Computational Strategies and Numerics for High-Fidelity Finite Element Models

with Application to Aero-Engine Design

Von der Fakultät für Maschinenbau, Elektro- und Energiesysteme der
Brandenburgischen Technischen Universität Cottbus – Senftenberg
zur Erlangung des akademischen Grades
Doktor der Ingenieurwissenschaften habilitatus (Dr.-Ing. habil.)
genehmigte Habilitationsschrift

von

Dr.-Ing. Markus Kober

geboren am 13.07.1979 in Cottbus

Das Habilitationsverfahren wurde durch den Fakultätsrat der Fakultät für Maschinenbau, Elektro- und Energiesysteme der Brandenburgischen Technischen Universität Cottbus – Senftenberg am 28.08.2019 eröffnet. Die Zuerkennung der Lehrbefähigung für das Fachgebiet „Technische Mechanik“ erfolgte am 26.02.2020.

Mitglieder der Habilitationskommission:

Prof. Dr.-Ing. Ralf Woll* (Vorsitzender)
Prof. Dr.-Ing. Arnold Kühhorn* (Gutachter)
Prof. Dr.-Ing. habil. Dieter Bestle* (Gutachter)
Prof. Dr.-Ing. habil. Maik Gude** (Gutachter)
Prof. Dr.-Ing. Christoph Egbers* (weiteres Mitglied)

---

* Brandenburgische Technische Universität Cottbus-Senftenberg

** Technische Universität Dresden

Preface

This habilitation thesis arose during my postdoctoral employment at the chair of Structural Mechanics and Vehicle Vibration Technology of the Brandenburg University of Technology Cottbus-Senftenberg. First of all, I would like to express my gratitude to Prof. Dr.-Ing. Arnold Kühhorn for giving me the opportunity for my research and for acting as supervisor. I would also like to thank my further supervisors Prof. Dr.-Ing. habil. Dieter Bestle especially for his ideas concerning improvements of this work and Prof. Dr.-Ing. habil. Maik Gude. Apart from the supervisors I would like to thank Prof. Dr.-Ing. Ralf Woll for chairing the habilitation committee and Prof. Dr.-Ing. Christoph Egbers as further member of the committee.

I am also pleased to thank my colleague Mark Golze for his support with regard to the necessary supercomputer computations, which were only possible due to his knowledge in this field.

Finally, I would like to thank my colleagues Dr. Akin Keskin, Andrew Layton, Dr. James Ong, Dr. Michael Lawson, Kai Singh and Michelle Tindall from Rolls-Royce for many fruitful discussions and provision of data and models.

Cottbus, May 2020

Content

Symbols ..... **XI**

Abbreviations ..... **XVII**

1 Introduction **19**

1.1 Classical Whole Engine Models 20

1.2 High-Fidelity Whole Engine Models 20

1.3 Focus and Outline of the Thesis 21

2 Contact Modeling **23**

2.1 Mechanical Basics 23

2.2 Consideration of Contact 26

2.2.1 Lagrange Multiplier Method 27

2.2.2 Penalty Method 29

2.2.3 Augmented Lagrange Method 31

2.2.4 Perturbed Lagrange Method 33

2.3 Determination of Gap Size 34

2.3.1 Node-to-Node Approach 34

2.3.2 Node-to-Surface Approach 35

2.3.3 Surface-to-Surface Approach 37

2.3.4 Mortar Approach 39

3 Choice of Proper Finite Element Formulations and Numerical Problems **41**

3.1 Locking of Continuum Elements 43

3.1.1 Shear Locking 43

3.1.2 Trapezoidal Locking 52

3.1.3 Volumetric Locking 55

3.2 Locking of Structural Elements 59

3.2.1 Curvature-Thickness Locking 61

3.2.2 Poisson Locking 61

3.2.3 Transverse Shear Locking 62

3.2.4 Membrane Locking 64

3.3 Concepts to Avoid Locking 65
3.3.1 Numerical Integration 65
3.3.2 Reduced Integration 69
3.3.3 Assumed Natural Strain Method 73
3.3.4 Discrete Strain Gap Method 73
3.3.5 Enhanced Assumed Strain Method 75
3.3.6 Problems of Reduced Integration and the Hourglassing Phenomenon 75
3.3.7 Concepts to Avoid Hourglassing 77
3.4 Three-Point-Bending Test Problem 80
3.4.1 Results of Displacement-Based Loading 81
3.4.2 Results of Forced-Based Loading 83

4 Modeling of Bearings in High Fidelity Engine Models 85
4.1 Necessary Mesh Density for Classical Contact Formulations 85
4.2 Mesh Density for Artificially Smoothed Contact Formulations 90
4.2.1 Construction of Quadratic Polynomials 91
4.2.2 Transformation of Polynomial to the *i*-th Edge of the Polygon 91
4.2.3 Smoothing of Second Contact Surface 94
4.2.4 Condition for Rotation of Smoothed Surfaces 95

5 Time Integration 99
5.1 Introduction 99
5.2 Explicit Time-Integration 100
5.2.1 One-Step Methods 101
5.2.2 Multi-Step Methods (The Central Difference Method) 102
5.3 Implicit Time-Integration 104
5.3.1 Newmark Algorithm 104
5.3.2 Hilber-Hughes-Taylor Algorithm 107
5.3.3 Newmark-Euler Algorithm 107
5.3.4 Solution of Nonlinear Equations 109
5.4 Examples 114
5.4.1 Time Integration of Equation of Motion for Moving Plate 114

5.4.2 1-DOF Vibration 118
5.4.3 Nonlinear 1-DOF Vibration 122

6 Stability and Accuracy Considerations of Time-Integration Algorithms 125

6.1 Stability of Linear Problems 126
6.1.1 Newmark Algorithm 128
6.1.2 HHT Algorithm 131
6.1.3 Newmark-Euler Algorithm (Composite Scheme) 133
6.1.4 Central Difference Method 137
6.2 Stability of Nonlinear Problems 140
6.3 Period Elongation 144
6.4 Amplitude Error 145

7 Implicit Time-Integration of Fast Rotating Structures 147

7.1 The Instability Problem 147
7.2 Explanation of the Instability Phenomenon 150
7.2.1 Rigid Rotating Pendulum 150
7.2.2 Elastic Rotating Pendulum 153
7.3 Influence of Stiffness and Time-Step Size 156
7.4 Possibilities for Resolving the Instability Problem 158
7.4.1 First Strategy 158
7.4.2 Second Strategy 160
7.5 Finite Element Examples 160
7.6 Optimization of Time-Integration Parameters 161

8 Model Reduction Strategy and Dummy Engine Model 167

8.1 Model Reduction via Solid-by-Shell Substitution 167
8.2 Considerations about Necessary Mesh Density 168
8.2.1 Beam with Constant Bending Moment 168
8.2.2 Beam with Variable Bending Moment 171
8.3 Tests and Examples of Solid-by-Shell Substitution 173
8.4 Dummy Engine Model 175
8.4.1 Applied Test Load Case and Results 176

8.4.2 Test of Model Reduction Strategy 178

9 Conclusions and Outlook 181

Appendix 183

A Computation of Eigenvalues for Stability Analysis of Newmark Algorithm..... 183

B Derivation of Stability Conditions for Newmark Time-Integration 184

C Computation of Eigenvalues for Stability Analysis of Newmark-Euler Algorithm. 190

D Derivation of Stability Conditions for Newmark-Euler Time-Integration 192

E Derivation of Stability Conditions for Central Difference Method..... 194

F Deflection Function of Thickness-Variable Beam 196

References 199

Symbols

Latin Symbols

$\boldsymbol{A}$	Director of shell element in undeformed configuration
$\boldsymbol{a}$	Director of shell element in deformed configuration
A	Area
	Polynomial constant
A_S	Shear deformation area
a	Distance
	Interval limit
a_i	Constant of one-step integration scheme
B	Polynomial constant
b	Distance
	Interval limit
	Width
C	Integration constant
	Polynomial constant
c	Edge length of polygon
	Wave propagation speed
$\boldsymbol{d}_A$, $\boldsymbol{d}_B$	Position vectors
d	Damping constant
d_i'	Constant of discrete transfer function
E	Young's modulus
E_a	Amplitude error
E_p	Period elongation
${}^{Q}E$	Fictive material parameter
$\boldsymbol{F}$	Vector of forces
F	Force
$\boldsymbol{f}^B$	Vector of body forces
$\boldsymbol{f}^B$	Vector of surface forces

$\boldsymbol{f}^{int}$, $\boldsymbol{f}^{ext}$	Vector of internal or external loads
F_c	Centripetal force
$\boldsymbol{g}$	Gravitational vector
G	Shear modulus
$G(z)$	Discrete transfer function
g_0	Initial normal distance
g_N	Normal distance
$\bar{g}_N$	Average normal distance
h	Height
	Interval width of segment in numerical integration
I	Moment of inertia
i	Control variable
	Imaginary unit
j	Control variable
k	Spring stiffness
k_t	Tangent stiffness
k_z	Shear correction parameter
$\boldsymbol{L}$	Load operator
$L(\boldsymbol{x}, \boldsymbol{\lambda})$	Lagrange function
l	Length
l_0	Undeformed length
M	Moment
m	Number of segments for numerical integration
	Mass
n	Number of finite elements
	Number of nodes in circumferential direction
	Number of support points for numerical integration
	Order of polynomial
$\boldsymbol{n}_A$, n_j	Normal unit vector
n_i'	Constant of discrete transfer function

p	Pressure
p_i	Constant of one-step integration scheme
Q	Shear force
Q_z	Shear force in z-direction
q_{ij}	Constant of one-step integration scheme
R	Radius
s	Slack variable
$\boldsymbol{t}$	Tangential unit vector
T	Period
t	Thickness
	Time
t_0, t_1	Beam thickness at end A and B
t_c	Constant beam thickness
t_{gap}	Gap size
t_n	Time point
${}^{in}\boldsymbol{u}$, ${}^{out}\boldsymbol{u}$	Translation vectors
V	Volume
v	Velocity
W_{AL}^C, W_L^C, W_P^C, W_{PL}^C	Contact energy for augmented Lagrange (AL), Lagrange (L), penalty (P) and perturbed Lagrange (PL) formulation
w	Displacement of beam in z-direction
w_k	Weighting factor for Gauss integration
w_γ	Shear displacement
w_φ	Bending displacement
$\boldsymbol{x}$	Displacement vector
x	Cartesian coordinate
	Displacement
${}^h x$	Interpolated displacement
x_i^s	Displacement constraint
x_k	Support point for numerical integration
y	Cartesian coordinate

z	Cartesian coordinate
	Substitution variable
z_1, z_2, z_3, z_4	Substitution variables

Greek Symbols

α	Angle
	Parameter of Composite algorithm
	Parameter of HHT algorithm
	Parameter of penalty contact formulation
	Taper angle
β	Inner angle of polygon
	Parameter of Newmark algorithm
β_N	Parameter of penalty contact formulation
γ	Parameter of Newmark algorithm
	Variable of Quasi-Newton approach
${}^{in}\gamma_i, {}^{out}\gamma_i$	Angle of rotation for *i*-th edge
$\gamma_{xy}, \gamma_{xz}, \gamma_{yz}$	Shear strain coordinates
$\widetilde{\boldsymbol{\gamma}}$	Vector of generalized strains
δ	Symbol indicating virtual quantities
	Variable of Quasi-Newton approach
ε	Penetration tolerance
	Parameter of penalty contact formulation
	Variable for determination of period elongation
$\varepsilon_x, \varepsilon_y, \varepsilon_z, \varepsilon_{xy}$	Strain coordinates
ζ	Coordinate of element coordinate system
	Coordinate of intermediate contact surface
η	Coordinate of element coordinate system
κ	Curvature

λ	Eigenvalue
	Lagrange multiplier
μ	Damping constant
ν	Poisson's ratio
ξ	Coordinate of element coordinate system
	Parameter of Newton-Raphson algorithm with line search
$\Pi(\boldsymbol{x})$	Energy functional
Π_{AL}, Π_{L}, Π_{P}, Π_{PL}	Total potential energy functional in augmented Lagrange (AL) , Lagrange (L), penalty (P) and perturbed Lagrange (PL) formulation
ρ	Density
	Spectral radius
σ_x, σ_y, σ_z	Normal stress coordinates
τ_{xy}, τ_{xz}, τ_{yz}	Shear stress coordinates
$\boldsymbol{\Phi}(\boldsymbol{x}_n, t_n, \Delta t)$	Function for definition of different Runge-Kutta methods
$\Phi(y)$	Integration function
φ	Rotation of node
	Angle
φ_0	Phase shift
$\Psi(x)$	Integration function
ω	Eigenfrequency
	Rotational velocity

Matrices and Tensors

$\boldsymbol{A}$	Matrix of time-integration algorithm
$\boldsymbol{B}$	Strain-displacement matrix
$\widetilde{\boldsymbol{B}}$	Modified strain-displacement matrix
$\boldsymbol{B}_B$	Bending part of strain-displacement matrix
$\boldsymbol{B}_S$	Shear part of strain-displacement matrix
$\boldsymbol{D}$	Damping matrix

$\boldsymbol{E}$	Elasticity matrix
$\boldsymbol{E}_S$	Shear part of elasticity matrix
$\boldsymbol{E}_B$	Bending part of elasticity matrix
$\widetilde{\boldsymbol{E}}$	Modified elasticity matrix
$\boldsymbol{\varepsilon}$, ε_{ij}	Strain tensor
$\boldsymbol{J}$	Jordan canonical form of matrix
$\boldsymbol{K}$	Stiffness matrix
$\boldsymbol{K}_e$	Element stiffness matrix
$\boldsymbol{M}$	Mass matrix
$\boldsymbol{N}$	Matrix of interpolation functions
$\boldsymbol{Q}$	Matrix of eigenvectors
$\boldsymbol{\sigma}$, σ_{ij}	Stress tensor
$\widetilde{\boldsymbol{\sigma}}$	Generalized stress tensor

Others

$\forall$	For all
$\wedge$	Logical AND
$\vee$	Logical OR
Δt	Time-step size
da	Total derivative of a
∂a	Partial derivative of a
δW^C, δW^{ext}, δW^{int}, δW^{kin}	Virtual contact energy, virtual external work, virtual internal work and virtual kinetic energy
$\delta \boldsymbol{x}$, δx_i	Virtual displacements
$\det$	Determinant
div	Divergence operator
$\in$	Element of
∇	Gradient operator
∞	Infinity

$\|\|\dots\|\|_2$	Euclidean norm
$\dot{a}$	Derivative of a with respect to time

Upper Indices

$^{-1}$	Inverse
^{T}	Transposed
$'$	Derivative

Abbreviations

ANS	Assumed natural strain
BFGS	Broyden-Fletcher-Goldfarb-Shanno
CPU	Central processing unit
DOF	Degree of freedom
DSG	Discrete strain gap
EAS	Enhanced assumed strain
FEM	Finite element method
HHT	Hilber-Hughes-Taylor
IPC	Intermediate pressure compressor
KKT	Karush-Kuhn-Tucker
LPT	Low pressure turbine
MMD	Multiple minimum degree
NDOF	Number of degrees of freedom
OGV	Outlet guide vanes
TBH	Thrust bearing housing
WEM	Whole engine model

1 Introduction

Due to the availability of enormous computational power and decreasing CPU costs, finite element models are becoming more and more complex in all fields of engineering. In this context, also aero-engine manufacturers try to model the thermo-mechanical behavior of their engines as accurately as possible. In the past, this was done by simplified Whole Engine models (classical WEM's), but nowadays high-fidelity Whole Engine Models are currently being developed. These models contain no simplifications or idealizations as used in classical WEM's, which leads on the one hand to the possibility of getting more accurate results in terms of displacements and stresses, but on the other hand causes a bunch of challenges with respect to the modeling strategy and numerics. Some of these challenges and numerical problems will be addressed and possible solution strategies will be developed.

In commercial finite element codes many different finite element formulations are available. Most of these standard elements work well for "normal" applications. But, for example, for poor aspect ratios the choice of the finite element formulation is essential to prevent unwanted phenomena like locking or hourglassing.

Since there are almost no simplifications or idealizations in high fidelity WEM's, all types of nonlinearity are present (material, geometrical and contact nonlinearities). Especially the contact behavior is a major challenge in models with high complexity such as aero engines. For this reason, it is important to understand the different algorithms and principles of contact mechanics. Also the time-step size of numerical integration is influenced significantly by the chosen contact algorithm which has a direct impact on the overall computational time. If there should be no simplifications or idealizations, special attention has to be given to parts like bearings of the engine rotor. Here, we will answer the question about the necessary mesh density for such parts depending on the used contact formulation.

For the solution of the equations of motion of a thermo-mechanical system, numerical time-integration schemes are necessary. There exist basically two different approaches for numerical integration, implicit and explicit schemes. The latter are suited in particular for highly dynamic processes, but could also be used for long term computations. This might be advantageous because of the simplicity and stability of explicit time-integration. On the other hand there is a need of extremely small time steps of these integration algorithms. Implicit time-integration is more complex, but allows for much bigger time-steps which, however, can cause difficulties with respect to contact problems. In this work both possibilities of time-integration are discussed in detail with all advantages and disadvantages especially in the context of rotating flexible structures.

Typical high-fidelity Whole Engine Models have millions of degrees of freedom, which leads to very high computational costs. There are different possibilities of "model reduction" techniques aiming at reducing the number of degrees of freedom while keeping the accuracy as high as possible. One of these techniques will be discussed and finally its capabilities for

academic examples as well as for the example of a Dummy Whole Engine Model will be investigated.

1.1 Classical Whole Engine Models

Classical Whole Engine Models mainly consist of shell and beam elements (Fig. 1.1, beam elements are not displayed). All components of the engine are simplified or idealized. There are also parts which are even not present in the model due to their minor impact on the structural behavior of the engine. Such models are typically used to compute eigenfrequencies and eigenmodes, the overall displacement behavior, but also details like tip clearances, which describe the gap size between rotor and casing. The latter has a big influence on the engine's efficiency for example.

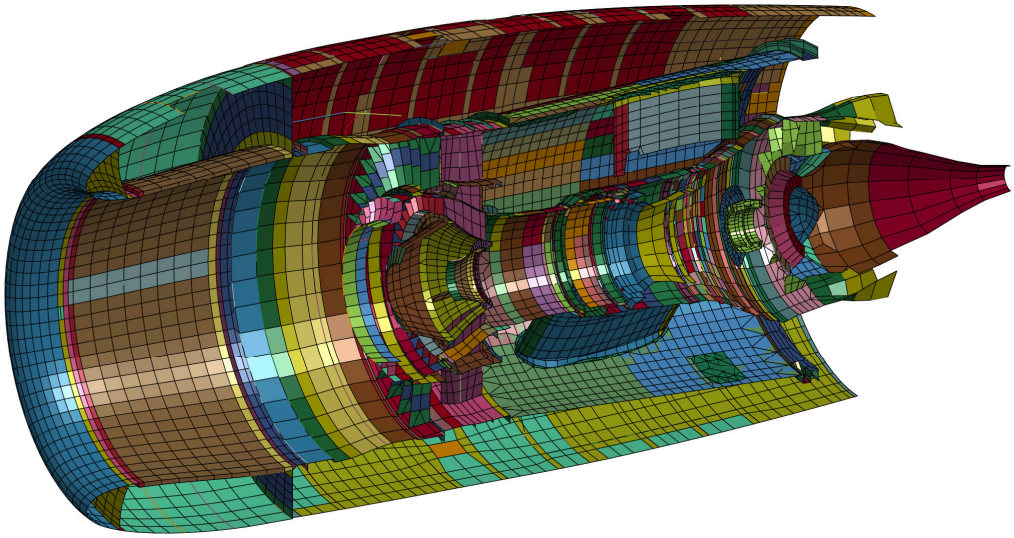


Fig. 1.1 Cut through a classical Whole Engine Model

Often model reduction concepts like static/dynamic condensation or super elements are used in classical Whole Engine Models [37]. The big advantage of such models is their low number of degrees of freedom (DOF) and, therefore, their low computational time. Linear static or dynamic computations can be performed within minutes or even seconds. Of course the accuracy of such models is not high and the results can only be used for general statements about displacements and eigenfrequencies.

1.2 High-Fidelity Whole Engine Models

In contrast to classical Whole Engine Models, a high-fidelity Whole Engine Model is defined as a model which requires almost no simplification or idealization. Every part of the engine, every bolt and every washer is modeled by solid elements to represent the geometry as accurately as possible (Fig. 1.2). The intention of such models is to compute the behavior of a running aero engine over a time-span of a few seconds by applying the correct transient pressure and temperature distributions. This strategy has many advantages but also some

drawbacks. The biggest advantage is that such a high-fidelity model offers the possibility of studying the transient behavior of an engine in detail. Another advantage is the higher accuracy that can be expected from such models because there are far less assumptions necessary than in a simplified model. Fewer assumptions and fewer idealizations mean also less possible errors and approximations of stiffness values, geometry and so on. It is also possible to take into account many nonlinear effects and phenomena which are just not visible in simplified models or do not appear. On the other hand the complexity of the model increases dramatically. The number of degrees of freedom is a few orders higher than in classical models, there exists nonlinear contact between hundreds or thousands of parts, nonlinear material behavior, and also nonlinear geometric effects are considered. By this, the computational time is much higher than for classical models, but with an increasing availability of CPU's and computational power the tendency is towards more complex high-fidelity models.

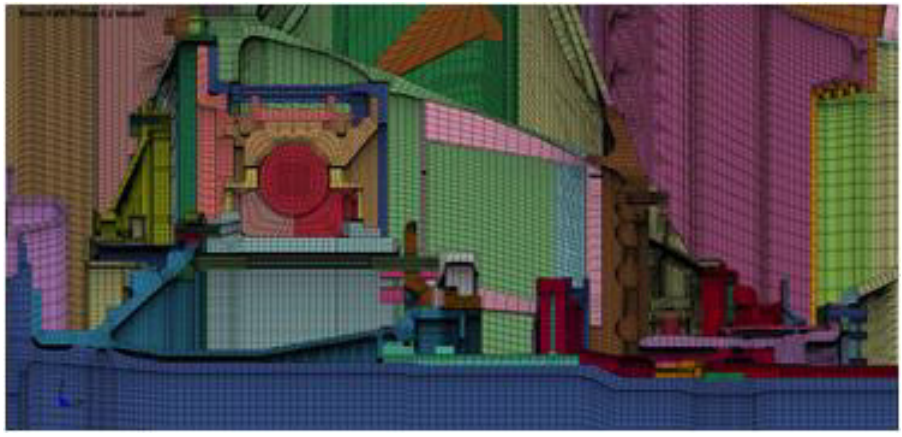


Fig. 1.2 Section of a high-fidelity Whole Engine Model

1.3 Focus and Outline of the Thesis

The intention of this thesis is to discuss the kind of problems a user is faced when dealing with transient high-fidelity models. To take into account nonlinear effects like contact, it is important to understand contact modeling algorithms to be able to detect possible modeling errors or convergence problems (Chapter 2). Also unwanted stiffening or softening effects like locking or hourglassing may appear. It is important to recognize the origins of such effects for being able to avoid them with an appropriate strategy. The main goal is to give the reader an idea about the mentioned origins and strategies (Chapter 3). For special topics and more detailed discussions the reader is referred to further publications. Particular attention is paid here to the time-integration algorithms and their stability (Chapter 5 and 6). The latter is a very important aspect if elastic rotating structures with implicit time-integration are considered (Chapter 7). Also the necessary mesh density in bearings and for a developed solid-by-shell substitution strategy is investigated in detail (Chapter 4). For all topics under discussion simple examples are given which should illustrate the described effect or theory.

Finally, a complex aero-engine model is used for demonstrating the effectiveness of the derived procedures and approaches (Chapter 8). Material models and related questions are not part of the considerations here.

2 Contact Modeling

A typical turbo-fan aero engine consists of thousands of single parts which are in contact with each other. Some of these contacts may be simplified by defining a tie contact, for example between bolt and washer, but there are also many sliding contacts, where a relative motion between both contact surfaces may occur. The latter ones may also open and close which makes their treatment much more complex than for tie contacts. For this reason, the mechanical basics and different contact descriptions are discussed in this section.

