<td>6.073120</td>
</tr>
<tr>
<td>310</td>
<td>3.138465</td>
<td>-3.138475</td>
<td>-2.002233</td>
<td>-2.001993</td>
<td>-0.000125</td>
<td>0.000006</td>
<td>6.276078</td>
<td>6.276931</td>
</tr>
</tbody>
</table>
Table 9.14  Analysis of cantilever beam with 20 elements, $f_e = 0.020$ and LFI = 0.200

<table>
<thead>
<tr>
<th>State</th>
<th>M</th>
<th>R</th>
<th>$d_1$</th>
<th>$d_{exact}$</th>
<th>$d_2$</th>
<th>$d_{2,exact}$</th>
<th>$\beta$</th>
<th>$\beta_{exact}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>0.099464</td>
<td>-0.100379</td>
<td>-0.013371</td>
<td>-0.013165</td>
<td>0.199916</td>
<td>0.198273</td>
<td>0.200290</td>
<td>0.198928</td>
</tr>
<tr>
<td>10</td>
<td>0.198837</td>
<td>-0.200539</td>
<td>-0.053044</td>
<td>-0.052299</td>
<td>0.395407</td>
<td>0.392460</td>
<td>0.400152</td>
<td>0.397673</td>
</tr>
<tr>
<td>15</td>
<td>0.298302</td>
<td>-0.300393</td>
<td>-0.117860</td>
<td>-0.116552</td>
<td>0.582530</td>
<td>0.579117</td>
<td>0.599545</td>
<td>0.596604</td>
</tr>
<tr>
<td>20</td>
<td>0.397893</td>
<td>-0.400000</td>
<td>-0.206138</td>
<td>-0.204508</td>
<td>0.757832</td>
<td>0.754667</td>
<td>0.798579</td>
<td>0.795786</td>
</tr>
<tr>
<td>25</td>
<td>0.497632</td>
<td>-0.499431</td>
<td>-0.315743</td>
<td>-0.314211</td>
<td>0.918200</td>
<td>0.915774</td>
<td>0.997395</td>
<td>0.995264</td>
</tr>
<tr>
<td>30</td>
<td>0.597534</td>
<td>-0.598800</td>
<td>-0.444198</td>
<td>-0.443200</td>
<td>1.060945</td>
<td>1.059437</td>
<td>1.196208</td>
<td>1.195068</td>
</tr>
<tr>
<td>35</td>
<td>0.697613</td>
<td>-0.698291</td>
<td>-0.588802</td>
<td>-0.587576</td>
<td>1.183836</td>
<td>1.183077</td>
<td>1.395354</td>
<td>1.395225</td>
</tr>
<tr>
<td>40</td>
<td>0.797889</td>
<td>-0.798135</td>
<td>-0.746658</td>
<td>-0.747084</td>
<td>1.285021</td>
<td>1.284614</td>
<td>1.595204</td>
<td>1.595778</td>
</tr>
<tr>
<td>45</td>
<td>0.898388</td>
<td>-0.898438</td>
<td>-0.914432</td>
<td>-0.915195</td>
<td>1.362961</td>
<td>1.362508</td>
<td>1.795860</td>
<td>1.796775</td>
</tr>
<tr>
<td>50</td>
<td>0.999126</td>
<td>-0.999098</td>
<td>-1.088247</td>
<td>-1.089181</td>
<td>1.416387</td>
<td>1.415794</td>
<td>1.997130</td>
<td>1.998253</td>
</tr>
<tr>
<td>55</td>
<td>1.100108</td>
<td>-1.099985</td>
<td>-1.264083</td>
<td>-1.265191</td>
<td>1.444922</td>
<td>1.444109</td>
<td>2.198836</td>
<td>2.200216</td>
</tr>
<tr>
<td>60</td>
<td>1.201331</td>
<td>-1.201051</td>
<td>-1.438022</td>
<td>-1.439373</td>
<td>1.448815</td>
<td>1.447719</td>
<td>2.400880</td>
<td>2.402662</td>
</tr>
<tr>
<td>65</td>
<td>1.302787</td>
<td>-1.302261</td>
<td>-1.606319</td>
<td>-1.607981</td>
<td>1.428994</td>
<td>1.427516</td>
<td>2.603236</td>
<td>2.605573</td>
</tr>
<tr>
<td>70</td>
<td>1.404463</td>
<td>-1.403611</td>
<td>-1.765500</td>
<td>-1.767480</td>
<td>1.386980</td>
<td>1.384996</td>
<td>2.805899</td>
<td>2.808925</td>
</tr>
<tr>
<td>75</td>
<td>1.506341</td>
<td>-1.505000</td>
<td>-1.912426</td>
<td>-1.914658</td>
<td>1.324830</td>
<td>1.322213</td>
<td>3.008869</td>
<td>3.012681</td>
</tr>
<tr>
<td>80</td>
<td>1.608399</td>
<td>-1.606739</td>
<td>-2.044377</td>
<td>-2.046714</td>
<td>1.245054</td>
<td>1.241715</td>
<td>3.212160</td>
<td>3.216798</td>
</tr>
<tr>
<td>85</td>
<td>1.710612</td>
<td>-1.708537</td>
<td>-2.159114</td>
<td>-2.161347</td>
<td>1.150542</td>
<td>1.146465</td>
<td>3.415789</td>