The intention of this chapter is to provide at least an idea about the different types of contact formulations and a general overview of algorithms, which are used in commercial FE codes. By this, a deeper understanding of contact problems should be gained that helps to solve possible convergence problems in contact computations. An extremely detailed discussion of contact descriptions between different types and dimensionality of surfaces (edge-to-beam, edge-to-surface, …) is given in special textbooks [79,80,148]. A historic overview of contact formulations used in the finite element method (FEM) may be found in [45].

At the end of this chapter special attention is paid to contact formulations that are suited for rotating contact surfaces. As it will be demonstrated, there may be a big influence of the used contact formulation onto the time-step size used in an implicit time-integration scheme.

2.1 Mechanical Basics

The foundation for the displacement-based finite element method [11,66] is the principle of virtual work. In the following, this principle is derived from the equilibrium conditions in the current configuration which hold for any point of a continuous body and read in index notation as

$$\operatorname{div} \sigma_{ij} + f_i^B = 0 \tag{2.1}$$

where σ_{ij} is the Cauchy stress tensor and f_i^B is the vector of body forces [65]. These equilibrium conditions follow directly from the equilibrium of forces under static conditions. A body under consideration may be subjected to force boundary conditions on its surface according to

$$\sigma_{ij} n_j = f_i^S \tag{2.2}$$

with n_j being the normal unit vector on its surface and f_i^S surface tractions, and displacement constraints

$$x_i = x_i^S \tag{2.3}$$

on the surface. Equation (2.1) is now multiplied by so-called virtual displacements δx_i (or more generally spoken test functions), which have to be kinematically admissible. This means that the test functions have to be C^0 continuous and vanish at displacement boundaries, which yields

$$\left(\operatorname{div}\sigma_{ij} + f_i^B\right)\delta x_i = 0. \tag{2.4}$$

If (2.4) is fulfilled also

$$\int_V \left(\operatorname{div}\sigma_{ij} + f_i^B\right)\delta x_i \,\mathrm{d}V = 0 \tag{2.5}$$

is satisfied. By applying the product rule of differentiation

$$\frac{\partial\left(\sigma_{ij}\delta x_i\right)}{\partial x_j} = \frac{\partial\sigma_{ij}}{\partial x_j}\delta x_i + \sigma_{ij}\frac{\partial\delta x_i}{\partial x_j} \tag{2.6}$$

Equation (2.5) can be written as

$$\int_V \left(\frac{\partial\left(\sigma_{ij}\delta x_i\right)}{\partial x_j} - \sigma_{ij}\frac{\partial\delta x_i}{\partial x_j} + f_i^B\delta x_i\right)\mathrm{d}V = 0. \tag{2.7}$$

Furthermore, Gauss's theorem [21]

$$\int_V \frac{\partial\left(\sigma_{ij}\delta x_i\right)}{\partial x_j}\mathrm{d}V = \int_A \left(\sigma_{ij}\delta x_i\right)n_j \,\mathrm{d}A \tag{2.8}$$

can be used to transform (2.7) into

$$\int_V \left(-\sigma_{ij}\frac{\partial\delta x_i}{\partial x_j} + f_i^B\delta x_i\right)\mathrm{d}V + \int_A \left(\sigma_{ij}\delta x_i\right)n_j \,\mathrm{d}A = 0. \tag{2.9}$$

With (2.2) and by taking (2.3) into account, this yields

$$\int_V \left(-\sigma_{ij}\frac{\partial\delta x_i}{\partial x_j} + f_i^B\delta x_i\right)\mathrm{d}V + \int_A f_i^S\delta x_i^S \,\mathrm{d}A = 0. \tag{2.10}$$

Due to the symmetry of the stress tensor,

$$\sigma_{ij}\frac{\partial\delta x_i}{\partial x_j} = \sigma_{ij}\left[\frac{1}{2}\left(\frac{\partial\delta x_i}{\partial x_j} + \frac{\partial\delta x_j}{\partial x_i}\right)\right] = \sigma_{ij}\delta\varepsilon_{ij} \tag{2.11}$$

holds and we get the so-called principle of virtual work

$$\int_V \sigma_{ij}\delta\varepsilon_{ij} \,\mathrm{d}V = \int_V f_i^B\delta x_i \mathrm{d}V + \int_A f_i^S\delta x_i^S \,\mathrm{d}A. \tag{2.12}$$

On the left-hand side of this equation we have the virtual inner work of the body δW^{int} and on the right hand side the virtual work of the externally applied loads δW^{ext} as sum of the work of body forces and surface tractions which finally results in

$$\delta W^{int} = \delta W^{ext}. \tag{2.13}$$

If we do not consider a static problem and high accelerations are present, the body forces also include the d'Alembert forces. Often it is useful to separate the virtual work of the d'Alembert forces δW^{kin} from the virtual work of the other body forces and the principle of virtual work becomes

$$\delta W(\delta \boldsymbol{x}, \boldsymbol{x}) \equiv \delta W^{int} - \delta W^{ext} + \delta W^{kin} = 0. \tag{2.14}$$

In the finite element world, the displacements ${}^{i}\boldsymbol{x}^{h}$ between the nodal displacements $\boldsymbol{x}$ are interpolated with the help of interpolation functions ${}^{i}\boldsymbol{N}$ as

$${}^{i}\boldsymbol{x}^{h} = {}^{i}\boldsymbol{N}\boldsymbol{x}, \tag{2.15}$$

where the upper index i indicates the i-th element and $\boldsymbol{x}$ is the nodal displacement vector of all node points. The strain matrix of the i-th element is defined as

$${}^{i}\boldsymbol{\varepsilon} = {}^{i}\boldsymbol{B}\boldsymbol{x}, \tag{2.16}$$

with ${}^{i}\boldsymbol{B}$ as the strain-displacement matrix (calculated with the help of the displacement interpolation functions). The stresses ${}^{i}\boldsymbol{\sigma}$ in an element (neglecting initial stresses) are related to the strains via the elasticity matrix ${}^{i}\boldsymbol{E}$:

$${}^{i}\boldsymbol{\sigma} = {}^{i}\boldsymbol{E}\,{}^{i}\boldsymbol{\varepsilon}. \tag{2.17}$$

Also the virtual quantities are calculated in a similar fashion according to

$$\delta {}^{i}\boldsymbol{x}^{h} = {}^{i}\boldsymbol{N}\delta\boldsymbol{x} \tag{2.18}$$

and

$$\delta {}^{i}\boldsymbol{\varepsilon} = {}^{i}\boldsymbol{B}\delta\boldsymbol{x}. \tag{2.19}$$

The principle of virtual work is now applied by computing the integrals of (2.12) for each element and summing up these quantities over all elements which leads to

$$\sum_{i} \int_{{}^{i}V} \delta {}^{i}\boldsymbol{\varepsilon}^{T}\,{}^{i}\boldsymbol{\sigma}\,\mathrm{dV} = \sum_{i} \int_{{}^{i}V} \delta {}^{i}\boldsymbol{x}^{h^{T}}\,{}^{i}\boldsymbol{f}^{B}\mathrm{d}V + \sum_{i} \int_{{}^{i}A} \delta {}^{i}\boldsymbol{x}^{h^{T}}\,{}^{i}\boldsymbol{f}^{S}\mathrm{d}A. \tag{2.20}$$

Using (2.18) and (2.19) finally yields together with (2.17) and (2.16)

$$\delta\boldsymbol{x}^{T}\left(\underbrace{\sum_{i} \underbrace{\int_{{}^{i}V} {}^{i}\boldsymbol{B}^{T}\,{}^{i}\boldsymbol{E}\,{}^{i}\boldsymbol{B}\mathrm{d}V}_{{}^{i}\boldsymbol{K}}}_{\boldsymbol{K}}\right)\boldsymbol{x} = \delta\boldsymbol{x}^{T}\left(\underbrace{\sum_{i} \underbrace{\int_{{}^{i}V} {}^{i}\boldsymbol{N}^{T}\,{}^{i}\boldsymbol{f}^{B}\mathrm{d}V}_{{}^{i}\boldsymbol{F}^{B}}}_{\boldsymbol{F}^{B}} + \underbrace{\sum_{i} \underbrace{\int_{{}^{i}A} {}^{i}\boldsymbol{N}^{T}\,{}^{i}\boldsymbol{f}^{S}\mathrm{d}A}_{{}^{i}\boldsymbol{F}^{S}}}_{\boldsymbol{F}^{S}}\right). \tag{2.21}$$

Since the coordinates of $\delta\boldsymbol{x}$ can be chosen arbitrarily and the load vector on the right-hand side of (2.33) can be summarized as $\boldsymbol{F} = \boldsymbol{F}^{B} + \boldsymbol{F}^{S}$ the well-known finite element equation

$$\boldsymbol{K}\boldsymbol{x} = \boldsymbol{F} \tag{2.22}$$

remains with the global stiffness matrix $\boldsymbol{K}$, the global displacement vector $\boldsymbol{x}$ and the global load vector $\boldsymbol{F}$.

If the considered body is accelerated, the body forces are extended by the d'Alembert forces according to

$$\boldsymbol{F}^{B} = \sum_{i} \int_{{}^{i}V} {}^{i}\boldsymbol{N}^{T}\left({}^{i}\boldsymbol{f}^{B} - {}^{i}\rho\,{}^{i}\boldsymbol{N}\ddot{\boldsymbol{x}}\right)\mathrm{d}V \tag{2.23}$$

with the acceleration vector $\ddot{\boldsymbol{x}}$ and the element density ${}^{i}\rho$. As mentioned before, it is useful for many applications to separate the d'Alembert forces from other body forces which can be done by extracting the mass matrix $\boldsymbol{M}$ from (2.23) for an element-wise constant density ${}^{i}\rho$ as

$$\boldsymbol{M} = \sum_{i} {}^{i}\rho \int_{{}^{i}V} {}^{i}\boldsymbol{N}^{T}\,{}^{i}\boldsymbol{N}\,\mathrm{d}{}^{i}V. \tag{2.24}$$

Analogously the damping matrix $\boldsymbol{D}$ can be computed in case that the damping is proportional to the velocity vector $\dot{\boldsymbol{x}}$, which yields after separation from the body forces

$$\boldsymbol{D} = \sum_{i} {}^{i}\mu \int_{{}^{i}V} {}^{i}\boldsymbol{N}^{T}\,{}^{i}\boldsymbol{N}\,\mathrm{d}{}^{i}V \tag{2.25}$$

with the damping constant ${}^{i}\mu$ of the i-th element. Taking into account d'Alembert forces and damping forces, the equations of motion for a finite element system become

$$\boldsymbol{M}\ddot{\boldsymbol{x}} + \boldsymbol{D}\dot{\boldsymbol{x}} + \boldsymbol{K}\boldsymbol{x} = \boldsymbol{F}. \tag{2.26}$$

2.2 Consideration of Contact

If contact has to be considered in the model, an extra term for the contact energy δW^{C} has to be added to Equation (2.14). Depending on this energy expression, four different contact formulations, that are often used, are briefly discussed:

1. Lagrange multiplier method,
2. Penalty method,
3. Augmented Lagrangian method, and
4. Perturbed Lagrangian method.

In the following, the basic ideas of these concepts are outlined and explained in a few words. For the sake of simplicity, we restrict all further explanations to static conservative systems (no energy dissipation). By this, the principle of the minimum of the potential energy is applicable, which can be directly derived from the principle of virtual work [11,21], and postulates that out of all possible deformations $\boldsymbol{x}_i$ the one that minimizes the energy functional $\Pi(\boldsymbol{x})$ defined as

$$\Pi(\boldsymbol{x}) = W^{int} - W^{ext} \tag{2.27}$$

with the internal work W^{int} and the external work W^{ext} fulfills the equilibrium conditions. In case of a contact problem, also the contact energy δW^{C} has to be taken into account in Equation (2.27), similar as in the principle of virtual work, as soon as the normal distance g_N of both contact surfaces becomes smaller or equal to zero ($g_N \leq 0$). The methods mentioned above differ especially with regard to this particular contact energy term. All methods need as an input variable the normal distance g_N between two arbitrary points $\boldsymbol{d}_A$ and $\boldsymbol{d}_B$ of the surfaces of the two bodies A and B in contact. This distance is defined according to Fig. 2.1 as

$$g_N = (\boldsymbol{d}_B - \boldsymbol{d}_A)^{T}\boldsymbol{n}_A, \tag{2.28}$$

which is positive or equal to zero in case of no penetration with $\boldsymbol{n}_A$ being the normal unit vector on the surface of body A. In a FEM computation monitoring a single pair of contact points only is not sufficient, but a whole set of FEM nodes defining the contact surfaces need to be considered. Through this, there is a vector $\boldsymbol{g}_N$ containing all necessary normal distances of the contact surfaces.

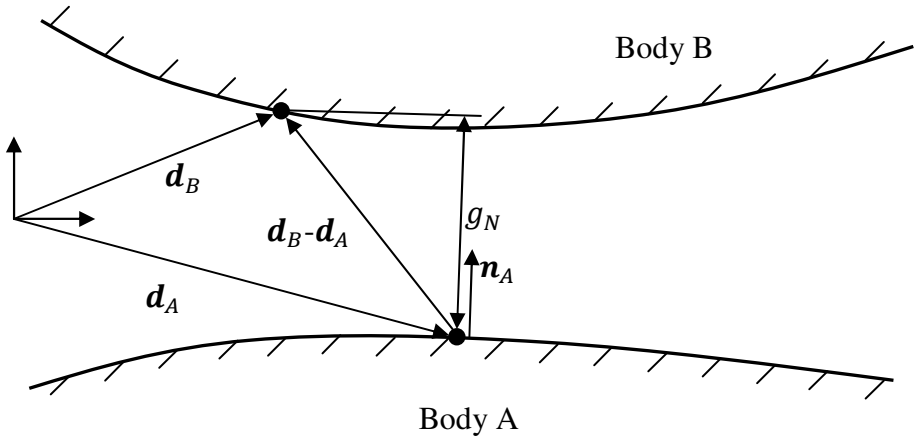


Fig. 2.1 Definition of gap size g_N between two contact surfaces

2.2.1 Lagrange Multiplier Method

Avoiding a penetration of the surfaces of two bodies in a FEM problem requires

$$\boldsymbol{g}_N \geq \boldsymbol{0}, \tag{2.29}$$

which is a constraint for the minimization problem (2.27). In this sense, condition (2.29) is introduced as an inequality condition into the minimization problem

$$\min_{\boldsymbol{x}} \Pi(\boldsymbol{x}) \quad \text{s.t. } \boldsymbol{g}_N \geq \boldsymbol{0}. \tag{2.30}$$

The Lagrange function for such a constrained optimization problem reads as

$$L(\boldsymbol{x}, \boldsymbol{\lambda}) = \Pi(\boldsymbol{x}) - \boldsymbol{\lambda}^T \boldsymbol{g}_N \tag{2.31}$$

and its solution must fulfill the Karush-Kuhn-Tucker conditions [73,88,51] as necessary conditions of first order:

$$\nabla_x L(\boldsymbol{x}, \boldsymbol{\lambda}) = \frac{\partial \Pi(\boldsymbol{x})}{\partial \boldsymbol{x}} - \boldsymbol{\lambda}^T \frac{\partial \boldsymbol{g}_N}{\partial \boldsymbol{x}} = \boldsymbol{0}, \tag{2.32a}$$

$$\nabla_\lambda L(\boldsymbol{x}, \boldsymbol{\lambda}) = -\boldsymbol{g}_N \leq \boldsymbol{0}, \tag{2.32b}$$

$$\lambda_i g_{i,N} = 0 \quad \forall\, i, \tag{2.32c}$$

$$\boldsymbol{\lambda} \geq \boldsymbol{0}. \tag{2.32d}$$

By this, the term $\boldsymbol{\lambda}^T \boldsymbol{g}_N$ in (2.31) can be interpreted as the contact energy

$$W_L^C = -\boldsymbol{\lambda}^T \boldsymbol{g}_N \tag{2.33}$$

where the vector of the Lagrange multipliers $\boldsymbol{\lambda}$ represents contact forces resulting in the total potential energy functional

$$\Pi_{\mathrm{L}}(\boldsymbol{x}, \boldsymbol{\lambda}) = \Pi(\boldsymbol{x}) + W_L^C = \Pi(\boldsymbol{x}) - \boldsymbol{\lambda}^T \boldsymbol{g}_N. \tag{2.34}$$

For a better understanding, the Lagrange multiplier method is demonstrated for a very simple example. Fig. 2.2 shows a linear spring with stiffness k elongated by a force F. On the right-hand side of the mechanical system a rigid wall limits the displacement x of the spring by the distance a between the rigid wall and the non-elongated position of the spring.

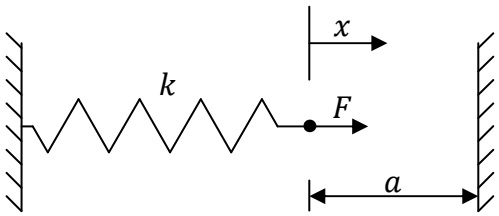


Fig. 2.2 Linear spring elongated by force F

Application of the energy functional (2.27) yields

$$\Pi(x) = \frac{1}{2} k x^2 - F x. \tag{2.35}$$

The contact condition (2.29) avoids penetration of the spring with the rigid wall

$$g_N = a - x \geq 0. \tag{2.36}$$

Finally, we get for the Lagrange function (2.31)

$$L(x, \lambda) = \frac{1}{2} k x^2 - F x - \lambda(a - x). \tag{2.37}$$

The corresponding Karush-Kuhn-Tucker (KKT) condition (2.32a) yields

$$k x - F + \lambda = 0 \tag{2.38}$$

and therefore

$$x = \frac{F - \lambda}{k}. \tag{2.39}$$

For the determination of the unknown Lagrange multiplier λ, KKT condition (2.32c) is used. Together with (2.39) we get

$$\lambda \left(a - \frac{F - \lambda}{k} \right) = 0 \tag{2.40}$$

with the two solutions

$$\lambda_1 = 0, \tag{2.41}$$

$$\lambda_2 = F - ak. \tag{2.42}$$

According to (2.32c), solution (2.41) is valid in case of no contact because of $g_N > 0$, whereas solution (2.42) is correct for an active contact $g_N = 0$. The corresponding displacements (2.39) are

$$x = \begin{cases} \dfrac{F - \lambda_1}{k} = \dfrac{F}{k} & \text{for } g_N > 0, \\ \dfrac{F - \lambda_2}{k} = a & \text{for } g_N \leq 0. \end{cases} \tag{2.43}$$

For the necessary determination of the gap size g_N, there exist different procedures which are explained in Section 2.3.

A simple FEM example, computed with the Lagrange multiplier method, is shown in Fig. 2.3. Body B (with size $1mm \times 1mm$) is pressed against body A (same size) with $1\,N/mm$, both meshed with just one solid element made of steel ($E = 210000\,N/mm^2$, $\nu = 0.3$). This leads to the displacement distribution plotted in Fig. 2.3b. The contact penetration of this nonlinear static computation is in the order of $10^{-19}mm$ which is zero with respect to numerical precision, and thus shows the exact fulfillment of the contact constraint.

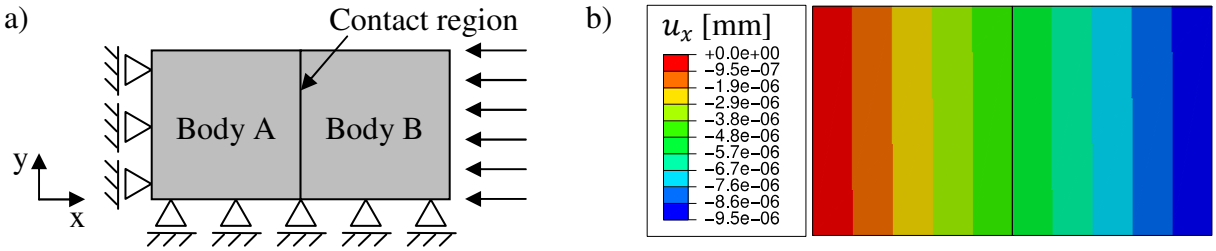


Fig. 2.3 Simple FEM contact problem (a) and resulting displacements (b) computed with Lagrange multiplier method

2.2.2 Penalty Method

By introducing a penalty term, a constraint optimization problem like (2.30) can be transformed into an unconstrained optimization problem [51]. In the penalty method, this term is defined as

$$W_P^C = \alpha \sum_i \beta_{i,N} \left| g_{i,N} \right|^{\varepsilon} \tag{2.44}$$

and added to the energy functional (2.27) in case of penetration, which gives

$$\Pi_P(\boldsymbol{x}) = \Pi(\boldsymbol{x}) + W_P^C = \Pi(\boldsymbol{x}) + \alpha \sum_i \beta_{i,N} \left| g_{i,N} \right|^{\varepsilon}. \tag{2.45}$$

The parameters $\beta_{i,N}$ are the penalty parameters penalizing any penetration $g_{i,N} < 0$ of contacting surfaces. Theoretically, for every i-th contact point there may be a different penalty parameter $\beta_{i,N}$, but in practical applications mostly the same parameter, depending on the material stiffness, is used for all contact pairs. The other two parameters α and ε are often chosen as $\alpha = 1/2$ and $\varepsilon = 2$. By this, the contact energy term becomes

$$W_P^C = \sum_i \frac{1}{2} \beta_{i,N} g_{i,N}^2 \tag{2.46}$$

and can be interpreted as the potential energy stored in a linear spring, where $\beta_{i,N}$ definines the stiffness of the i-th contact point. Fig. 2.4 shows the interpretation of the penalty contact formulation (2.46) for two meshed bodies in contact. The springs, which are introduced between the contact surfaces determine the penetration of the two bodies. The stiffness values of these springs can be adjusted to the gap size. The more penetration, the higher is the spring

stiffness. As a result the penetration should be reduced in the next time-step.

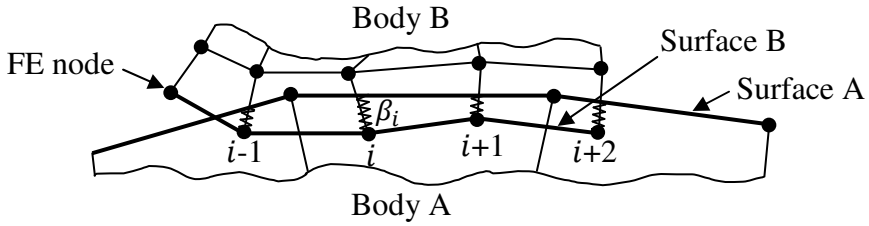


Fig. 2.4 Interpretation of penalty method as introduction of springs between the contact surfaces of body A and body B

Returning to the simple contact spring example of Fig. 2.2 and assuming $\alpha = 1/2$ and $\varepsilon = 2$, we get for the energy functional (2.45) for the penetration case $x > a$

$$\Pi_{\mathrm{P}}(x) = \frac{1}{2}kx^2 - Fx + \frac{1}{2}\beta_N g_N^2 \qquad \text{where} \qquad g_N = a - x. \tag{2.47}$$

Since we do not introduce constraints as in the Lagrange multiplier method, it is not necessary to apply KKT conditions but it is sufficient to compute the first derivative of (2.47) with respect to x and set the result to zero:

$$\frac{\partial \Pi_{\mathrm{P}}(x)}{\partial x} = kx - F + \beta_N(x - a) = 0. \tag{2.48}$$

Solving (2.48) for the unknown displacement x yields

$$x = \frac{F + a\beta_N}{k + \beta_N}. \tag{2.49}$$

It is obvious that the solution becomes $x = F/k$ for $\beta_N = 0$ and goes to $x = a$ for $\beta_N \to \infty$. In the former case the contact is just ignored, the latter is the exact solution.

The diagram of Fig. 2.5 shows the energy term (2.47) for different values of the contact stiffness β_N. In this example, $F = 2N$, $a = 0.1m$ and $k = 10\,N/m$ are chosen. Since $x = F\,/\,k = 2N/(10\ N/m) = 0.2m$, the correct solution is $x = a = 0.1m$. As one can observe, the penalty stiffness always allows a certain penetration, and therefore a violation of the contact constraint. Only for $\beta_N \to \infty$ the correct solution is obtained. This may lead to the conclusion that better solutions are achieved by higher contact stiffness values, which is of course correct at a first glance. But in computational mechanics, the contact stiffness shouldn't be chosen too high, otherwise this may lead to numerical problems due too ill-conditioned stiffness matrices. Choosing the right contact stiffness is an art for the penalty method. For practical applications, a value in the order of the representative underlying element stiffness of the contact surfaces is often a good choice.

Also for the FEM example in Fig. 2.3, the difference between the penalty method and the Lagrange multiplier method is optically visible in terms of a small penetration for the penalty method, which is in the order of $10^{-6}mm$ in this example for a typical contact stiffness value of $\beta_N = 1.1 \cdot 10^6\ N/mm$ (Fig. 2.6).

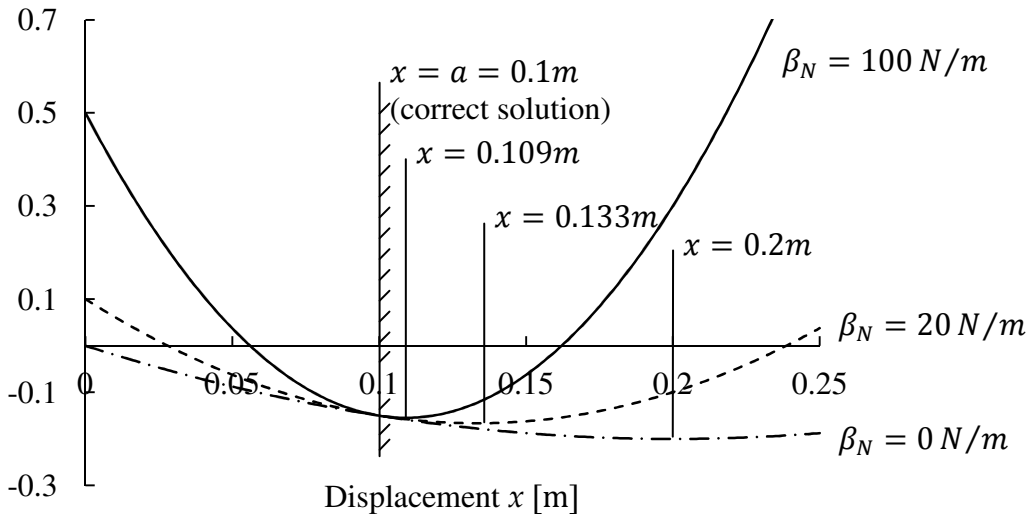


Fig. 2.5 Correct solution for the x displacement and solutions of the penalty method for different values of the contact stiffness parameter β_N

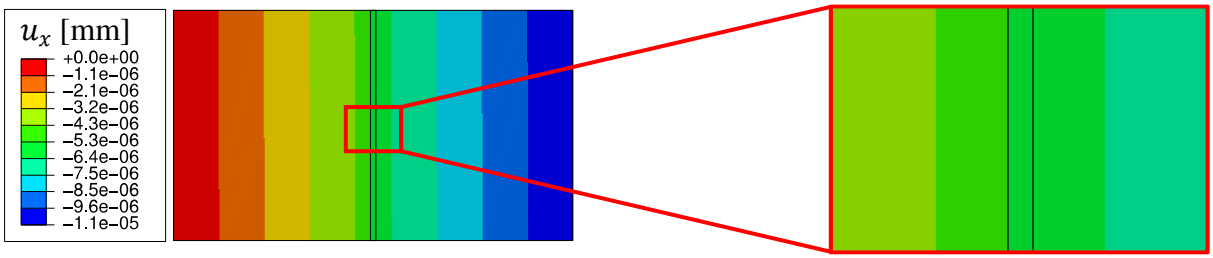


Fig. 2.6 Resulting displacement of simple contact problem computed with penalty method

Besides the disadvantage of the penalty method that the contact constraint is only fulfilled approximately, there is also an advantage. In comparison to the Lagrange multiplier method, no additional equations for the contact constraints are necessary which means that the computational effort is not increased and the system of equations does not grow.

2.2.3 Augmented Lagrange Method

The augmented Lagrange method combines Lagrange multiplier and penalty method. The contact energy is defined as

$$W_{AL}^{C} = \sum_i \frac{1}{2}\beta_{i,N} g_{i,N}^2 - \boldsymbol{\lambda}^T \boldsymbol{g}_N, \tag{2.50}$$

which leads in case of penetration to the energy functional

$$\Pi_{\mathrm{AL}}(\boldsymbol{x}, \boldsymbol{\lambda}) = \Pi(\boldsymbol{x}) + W_{AL}^{C} = \Pi(\boldsymbol{x}) + \sum_i \frac{1}{2}\beta_{i,N} g_{i,N}^2 - \boldsymbol{\lambda}^T \boldsymbol{g}_N. \tag{2.51}$$

For the augmented Lagrange method two different interpretations exist. The first one applies the KKT conditions to Equation (2.51). This procedure is described for example in [21] or [38] and leads again to additional equations similar as for the Lagrange multiplier method in

Section 2.2.1, which increases the computational effort. The second interpretation is of more practical relevance and is also known as method of multipliers, which in the field of optimization was firstly discussed by Hestenes [61] and Powell [119]. In this sense, the minimization of (2.51) is treated as an unconstrained optimization problem. As such, the iterative optimization process starts with a sensible guess for the unknown $\boldsymbol{\lambda}$ (even $\boldsymbol{\lambda} = \mathbf{0}$ is possible). The contact stiffnesses $\beta_{i,N}$ have to be chosen too, but their values may be smaller than within the pure penalty method which therefore leads to a better numerical conditioning of the problem. After the first iteration, the contact stiffnesses $\beta_{i,N}$ may be updated, but this is not necessary. More important is the update of $\boldsymbol{\lambda}$ following the rule

$$^{k+1}\lambda_i = {}^{k}\lambda_i - \beta_{i,N}\,{}^{k}g_{i,N}. \tag{2.52}$$

By this, the contact pressure is increased in every iteration, which leads to a decrease of the penetration. The iterative process may be stopped if a certain penetration tolerance ε (e.g. 0.1% of the characteristic interface length) is reached.

For the spring problem in Fig. 2.2, the augmented Lagrange method yields for $x > a$ the energy functional

$$\Pi_{\mathrm{AL}}(x,\lambda) = \frac{1}{2}kx^2 - Fx + \frac{1}{2}\beta_N g_N^2 - \lambda g_N. \tag{2.53}$$

Consequently, its derivative with respect to x yields with the substitution $g_N = a - x$

$$\frac{\partial \Pi_{\mathrm{AL}}(x,\lambda)}{\partial \mathrm{x}} = kx - F + \beta_N(x-a) + \lambda. \tag{2.54}$$

Setting this derivative to zero and solving for the unknown displacement x leads to

$$x = \frac{F - \lambda + \beta_N a}{k + \beta_N}. \tag{2.55}$$

In contrast to the Lagrange multiplier and penalty method now some contact iterations have to be performed. Table 2.1 shows the results of these iterations for $\beta_N = 100\,N/m = const.$ All other values are chosen as in the section before for this problem. Obviously convergence to the correct solution $x = 0.1m$ is reached rather fast.

For the FEM example in Fig. 2.3 a contact penetration in the order of $10^{-6}mm$ is computed

Table 2.1 Displacements of simple spring contact problem computed with augmented Lagrange method

Iteration	$\lambda\ [N]$	Displacement $x\ [m]$
0	0	0.10909091
1	0.90909091	0.10082645
2	0.99173554	0.10007513
3	0.99924869	0.10000683
4	0.9999317	0.10000062
5	0.99999379	0.10000006
6	0.99999944	0.10000001

with a commercial FE tool using a penetration tolerance of 0.1% of the characteristic element length of 1mm, which gives $\varepsilon = 10^{-3}mm$ (0.1% of 1mm). This result was achieved in just one iteration because the penalty stiffness β_N was chosen identically as for the penalty method, which finally leads to the same result as for the penalty method.

2.2.4 Perturbed Lagrange Method

The perturbed Lagrange formulation [131] is a regularization of the Lagrange multiplier method, which remedies some unfavorable numerical properties of the Lagrange multiplier method (e.g. discontinuity of displacements at time of impact). The contact energy is defined as

$$W_{PL}^{C} = -\frac{1}{2\beta_N}\boldsymbol{\lambda}^T\boldsymbol{\lambda} - \boldsymbol{\lambda}^T\boldsymbol{g}_N, \tag{2.56}$$

where the first term is the regularization term containing a constant β_N with typically very high values similar as in the penalty method. The energy functional of the principle of the minimum of the potential energy for a mechanical system with contact conditions then becomes

$$\Pi_{\mathrm{PL}}(\boldsymbol{x},\boldsymbol{\lambda}) = \Pi(\boldsymbol{x}) + W_{PL}^{C} = \Pi(\boldsymbol{x}) - \frac{1}{2\beta_N}\boldsymbol{\lambda}^T\boldsymbol{\lambda} - \boldsymbol{\lambda}^T\boldsymbol{g}_N. \tag{2.57}$$

If the KKT conditions

$$\nabla_x L(\boldsymbol{x},\boldsymbol{\lambda}) = \frac{\partial\Pi(\boldsymbol{x})}{\partial\boldsymbol{x}} - \frac{\partial\boldsymbol{g}_N{}^T}{\partial\boldsymbol{x}}\boldsymbol{\lambda} = \mathbf{0}, \tag{2.58a}$$

and

$$\nabla_\lambda L(\boldsymbol{x},\boldsymbol{\lambda}) = -\frac{1}{\beta_N}\boldsymbol{\lambda} - \boldsymbol{g}_N \leq \mathbf{0} \tag{2.58b}$$

are applied, we get equations for the determination of the unknowns $\boldsymbol{x}$ and $\boldsymbol{\lambda}$.

To illustrate the perturbed Lagrange method, the simple spring example (2.35), (2.36) is used again. The Lagrange function reads as

$$L(x,\lambda) = \frac{1}{2}kx^2 - Fx - \frac{1}{2\beta_N}\lambda^2 - \lambda(a-x). \tag{2.59}$$

Conditions (2.58a) and (2.58b) give

$$\frac{\partial L(x,\lambda)}{\partial x} = kx - F + \lambda = 0, \tag{2.60}$$

$$\frac{\partial L(x,\lambda)}{\partial \lambda} = -\frac{\lambda}{\beta_N} + x - a \leq 0. \tag{2.61}$$

Supposing equality in the last equation, λ can be determined as

$$\lambda = \beta_N(x-a). \tag{2.62}$$

Substituting this in (2.60) results in

$$x = \frac{F + \beta_N a}{k + \beta_N}. \tag{2.63}$$

For the values used before ($F = 2N$, $a = 0.1m$ and $k = 10\,N/m$) and a penalty parameter of $\beta_N = 100\,N/m$ a displacement of $x \approx 0.109$m is calculated. For $\beta_N \to \infty$ the exact solution $x = a = 0.1$m is obtained. Obviously (2.63) is identical to the result of the penalty method (2.49). Therefore the perturbed Lagrange method is more of theoretical interest and not implemented in widely used commercial FE codes.

2.3 Determination of Gap Size

In all the strategies and algorithms discussed above the gap size g_N is an important quantity. In a FEM computation, the determination of this parameter is not as trivial as it seems to be at a first glance. Therefore, the most important methods for the calculation of the gap size between two bodies or surfaces in contact are presented in the following. For reasons of simplicity all considerations are restricted to the two-dimensional case.

2.3.1 Node-to-Node Approach

The node-to-node contact approach is the simplest possibility for determining the gap size. However, it is only applicable if identical meshes with coincident nodes are used at both contact pairs (see Fig. 2.7) and, therefore, has limited application. Since a large tangential movement is not allowed in this formulation, it can only be applied to geometrically linear problems. According to Fig. 2.7 the gap size between two nodes is computed as

$$g_{i,N} = \left(\boldsymbol{x}_i^B - \boldsymbol{x}_i^A\right)^T \boldsymbol{n}_i^A + g_{i0} \tag{2.64}$$

with g_{i0} being the initial normal distance of the *i*-th node pair, displacement vectors $\boldsymbol{x}_i^A$ and $\boldsymbol{x}_i^B$ of the associated nodes and unit normal vector $\boldsymbol{n}_i^A$ on body A at node i. Due to the mentioned limitations of the node-to-node approach, it is not used anymore in commercial tools nowadays.

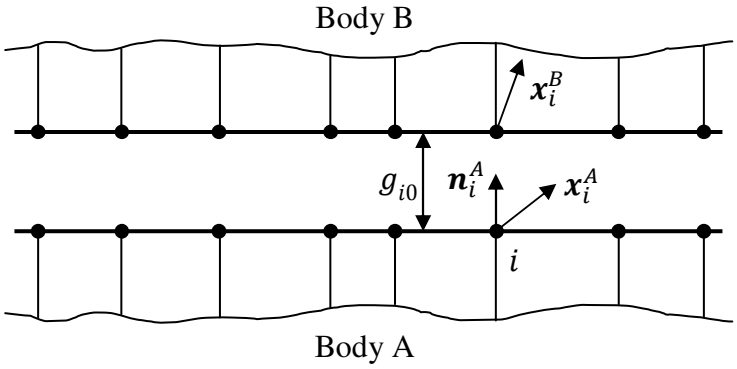


Fig. 2.7 Computation of gap-size g_N with node-to-node approach

2.3.2 Node-to-Surface Approach

In the node-to-surface approach it is necessary to classify the two surfaces of the bodies getting into contact as master and slave surfaces. The distance g_n between the two surfaces is then computed as the normal distance of a node of the slave surface to a segment of the master surface (Fig. 2.8). The master segment is defined by two master nodes (in the two-dimensional case). By this procedure, only penetrations of the slave surface into the master surface are checked. It is not noticed by the algorithm if a master node penetrates the slave surface. For this reason, better results are typically obtained by choosing the surface with the finer mesh to be the slave surface.

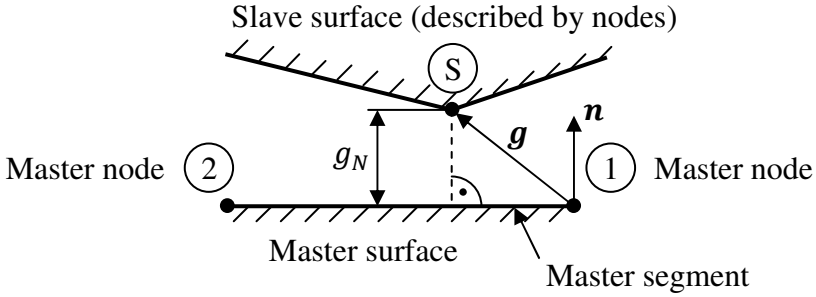


Fig. 2.8 Computation of gap-size g_N with node-to-surface approach

If the vector $\boldsymbol{g}$ is defined as the vector pointing from master node 1 to the slave node S, the gap size g_N can be calculated with the help of the unit normal vector $\boldsymbol{n}$ on the master surface as

$$g_N = \boldsymbol{g}^T \boldsymbol{n}. \tag{2.65}$$

For some configurations this algorithm may fail or cause problems [151] as shown in Fig. 2.9. In Fig. 2.9a no projection of the slave node onto a specific master segment is possible, which means that the slave node is a kind of blind spot. If the master surface moves towards the slave surface within the dashed cone there will be a penetration without detection. Another difficult situation is shown in Fig. 2.9b where the gap size is not unique and can oscillate, which means an oscillation of the contact stiffness. By this, convergence problems may be caused. In case of Fig. 2.9c the gap size cannot be determined because the normal from the slave node to the master surface lies outside the master segment.

The last effect can be demonstrated for a very simple FE example where the Lagrange multiplier method is used as contact algorithm. The upper two elements in Fig. 2.10a (body

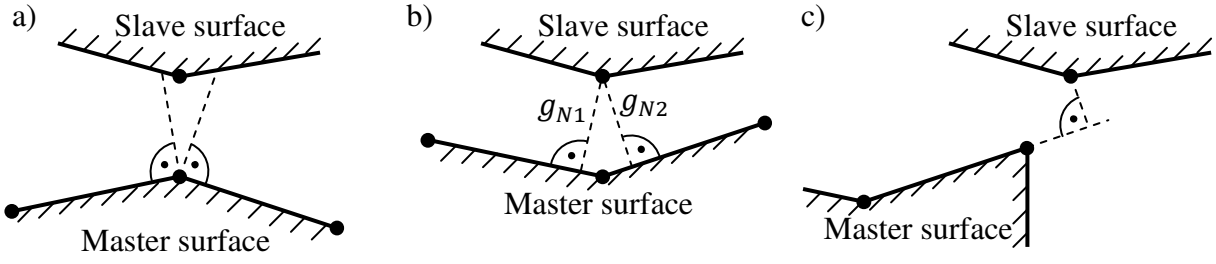


Fig. 2.9 Difficult situations for node-to-surface approach [151]

B) are pressed against the lower element (body A) by load p, which is fixed at the ground. At first the slave surface is the contact surface of the upper two elements. As it can be seen in Fig. 2.10b, this leads to a penetration of body B into body A due to the fact that no gap size can be determined for the two outer slave nodes. Also a change of master and slave surface (Fig. 2.10c) does not completely resolve the problem since the master surface has now a finer mesh and may penetrate the slave surface. Only a finer mesh of body B leads to a qualitatively better solution (Fig. 2.10d). The deformation scale factor, which is a constant multiplied to all nodal displacement values, is the same for all figures.

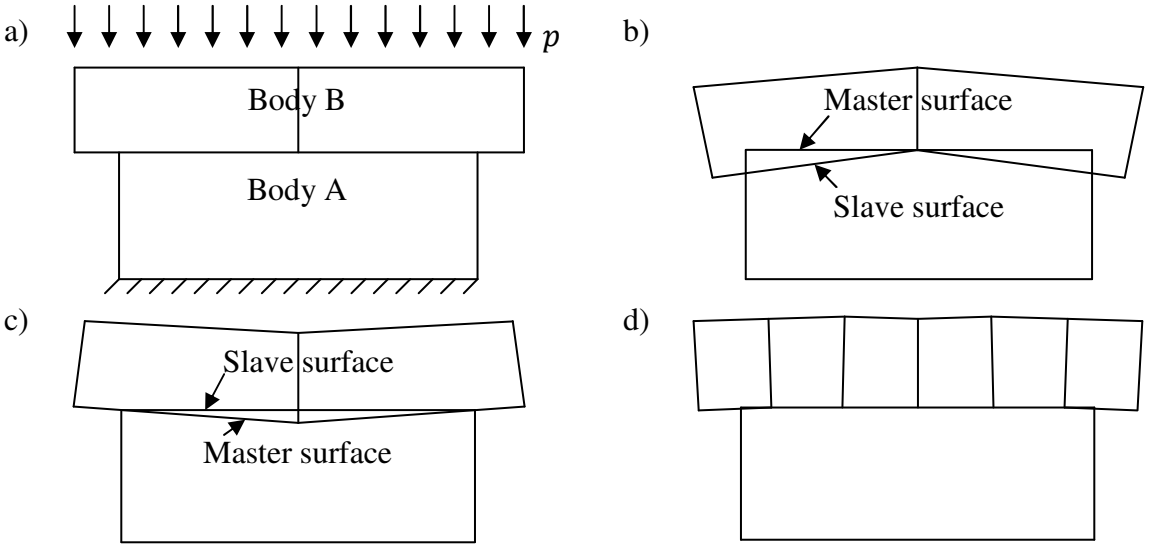


Fig. 2.10 Simple contact problem of two bodies pressed against each other (a) with solutions of node-to-surface approach with b) body A as master and body B as slave, c) body B as master and body A as slave, and d) with finer mesh (d)

Instead of using a finer mesh for the contact surfaces, also the application of a so-called two-pass-contact strategy can improve the result [131,136]. In such a formulation master and slave surfaces are exchanged and the contact situation is checked again. Of course, this has the disadvantage of an increased computational effort.

Another problem of the node-to-surface approach is related to the correct load distribution in the contact interface which appears in combination with the penalty algorithm. Let us consider the simple contact problem of Fig. 2.11a consisting of three finite elements. The two upper elements are pressed against the lower element with an equally distributed pressure $p = const$. The correct load distribution for the contact forces, which would lead to the expected deformation and penetration, is shown in Fig. 2.11b. However, by applying the penalty method we obtain the uniform load distribution of Fig. 2.11c since the contact stiffness of the penalty method is distributed equally to all slave nodes. Such a distribution of the contact forces results in the deformation of Fig. 2.11d which is physically not correct because the middle node penetrates the master surface deeper than the outer nodes.

Of course, the described effects become much smaller for finer meshes. Nevertheless, these disadvantages of the node-to-surface approach led to the development of the surface-to-

surface approach (sometimes also called segment-to-segment approach). On the other hand, the simplicity and computational effectiveness of the node-to-surface algorithm gave rise to the development of remedies for the described difficulties of the node-to-surface approach. Some of these are presented for example in [151] and [152].

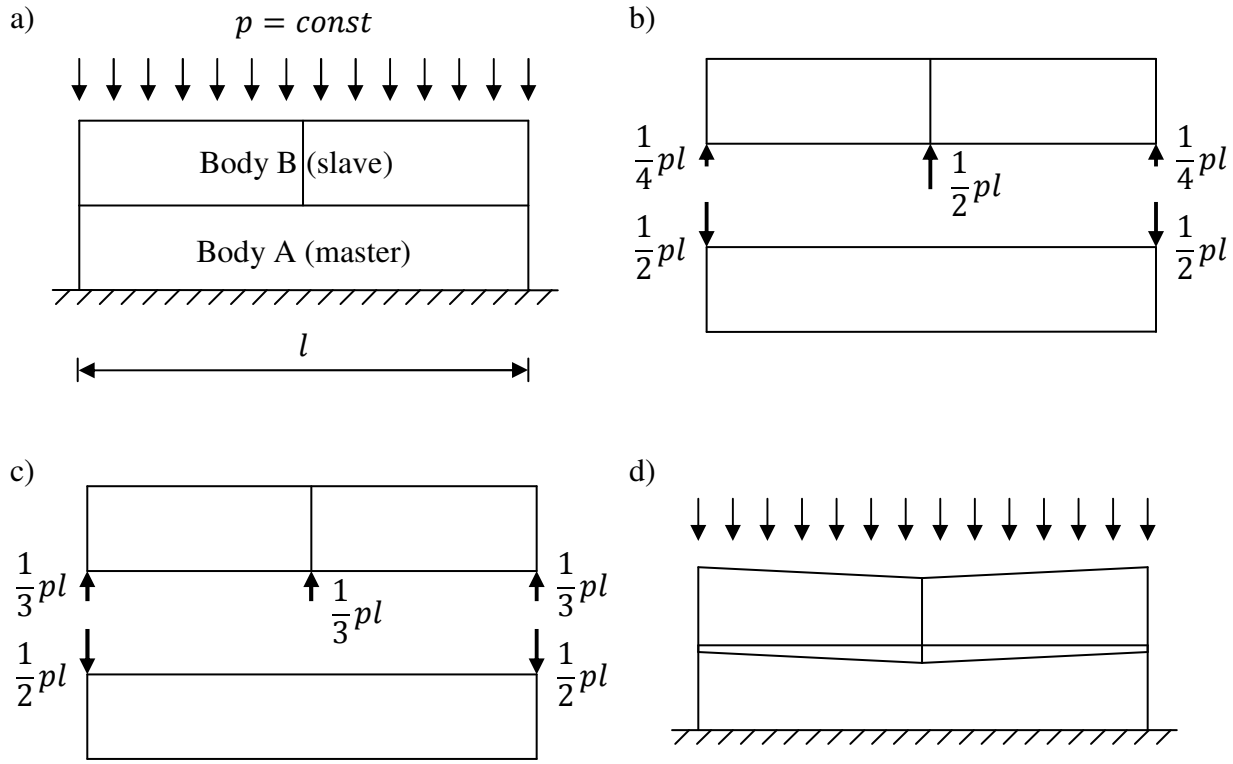


Fig. 2.11 Simple contact problem (a) with correct distribution of contact forces (b) and contact force distribution of classical node-to-surface approach in combination with penalty algorithm (c) leading to a deformed configuration (d)

2.3.3 Surface-to-Surface Approach

This approach was developed due to the disadvantages and inaccuracies of the simpler node-to-surface approach. First ideas about the surface-to-surface strategy, which is sometimes also called segment-to-segment strategy, can be found in [131,117,116] with further developments in [153] and [48]. The basic idea is to use a piecewise constant approximation of the contact pressure which is discontinuous at the borders between so-called contact segments. By this the contact constraints are enforced over a contact segment in an average sense. A contact segment is defined by the normal projections of two nodes onto the surface of the other contact partner (see Fig. 2.12).