<td>3.421225</td>
</tr>
<tr>
<td>90</td>
<td>1.812953</td>
<td>-1.810508</td>
<td>-2.254929</td>
<td>-2.256819</td>
<td>1.044470</td>
<td>1.039738</td>
<td>3.619780</td>
<td>3.625905</td>
</tr>
<tr>
<td>95</td>
<td>1.915389</td>
<td>-1.912669</td>
<td>-2.330682</td>
<td>-2.332000</td>
<td>0.930213</td>
<td>0.925016</td>
<td>3.824153</td>
<td>3.830777</td>
</tr>
<tr>
<td>100</td>
<td>2.017888</td>
<td>-2.015034</td>
<td>-2.385815</td>
<td>-2.386393</td>
<td>0.811242</td>
<td>0.805871</td>
<td>4.028918</td>
<td>4.035775</td>
</tr>
<tr>
<td>105</td>
<td>2.120413</td>
<td>-2.117610</td>
<td>-2.420365</td>
<td>-2.420135</td>
<td>0.691040</td>
<td>0.685847</td>
<td>4.234060</td>
<td>4.240827</td>
</tr>
<tr>
<td>110</td>
<td>2.222928</td>
<td>-2.220373</td>
<td>-2.434954</td>
<td>-2.433973</td>
<td>0.573004</td>
<td>0.568345</td>
<td>4.439510</td>
<td>4.445855</td>
</tr>
<tr>
<td>115</td>
<td>2.325391</td>
<td>-2.323271</td>
<td>-2.430785</td>
<td>-2.429219</td>
<td>0.460366</td>
<td>0.456512</td>
<td>4.645117</td>
<td>4.650782</td>
</tr>
<tr>
<td>120</td>
<td>2.427768</td>
<td>-2.426192</td>
<td>-2.409615</td>
<td>-2.407688</td>
<td>0.356079</td>
<td>0.353140</td>
<td>4.850632</td>
<td>4.855536</td>
</tr>
<tr>
<td>125</td>
<td>2.530034</td>
<td>-2.528997</td>
<td>-2.373680</td>
<td>-2.371602</td>
<td>0.262638</td>
<td>0.260583</td>
<td>5.055866</td>
<td>5.060068</td>
</tr>
<tr>
<td>130</td>
<td>2.632179</td>
<td>-2.631591</td>
<td>-2.325563</td>
<td>-2.323494</td>
<td>0.182018</td>
<td>0.180700</td>
<td>5.260664</td>
<td>5.264359</td>
</tr>
<tr>
<td>135</td>
<td>2.734201</td>
<td>-2.733946</td>
<td>-2.268059</td>
<td>-2.266102</td>
<td>0.115560</td>
<td>0.114832</td>
<td>5.465023</td>
<td>5.468402</td>
</tr>
<tr>
<td>140</td>
<td>2.836105</td>
<td>-2.836116</td>
<td>-2.203998</td>
<td>-2.202272</td>
<td>0.064033</td>
<td>0.063788</td>
<td>5.669101</td>
<td>5.672211</td>
</tr>
<tr>
<td>145</td>
<td>2.937897</td>
<td>-2.938157</td>
<td>-2.136262</td>
<td>-2.134864</td>
<td>0.027721</td>
<td>0.027857</td>
<td>5.872956</td>
<td>5.875794</td>
</tr>
<tr>
<td>150</td>
<td>3.039586</td>
<td>-3.040117</td>
<td>-2.067629</td>
<td>-2.066654</td>
<td>0.006391</td>
<td>0.006823</td>
<td>6.076665</td>
<td>6.079172</td>
</tr>
<tr>
<td>155</td>
<td>3.141186</td>
<td>-3.141999</td>
<td>-2.000748</td>
<td>-2.000259</td>
<td>-0.00573</td>
<td>0.000000</td>
<td>6.280238</td>
<td>6.282371</td>
</tr>
</tbody>
</table>
Table 9.15  Analysis of cantilever beam with 20 elements, $f_e = 0.100$ and LFI = 0.300

<table>
<thead>
<tr>
<th>State</th>
<th>M</th>
<th>R</th>
<th>$d_1$</th>
<th>$d_{1\text{exact}}$</th>
<th>$d_2$</th>
<th>$d_{2\text{exact}}$</th>
<th>$\beta$</th>
<th>$\beta_{\text{exact}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>0.145 523</td>
<td>-0.153 716</td>
<td>-0.031 043</td>
<td>-0.028 117</td>
<td>0.303 913</td>
<td>0.288 998</td>
<td>0.304 001</td>
<td>0.291 047</td>
</tr>
<tr>
<td>10</td>
<td>0.290 556</td>
<td>-0.306 627</td>
<td>-0.121 641</td>
<td>-0.110 679</td>
<td>0.591 991</td>
<td>0.564 942</td>
<td>0.605 780</td>
<td>0.581 113</td>
</tr>
<tr>
<td>15</td>
<td>0.437 281</td>
<td>-0.456 482</td>
<td>-0.263 659</td>
<td>-0.245 378</td>
<td>0.848 573</td>
<td>0.820 221</td>
<td>0.902 745</td>
<td>0.874 562</td>
</tr>
<tr>
<td>20</td>
<td>0.585 981</td>
<td>-0.602 752</td>
<td>-0.446 346</td>