For the definition of a contact segment it is important to determine the projections of the nodes of each surface onto the opposite surface, which can be done according to Fig. 2.12a and Fig. 2.12b by

$$ {}^{A}\alpha \cdot |{}^{A}\boldsymbol{d}_2 - {}^{A}\boldsymbol{d}_1| = ({}^{B}\boldsymbol{d}_2 - {}^{A}\boldsymbol{d}_1)^{T\,A}\boldsymbol{t} \tag{2.66} $$

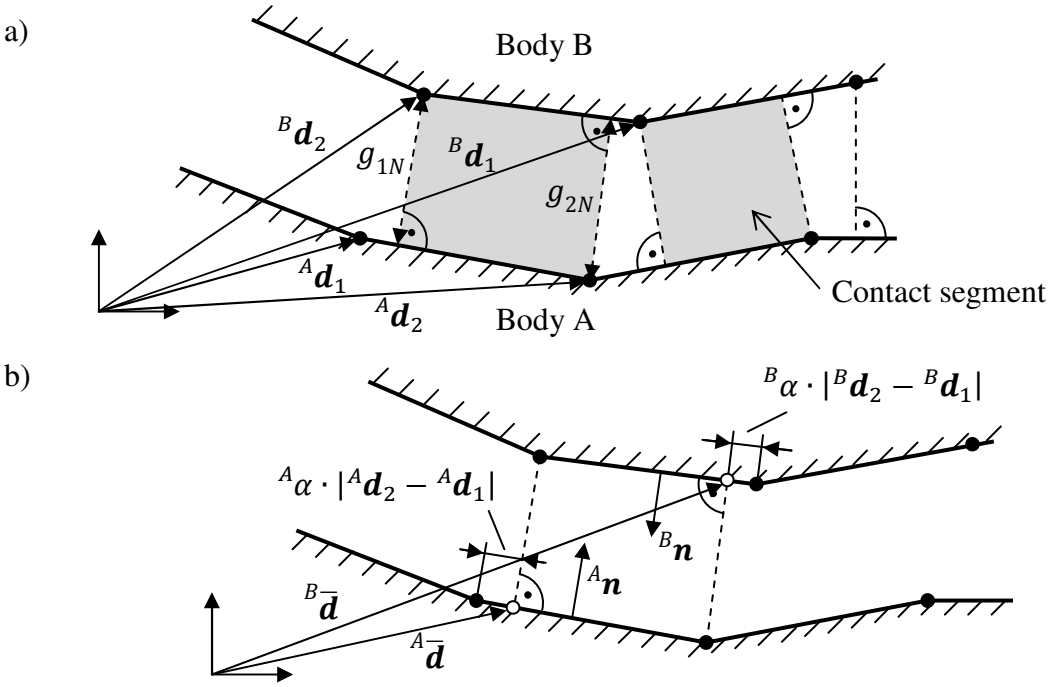


Fig. 2.12 Construction of contact segments (a) and geometry of contact segment (b)

with the tangential unit vector ${}^A\boldsymbol{t} = ({}^A\boldsymbol{d}_2 - {}^A\boldsymbol{d}_1)/|{}^A\boldsymbol{d}_2 - {}^A\boldsymbol{d}_1|$ and

$${}^B\alpha \cdot |{}^B\boldsymbol{d}_2 - {}^B\boldsymbol{d}_1| = ({}^A\boldsymbol{d}_2 - {}^B\boldsymbol{d}_1)^T {}^B\boldsymbol{t} \tag{2.67}$$

with the tangential unit vector ${}^B\boldsymbol{t} = ({}^B\boldsymbol{d}_2 - {}^B\boldsymbol{d}_1)/|{}^B\boldsymbol{d}_2 - {}^B\boldsymbol{d}_1|$. From this, the coordinates of the artificial or virtual nodes, resulting from the projection, become

$${}^A\bar{\boldsymbol{d}} = {}^A\boldsymbol{d}_1 + {}^A\alpha \cdot |{}^A\boldsymbol{d}_2 - {}^A\boldsymbol{d}_1| {}^A\boldsymbol{t} = {}^A\alpha {}^A\boldsymbol{d}_2 + (1 - {}^A\alpha) {}^A\boldsymbol{d}_1, \tag{2.68}$$

$${}^B\bar{\boldsymbol{d}} = {}^B\boldsymbol{d}_1 + {}^B\alpha \cdot |{}^B\boldsymbol{d}_2 - {}^B\boldsymbol{d}_1| {}^B\boldsymbol{t} = {}^B\alpha {}^B\boldsymbol{d}_2 + (1 - {}^B\alpha) {}^B\boldsymbol{d}_1 \tag{2.69}$$

with parameters ${}^A\alpha$ and ${}^B\alpha$ computed from (2.66) and (2.67), respectively. Similarly, the displacement vectors of the artificial nodes are computed as

$${}^A\bar{\boldsymbol{x}} = {}^A\alpha {}^A\boldsymbol{x}_2 + (1 - {}^A\alpha) {}^A\boldsymbol{x}_1, \tag{2.70}$$

$${}^B\bar{\boldsymbol{x}} = {}^B\alpha {}^B\boldsymbol{x}_2 + (1 - {}^B\alpha) {}^B\boldsymbol{x}_1 \tag{2.71}$$

from the displacements ${}^A\boldsymbol{x}_1$, ${}^A\boldsymbol{x}_2$, ${}^B\boldsymbol{x}_2$ and ${}^B\boldsymbol{x}_1$ of the adjacent FE nodes. Finally, it is possible to determine the gap sizes g_{1N} and g_{2N} as

$$g_{1N} = ({}^B\boldsymbol{x}_2 - {}^A\bar{\boldsymbol{x}})^T {}^A\boldsymbol{n} + g_{1N}^0, \tag{2.72}$$

$$g_{2N} = ({}^A\boldsymbol{x}_2 - {}^B\bar{\boldsymbol{x}})^T {}^B\boldsymbol{n} + g_{2N}^0 \tag{2.73}$$

with the initial gap sizes $g_{1N}^0 = \left|{}^A\bar{\boldsymbol{d}} - {}^B\boldsymbol{d}_2\right|$ and $g_{2N}^0 = \left|{}^A\boldsymbol{d}_2 - {}^B\bar{\boldsymbol{d}}\right|$ under the assumption that the gap is open initially and the initial gap size positive therefore. For the contact computation the average gap size

$$\bar{g}_N = \frac{1}{2}(g_{1N} + g_{2N}) \tag{2.74}$$

is used.

Applying the surface-to-surface approach for the example problem in Fig. 2.10a yields the result in Fig. 2.13a. Here, the lower surface is defined as master and the upper surface as slave surface (analogously to Fig. 2.10b) and the Lagrange multiplier algorithm is used. Obviously the result is very good (same deformation scale factor as in Fig. 2.10).

Also for the example problem in Fig. 2.11a the results are now correct in combination with the penalty algorithm, since the gap-size averaging of the surface-to-surface approach over the contact segments automatically leads to the correct load distribution in this case (Fig. 2.13b).

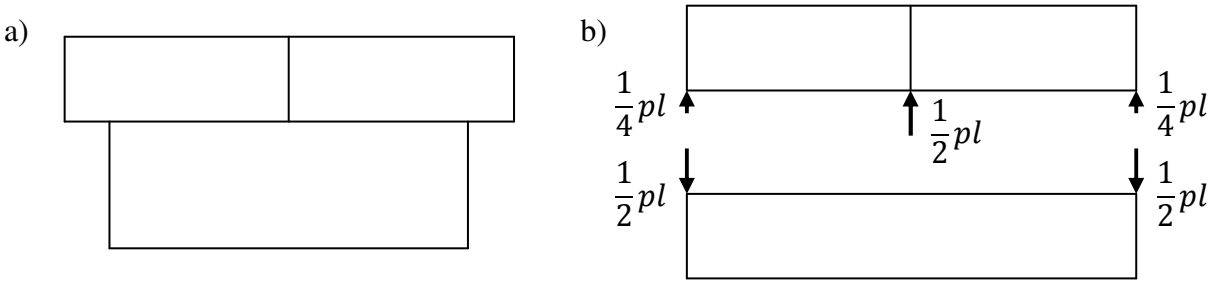


Fig. 2.13 Result of surface-to-surface approach in combination with Lagrange multiplier contact algorithm for example problem in Fig. 2.10a (a) and contact force distribution of surface-to-surface approach in combination with penalty algorithm for example problem in Fig. 2.11a (b)

Although the surface-to-surface approach has many advantages in comparison to the node-to-surface approach and generally leads to better results, there are also some cases where the surface-to-surface approach has difficulties and may fail without special extensions [148,150].

2.3.4 Mortar Approach

So-called mortar methods originate from domain decomposition problems. The contact constraints are enforced in a weak sense which shows some parallels to the surface-to-surface approach. Nevertheless, there is a strong mathematical background, which is derived for example in [18,64] and [103]. In the latter paper also a review about the development of mortar methods is given.

The basic idea of mortar methods is to use an intermediate contact interface which can be constructed from contact segments like the ones used in the surface-to-surface approach (Fig. 2.14). It can be also useful to use directly one of the contact surfaces as intermediate surface as shown in [147]. For the interpolation of the gap function $g_N(\zeta)$ along the intermediate surface, the interpolation functions of the contact surfaces have to be used, but for the interpolation of the Lagrange multipliers $\lambda(\zeta)$ the functions can be chosen freely. However, this does not mean that the mortar approach is restricted to the Lagrange multiplier method. The penalty algorithm can also be applied, as in [120] for example.

An advantage of the mortar approach is observable in the simulation of rotating contact surfaces (Fig. 2.15a). In this example a simple sliding contact bearing is modeled where the outer ring of the bearing is fixed at its outer nodes in all directions and the inner shaft is fixed

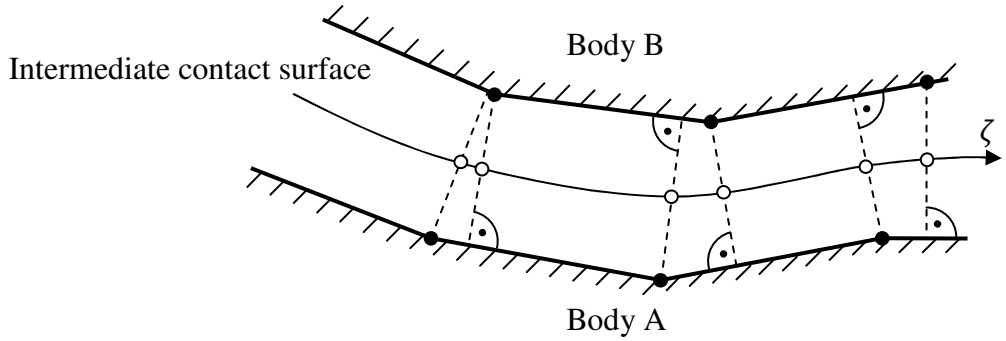


Fig. 2.14 Intermediate contact surface of mortar method

at its front in axial direction. Between the inner shaft and the outer ring a frictionless sliding contact is defined. By this, the shaft rotates due to a pressure load applied to the "blades". For the contact computation the penalty algorithm in combination with different methods for the gap determination is used. The diagram in Fig. 2.15b shows the time-step size for a simulation with an implicit time-integration scheme over simulation time. The plotted time-step size per time-step is adjusted for all computations by the same algorithm of a commercial FEM tool, which is similar to the one presented in Section 7.4. The biggest time steps, and therefore the lowest overall computational time is achieved by the simulation using the mortar approach.

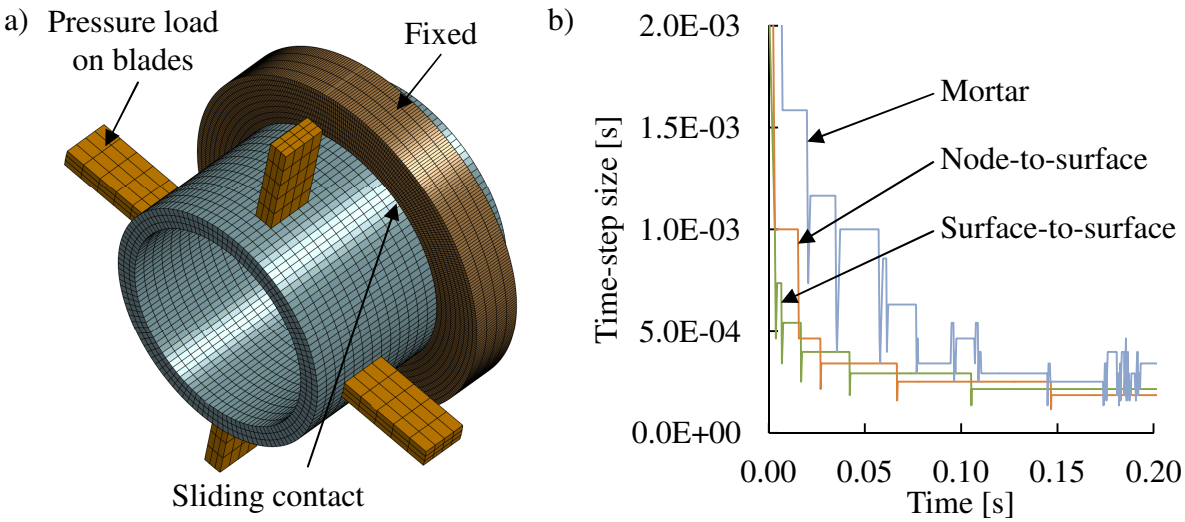


Fig. 2.15 Example problem with sliding contact (a) and comparison of time-step sizes for different contact approaches (b)

3 Choice of Proper Finite Element Formulations and Numerical Problems

At the beginning of the solution process of arbitrary boundary value problems by the finite element method the computational engineer has to choose a proper finite element formulation for the discretization of the structural domain. During more than 60 years of FEM history, many different finite element formulations have been developed. The first simple and straight forward approaches led to numerical problems like the locking phenomenon, which means that the element behaves far too stiff than actually intended. As a reaction, the concept of under-integrated elements has been developed, which leads to locking-free elements. On the other hand, these types of elements may behave too soft in certain situations and suffer from so called zero-energy modes, also called hourglass modes, since several elements in a mesh may form an hourglass-like shape in these modes.

The described problems can be demonstrated by the very simple example of a plane cantilever beam (Fig. 3.1). For small deformations and slender beams the linearized Euler-Bernoulli beam theory is applicable and the maximum beam deflection is computed as

$$^{B}x_{max} = \frac{F \cdot l^3}{3E \cdot I} \tag{3.1}$$

with E as Young's modulus of the used material and $I = bh^3/12$ as area moment of inertia for a rectangular cross section. By choosing $E = 210000\,N/mm^2$, $F = 100N$, $l = 100mm$, $b = 1mm$ and $h = 10mm$, a maximum bending displacement of $^{B}x_{max} \approx 1.905mm$ is calculated according to Equation (3.1). Since the shear force is not zero for this problem, we also can take into account the shear displacement to get a more accurate solution. According to Timoshenko's beam theory [137,85] the maximum shear displacement is given by

$$^{S}x_{max} = \frac{F \cdot l}{G \cdot A_S} \tag{3.2}$$

with shear modulus $G = E/[2(1+\nu)]$ and the shear area $A_S = A/k_z$. The shear correction parameter k_z depends on the cross section geometry and equals $6/5$ for a rectangular cross section. For a Poisson's ratio of $\nu = 0.3$ and $A = bh$ we obtain a shear displacement of $^{S}x_{max} \approx 0.015mm$. By using the principle of superposition, the overall deflection for the beam is $x_{max} = {}^{B}x_{max} + {}^{S}x_{max} \approx 1.92mm$.

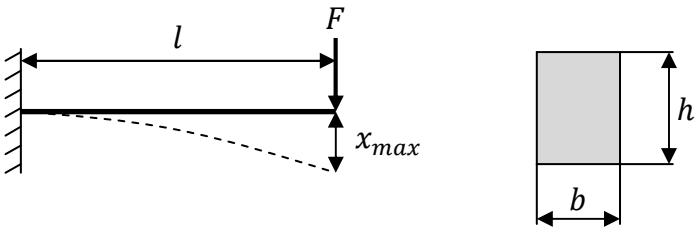


Fig. 3.1 Cantilever beam with rectangular cross section

For solving the same problem with the help of the finite element method, the beam geometry has to be meshed and a proper mesh density and finite element formulation have to be chosen. Let's assume that we use two-dimensional solid elements for the meshing of the structure. Although the boundary conditions are identical for all computations in Fig. 3.2, the resulting maximum displacements are very diverse for the different models and used standard finite element formulations. With linear fully-integrated elements (Fig. 3.2a-c) the displacement is underestimated, but with increasing mesh density a convergence towards the correct result is recognizable. The under-integrated elements in Fig. 3.2d-f overestimate the displacement without converging to the correct result for a finer mesh. The worst result is obtained by using triangular elements (Fig. 3.2g-i). The most accurate and fast converging result is produced by elements with quadratic interpolation functions (Fig. 3.2j-l). It is interesting to note that the number of degrees of freedom (DOF) and thus the computational time is about the same for every column of Fig. 3.2 except for the last row.

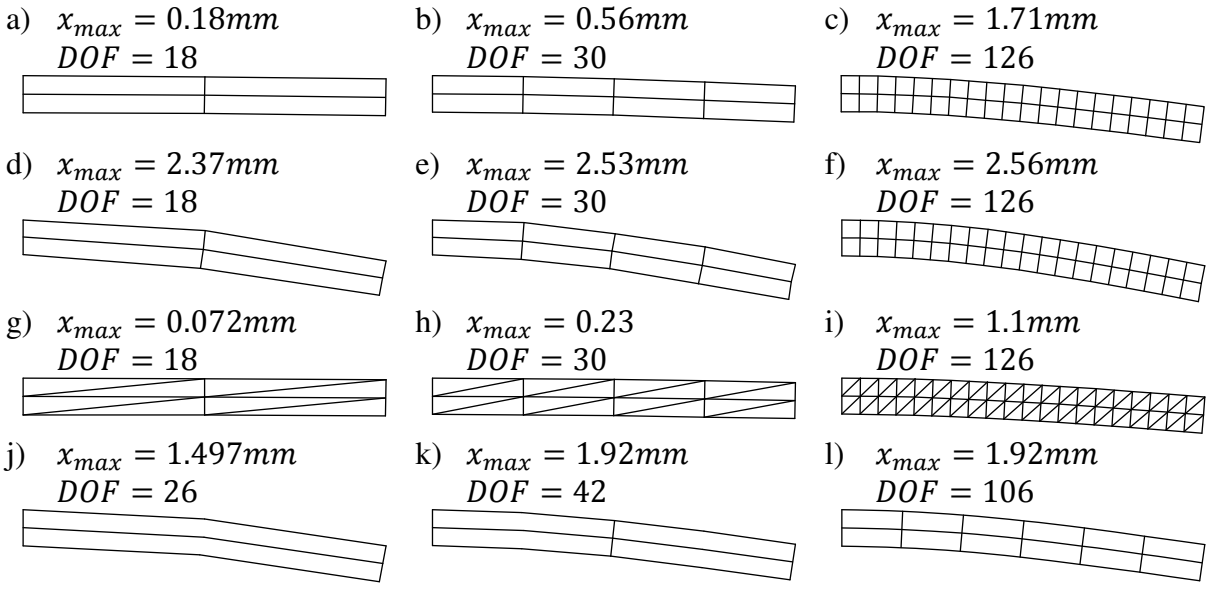


Fig. 3.2 Simulation results with different mesh densities and finite element formulations: linear fully-integrated elements (a-c), linear underintegrated elements (d-f), linear triangular elements (g-i) and quadratic elements (j-l)

In the following the reasons for the mentioned phenomena locking and hourglassing as well as for the differences in the computations of the introductory beam example are explained and successful strategies for avoiding these unwanted numerical problems are described. By this, the reader should gain a deeper understanding of possible numerical problems and the solution concepts, which is very important for the assessment of the computational results and their quality. Finally, three-point bending tests with different element types and anti-hourglassing formulations are presented to give some examples for the mentioned phenomena and possible solutions.

3.1 Locking of Continuum Elements

Continuum elements are the simplest formulations for finite elements because they do not contain any assumptions or simplifications about deformations or stress/strain behavior as it is done for shell elements or plane strain elements, for example. Typical continuum elements are three- or two-dimensional solid elements, which have only displacement degrees of freedom at their nodes. Such standard element formulations are the basis of every finite element code and consequently widely used. The locking phenomena of such elements can be separated into three groups and are explained in the following.

3.1.1 Shear Locking

Shear locking may appear for two- and three dimensional continuum elements as well as in the membrane part of shell-like elements. The effect can be illustrated for the example of two-dimensional pure bending under the assumption of plane stress. Let us first derive the analytical solution for such a problem. We consider a rectangular plate with size $b \times h$ and thickness t which is subjected to a constant bending moment M (Fig. 3.3a). To derive an analytical solution with regard to stresses and displacements, the classical theory of elasticity and an Airy stress function (named after George Biddell Airy (1801-1892) [1]) of the form

$$F = \frac{2M}{th^3} y^3 \tag{3.3}$$

can be used [57]. The stresses are then calculated as

$$\sigma_x = \frac{\partial^2 F}{\partial y^2} = \frac{12M}{th^3} y, \tag{3.4a}$$

$$\sigma_y = \frac{\partial^2 F}{\partial x^2} = 0, \tag{3.4b}$$

$$\tau_{xy} = -\frac{\partial^2 F}{\partial x \partial y} = 0 \tag{3.4c}$$

(see also Fig. 3.3b). Following from this, the strains

$$\varepsilon_x = \frac{1}{E}\left(\sigma_x - \nu\sigma_y\right) = \frac{12M}{Eth^3} y, \tag{3.5a}$$

$$\varepsilon_y = \frac{1}{E}\left(\sigma_y - \nu\sigma_x\right) = -\frac{12M\nu}{Eth^3} y, \tag{3.5b}$$

$$\varepsilon_{xy} = \frac{1}{2G}\tau_{xy} = 0 \tag{3.5c}$$

are obtained by applying the equations of linear elasticity with Young's modulus E, shear modulus G and Poisson's ratio ν. The displacements x_x and x_y result from the integration of Equations (3.5) as

$$x_x = \int \varepsilon_x \, dx = \frac{1}{E}\left(\frac{12M}{th^3} xy + \Phi(y)\right) \tag{3.6a}$$

and

$$x_y = \int \varepsilon_y \, dy = \frac{1}{E}\left(-\frac{6M\nu}{th^3}y^2 + \Psi(x)\right). \tag{3.6b}$$

From $x_x(x = 0) = 0$ follows immediately $\Phi(y) = 0$. Furthermore, the shear stresses (3.4c) vanish in a pure bending state, which means according to (3.5c)

$$\gamma_{xy} = 2\varepsilon_{xy} = \frac{\partial x_x}{\partial y} + \frac{\partial x_y}{\partial x} = \frac{1}{E}\left(\frac{12M}{th^3}x + \frac{\partial \Psi(x)}{\partial x}\right) = 0. \tag{3.7}$$

By integrating Equation (3.7) w.r.t. x we get

$$\Psi(x) = -\frac{6M}{th^3}x^2 + C. \tag{3.8}$$

The still unknown integration constant C can be determined from the boundary condition $x_y(x = 0, y = 0) = 0$ as $C = 0$ and we finally get the resulting displacement field

$$x_x = \frac{12M}{Eth^3}xy, \tag{3.9a}$$

$$x_y = \frac{1}{E}\left(-\frac{6M\nu}{th^3}y^2 - \frac{6M}{th^3}x^2\right). \tag{3.9b}$$

By treating $12M/Eth^3$ as a constant value we can simply write

$$x_x \sim xy, \tag{3.10a}$$

$$x_y \sim \left(-\frac{\nu y^2}{2} - \frac{x^2}{2}\right) \tag{3.10b}$$

for a displacement field describing pure bending.

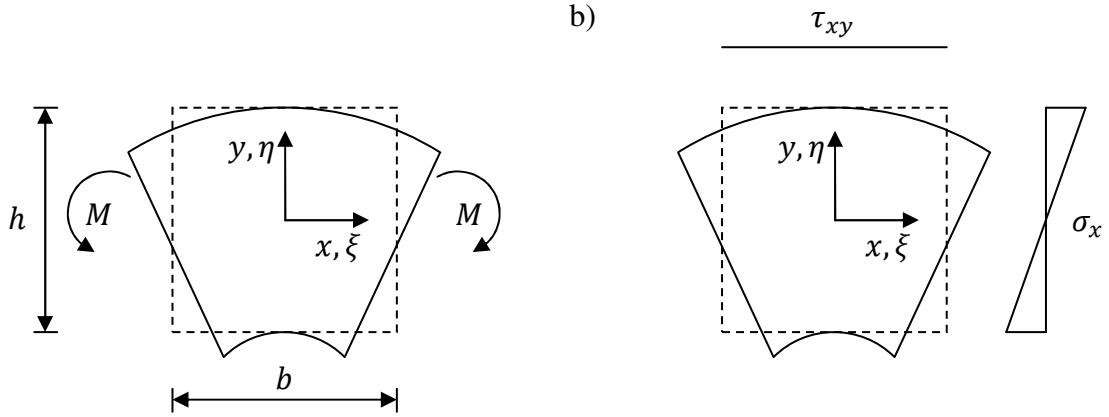


Fig. 3.3 Pure bending situation of rectangular plate (a) and resulting stress distribution (b)

Let us now address the question what happens if the derived displacement field for pure bending is applied to different types of finite elements. In other words, how does a finite element behave if the space around it is bended according to Equations (3.10)? To answer this question for a finite element type in general, it is useful to consider the deformations in the natural ξ, η element coordinate system. For this reason we introduce a simple transformation from an arbitrary rectangular finite element to the basic finite element according to Fig. 3.4

and get

$$\xi = \frac{2x}{b} \tag{3.11}$$

and

$$\eta = \frac{2y}{h}, \tag{3.12}$$

respectively.

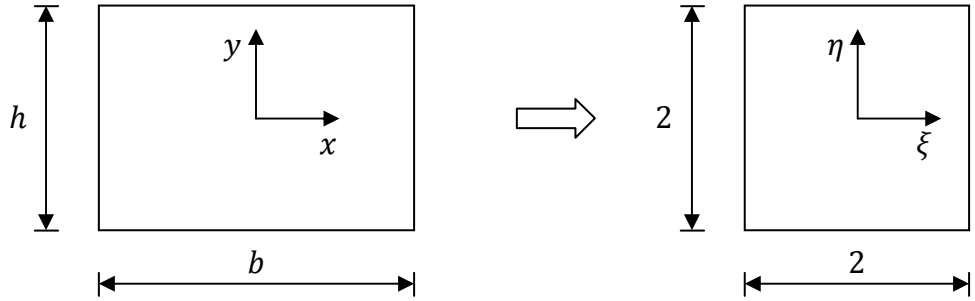


Fig. 3.4 Transformation from x, y coordinate system into natural ξ, η element coordinate system

Before proceeding with applying the deformation field to the finite element we have to think about the so-called aliasing problem [102]. This term is borrowed from sample data theory and describes in our case the incorrect interpolation between nodal values. For example, a linear element cannot correctly map a quadratic deformation state. This effect is called aliasing and shown in Fig. 3.5, which shows the mapping of higher order functions in case of a linear element. The nodal values (circles) are transferred correctly but in between there is only a linear interpolation in the linear element. This knowledge is important for our further considerations.

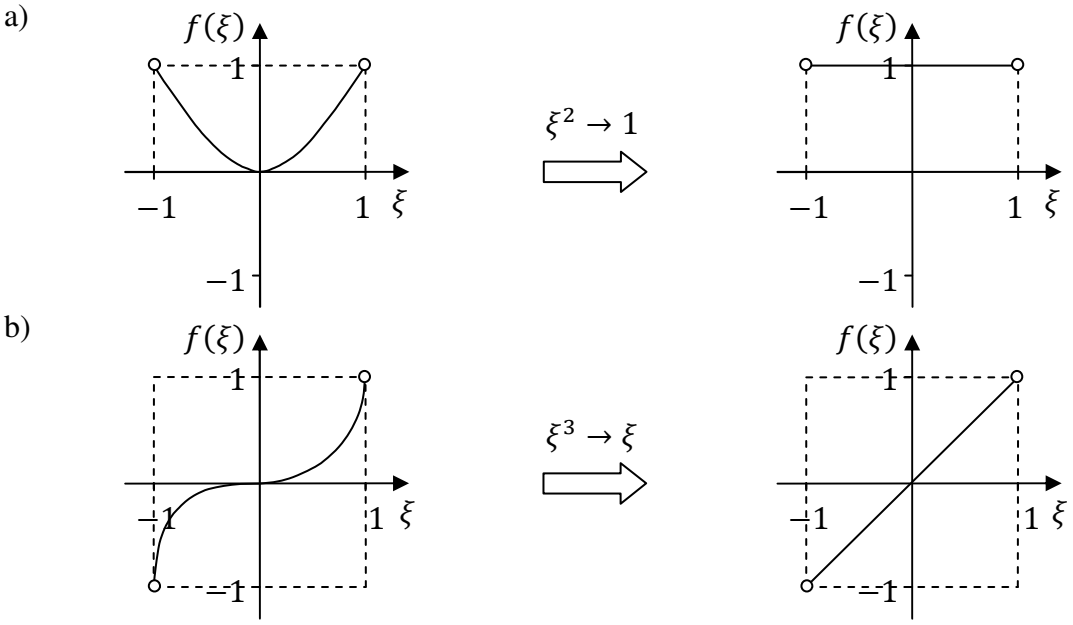


Fig. 3.5 Aliasing of quadratic (a) and cubic (b) functions

In the next step we apply the bending deformation field (3.10) to a bilinear two-dimensional continuum element. This is for example possible by using a pair of forces F at two opposite edges of the element (Fig. 3.6). We assume the bending deformation field causes the displacements

$$x_x = xy = \frac{\xi b}{2} \cdot \frac{\eta h}{2} \tag{3.13a}$$

and

$$x_y = -\frac{\nu y^2}{2} - \frac{x^2}{2} = -\frac{\nu \eta^2 h^2}{8} - \frac{\xi^2 b^2}{8} \tag{3.13b}$$

in the continuum. In the finite element these are mapped to the alias displacements

$${}^a x_x = \frac{\xi b \eta h}{4} \tag{3.14a}$$

and

$${}^a x_y = -\frac{\nu h^2}{8} - \frac{b^2}{8} \tag{3.14b}$$

by using the substitutions $\xi^2 \to 1$ and $\eta^2 \to 1$ because of the linear interpolation functions of the element. To clarify the situation again: the linear finite element undergoes the alias displacements of Equations (3.14) although a quadratic displacement field is applied to the surrounding continuum. From these alias displacements we can compute the resulting strains in the element as

$$\varepsilon_x = \frac{d^a x_x}{dx} = \frac{\partial^a x_x}{\partial \xi}\frac{\partial \xi}{\partial x} + \frac{\partial^a x_x}{\partial \eta}\frac{\partial \eta}{\partial x} = \frac{b\eta h}{4}\frac{2}{b} + \frac{\xi b h}{4} \cdot 0 = \frac{\eta h}{2} = y, \tag{3.15a}$$

$$\varepsilon_y = \frac{d^a x_y}{dy} = \frac{\partial^a x_y}{\partial \xi}\frac{\partial \xi}{\partial y} + \frac{\partial^a x_y}{\partial \eta}\frac{\partial \eta}{\partial y} = 0, \tag{3.15b}$$

$$\begin{aligned} \varepsilon_{xy} &= \frac{1}{2}\left(\frac{d^a x_x}{dy} + \frac{d^a x_y}{dx}\right) \\ &= \frac{1}{2}\left(\frac{\partial^a x_x}{\partial \xi}\frac{\partial \xi}{\partial y} + \frac{\partial^a x_x}{\partial \eta}\frac{\partial \eta}{\partial y} + \frac{\partial^a x_y}{\partial \xi}\frac{\partial \xi}{\partial x} + \frac{\partial^a x_y}{\partial \eta}\frac{\partial \eta}{\partial x}\right) \\ &= \frac{1}{2}\left(\frac{b\eta h}{4} \cdot 0 + \frac{\xi b h}{4}\frac{2}{h} + 0 \cdot \frac{2}{b} + 0 \cdot 0\right) = \frac{\xi b}{4} = \frac{x}{2}. \end{aligned} \tag{3.15c}$$

With Hooke's law [138] for the plane stress state

$$\sigma_x = \frac{E}{1-\nu^2}\left(\varepsilon_x + \nu\varepsilon_y\right), \;\; \sigma_y = \frac{E}{1-\nu^2}\left(\varepsilon_y + \nu\varepsilon_x\right), \;\; \tau_{xy} = \frac{E}{1+\nu}\varepsilon_{xy} \tag{3.16}$$

we get the stresses

$$\sigma_x = \frac{E}{1-\nu^2} y, \tag{3.17a}$$

$$\sigma_y = \frac{E\nu}{1-\nu^2} y, \tag{3.17b}$$

$$\tau_{xy} = \frac{E}{2(1+\nu)} x. \tag{3.17c}$$

Comparing these results with the analytical results from Equations (3.4) we have to experience that in contrast to the analytical solution the stresses σ_y and τ_{xy} are not zero in the finite element (compare also Fig. 3.3b and Fig. 3.6). These additional stresses, which are called parasitic stresses, do also contribute to the strain energy

$$W^{int} = \frac{1}{2} \sigma_{ij} \varepsilon_{ij} \tag{3.18}$$

if the associated strains are non-zero. This is obviously the case for τ_{xy}. By this, the strain energy stored in the element is higher than it actually should be, which on the other hand must result in smaller bending deformations, i.e. the element locks. In other words, the bending deformations are smaller due to the presence of parasitic shear strains and stresses.

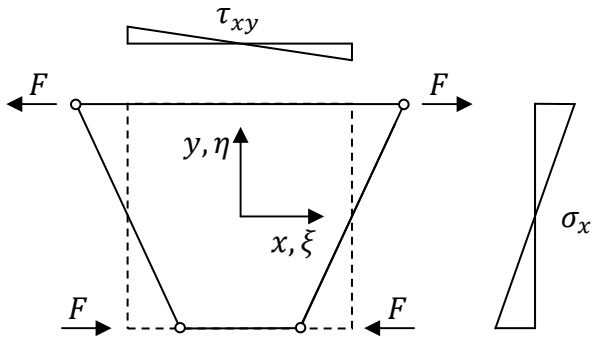


Fig. 3.6 Deformation behavior and stresses of a bilinear finite element under pure bending

The derived behavior of the bilinear finite element can also be observed in real finite element computations. Fig. 3.7 shows a simple beam model subjected to a bending load case. The bending moment is applied by a pair of forces ($F = 1N$) at the right end of the model and the reaction forces at the left end create the second moment. The structure is meshed with five quadratic bilinear finite elements of equal size and form and a linear elastic material model with $E = 210000N/mm^2$ and $\nu = 0.3$ is used.

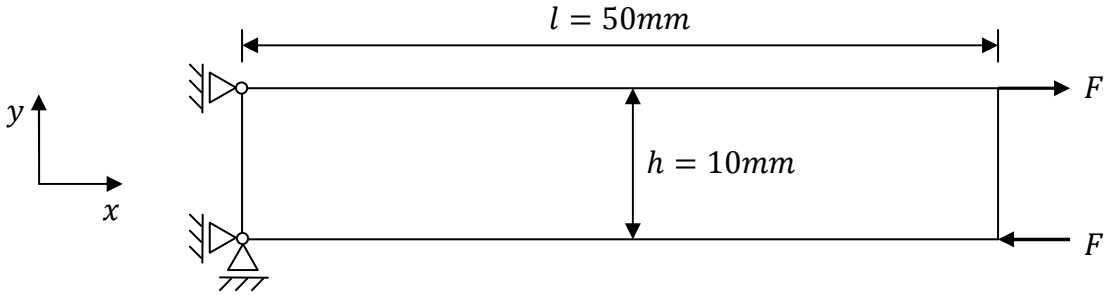


Fig. 3.7 Beam under pure bending

The computed strains in Fig. 3.8 confirm the results of the hand computation. The strain coordinate ε_x varies linearly in y-direction, ε_y equals zero from a numerical point of view and the parasitic strain ε_{xy} is distributed linearly in x-direction in every element as predicted by Equations (3.15a-c).

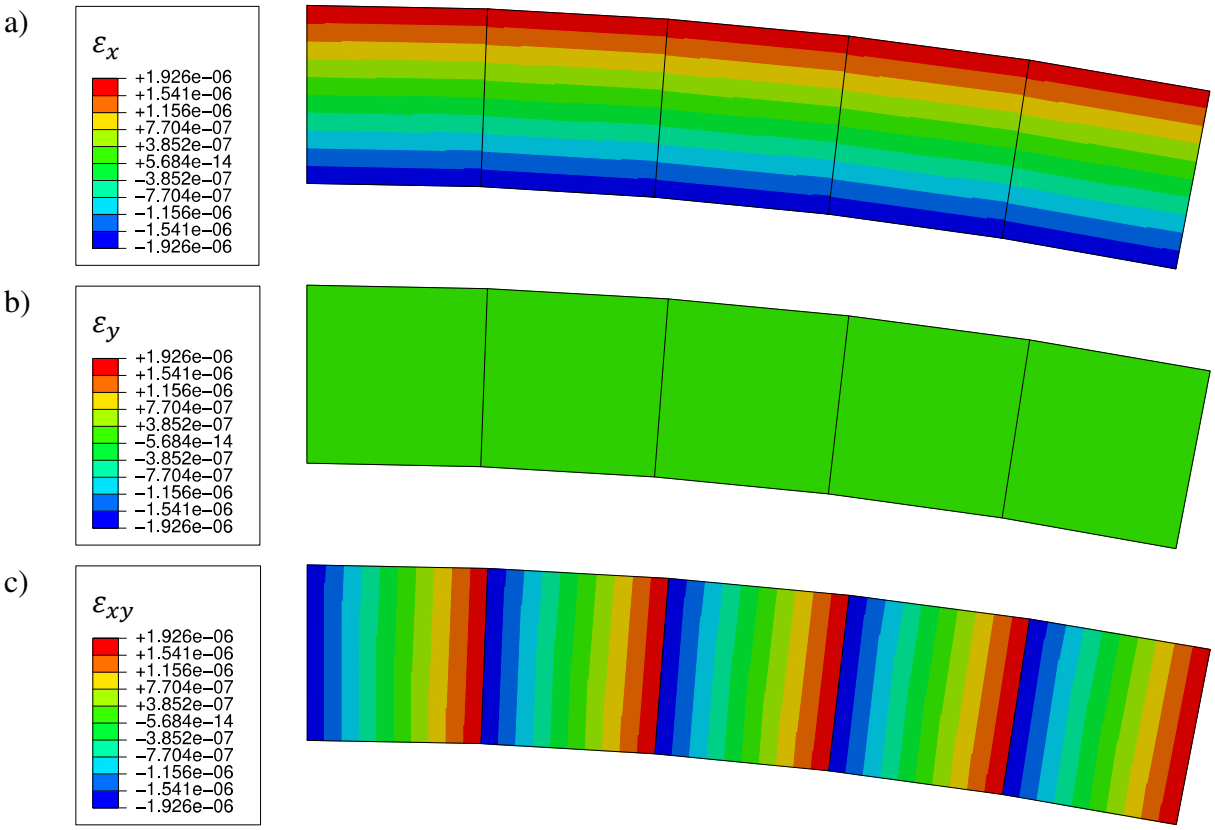


Fig. 3.8 Computed strains for beam subjected to bending with bilinear finite elements

Also the stresses in Fig. 3.9 confirm the theoretical results of Equations (3.17a-c). The stresses σ_x and σ_y vary linearly over the beam thickness in y-direction, where the difference between these two stress coordinates is just the factor ν. The parasitic stress τ_{xy} is linearly distributed in x-direction in every element. It is interesting to observe that the parasitic stresses τ_{xy} as well as the parasitic strains ε_{xy} become zero for $\xi = 0$. This means the computed stresses and strains are correct only in this region of the finite element. Stresses and strains are also evaluated at discrete points in an element, the so-called integration points [11]. These integration or Gauss points have fixed positions in the element. Their number depends on the dimensionality of the element and the shape functions. The used two-dimensional bilinear elements of the example above need four integration points for an exact integration of the functions which are necessary for the computation of the element stiffness matrix. The positions of the integration points for the bilinear element are at $\xi = \pm 1/\sqrt{3}$ and $\eta = \pm 1/\sqrt{3}$ with respect to Fig. 3.4b. Hence, the computed stresses τ_{xy} and strains ε_{xy} are not zero. But this knowledge gives us a first idea for the construction of a shear locking free finite element. If the numerical integration for the element stiffness matrix and therefore the stress and strain evaluation would be done at $\xi = 0$ and $\eta = 0$ the computed stresses would be correct. We keep this in mind and will discuss this so-called principle of underintegrated elements in later sections.

Next, we take a look at the behavior of other two-dimensional finite element formulations with respect to parasitic strains and stresses. It is also possible to mesh the beam problem with

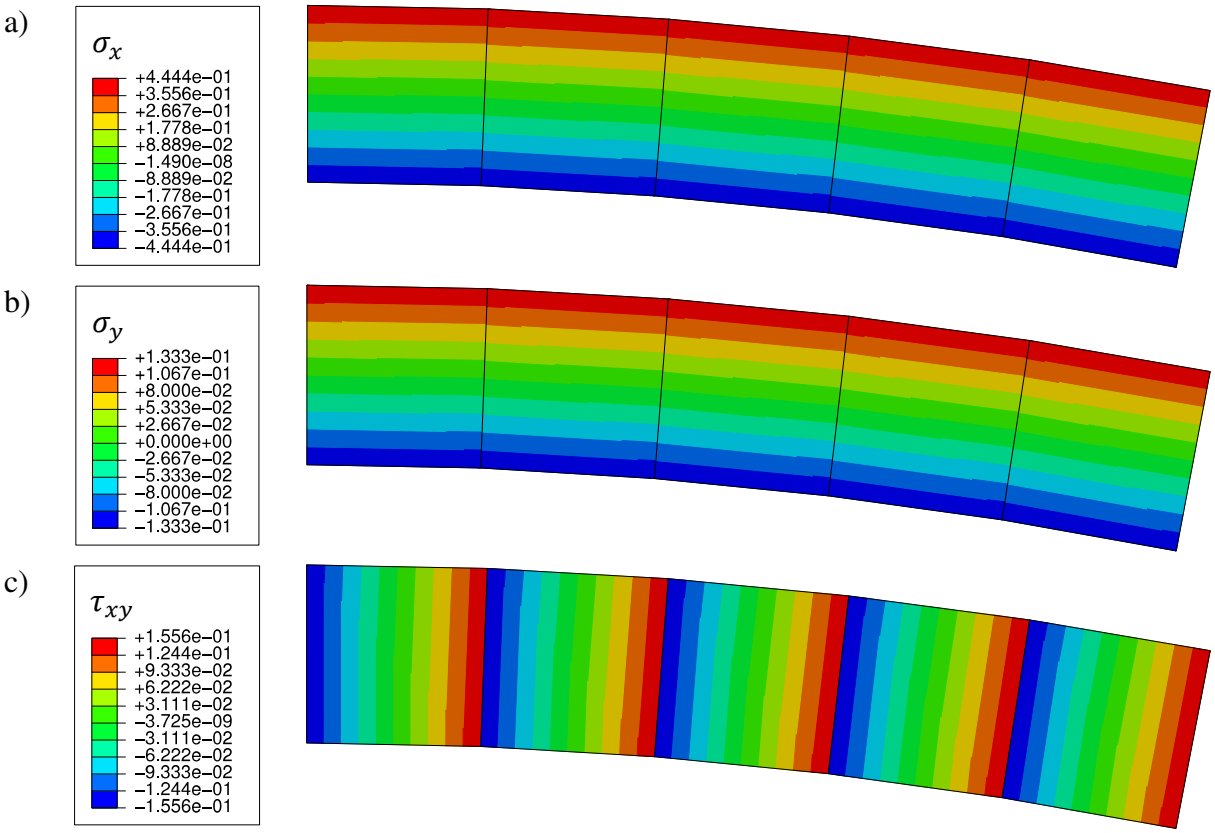


Fig. 3.9 Computed stresses for beam subjected to bending with bilinear finite elements

linear triangular elements. In this case each bilinear rectangular element is divided into two triangular elements. Fig. 3.10 shows two such linear triangular elements subjected to the quadratic deformation field (3.10). These triangular elements are exactly integrated by just one integration point per element [102] although also other integration rules with more integration points are possible but more costly and therefore rarely used [35]. Basically the deformation behavior is the same as for rectangular elements since the same linear interpolation functions are used. This leads to a constant stress state in the element for all stress coordinates (Fig. 3.10), and thus not even the linear behavior of the bending stress σ_x is correctly captured within the element. Consequently the locking effect is even stronger than for rectangular elements, which explains the stiffer behavior of the beam model with

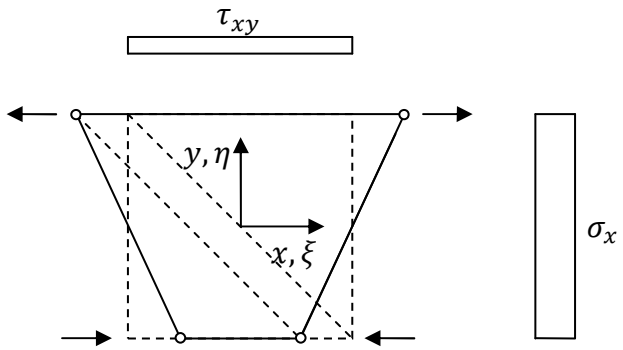


Fig. 3.10 Deformation behavior and stresses of two linear triangular finite elements under pure bending

triangular elements in Fig. 3.2. For this reason linear triangular elements (in the two-dimensional case) and linear tetrahedral elements (in the three-dimensional case) should be avoided for bending problems.

Finally, we want to consider a biquadratic rectangular element (Fig. 3.11) and apply again the displacement field (3.13) to the surrounding continuum. In this case of a finite element with quadratic interpolation functions, these displacements can be captured by the element without any aliasing. Consequently, the strains can be computed directly and become

$$\varepsilon_x = \frac{dx_x}{dx} = y, \tag{3.19a}$$

$$\varepsilon_y = \frac{dx_y}{dy} = -\nu y, \tag{3.19b}$$

$$\varepsilon_{xy} = \frac{1}{2}\left(\frac{dx_x}{dy} + \frac{dx_y}{dx}\right) = \frac{1}{2}(x - x) = 0 \tag{3.19c}$$

and by using Hooke's law (3.16) the stresses

$$\sigma_x = Ey, \tag{3.20a}$$

$$\sigma_y = 0, \tag{3.20b}$$

$$\tau_{xy} = 0. \tag{3.20c}$$

This strain and stress distribution, resulting from a prescribed displacement, corresponds qualitatively to the analytical results of Equations (3.4), resulting from a prescribed bending load, which shows that biquadratic finite elements are able to compute bending strains and stresses correctly because of their higher-order shape functions (compare Fig. 3.11 and Fig. 3.3b). Because there are no parasitic strains or stresses, locking will not appear for such bending load cases.

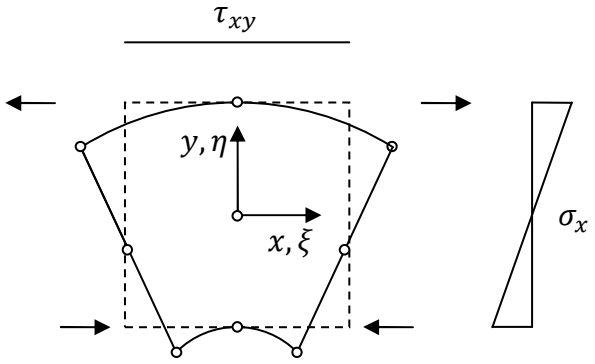


Fig. 3.11 Deformation behavior and stresses of a biquadratic finite element under pure bending

Also the FEM computation of the beam example in Fig. 3.12 confirms these results. The strain coordinate ε_x varies linearly in y-direction, ε_y differs from ε_x by a factor of $(-\nu)$ and ε_{xy} equals zero. The corresponding stresses are displayed in Fig. 3.13. The only non-zero stress coordinate is σ_x as predicted in Equations (3.20a-c).