<td>-0.427 399</td>
<td>1.064 109</td>
<td>1.043 815</td>
<td>1.194 844</td>
<td>1.171 962</td>
</tr>
<tr>
<td>25</td>
<td>0.736 129</td>
<td>-0.746 104</td>
<td>-0.659 507</td>
<td>-0.648 132</td>
<td>1.233 890</td>
<td>1.224 814</td>
<td>1.482 868</td>
<td>1.472 258</td>
</tr>
<tr>
<td>30</td>
<td>0.886 804</td>
<td>-0.890 032</td>
<td>-0.895 467</td>
<td>-0.895 467</td>
<td>1.357 046</td>
<td>1.354 780</td>
<td>1.774 876</td>
<td>1.773 608</td>
</tr>
<tr>
<td>35</td>
<td>1.038 255</td>
<td>-1.038 336</td>
<td>-1.155 903</td>
<td>-1.157 405</td>
<td>1.430 522</td>
<td>1.429 737</td>
<td>2.074 034</td>
<td>2.076 510</td>
</tr>
<tr>
<td>40</td>
<td>1.191 111</td>
<td>-1.188 947</td>
<td>-1.416 932</td>
<td>-1.421 997</td>
<td>1.449 794</td>
<td>1.448 454</td>
<td>2.375 668</td>
<td>2.382 221</td>
</tr>
<tr>
<td>45</td>
<td>1.345 371</td>
<td>-1.339 840</td>
<td>-1.666 175</td>
<td>-1.676 125</td>
<td>1.416 149</td>
<td>1.412 307</td>
<td>2.677 798</td>
<td>2.690 742</td>
</tr>
<tr>
<td>50</td>
<td>1.500 889</td>
<td>-1.490 629</td>
<td>-1.892 013</td>
<td>-1.907 149</td>
<td>1.335 043</td>
<td>1.326 042</td>
<td>2.980 196</td>
<td>3.001 779</td>
</tr>
<tr>
<td>60</td>
<td>1.815 170</td>
<td>-1.792 401</td>
<td>-2.239 051</td>
<td>-2.258 665</td>
<td>1.063 614</td>
<td>1.037 325</td>
<td>3.587 242</td>
<td>3.630 340</td>
</tr>
<tr>
<td>65</td>
<td>1.973 896</td>
<td>-1.943 858</td>
<td>-2.348 995</td>
<td>-2.365 603</td>
<td>0.893 261</td>
<td>0.857 313</td>
<td>3.893 245</td>
<td>3.947 792</td>
</tr>
<tr>
<td>70</td>
<td>2.134 134</td>
<td>-2.096 113</td>
<td>-2.413 119</td>
<td>-2.423 117</td>
<td>0.714 261</td>
<td>0.669 904</td>
<td>4.202 440</td>
<td>4.268 268</td>
</tr>
<tr>
<td>75</td>
<td>2.297 395</td>
<td>-2.249 531</td>
<td>-2.431 654</td>
<td>-2.432 269</td>
<td>0.537 272</td>
<td>0.486 346</td>
<td>4.516 981</td>
<td>4.594 789</td>
</tr>
<tr>
<td>80</td>
<td>2.434 796</td>
<td>-2.421 866</td>
<td>-2.409 919</td>
<td>-2.405 648</td>
<td>0.360 266</td>
<td>0.346 413</td>
<td>4.845 382</td>
<td>4.869 591</td>
</tr>
<tr>
<td>85</td>
<td>2.590 524</td>
<td>-2.577 395</td>
<td>-2.350 971</td>
<td>-2.344 399</td>
<td>0.229 291</td>
<td>0.211 660</td>
<td>5.156 729</td>
<td>5.181 048</td>
</tr>
<tr>
<td>90</td>
<td>2.745 091</td>
<td>-2.734 422</td>
<td>-2.266 779</td>
<td>-2.259 541</td>
<td>0.115 563</td>
<td>0.108 663</td>
<td>5.469 016</td>
<td>5.490 183</td>
</tr>
<tr>
<td>95</td>
<td>2.897 956</td>
<td>-2.891 646</td>
<td>-2.167 435</td>
<td>-2.161 568</td>
<td>0.042 726</td>
<td>0.040 162</td>
<td>5.780 225</td>
<td>5.795 912</td>
</tr>
<tr>
<td>100</td>
<td>3.049 183</td>
<td>-3.048 429</td>
<td>-2.062 999</td>
<td>-2.060 269</td>
<td>0.005 068</td>
<td>0.005 585</td>
<td>6.090 111</td>
<td>6.098 365</td>
</tr>
<tr>
<td>103</td>
<td>3.139 310</td>
<td>-3.142 078</td>
<td>-2.001 892</td>
<td>-2.001 454</td>
<td>-0.001 249</td>
<td>0.000 030</td>
<td>6.275 738</td>
<td>6.278 620</td>
</tr>
</tbody>
</table>
### Table 9.16  Analysis of cantilever beam with 8 eight elements, $f_e = 0.020$ and LFI = 0.200

<table>
<thead>
<tr>
<th>state</th>
<th>$M$</th>
<th>$R$</th>
<th>$d_1$</th>
<th>$d_{\text{exact}}$</th>
<th>$d_2$</th>
<th>$d_{2,\text{exact}}$</th>
<th>$\beta$</th>
<th>$\beta_{\text{exact}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>0.09989</td>
<td>-0.099615</td>
<td>-0.013153</td>