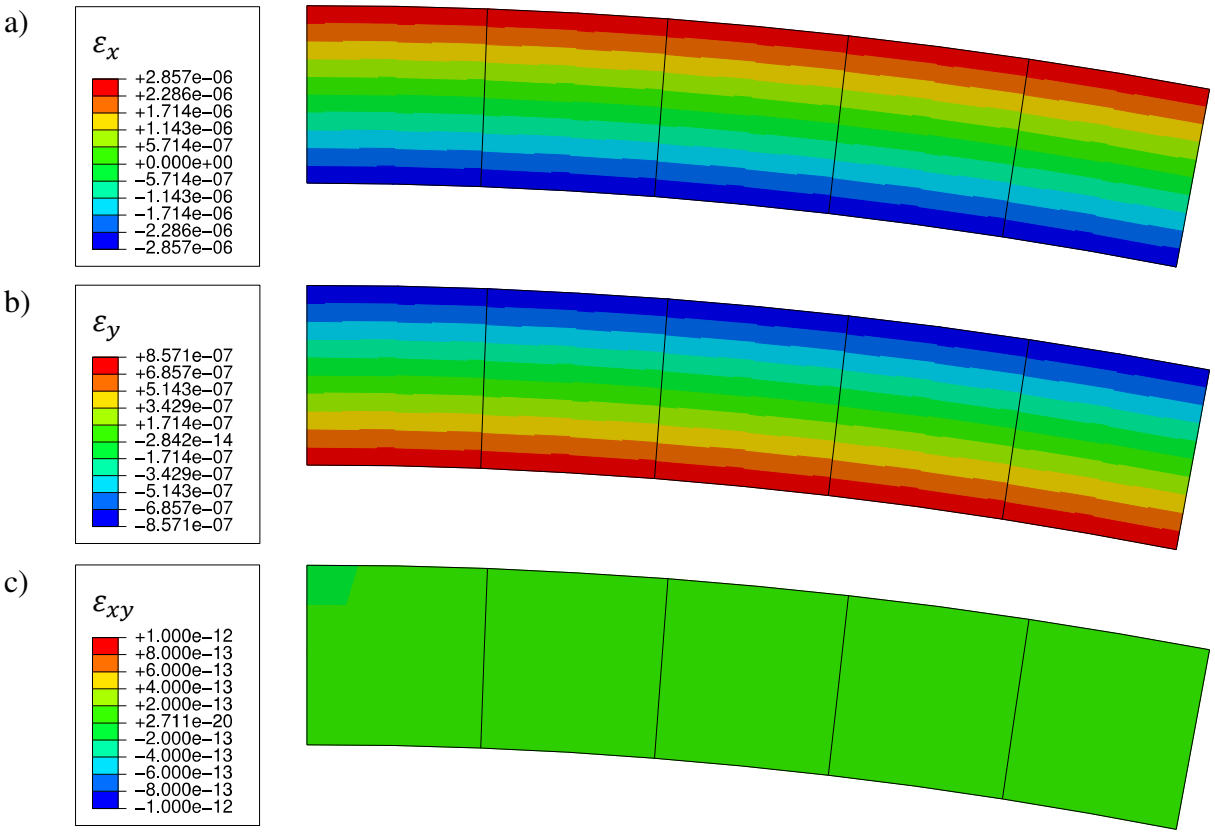


Fig. 3.12 Computed strains for beam subjected to bending with biquadratic finite elements

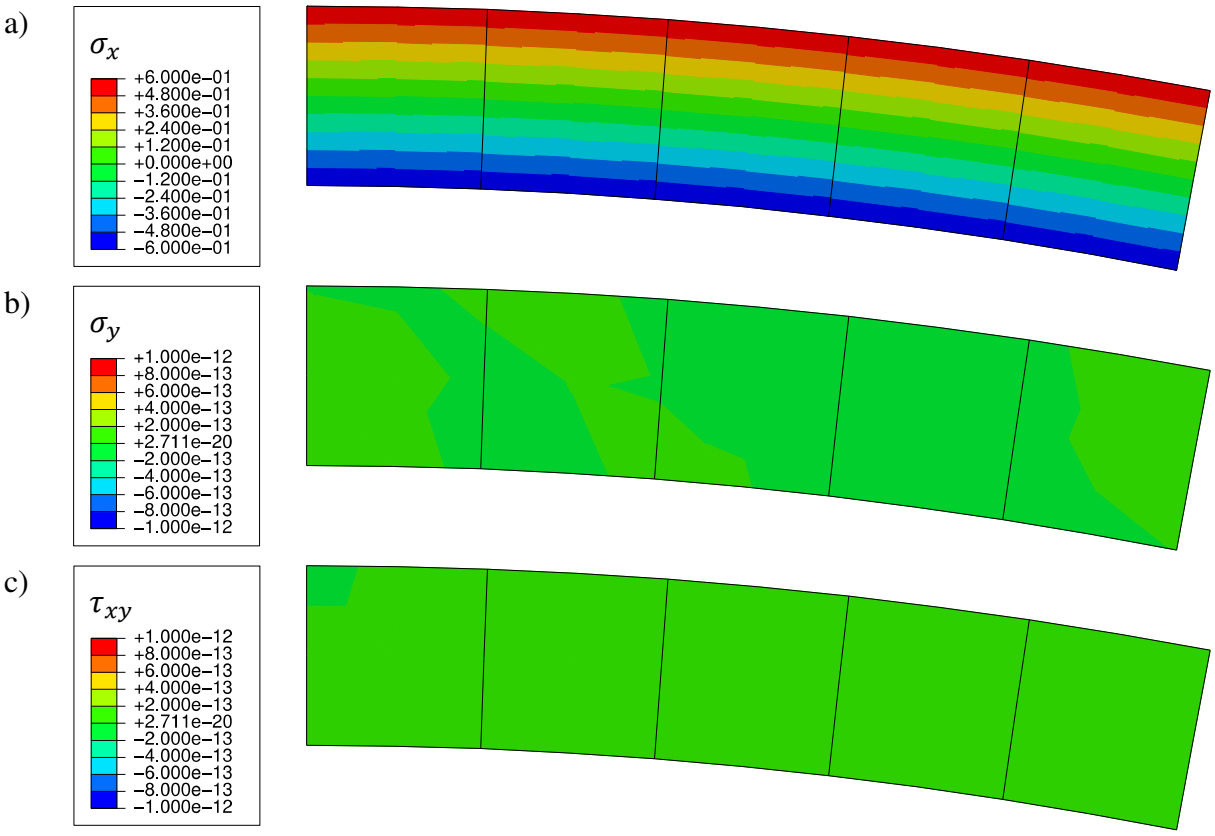


Fig. 3.13 Computed stresses for beam subjected to bending with biquadratic finite elements

3.1.2 Trapezoidal Locking

Trapezoidal locking appears for example if a curved structure is subjected to bending loads or if non-regular meshes are used. In both cases the mesh contains trapezoidal shaped finite elements (Fig. 3.14).

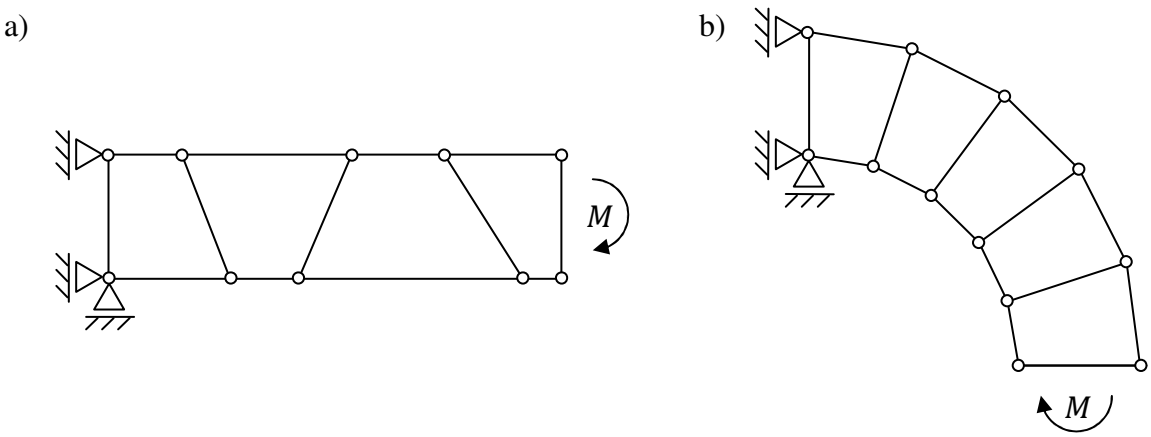


Fig. 3.14 Non-regular mesh (a) and mesh of a curved structure (b)

To recognize the effect of the trapezoidal element shape onto parasitic stresses it is again necessary to perform a transformation from the x, y coordinate system into the natural ξ, η element coordinate system. According to Fig. 3.15 we get for small angles α

$$\eta = \frac{2y}{h} \tag{3.21}$$

and

$$\xi = \frac{2x}{b(y)} \approx \frac{2x}{b - 2\alpha y} \quad \Rightarrow \quad x = \frac{\xi(b - \alpha\eta h)}{2}. \tag{3.22}$$

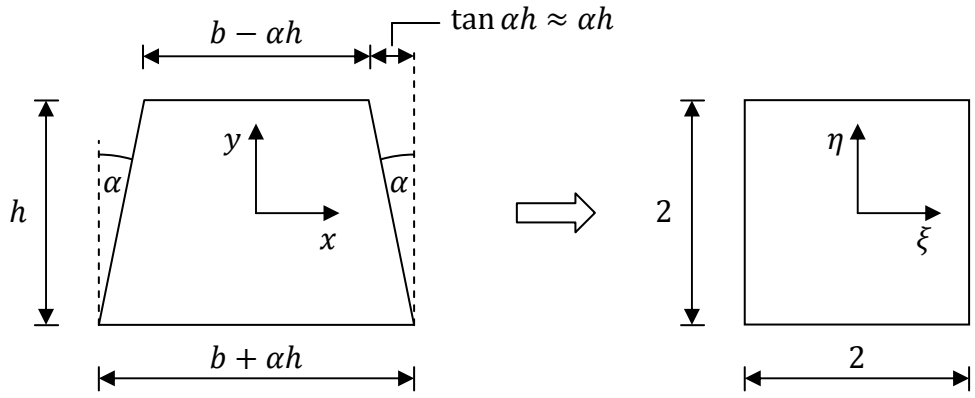


Fig. 3.15 Transformation from x, y coordinate system into natural ξ, η element coordinate system for trapezoidal element

If we assume a pure bending deformation of the continuum according to (3.13) like for the rectangular element before, the displacements

$$x_x = xy = \frac{\xi(b-\alpha\eta h)}{2}\cdot\frac{\eta h}{2} = \frac{\xi}{4}(bh\eta - \alpha h^2\eta^2) \tag{3.23a}$$

and

$$\begin{aligned} x_y &= -\frac{\nu y^2}{2} - \frac{x^2}{2} = -\frac{\nu\eta^2 h^2}{8} - \frac{\xi^2(b-\alpha\eta h)^2}{8} \\ &= -\frac{\nu h^2}{8}\eta^2 - \frac{\xi^2}{8}(b^2 - 2b\alpha\eta h + \alpha^2 h^2\eta^2) \end{aligned} \tag{3.23b}$$

should follow in the element. But due to the linear shape functions we get with the substitutions $\xi^2 \to 1$ and $\eta^2 \to 1$ according to Fig. 3.5 the alias displacements

$${}^a x_x = \frac{\xi b\eta h - \alpha\xi h^2}{4} \tag{3.24a}$$

and

$${}^a x_y = -\frac{\nu h^2}{8} - \frac{b^2 - 2b\alpha\eta h + \alpha^2 h^2}{8}. \tag{3.24b}$$

in the element. Following from this, the strains

$$\begin{aligned} \varepsilon_x &= \frac{d^a x_x}{dx} = \frac{\partial {}^a x_x}{\partial\xi}\frac{\partial\xi}{\partial x} + \frac{\partial {}^a x_x}{\partial\eta}\frac{\partial\eta}{\partial x} = \frac{b\eta h - \alpha h^2}{4}\cdot\frac{2}{b-2\alpha y} + \frac{\xi bh}{4}\cdot 0 \\ &= \frac{b\eta h - \alpha h^2}{2(b-2\alpha y)} = \frac{2yb - \alpha h^2}{2(b-2\alpha y)}, \end{aligned} \tag{3.25a}$$

$$\varepsilon_y = \frac{d^a x_y}{dy} = \frac{\partial {}^a x_y}{\partial\xi}\frac{\partial\xi}{\partial y} + \frac{\partial {}^a x_y}{\partial\eta}\frac{\partial\eta}{\partial y} = 0 + \frac{b\alpha h}{4}\cdot\frac{2}{h} = \frac{b\alpha}{2}, \tag{3.25b}$$

$$\begin{aligned} \varepsilon_{xy} &= \frac{1}{2}\left(\frac{d^a x_x}{dy} + \frac{d^a x_y}{dx}\right) = \frac{1}{2}\left(\frac{\partial {}^a x_x}{\partial\xi}\frac{\partial\xi}{\partial y} + \frac{\partial {}^a x_x}{\partial\eta}\frac{\partial\eta}{\partial y} + \frac{\partial {}^a x_y}{\partial\xi}\frac{\partial\xi}{\partial x} + \frac{\partial {}^a x_y}{\partial\eta}\frac{\partial\eta}{\partial x}\right) \\ &= \frac{1}{2}\left(\frac{b\eta h - \alpha h^2}{4}\cdot\frac{4\alpha x}{(b-2\alpha y)^2} + \frac{\xi bh}{4}\frac{2}{h} + 0 + 0\right) = \frac{x(b^2 - \alpha^2 h^2)}{2(b-2\alpha y)^2} \end{aligned} \tag{3.25c}$$

are obtained. The corresponding stresses can be computed with (3.16) as

$$\sigma_x = \frac{E}{2(1-\nu^2)}\left(\frac{2yb - \alpha h^2}{b-2\alpha y} + \nu b\alpha\right), \tag{3.26a}$$

$$\sigma_y = \frac{E}{2(1-\nu^2)}\left(b\alpha + \nu\cdot\frac{2yb - \alpha h^2}{b-2\alpha y}\right), \tag{3.26b}$$

$$\tau_{xy} = \frac{E}{2(1-\nu^2)}\left(\frac{x(b^2 - \alpha^2 h^2)}{(b-2\alpha y)^2}\right). \tag{3.26c}$$

Obviously, the angle α describing the initial trapezoidal element distortion influences the amount of the parasitic stresses. With respect to the strains one can notice that in contrast to a rectangular finite element the strain component ε_y is not zero anymore, but takes a constant value for $\alpha \neq 0$.

For the visualization of these results we use again the FEM example of Fig. 3.7, but in opposite to the regular meshes used for example in Fig. 3.8, the middle element is initially distorted such that it takes a trapezoidal shape with $\alpha = 0.1rad$ as shown in Fig. 3.16. The computed strains Fig. 3.17 show the non-zero strain coordinate ε_y in the trapezoidal shaped middle element. The behavior of the other strain coordinates is similar as in Fig. 3.8 since the angle α is small. The stress distribution is also similar to the one of the structured mesh (compare Fig. 3.18 and Fig. 3.9). Since the strain coordinate ε_y is non-zero for the trapezoidal shaped element(s), there is a contribution to the strain energy according to Equation (3.18) and the locking phenomenon is increased, which leads to a smaller maximum deflection of the model (Fig. 3.19).

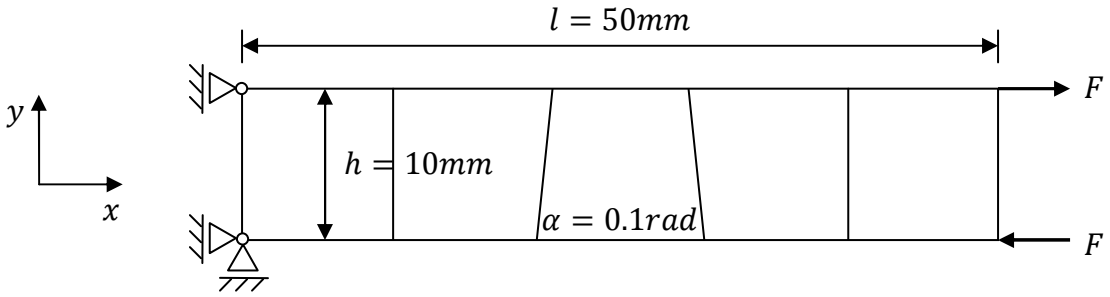


Fig. 3.16 Beam under pure bending with non-regular mesh

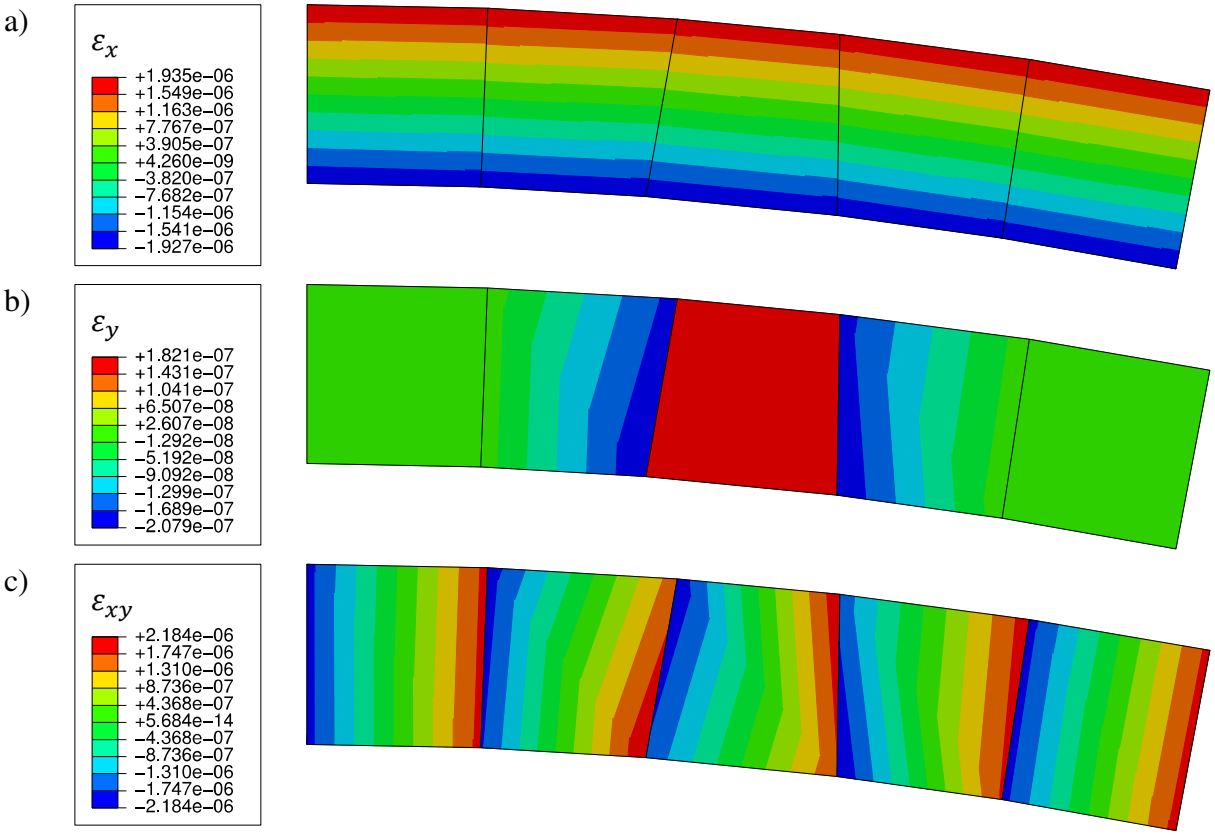


Fig. 3.17 Computed strains for beam subjected to bending with bilinear finite elements in a non-regular mesh

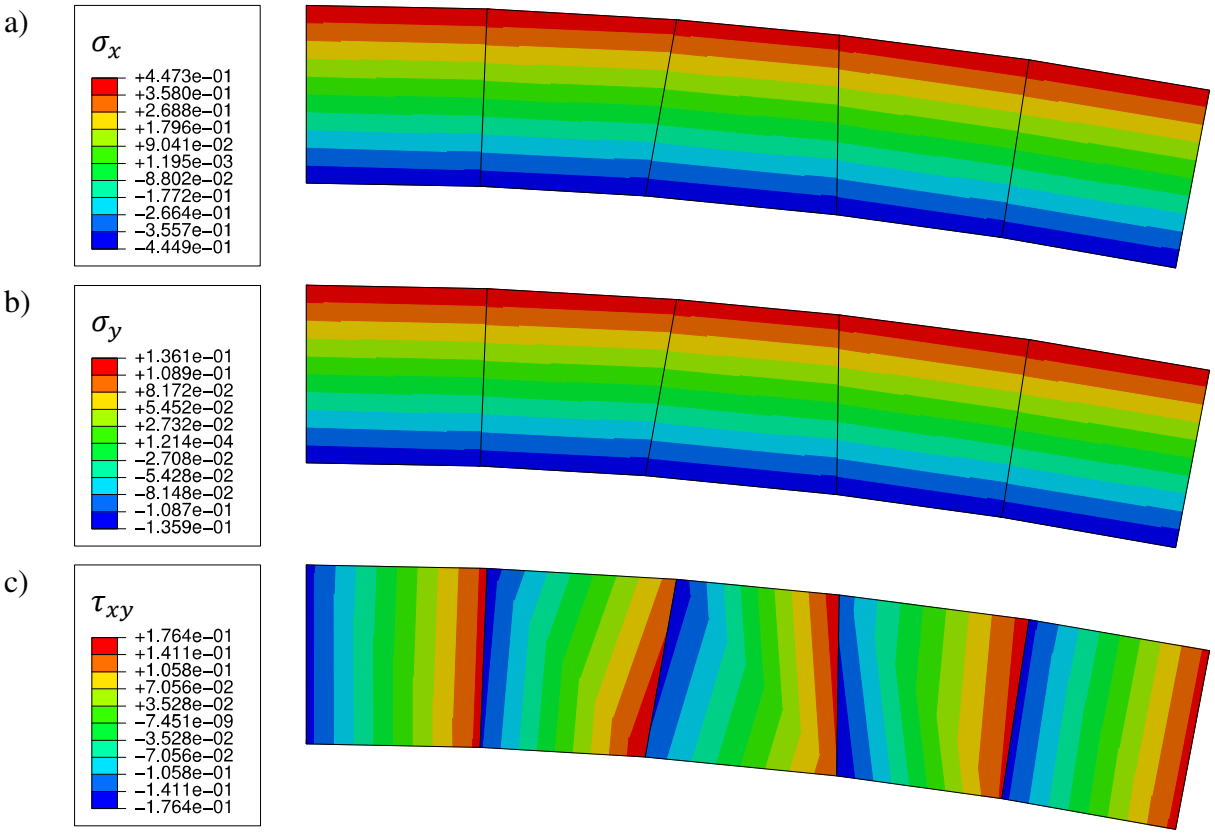


Fig. 3.18 Computed stresses for beam subjected to bending with bilinear finite elements in a non-regular mesh

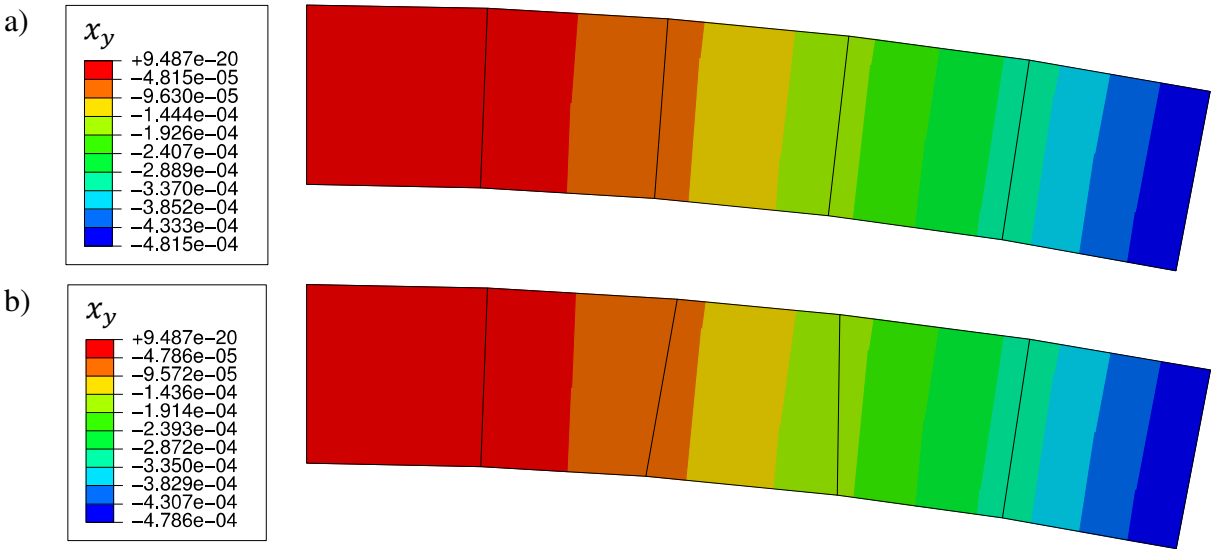


Fig. 3.19 Displacements of beam for regular (a) and non-regular mesh (b)

3.1.3 Volumetric Locking

Volumetric locking is sometimes also called dilatation locking and is caused by the Poisson's ratio of the used material. It appears for example in bending problems. For $\nu = 0$ no

volumetric locking is visible, but the stronger the materials incompressibility becomes the stronger becomes the stiffening effect and reaches its maximum for an incompressible material with $\nu = 0.5$.