<td>-0.013200</td>
<td>0.198575</td>
<td>0.198539</td>
<td>0.199223</td>
<td>0.199196</td>
</tr>
<tr>
<td>10</td>
<td>0.199127</td>
<td>-0.199142</td>
<td>-0.052257</td>
<td>-0.052451</td>
<td>0.393085</td>
<td>0.393018</td>
<td>0.398277</td>
<td>0.398254</td>
</tr>
<tr>
<td>15</td>
<td>0.298722</td>
<td>-0.298734</td>
<td>-0.116445</td>
<td>-0.116874</td>
<td>0.580038</td>
<td>0.579883</td>
<td>0.597462</td>
<td>0.597444</td>
</tr>
<tr>
<td>20</td>
<td>0.398416</td>
<td>-0.398424</td>
<td>-0.204293</td>
<td>-0.205029</td>
<td>0.755877</td>
<td>0.755553</td>
<td>0.796843</td>
<td>0.796832</td>
</tr>
<tr>
<td>25</td>
<td>0.498239</td>
<td>-0.498244</td>
<td>-0.313855</td>
<td>-0.314940</td>
<td>0.917303</td>
<td>0.916704</td>
<td>0.996484</td>
<td>0.996479</td>
</tr>
<tr>
<td>30</td>
<td>0.598221</td>
<td>-0.598658</td>
<td>-0.442955</td>
<td>-0.444147</td>
<td>1.061694</td>
<td>1.060358</td>
<td>1.196426</td>
<td>1.196442</td>
</tr>
<tr>
<td>35</td>
<td>0.698294</td>
<td>-0.698739</td>
<td>-0.588043</td>
<td>-0.589614</td>
<td>1.185632</td>
<td>1.183845</td>
<td>1.396420</td>
<td>1.396589</td>
</tr>
<tr>
<td>40</td>
<td>0.798586</td>
<td>-0.798797</td>
<td>-0.746001</td>
<td>-0.748221</td>
<td>1.287414</td>
<td>1.285237</td>
<td>1.596559</td>
<td>1.597171</td>
</tr>
<tr>
<td>45</td>
<td>0.898108</td>
<td>-0.899157</td>
<td>-0.913735</td>
<td>-0.916425</td>
<td>1.365816</td>
<td>1.362978</td>
<td>1.797296</td>
<td>1.798217</td>
</tr>
<tr>
<td>50</td>
<td>0.998870</td>
<td>-0.999849</td>
<td>-1.087556</td>
<td>-1.090477</td>
<td>1.419794</td>
<td>1.416095</td>
<td>1.998628</td>
<td>1.999741</td>
</tr>
<tr>
<td>55</td>
<td>1.100884</td>
<td>-1.100884</td>
<td>-1.264220</td>
<td>-1.266539</td>
<td>1.448798</td>
<td>1.444230</td>
<td>2.201768</td>
<td>2.201768</td>
</tr>
<tr>
<td>60</td>
<td>1.202052</td>
<td>-1.202051</td>
<td>-1.438490</td>
<td>-1.440597</td>
<td>1.453119</td>
<td>1.447657</td>
<td>2.404105</td>
<td>2.404104</td>
</tr>
<tr>
<td>65</td>
<td>1.303411</td>
<td>-1.303410</td>
<td>-1.607261</td>
<td>-1.608992</td>
<td>1.433653</td>
<td>1.427321</td>
<td>2.606825</td>
<td>2.606822</td>
</tr>
<tr>
<td>70</td>
<td>1.404946</td>
<td>-1.404945</td>
<td>-1.767019</td>
<td>-1.768211</td>
<td>1.391885</td>
<td>1.384743</td>
<td>2.809896</td>
<td>2.809892</td>
</tr>
<tr>
<td>75</td>
<td>1.506643</td>
<td>-1.506642</td>
<td>-1.914575</td>
<td>-1.915073</td>
<td>1.329841</td>
<td>1.321999</td>
<td>3.013293</td>
<td>3.013286</td>
</tr>
<tr>
<td>80</td>
<td>1.608484</td>
<td>-1.608484</td>
<td>-2.047140</td>
<td>-2.046817</td>
<td>1.250709</td>
<td>1.241641</td>
<td>3.216977</td>
<td>3.216968</td>
</tr>
<tr>
<td>85</td>
<td>1.710451</td>
<td>-1.710451</td>
<td>-2.162421</td>
<td>-2.161181</td>
<td>1.155400</td>
<td>1.146625</td>
<td>3.420910</td>
<td>3.420902</td>
</tr>
<tr>
<td>90</td>
<td>1.812522</td>
<td>-1.812522</td>
<td>-2.258670</td>
<td>-2.256459</td>
<td>1.049153</td>
<td>1.040206</td>
<td>3.625049</td>
<td>3.625044</td>
</tr>
<tr>
<td>95</td>
<td>1.914676</td>
<td>-1.914676</td>
<td>-2.334739</td>
<td>-2.331548</td>
<td>0.934722</td>
<td>0.925833</td>
<td>3.829360</td>
<td>3.829352</td>
</tr>
<tr>
<td>100</td>
<td>2.016889</td>
<td>-2.016889</td>
<td>-2.390085</td>
<td>-2.385964</td>
<td>0.815645</td>
<td>0.807041</td>
<td>4.033784</td>
<td>4.033778</td>
</tr>
<tr>
<td>105</td>
<td>2.119138</td>
<td>-2.119137</td>
<td>-2.424792</td>
<td>-2.419839</td>
<td>0.695428</td>
<td>0.687332</td>
<td>4.238276</td>
<td>4.238275</td>
</tr>
<tr>
<td>110</td>
<td>2.221997</td>
<td>-2.221996</td>
<td>-2.439535</td>
<td>-2.433907</td>
<td>0.577450</td>
<td>0.570065</td>
<td>4.442788</td>
<td>4.442794</td>
</tr>
<tr>
<td>115</td>
<td>2.323643</td>
<td>-2.323642</td>
<td>-2.435548</td>
<td>-2.429447</td>
<td>0.464860</td>
<td>0.458357</td>
<td>4.647271</td>
<td>4.647286</td>
</tr>
<tr>
<td>120</td>
<td>2.425853</td>
<td>-2.425852</td>
<td>-2.414560</td>
<td>-2.408232</td>
<td>0.360477</td>
<td>0.354982</td>
<td>4.851679</td>
<td>4.851706</td>
</tr>
<tr>
<td>125</td>
<td>2.528010</td>
<td>-2.528009</td>
<td>-2.378717</td>
<td>-2.372442</td>
<td>0.266712</td>
<td>0.262298</td>
<td>5.056009</td>
<td>5.056021</td>
</tr>
<tr>
<td>130</td>
<td>2.630096</td>
<td>-2.630095</td>
<td>-2.330508</td>
<td>-2.324578</td>
<td>0.185513</td>
<td>0.182194</td>
<td>5.260145</td>
<td>5.260191</td>
</tr>
<tr>
<td>135</td>
<td>2.732114</td>
<td>-2.732115</td>
<td>-2.272624</td>
<td>-2.267351</td>
<td>0.118313</td>
<td>0.116033</td>
<td>5.464191</td>
<td>5.464229</td>
</tr>
<tr>
<td>140</td>
<td>2.834062</td>
<td>-2.834063</td>
<td>-2.207918</td>
<td>-2.203598</td>
<td>0.066029</td>
<td>0.064664</td>
<td>5.668102</td>
<td>5.668124</td>
</tr>
<tr>
<td>145</td>
<td>2.935935</td>
<td>-2.935940</td>
<td>-2.139280</td>
<td>-2.136180</td>
<td>0.029046</td>
<td>0.028408</td>
<td>5.871856</td>
<td>5.871871</td>
</tr>
<tr>
<td>150</td>
<td>3.037730</td>
<td>-3.037735</td>
<td>-2.069538</td>
<td>-2.067891</td>
<td>0.007244</td>
<td>0.007077</td>
<td>6.075464</td>
<td>6.075460</td>
</tr>
<tr>
<td>155</td>
<td>3.139457</td>
<td>-3.139464</td>
<td>-2.001380</td>
<td>-2.001361</td>
<td>0.000000</td>
<td>0.000003</td>
<td>6.278929</td>
<td>6.278913</td>
</tr>
</tbody>
</table>
### Table 9.17  Analysis of cantilever beam with 4 four elements, $f_e = 0.020$ and LFI = 0.200

<table>
<thead>
<tr>
<th>State</th>
<th>$M$</th>
<th>$R$</th>
<th>$d_1$</th>
<th>$d_{exact}$</th>
<th>$d_2$</th>
<th>$d_{exact}$</th>
<th>$\beta$</th>
<th>$\beta_{exact}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>0.099620</td>
<td>-0.099634</td>
<td>-0.013004</td>
<td>-0.013206</td>
<td>0.198628</td>
<td>0.198581</td>
<td>0.199262</td>
<td>0.199240</td>
</tr>
<tr>
<td>10</td>
<td>0.199175</td>
<td>-0.199188</td>
<td>-0.051677</td>
<td>-0.052476</td>
<td>0.393296</td>
<td>0.393111</td>
<td>0.398370</td>
<td>0.398351</td>
</tr>
<tr>
<td>15</td>
<td>0.298796</td>
<td>-0.298806</td>
<td>-0.115186</td>
<td>-0.116931</td>
<td>0.580575</td>
<td>0.580019</td>
<td>0.597607</td>
<td>0.597592</td>
</tr>
<tr>
<td>20</td>
<td>0.398514</td>
<td>-0.398521</td>
<td>-0.202260</td>
<td>-0.205126</td>
<td>0.756981</td>
<td>0.755719</td>
<td>0.797038</td>
<td>0.797028</td>
</tr>
<tr>
<td>25</td>
<td>0.498359</td>
<td>-0.498364</td>
<td>-0.310722</td>
<td>-0.315084</td>
<td>0.919270</td>
<td>0.916888</td>
<td>0.996724</td>
<td>0.996719</td>
</tr>
<tr>
<td>30</td>
<td>0.598359</td>
<td>-0.598361</td>
<td>-0.438522</td>
<td>-0.444337</td>
<td>1.064511</td>
<td>1.060542</td>
<td>1.196718</td>
<td>1.196717</td>
</tr>
<tr>
<td>35</td>
<td>0.698534</td>
<td>-0.698535</td>
<td>-0.582787</td>
<td>-0.589979</td>
<td>1.190155</td>
<td>1.184114</td>
<td>1.397065</td>
<td>1.397067</td>
</tr>
<tr>
<td>40</td>