An illustrative explanation for the stiffening effect can be given by Fig. 3.20. The bending moment applied to the bilinear element causes a thickening of the fibers in the compression zone and a thinning in the tension zone due to the Poisson effect. Following from this, the middle line of the element should move upwards, but since the nodes don't undergo any vertical movement (see Fig. 3.8b), a linear shape function is not able to capture this behavior. The middle line is therefore fixed (locked) at its middle position which causes the locking.

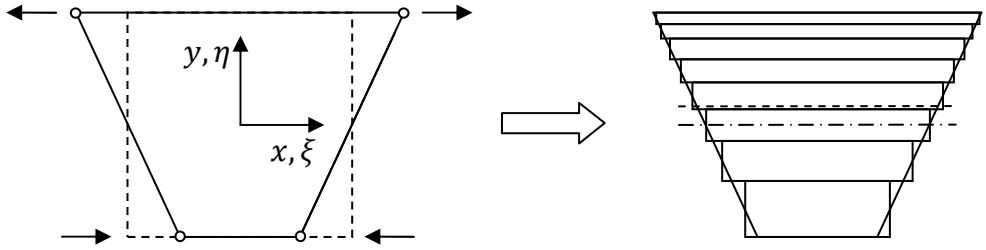


Fig. 3.20 Poisson effect under bending moment [81]

The biggest effect of the Poisson's ratio with respect to locking is observable if a plane strain state ($\varepsilon_z = 0$) is assumed. In this case Hooke's law [138] becomes

$$\sigma_x = \frac{E}{(1+\nu)(1-2\nu)}\Big((1-\nu)\varepsilon_x + \nu\varepsilon_y\Big), \tag{3.27a}$$

$$\sigma_y = \frac{E}{(1+\nu)(1-2\nu)}\Big((1-\nu)\varepsilon_y + \nu\varepsilon_x\Big), \tag{3.27b}$$

$$\sigma_z = \nu\big(\sigma_x + \sigma_y\big), \tag{3.27c}$$

$$\tau_{xy} = \frac{E}{1+\nu}\varepsilon_{xy} \tag{3.27d}$$

or in strain explicit form

$$\varepsilon_x = \frac{1-\nu^2}{E}\Big(\sigma_x - \frac{\nu}{1-\nu}\sigma_y\Big), \tag{3.28a}$$

$$\varepsilon_y = \frac{1-\nu^2}{E}\Big(\sigma_y - \frac{\nu}{1-\nu}\sigma_x\Big), \tag{3.28b}$$

$$\varepsilon_z = 0, \tag{3.28c}$$

$$\varepsilon_{xy} = \frac{1+\nu}{E}\tau_{xy}. \tag{3.28d}$$

Similar as in Section 3.1.1, a displacement field describing pure bending can be derived by applying (3.4) in (3.28) and subsequent integration as

$$x_x = xy, \tag{3.29a}$$

$$x_y = -\frac{x^2}{2} - \frac{\nu y^2}{2(1-\nu)}. \tag{3.29b}$$

From this we obtain the correct strains

$$\varepsilon_x = \frac{dx_x}{dx} = y, \tag{3.30a}$$

$$\varepsilon_y = \frac{dx_y}{dy} = -\frac{\nu}{1-\nu} y, \tag{3.30b}$$

$$\varepsilon_z = 0 \tag{3.30c}$$

$$\varepsilon_{xy} = \frac{1}{2}\left(\frac{dx_x}{dy} + \frac{dx_y}{dx}\right) = 0, \tag{3.30d}$$

and with (3.27) the stresses

$$\sigma_x = \frac{E}{1-\nu^2} y, \tag{3.31a}$$

$$\sigma_y = 0, \tag{3.31b}$$

$$\sigma_z = \frac{E\nu}{1-\nu^2} y \tag{3.31c}$$

$$\tau_{xy} = 0. \tag{3.31d}$$

For a bilinear rectangular element, we firstly have to transform the displacement field (3.29) to the natural element coordinate system according to (3.11) and (3.12) as

$$x_x = \frac{\xi b}{2} \cdot \frac{\eta h}{2}, \tag{3.32a}$$

$$x_y = -\frac{\xi^2 b^2}{8} - \frac{\nu \eta^2 h^2}{8(1-\nu)}. \tag{3.32b}$$

Following from this the alias displacements for a bilinear element, which are the displacements the element can actually describe, are again obtained by the substitutions $\xi^2 \to 1$ and $\eta^2 \to 1$ according to Fig. 3.5 as

$${}^a x_x = \frac{\xi b \eta h}{4}, \tag{3.33a}$$

$${}^a x_y = -\frac{b^2}{8} - \frac{\nu h^2}{8(1-\nu)}. \tag{3.33b}$$

Computing the strains from these alias displacements yields

$$\varepsilon_x = \frac{d^a x_x}{dx} = \frac{\partial {}^a x_x}{\partial \xi}\frac{\partial \xi}{\partial x} + \frac{\partial {}^a x_x}{\partial \eta}\frac{\partial \eta}{\partial x} = \frac{b\eta h}{4} \cdot \frac{2}{b} + \frac{\xi b h}{4} \cdot 0 = \frac{\eta h}{2} = y, \tag{3.34a}$$

$$\varepsilon_y = \frac{d^a x_y}{dy} = \frac{\partial {}^a x_y}{\partial \xi}\frac{\partial \xi}{\partial y} + \frac{\partial {}^a x_y}{\partial \eta}\frac{\partial \eta}{\partial y} = 0, \tag{3.34b}$$

$$\varepsilon_z = 0, \tag{3.34c}$$

$$\begin{aligned}\varepsilon_{xy} &= \frac{1}{2}\left(\frac{d^a x_x}{dy} + \frac{d^a x_y}{dx}\right)\\ &= \frac{1}{2}\left(\frac{\partial^a x_x}{\partial \xi}\frac{\partial \xi}{\partial y} + \frac{\partial^a x_x}{\partial \eta}\frac{\partial \eta}{\partial y} + \frac{\partial^a x_y}{\partial \xi}\frac{\partial \xi}{\partial x} + \frac{\partial^a x_y}{\partial \eta}\frac{\partial \eta}{\partial x}\right)\\ &= \frac{1}{2}\left(\frac{b\eta h}{4}\cdot 0 + \frac{\xi b h}{4}\frac{2}{h} + 0 + 0\right) = \frac{\xi b}{4} = \frac{x}{2}.\end{aligned} \tag{3.34d}$$

Consequently, with (3.27) we get for the stresses in the element

$$\sigma_x = \frac{E(1-\nu)}{(1+\nu)(1-2\nu)}y, \tag{3.35a}$$

$$\sigma_y = \frac{E\nu}{(1+\nu)(1-2\nu)}y, \tag{3.35b}$$

$$\sigma_z = \nu(\sigma_x + \sigma_y) = \frac{E\nu}{(1+\nu)(1-2\nu)}y, \tag{3.35c}$$

$$\tau_{xy} = \frac{E}{2(1+\nu)}x. \tag{3.35d}$$

Especially the shear stress coordinate τ_{xy} is a parasitic stress since it contributes to the strain energy due to $\varepsilon_{xy} \neq 0$. Also σ_x is not computed correctly in comparison to the analytical solution (3.31a). This becomes even clearer if the analytically computed strain energy

$$\begin{aligned}{}^{anal}W^{int}_{plane\ strain} &= \frac{1}{2}\sigma_{ij}\varepsilon_{ij} = \frac{1}{2}\left(\frac{E}{1-\nu^2}y^2 + 0 + 0 + 0\right)\\ &= \frac{Ey^2}{2(1-\nu^2)}\end{aligned} \tag{3.36}$$

and the numerically computed strain energy of the bilinear element

$$\begin{aligned}{}^{num}W^{int}_{plane\ strain} &= \frac{1}{2}\left(\frac{E(1-\nu)}{(1+\nu)(1-2\nu)}y^2 + 0 + 0 + 2\cdot\frac{E}{2(1+\nu)}\frac{x^2}{2}\right)\\ &= \frac{E}{2(1+\nu)}\left(\frac{1-\nu}{1-2\nu}y^2 + \frac{x^2}{2}\right)\end{aligned} \tag{3.37}$$

are compared. Now both sources of errors are visible: the x^2 term results from spurious shear stresses/strains and is responsible for the shear locking and the term in front of y^2 is responsible for volumetric locking.

For the plane stress state (3.16) the situation is not that critical. The displacement field (3.13) leads to the analytically computed strains

$$\varepsilon_x = \frac{dx_x}{dx} = y, \tag{3.38a}$$

$$\varepsilon_y = \frac{dx_y}{dy} = -\nu y, \tag{3.38b}$$

$$\varepsilon_{xy} = \frac{1}{2}\left(\frac{dx_x}{dy} + \frac{dx_y}{dx}\right) = 0, \tag{3.38c}$$

and with (3.16) to the plane stresses

$$\sigma_x = \frac{E}{1-\nu^2}\left(\varepsilon_x + \nu\varepsilon_y\right) = Ey, \tag{3.39a}$$

$$\sigma_y = \frac{E}{1-\nu^2}\left(\varepsilon_y + \nu\varepsilon_x\right) = 0, \tag{3.39b}$$

$$\tau_{xy} = \frac{E}{1+\nu}\varepsilon_{xy} = 0 \tag{3.39c}$$

and finally to the strain energy

$${}^{anal}W^{int}_{plane\ stress} = Ey^2 + 0 + 0 + 0 = Ey^2. \tag{3.40}$$

If a bilinear element is subjected to the same displacement field we get the strains and stresses of Equations (3.15) and (3.17) and consequently the strain energy stored in the bilinear element

$$\begin{aligned} {}^{num}W^{int}_{plane\ stress} &= \frac{1}{2}\left(\frac{Ey}{1-\nu^2}y + 0 + 0 + 2\cdot\frac{Ex}{2(1+\nu)}\frac{x}{2}\right) \\ &= \frac{E}{2}\left(\frac{y^2}{1-\nu^2} + \frac{x^2}{2(1+\nu)}\right). \end{aligned} \tag{3.41}$$

Fig. 3.21 shows the strain energy dependence of the Poisson's ratio ν for the plane stress state and the plane strain state for constant values $E = 1$ and $x = y = 5$. As it can be seen, the strain energy in the bilinear plane strain element ("Numerical plane strain") increases dramatically for $\nu \to 0.5$. It should also be noted that volumetric locking does not only appear for bending problems [102].

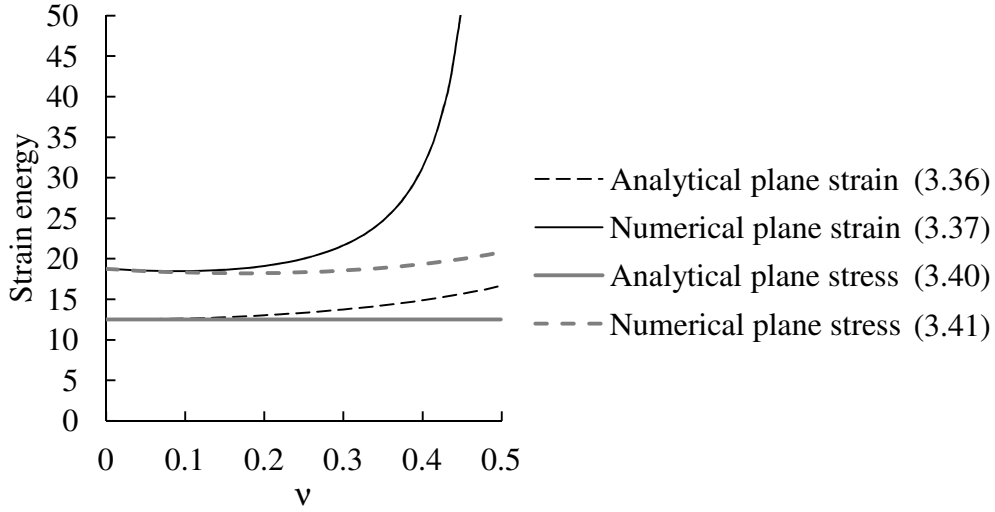


Fig. 3.21 Strain energy under pure bending in dependence of Poisson's ratio

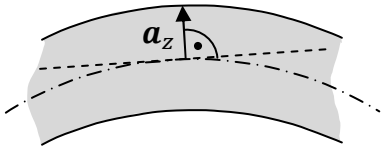
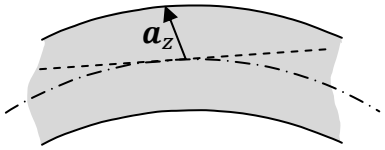
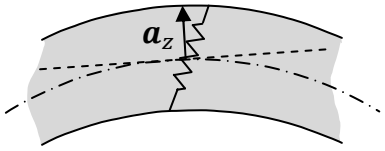
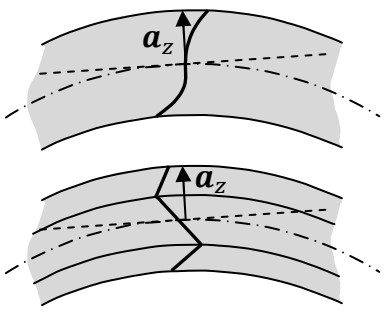
3.2 Locking of Structural Elements

In contrast to continuum elements, structural elements are based on certain assumptions about the stress state or the kinematics and properties of a structure. This leads on the one hand to

efficient element formulations and faster computations, but on the other hand the computational results are only accurate if the assumptions used in the element formulation are at least approximately fulfilled also for the real structure. Well-known examples for structural elements are beam and shell elements. Both incorporate assumptions about the governing stress state as well as for the kinematic behavior, which are approximately fulfilled for slender beams and thin shell structures without loading in thickness direction and moderate curvature.

Especially for shell elements there is an enormous number of different formulations that has been developed during the last decades. One of the first systematizations of the "zoo" of shell element formulations can be found in [106]. Due to the huge number of publications in this field a review about shell element formulations and their history would by far go beyond the scope of this section. More information can be found in some special textbooks, e.g. [29]. Nevertheless, a brief overview about different shell element formulations is given in Table 3.1, which is an attempt of grouping the diverse formulations with regard to their number of

Table 3.1 Categories of shell element formulations

3-parameter models (Kirchhoff-Love) • Cross sections stay plane and undeformed • No shear deformation	$\sigma_z = 0,$ $\gamma_{xz} = \gamma_{yz} = 0$	Plane stress state	
5-parameter models (Reissner-Mindlin) • Cross sections stay plane and undeformed • Constant shear distribution	$\sigma_z = 0,$ $\gamma_{xz} \neq 0, \gamma_{yz} \neq 0$		
6- or 7-parameter models • Cross sections stay plane but may deform	$\sigma_z \neq 0,$ $\varepsilon_z \neq 0,$ $\gamma_{xz} \neq 0, \gamma_{yz} \neq 0$	Unmodified 3D constitutive laws	
Higher order models (multi layer or director) • Cross sections need not stay plane and may deform	$\sigma_z \neq 0,$ $\varepsilon_z \neq 0,$ $\gamma_{xz} \neq 0, \gamma_{yz} \neq 0$		

parameters following [60]. Typically, 5-parameter models based on the Reissner-Mindlin shear deformation theory [123,105] are used in commercial FE codes. The simpler Kirchhoff-Love formulations [76,99] are also implemented sometimes. It might be surprising, but from a numerical point of view the implementation of the shear deformation-free Kirchhoff-Love theory for shells or Euler-Bernoulli theory for beams [91] is more elaborate than for the constant shear Reissner-Mindlin shells or Timoshenko beams [137]. This is grounded in the demand for constant shear force distributions. In the Euler-Bernoulli beam the shear force is proportional to the third derivative of the deflection $Q(x) \sim w'''(x)$ which is why at least a third-order interpolation polynomial for the displacement function $w(x)$ is necessary. For a shear flexible theory, a first-order polynomial is sufficient. The same applies to shell elements: Kirchhoff-Love elements request C^1 continuity of the interpolation function (compatibility of displacements and rotations), whereas Reissner-Mindlin formulations need just C^0 continuity (compatibility of displacements).

3.2.1 Curvature-Thickness Locking

Curvature-thickness locking is a special case of trapezoidal locking. It is limited to three-dimensional shell elements with 7-parameter formulations and thickness flexibility if applied in the mesh of curved structures (Fig. 3.22). The described elements are derived from degenerated continuum elements and the same kinematics are used. By this, the locking behavior is very similar as described in Section 0. The parasitic strains and stresses in thickness direction lead to a stiffening of the element, which does not appear for initially plane structures and can be minimized by a finer mesh. A more detailed description of this effect can be found in [122] and [23].

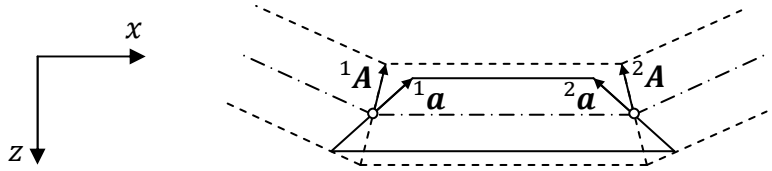


Fig. 3.22 Curved structure meshed with shell elements and directors of undeformed state ($\boldsymbol{A}$) and deformed state ($\boldsymbol{a}$) under bending

3.2.2 Poisson Locking

Poisson locking is a similar effect like the volumetric locking for continuum elements. In 5-parameter formulations (Reissner-Mindlin formulations) there is no thickness flexibility which makes it impossible for the element to capture the thickness strain distribution according to

$$\varepsilon_z = \frac{1}{E}\left[\sigma_z - \nu\left(\sigma_x + \sigma_y\right)\right] \tag{3.42}$$

for example in bending situations, caused by the Poisson effect, correctly. The obstruction of the thickness strain would actually cause parasitic stresses in thickness direction but due to the

premise $\sigma_z = 0$, which holds for Reissner-Mindlin formulations, locking is avoided.

For shell element formulations with thickness flexibility (6 or more parameters), which are using a linear displacement interpolation function in thickness direction and thus are able to produce constant shear distributions, locking will appear. The reason for this behavior is that these elements are able to compute stresses in thickness direction (similar as described in Section 3.1.1) which finally results in a contribution to the strain energy. Only for membrane stress states the constant strain distribution fits to the constant stress distribution in thickness direction. A remedy for the described phenomenon is the introduction of a quadratic displacement interpolation function in thickness direction allowing linear strain distributions.

3.2.3 Transverse Shear Locking

One of the most important stiffening effects for structural elements is the transverse shear locking. In contrast to the shear locking of continuum elements, the phrase "transverse" indicates the direction of the effect for structural elements having a particular thickness direction which is not the case for continuum elements. Affected are structural elements based on Reissner-Mindlin kinematics (5-parameter shells or Timoshenko beams). The phenomenon does not appear for shear deformation free elements like Kirchhoff-Love shells or Euler-Bernoulli beams.

In order to describe the locking effect, the example of a Timoshenko beam element is used. According to Fig. 3.23a, the interpolation functions for the transverse displacement w and rotations φ for a linear plane element are

$$^{h}w(x) = \left(1 - \frac{x}{l}\right) w_1 + \frac{x}{l} w_2 \tag{3.43}$$

and

$$^{h}\varphi(x) = \left(1 - \frac{x}{l}\right) \varphi_1 + \frac{x}{l} \varphi_2 \tag{3.44}$$

using the kinematics

$$\gamma = \varepsilon_{xz} + \varepsilon_{zx} = \frac{\partial w}{\partial x} + \varphi, \quad \kappa = -\frac{\partial \varphi}{\partial x} \tag{3.45}$$

with the shear strain γ and the curvature κ. By introducing the shear deformation area A_s, the constitutive law becomes

$$Q_z = GA_s\gamma, \quad M = EI\kappa. \tag{3.46}$$

The shear deformation area $A_s = A/k_z$ is defined as the cross section area A of the element divided by a shear correction parameter k_z which depends on the shape of the cross section. It takes into account and corrects the simplified shear deformation assumption of the Reissner-Mindlin or Timoshenko theory, and can be calculated as

$$k_z = \frac{A}{Q_z^2} \int_A \tau_{xz}^2 \, \mathrm{d}A \tag{3.47}$$

with Q_z being the shearing force [11]. Equation (3.47) is based on the assumption about the equality of shear deformation energies. Other possibilities for the computation of shear correction parameters are given in [36].

For a pure bending state ($w_1 = w_2 = 0,\ -\varphi_1 = \varphi_2 = \varphi$) we get for the shear strain distribution

$$ {}^h\gamma(x) = \frac{\partial\, {}^h w}{\partial x} + {}^h\varphi = \left(\frac{x}{l} - 1\right)\varphi + \frac{x}{l}\varphi = \varphi\left(\frac{2x}{l} - 1\right) \tag{3.48} $$

in the element and thus a linear distribution of the shear strains (Fig. 3.23b) which actually should be zero for a pure bending state. The reason for the non-zero and therefore parasitic shear distribution is again the used linear interpolation function for the displacements and rotations. According to (3.48) the shear distribution equals zero for all x only for $\varphi = 0$ which means that no deformation is present at all.