<td>0.798883</td>
<td>-0.798951</td>
<td>-0.740149</td>
<td>-0.747070</td>
<td>1.294083</td>
<td>1.285502</td>
<td>1.597113</td>
<td>1.597765</td>
</tr>
<tr>
<td>45</td>
<td>0.899409</td>
<td>-0.899463</td>
<td>-0.907351</td>
<td>-0.916938</td>
<td>1.374747</td>
<td>1.363174</td>
<td>1.797931</td>
<td>1.798817</td>
</tr>
<tr>
<td>50</td>
<td>1.000174</td>
<td>-1.000164</td>
<td>-1.080919</td>
<td>-1.091005</td>
<td>1.431074</td>
<td>1.416217</td>
<td>1.999266</td>
<td>2.000348</td>
</tr>
<tr>
<td>55</td>
<td>1.101166</td>
<td>-1.101166</td>
<td>-1.257689</td>
<td>-1.267030</td>
<td>1.462679</td>
<td>1.444274</td>
<td>2.202326</td>
<td>2.202333</td>
</tr>
<tr>
<td>60</td>
<td>1.202346</td>
<td>-1.202346</td>
<td>-1.432592</td>
<td>-1.441097</td>
<td>1.469664</td>
<td>1.447632</td>
<td>2.404690</td>
<td>2.404693</td>
</tr>
<tr>
<td>65</td>
<td>1.303716</td>
<td>-1.303715</td>
<td>-1.602491</td>
<td>-1.609487</td>
<td>1.452809</td>
<td>1.427225</td>
<td>2.607436</td>
<td>2.607433</td>
</tr>
<tr>
<td>70</td>
<td>1.405262</td>
<td>-1.405261</td>
<td>-1.763865</td>
<td>-1.768688</td>
<td>1.413473</td>
<td>1.384579</td>
<td>2.810527</td>
<td>2.810524</td>
</tr>
<tr>
<td>75</td>
<td>1.506968</td>
<td>-1.506966</td>
<td>-1.913491</td>
<td>-1.915518</td>
<td>1.353558</td>
<td>1.321769</td>
<td>3.013941</td>
<td>3.013935</td>
</tr>
<tr>
<td>80</td>
<td>1.608816</td>
<td>-1.608815</td>
<td>-2.048522</td>
<td>-2.047219</td>
<td>1.275460</td>
<td>1.241354</td>
<td>3.217640</td>
<td>3.217633</td>
</tr>
<tr>
<td>85</td>
<td>1.710789</td>
<td>-1.710789</td>
<td>-2.166566</td>
<td>-2.161529</td>
<td>1.181991</td>
<td>1.146289</td>
<td>3.421587</td>
<td>3.421579</td>
</tr>
<tr>
<td>90</td>
<td>1.812866</td>
<td>-1.812866</td>
<td>-2.265756</td>
<td>-2.256746</td>
<td>1.076294</td>
<td>1.039832</td>
<td>3.625741</td>
<td>3.625732</td>
</tr>
<tr>
<td>95</td>
<td>1.915025</td>
<td>-1.915024</td>
<td>-2.344794</td>
<td>-2.331769</td>
<td>0.961747</td>
<td>0.925433</td>
<td>3.830057</td>
<td>3.830049</td>
</tr>
<tr>
<td>100</td>
<td>2.017243</td>
<td>-2.017242</td>
<td>-2.402984</td>
<td>-2.386116</td>
<td>0.841852</td>
<td>0.806627</td>
<td>4.034492</td>
<td>4.034485</td>
</tr>
<tr>
<td>105</td>
<td>2.119498</td>
<td>-2.119497</td>
<td>-2.440240</td>
<td>-2.419923</td>
<td>0.720129</td>
<td>0.686912</td>
<td>4.239004</td>
<td>4.238996</td>
</tr>
<tr>
<td>110</td>
<td>2.221768</td>
<td>-2.221767</td>
<td>-2.457067</td>
<td>-2.433923</td>
<td>0.600022</td>
<td>0.569648</td>
<td>4.443535</td>
<td>4.443536</td>
</tr>
<tr>
<td>115</td>
<td>2.324032</td>
<td>-2.324031</td>
<td>-2.454544</td>
<td>-2.429397</td>
<td>0.484758</td>
<td>0.457946</td>
<td>4.648056</td>
<td>4.648064</td>
</tr>
<tr>
<td>120</td>
<td>2.426270</td>
<td>-2.426268</td>
<td>-2.434262</td>
<td>-2.408114</td>
<td>0.377292</td>
<td>0.354581</td>
<td>4.852519</td>
<td>4.852539</td>
</tr>
<tr>
<td>125</td>
<td>2.528460</td>
<td>-2.528459</td>
<td>-2.398262</td>
<td>-2.372256</td>
<td>0.280207</td>
<td>0.261916</td>
<td>5.056112</td>
<td>5.056921</td>
</tr>
<tr>
<td>130</td>
<td>2.630584</td>
<td>-2.630583</td>
<td>-2.348954</td>
<td>-2.324324</td>
<td>0.195644</td>
<td>0.181843</td>
<td>5.261110</td>
<td>5.261168</td>
</tr>
<tr>
<td>135</td>
<td>2.732646</td>
<td>-2.732646</td>
<td>-2.289018</td>
<td>-2.267033</td>
<td>0.125241</td>
<td>0.115726</td>
<td>5.465202</td>
<td>5.465293</td>
</tr>
<tr>
<td>140</td>
<td>2.834617</td>
<td>-2.834617</td>
<td>-2.221316</td>
<td>-2.203238</td>
<td>0.070145</td>
<td>0.064646</td>
<td>5.669173</td>
<td>5.669234</td>
</tr>
<tr>
<td>145</td>
<td>2.936524</td>
<td>-2.936526</td>
<td>-2.148801</td>
<td>-2.135785</td>
<td>0.030939</td>
<td>0.028242</td>
<td>5.873008</td>
<td>5.873048</td>
</tr>
<tr>
<td>150</td>
<td>3.038365</td>
<td>-3.038368</td>
<td>-2.074418</td>
<td>-2.067468</td>
<td>0.007701</td>
<td>0.007701</td>
<td>6.076711</td>
<td>6.076731</td>
</tr>
<tr>
<td>155</td>
<td>3.140141</td>
<td>-3.140147</td>
<td>-2.000924</td>
<td>-2.000005</td>
<td>0.000001</td>
<td>0.000001</td>
<td>6.280280</td>
<td>6.280282</td>
</tr>
<tr>
<td>State</td>
<td>M</td>
<td>R</td>
<td>d₁</td>
<td>d₁_exact</td>
<td>d₂</td>
<td>d₂_exact</td>
<td>β</td>
<td>β_exact</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
<td>---------</td>
<td>--------</td>
<td>----------</td>
<td>--------</td>
<td>----------</td>
<td>---------</td>
<td>---------</td>
</tr>
<tr>
<td>5</td>
<td>0.049 997</td>
<td>0.050 000</td>
<td>-0.002 499</td>
<td>-0.003 331</td>
<td>0.099 947</td>
<td>0.099 911</td>
<td>0.099 987</td>
<td>0.099 994</td>
</tr>
<tr>
<td>10</td>
<td>0.099 994</td>
<td>-0.099 997</td>
<td>-0.009 989</td>
<td>-0.013 305</td>
<td>0.199 640</td>
<td>0.199 323</td>
<td>0.199 972</td>
<td>0.199 988</td>
</tr>
<tr>
<td>15</td>
<td>0.149 993</td>
<td>-0.149 996</td>
<td>-0.022 452</td>
<td>-0.029 863</td>
<td>0.298 837</td>
<td>0.297 743</td>
<td>0.299 959</td>
<td>0.299 986</td>
</tr>
<tr>
<td>20</td>
<td>0.199 995</td>
<td>-0.199 998</td>
<td>-0.039 858</td>
<td>-0.05 2905</td>
<td>0.397 293</td>
<td>0.39 4684</td>
<td>0.399 952</td>
<td>0.399 989</td>
</tr>
<tr>
<td>25</td>
<td>0.250 000</td>
<td>-0.250 003</td>
<td>-0.062 164</td>
<td>-0.082 298</td>
<td>0.494 763</td>
<td>0.489 669</td>
<td>0.499 953</td>
<td>0.499 999</td>
</tr>
<tr>
<td>30</td>
<td>0.300 010</td>
<td>-0.300 013</td>
<td>-0.089 316</td>
<td>-0.117 866</td>
<td>0.591 005</td>
<td>0.587 232</td>
<td>0.599 962</td>
<td>0.600 019</td>
</tr>
<tr>
<td>35</td>
<td>0.350 025</td>
<td>-0.350 028</td>
<td>-0.121 249</td>
<td>-0.159 400</td>
<td>0.685 780</td>
<td>0.671 923</td>
<td>0.699 983</td>
<td>0.700 050</td>
</tr>
<tr>
<td>40</td>
<td>0.400 047</td>
<td>-0.400 050</td>
<td>-0.157 885</td>
<td>-0.206 657</td>
<td>0.788 853</td>
<td>0.774 853</td>
<td>0.800 166</td>
<td>0.800 094</td>
</tr>
<tr>
<td>45</td>
<td>0.450 076</td>
<td>-0.450 079</td>
<td>-0.199 134</td>
<td>-0.259 358</td>
<td>0.889 990</td>
<td>0.884 990</td>
<td>0.900 065</td>
<td>0.900 152</td>
</tr>
<tr>
<td>50</td>
<td>0.500 114</td>
<td>-0.500 116</td>
<td>-0.244 897</td>
<td>-0.317 195</td>
<td>0.990 666</td>
<td>0.985 666</td>
<td>1.000 130</td>
<td>1.000 227</td>
</tr>
<tr>
<td>55</td>
<td>0.550 161</td>
<td>-0.550 163</td>
<td>-0.295 063</td>
<td>-0.379 831</td>
<td>1.095 557</td>
<td>1.089 557</td>
<td>1.100 213</td>
<td>1.100 321</td>
</tr>
<tr>
<td>60</td>
<td>0.600 217</td>
<td>-0.600 219</td>
<td>-0.349 508</td>
<td>-0.446 902</td>
<td>1.199 442</td>
<td>1.193 442</td>
<td>1.200 317</td>
<td>1.200 435</td>
</tr>
<tr>
<td>65</td>
<td>0.650 285</td>
<td>-0.650 287</td>
<td>-0.408 100</td>
<td>-0.518 018</td>
<td>1.302 367</td>
<td>1.296 367</td>
<td>1.300 442</td>
<td>1.300 570</td>
</tr>
<tr>
<td>70</td>
<td>0.700 365</td>
<td>-0.700 366</td>
<td>-0.470 696</td>
<td>-0.592 771</td>
<td>1.405 293</td>
<td>1.400 293</td>
<td>1.400 590</td>
<td>1.400 729</td>
</tr>
<tr>
<td>75</td>
<td>0.750 457</td>
<td>-0.750 458</td>
<td>-0.537 144</td>
<td>-0.670 730</td>
<td>1.508 229</td>
<td>1.498 229</td>
<td>1.500 764</td>
<td>1.500 913</td>
</tr>
<tr>
<td>80</td>
<td>0.800 562</td>
<td>-0.800 563</td>
<td>-0.607 279</td>
<td>-0.751 452</td>
<td>1.611 195</td>
<td>1.601 195</td>
<td>1.600 964</td>
<td>1.601 124</td>
</tr>
<tr>
<td>85</td>
<td>0.850 681</td>
<td>-0.850 682</td>
<td>-0.680 930</td>
<td>-0.834 477</td>
<td>1.714 161</td>
<td>1.704 161</td>
<td>1.701 193</td>
<td>1.701 362</td>
</tr>
<tr>
<td>90</td>
<td>0.900 815</td>
<td>-0.900 816</td>
<td>-0.757 917</td>
<td>-0.919 339</td>
<td>1.817 133</td>
<td>1.807 133</td>
<td>1.801 450</td>
<td>1.801 630</td>
</tr>
<tr>
<td>95</td>
<td>0.950 965</td>
<td>-0.950 966</td>
<td>-0.838 049</td>
<td>-1.005 563</td>
<td>1.920 105</td>
<td>1.906 105</td>
<td>1.900 739</td>
<td>1.901 929</td>
</tr>
<tr>
<td>100</td>
<td>1.001 130</td>
<td>-1.001 131</td>
<td>-0.921 129</td>
<td>-1.092 671</td>
<td>2.023 087</td>
<td>2.012 087</td>
<td>2.002 060</td>
<td>2.002 260</td>
</tr>
<tr>
<td>105</td>
<td>1.051 312</td>
<td>-1.051 313</td>
<td>-1.066 952</td>
<td>-1.180 185</td>
<td>2.126 024</td>
<td>2.115 024</td>
<td>2.102 414</td>
<td>2.102 625</td>
</tr>
<tr>
<td>110</td>
<td>1.101 512</td>
<td>-1.101 512</td>
<td>-1.095 306</td>
<td>-1.267 631</td>
<td>2.228 973</td>
<td>2.218 973</td>
<td>2.202 802</td>
<td>2.202 024</td>
</tr>
<tr>
<td>115</td>
<td>1.151 729</td>
<td>-1.151 730</td>
<td>-1.185 971</td>
<td>-1.354 539</td>
<td>2.331 922</td>
<td>2.321 922</td>
<td>2.303 226</td>
<td>2.303 458</td>
</tr>
<tr>
<td>120</td>
<td>1.201 965</td>
<td>-1.201 965</td>
<td>-1.278 722</td>
<td>-1.440 449</td>
<td>2.434 876</td>
<td>2.424 876</td>
<td>2.403 687</td>
<td>2.403 930</td>
</tr>
<tr>
<td>125</td>
<td>1.252 219</td>
<td>-1.252 219</td>
<td>-1.373 329</td>
<td>-1.524 915</td>
<td>2.537 827</td>
<td>2.527 827</td>
<td>2.504 185</td>
<td>2.504 438</td>
</tr>
<tr>
<td>130</td>
<td>1.302 493</td>
<td>-1.302 493</td>
<td>-1.469 554</td>
<td>-1.607 504</td>
<td>2.640 777</td>
<td>2.630 777</td>
<td>2.604 985</td>
<td>2.604 985</td>
</tr>
<tr>
<td>135</td>
<td>1.352 786</td>
<td>-1.352 786</td>
<td>-1.567 157</td>
<td>-1.687 803</td>
<td>2.743 740</td>
<td>2.733 740</td>
<td>2.705 298</td>
<td>2.705 572</td>
</tr>
<tr>
<td>140</td>
<td>1.403 099</td>
<td>-1.403 099</td>
<td>-1.665 895</td>
<td>-1.765 418</td>
<td>2.847 712</td>
<td>2.837 712</td>
<td>2.805 914</td>
<td>2.806 198</td>
</tr>
<tr>
<td>145</td>
<td>1.453 432</td>
<td>-1.453 432</td>
<td>-1.765 518</td>
<td>-1.839 979</td>
<td>2.950 684</td>
<td>2.940 684</td>
<td>2.906 570</td>
<td>2.906 864</td>
</tr>
<tr>
<td>150</td>
<td>1.503 786</td>
<td>-1.503 786</td>
<td>-1.865 775</td>
<td>-1.911 144</td>
<td>3.053 666</td>
<td>3.043 666</td>
<td>2.906 864</td>
<td>2.906 864</td>
</tr>
</tbody>
</table>
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


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, 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
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 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
state 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