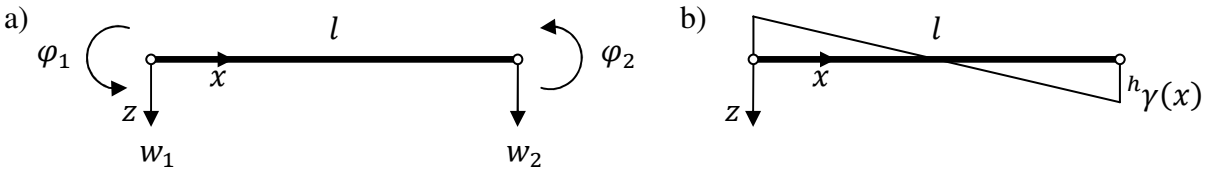


Fig. 3.23 Plane Timoshenko beam element with bending degrees of freedom (a) and distribution of shear strain for pure bending load case (b)

The parasitic shear strain ${}^h\gamma$ and the resulting parasitic shear stress $\tau_{xz}(x) = G\,{}^h\gamma(x)$ can be eliminated by using a quadratic interpolation function

$$ {}^h w(x) = \left(1 - 3\frac{x}{l} + 2\left(\frac{x}{l}\right)^2\right)w_1 + \left(4\frac{x}{l} - 4\left(\frac{x}{l}\right)^2\right)w_2 + \left(-\frac{x}{l} + 2\left(\frac{x}{l}\right)^2\right)w_3 \tag{3.49} $$

for the transverse displacement combined with a linear rotation interpolation (3.44) which results in a quadratic element (Fig. 3.24). For the same boundary conditions as before ($w_1 = w_3 = 0, -\varphi_1 = \varphi_3 = \varphi$) this leads to the shear strain distribution

$$ \begin{aligned} {}^h\gamma(x) &= \frac{\partial\, {}^h w}{\partial x} + {}^h\varphi \\ &= \left(-\frac{3}{l} + 4\frac{x}{l^2}\right)w_1 + \left(\frac{4}{l} - 8\frac{x}{l^2}\right)w_2 + \left(-\frac{1}{l} + 4\frac{x}{l^2}\right)w_3 \\ &\quad + \left(1 - \frac{x}{l}\right)\varphi_1 + \frac{x}{l}\varphi_3 \\ &= \left(\frac{4}{l} - 8\frac{x}{l^2}\right)w_2 - \left(1 - 2\frac{x}{l}\right)\varphi. \end{aligned} \tag{3.50} $$

Obviously, it is now possible to fulfill the condition ${}^h\gamma(x) = 0$ also for $x = 0$ and $x = l$ since

$$ {}^h\gamma(x = 0) = \frac{4}{l}w_2 - \varphi \tag{3.51} $$

and

$$^{h}\gamma(x = l) = -\frac{4}{l}w_2 + \varphi \tag{3.52}$$

by

$$\varphi = \frac{4}{l}w_2, \tag{3.53}$$

which is a nonzero pure bending and shear-free deformation.

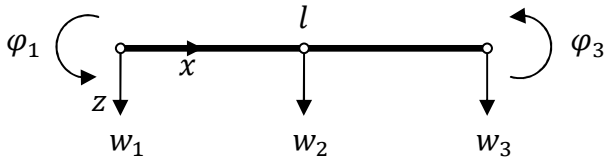


Fig. 3.24 Plane Timoshenko beam element with quadratic interpolation function and bending degrees of freedom

3.2.4 Membrane Locking

Unfortunately, using second order shell or beam elements does not solve all locking problems. Especially for curved structures, membrane locking may appear. The effect is limited to a certain type of elements. Exposed to this phenomenon are beam and shell elements with quadratic interpolation functions. Linear beam elements or linear triangular shell elements are membrane locking free. As described in detail in [133,81,118], the reason for locking is the appearance of parasitic membrane strains and stresses due to the curvature of the structure, although the deformation should be free of these (Fig. 3.25). Affected are shear deformation free element formulations as well as elements including shear deformation, because the parasitic membrane strains do not depend on assumptions about the shear deformation behavior. As visible in Fig. 3.25, the membrane locking effect is much stronger for quadratic element formulations than for linear ones, although the bilinear element is initially distorted to achieve the curvature in the element, which is necessary for the locking. Additionally, the bilinear element is already modified such that transverse shear locking is avoided.

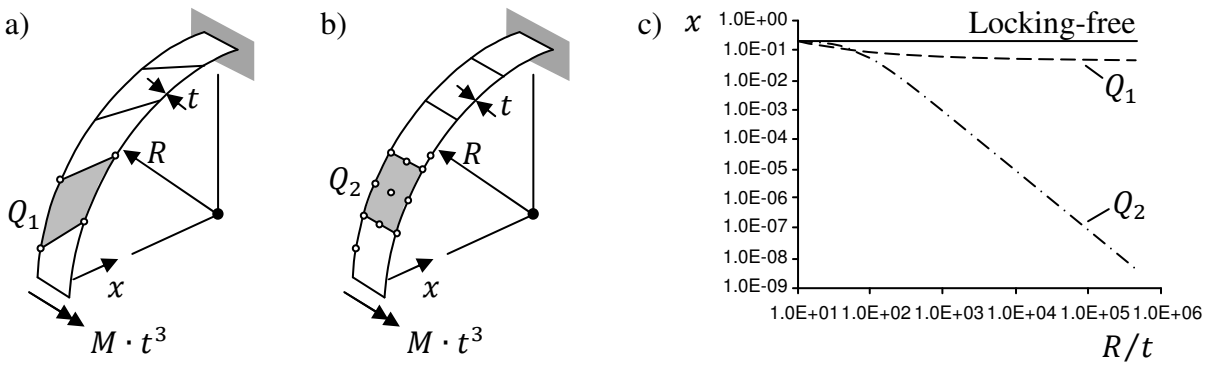


Fig. 3.25 Curved structure meshed with a) linear and b) quadratic shell elements and c) resulting stiffness behavior (from [81])

3.3 Concepts to Avoid Locking

As demonstrated, the different types of locking can produce big errors in terms of displacements, stresses and strains. Thus, it is necessary to develop concepts for avoiding these unwanted effects. Probably the most important of these concepts is the concept of underintegration. As we will see, this strategy is quite simple and leads to very good results in many cases, but sometimes causes another unwanted effect, the so-called "Hourglassing". Further approaches to circumvent locking are the Assumed Natural Strain method (ANS method), the Discrete Strain Gap method (DSG method) and the Enhanced Assumed Strain method (EAS method) which are also briefly discussed.

3.3.1 Numerical Integration

To understand the important concept of underintegration (or reduced integration) we start with a brief introduction to numerical integration, which is very important in the field of FEM to calculate stiffness matrices, mass matrices and element load vectors for example. Often the term quadrature is also used for numerical integration. This has some historical reasons and is based on the fact that ancient Greek mathematicians understood by quadrature the construction of a square having the same area as a figure. In dynamic computations, an additional numerical time-integration is necessary to solve the equations of motion, which is discussed in Chapter 5. Although for many simple problems analytical solutions are available, the numerical integration is often more effective and faster.

The basic idea of many numerical integration procedures is to approximate the integral of a function $f(x)$ in an interval $[a, b]$ by a summation of a number of function values $f(x_k)$ at certain support points x_k multiplied by weighting factors w_k according to

$$\int_a^b f(x)\mathrm{d}x \approx (b-a)\sum_{k=0}^{n} w_k f(x_k). \tag{3.54}$$

In Fig. 3.26a the integral of the function $f(x)$ between b and a can be approximated for instance by $\int_a^b f(x)\mathrm{d}x \approx (b-a)\cdot\frac{1}{2}\big(f(x_0)+f(x_1)\big)$. The accuracy of this result can be improved in two ways. The first possibility is to divide the interval $[a, b]$ into more segments $m > 1$ of width $h = (b-a)/m$ (Fig. 3.26b) and summing up the integration results of all segments. The second possibility is to use more support points x_k within the interval.

With respect to the number and position of the support points within an interval, different integration or quadrature rules can be derived. If the support points x_k are equally distributed across the interval and also the interval limits are used ($x_0 = a$, $x_n = b$), the integration rule belongs to the group of closed Newton-Cotes formulas. Within this group a number of different algorithms, depending on the number of necessary support points, described by the parameter n, exist [30]. Table 3.2 shows some integration rules with the necessary weighting factors w_k and the expected errors for the integration of a polynomial of order n (with $\xi\epsilon[a, b]$).

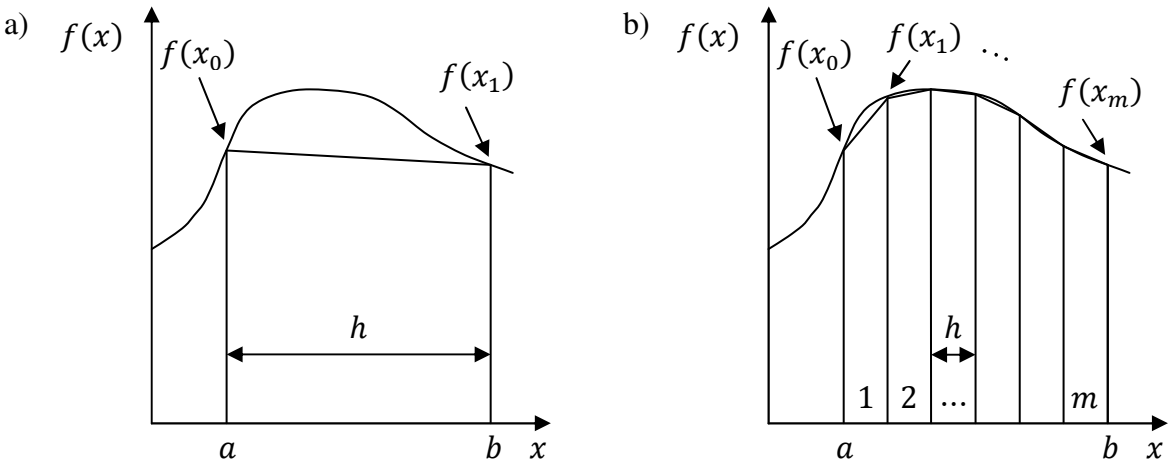


Fig. 3.26 Numerical integration of a function in interval $[a, b]$ divided into one segment (a) and many segments (b)

Table 3.2 Newton-Cotes quadrature rules

n	Formula/weighting factors	Error	Name
1	$(b-a)\dfrac{f(x_0)+f(x_1)}{2}$	$\dfrac{1}{12}h^3 f''(\xi)$	Trapezoid rule
2	$(b-a)\dfrac{f(x_0)+4f(x_1)+f(x_2)}{6}$	$\dfrac{1}{90}h^5 f^{(4)}(\xi)$	Simpson's rule
3	$(b-a)\dfrac{f(x_0)+3f(x_1)+3f(x_2)+f(x_3)}{8}$	$\dfrac{3}{80}h^5 f^{(4)}(\xi)$	Simpson's 3/8 rule
4	$(b-a)\left[\dfrac{7f(x_0)+32f(x_1)+12f(x_2)}{90} + \dfrac{32f(x_3)+7f(x_4)}{90}\right]$	$\dfrac{8}{945}h^7 f^{(6)}(\xi)$	Milne's/Boole's rule
5	$(b-a)\left[\dfrac{19f(x_0)+75f(x_1)+50f(x_2)}{90} + \dfrac{50f(x_3)+75f(x_4)+19f(x_5)}{90}\right]$	$\dfrac{275}{12096}h^7 f^{(6)}(\xi)$	6-point rule
6	$(b-a)\left[\dfrac{41f(x_0)+216f(x_1)+27f(x_2)}{840} + \dfrac{272f(x_3)+27f(x_4)+216f(x_5)+41f(x_6)}{840}\right]$	$\dfrac{9}{1400}h^9 f^{(8)}(\xi)$	Weddle's rule

One can show that the Newton-Cotes formulas are exact for the integration of polynomials of order $n+1$ for even n and of order n for uneven n [30]. For this reason Newton-Cotes formulas with an even number of n (e.g. Simpson rule) are preferable to formulas with uneven n (e.g. 3/8 rule).

To demonstrate the application of the Newton-Cotes formulas we want to numerically integrate a third order polynomial where the analytical solution is known: $\int_0^1 x^3 \mathrm{d}x = 1/4$. By using Simpson's rule for this problem one gets $\int_0^1 x^3 \mathrm{d}x = (1-0) \cdot \frac{1}{6} \cdot (0^3 + 4 \cdot 0.5^3 + 1^3) = 1/4$, which is the exact solution. The integration of a fourth order polynomial, e.g. $\int_0^1 x^4 \mathrm{d}x = 1/5 = 0.2$, by Simpson's rule gives $\int_0^1 x^4 \mathrm{d}x \approx (1-0) \cdot \frac{1}{6} \cdot (0^4 + 4 \cdot 0.5^4 + 1^4) = 5/24 \approx 0.208$, which is not the exact solution anymore. To reduce the error, an integration rule using more support points has to be used (e.g. Milne's/Boole's rule) or the interval $[a, b]$ has to be divided into segments, which are separately integrated by Simpson's rule. The first strategy leads e.g. by applying Milne's/Boole's rule to the exact result, the latter for two segments to $\int_0^1 x^4 \mathrm{d}x = \int_0^{0.5} x^4 \mathrm{d}x + \int_{0.5}^1 x^4 \mathrm{d}x \approx (0.5) \cdot \frac{1}{6} \cdot (0^4 + 4 \cdot 0.25^4 + 0.5^4) + (0.5) \cdot \frac{1}{6} \cdot (0.5^4 + 4 \cdot 0.75^4 + 1^4) \approx 0.201$.

Another improvement of the integration accuracy is possible by not using equally distributed support points but particular ones. This leads to the idea of Gauss integration, where support points are determined such that the error of the algorithm is minimized. By this, the integration rule has more "degrees of freedom" and leads to more accurate results. The integration Equation (3.54) is still the same, but the points x_k of the function evaluation and the weighting factors w_k are different (Fig. 3.27).

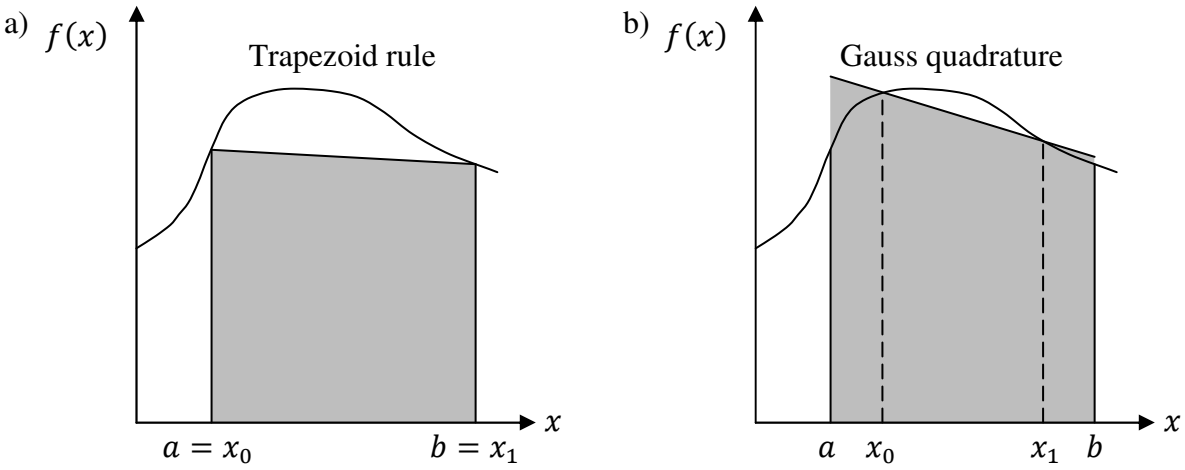


Fig. 3.27 Numerical integration of a function in interval $[a, b]$ with trapezoid rule (a) and two-point Gauss quadrature (b)

The weighting factors and positions of function evaluations can be determined analytically from the exact solutions of the integration of polynomials of order $n+2$ and lower [30]. Table 3.3 lists the necessary weighting factors and support point positions, i.e. Gauss points, for different numbers of evaluation points $n+1$ within the interval $[-1,1]$.

The Gauss quadrature is exact for polynomials up to an order of $2n+1$. It should be noted that the values in Table 3.3 apply to the interval $[-1,1]$. To use these data for an arbitrary

interval $[a, b]$, a transformation of the Gauss points according to

$$\int_a^b f(x)\mathrm{d}x \approx \frac{b-a}{2}\sum_{k=0}^{n} w_k f\left(\frac{b-a}{2}x_k + \frac{a+b}{2}\right) \tag{3.55}$$

is necessary.

Table 3.3 Position of Gauss points and weighting factors

n	Support points x_k	Weighting factors w_k
0	$x_0 = 0$	$w_0 = 2$
1	$x_1 = -x_0 = \frac{1}{\sqrt{3}}$	$w_0 = w_1 = 1$
2	$x_2 = -x_0 = \sqrt{\frac{3}{5}},\ \ x_1 = 0$	$w_0 = w_2 = \frac{5}{9},\ \ w_1 = \frac{8}{9}$
3	$x_3 = -x_0 = 0.8611363116$ $x_2 = -x_1 = 0.3399810436$	$w_0 = w_3 = 0.3478548451$ $w_1 = w_2 = 0.6521451549$
4	$x_4 = -x_0 = 0.9061798459$ $x_3 = -x_1 = 0.5384693102$ $x_2 = 0$	$w_0 = w_4 = 0.2369268851$ $w_1 = w_3 = 0.4786286705$ $w_2 = 0.5688888889$

For demonstration of the Gauss quadrature, the same examples as for the Newton-Cotes integration are used. We start with the numerical integration of the simple cubic function. For $n = 1$ we get

$$\int_0^1 x^3\mathrm{d}x \approx \frac{1}{2}\left[\left(\frac{1}{2}\cdot\frac{-1}{\sqrt{3}} + \frac{1}{2}\right)^3 + \left(\frac{1}{2}\cdot\frac{1}{\sqrt{3}} + \frac{1}{2}\right)^3\right] = \frac{1}{4} \tag{3.56}$$

which is the exact solution obtained by evaluating the function $f(x)$ at only two points instead of three as necessary for Simpson's rule. For integrating the fourth-order polynomial exactly, three Gauss points are necessary which yields

$$\int_0^1 x^4\mathrm{d}x \approx \frac{1}{2}\left[\frac{5}{9}\cdot\left(-\frac{1}{2}\cdot\sqrt{\frac{3}{5}} + \frac{1}{2}\right)^4 + \frac{8}{9}\cdot\left(\frac{1}{2}\right)^4 + \frac{5}{9}\cdot\left(\frac{1}{2}\cdot\sqrt{\frac{3}{5}} + \frac{1}{2}\right)^4\right] = \frac{1}{5}. \tag{3.57}$$

To achieve the same result with the help of Newton-Cotes integration (Milne's/Boole's rule), five function evaluations are required.

Due to the higher accuracy of Gauss quadrature this integration scheme is preferred in FE systems for the computation of stiffness matrices, load vectors and so on. The necessary order of the integration depends on the used interpolation function of the integrand. If the used

number of Gauss points leads to an exact integration the procedure is called “full integration”. But strictly speaking, this full integration is only exact for perfectly shaped undeformed finite elements. The stronger the element deformation, the more inaccurate becomes the numerical integration, which is why a good element quality is important for an accurate result of FE computations.

3.3.2 Reduced Integration

As mentioned briefly, many locking phenomena can be avoided by the “trick” of underintegration, which means to use a lower integration order (less Gauss points) than actually necessary for an exact integration. For example, a plane bilinear solid element is not integrated with four but only with one Gauss point. This Gauss point is placed according to Table 3.3 in the middle of the element. Thus, an exact integration of the respective matrices is not possible anymore, which leads to small errors in the computation of the element’s stiffness matrix, and following from this the elimination of higher-order terms in the computed strain distribution. By this, for the bilinear element only constant strain distributions can be computed. As we have seen, the parasitic shear stresses in case of shear locking have a linear distribution across the element with a zero in the elements midpoint (Section 3.1.1). In the reduced integration process, the strains are now evaluated at this point and the computed shear stress ε_{xy} will be zero for a pure bending load which is correct. Unfortunately, also the bending strain ε_x will be evaluated at the same point and becomes zero too, which is not correct. This means that the strain energy in the element is zero and the applied loads lead to arbitrarily high displacements due to the principle of virtual work which postulates equality of internal and external work. As a consequence, at least two underintegrated solid elements have to be used in bending direction (thickness direction) to avoid infinitely high displacements (as done in Fig. 3.2). On the other hand, there are two important advantages in the concept of underintegration. The first one is that the computed displacements are not influenced by the shear locking phenomenon anymore, and the second one is an increased computational speed due to the reduced number of necessary function evaluations in the integration process. The latter one is especially important for nonlinear computations where the elements stiffness matrix is recalculated after every iteration or time-step.

For demonstration this principle is applied to the example of a linear Timoshenko beam element with length l. Starting from the principle of virtual work [11], the weak form of the virtual inner work for a beam with Timoshenko kinematics [27] is

$$\delta W^{int} = \int_0^l GA_S(x)[w' + \varphi]\delta[w' + \varphi]\mathrm{d}x + \int_0^l EI(x)\varphi'\delta\varphi'\mathrm{d}x. \tag{3.58}$$

On the finite element basis this becomes

$$\delta^h W^{int} = \int_0^l \delta[{}^h w' + {}^h\varphi]GA_S(x)[{}^h w' + {}^h\varphi]\mathrm{d}x + \int_0^l \delta^h\varphi' EI(x){}^h\varphi'\mathrm{d}x \tag{3.59}$$

where all functions with index h are interpolated functions in the element. By using the linear interpolation functions (3.43) and (3.44) between displacements w_i and rotations φ_i at the two element nodes we introduce the strain-displacement matrix $\boldsymbol{B}_S$ for shear such that

$$\boldsymbol{B}_S\boldsymbol{x} = \boldsymbol{B}_S\begin{bmatrix} w_1 \\ \varphi_1 \\ w_2 \\ \varphi_2 \end{bmatrix} = {}^h w' + {}^h\varphi = -\frac{1}{l}w_1 + \left(1 - \frac{x}{l}\right)\varphi_1 + \frac{1}{l}w_2 + \frac{x}{l}\varphi_2 \tag{3.60}$$

which yields by comparison

$$\boldsymbol{B}_S = \begin{bmatrix} -\frac{1}{l} & 1 - \frac{x}{l} & \frac{1}{l} & \frac{x}{l} \end{bmatrix}. \tag{3.61}$$

For the bending term in Equation (3.59) we get

$$\boldsymbol{B}_B\boldsymbol{x} = \boldsymbol{B}_B\begin{bmatrix} w_1 \\ \varphi_1 \\ w_2 \\ \varphi_2 \end{bmatrix} = {}^h\varphi' = -\frac{1}{l}\varphi_1 + \frac{1}{l}\varphi_2 \tag{3.62}$$

which gives

$$\boldsymbol{B}_B = \begin{bmatrix} 0 & -\frac{1}{l} & 0 & \frac{1}{l} \end{bmatrix}. \tag{3.63}$$

Now we are able to write the inner virtual work of the element as

$$\delta^h W^{int} = \delta\boldsymbol{x}^T \int_0^l \boldsymbol{B}_S^T G A_S(x) \boldsymbol{B}_S \mathrm{d}x \, \boldsymbol{x} + \delta\boldsymbol{x}^T \int_0^l \boldsymbol{B}_B^T EI(x) \boldsymbol{B}_B \mathrm{d}x \, \boldsymbol{x}, \tag{3.64}$$

which can be reformulated by introducing elasticity matrices (1x1 matrices for homogenous, isotropic material)

$$\boldsymbol{E}_S = [GA_S(x)], \tag{3.65}$$

$$\boldsymbol{E}_B = [EI(x)] \tag{3.66}$$

as

$$\delta^h W^{int} = \delta\boldsymbol{x}^T \int_0^l \boldsymbol{B}_S^T \boldsymbol{E}_S \boldsymbol{B}_S \mathrm{d}x \, \boldsymbol{x} + \delta\boldsymbol{x}^T \int_0^l \boldsymbol{B}_B^T \boldsymbol{E}_B \boldsymbol{B}_B \mathrm{d}x \, \boldsymbol{x} = \delta\boldsymbol{x}^T \boldsymbol{K}_e \boldsymbol{x} \tag{3.67}$$

where

$$\boldsymbol{K}_e = \boldsymbol{K}_S + \boldsymbol{K}_B = \int_0^l \boldsymbol{B}_S^T \boldsymbol{E}_S \boldsymbol{B}_S \mathrm{d}x + \int_0^l \boldsymbol{B}_B^T \boldsymbol{E}_B \boldsymbol{B}_B \mathrm{d}x \tag{3.68}$$

is the element stiffness matrix. For reasons of simplicity we assume that the elasticity matrices $\boldsymbol{E}_S$ and $\boldsymbol{E}_B$ are constant (cross section does not change along the beam element in longitudinal direction). The bending part of the strain-displacement matrix $\boldsymbol{B}_B$ contains constant derivatives of the linear interpolation functions, which means that just one Gauss point is necessary for exact integration. The shear part of the strain-displacement matrix $\boldsymbol{B}_S$ includes linear functions which leads to quadratic integrals due to the multiplication in

Equation (3.68). Thus, for an exact integration at least two Gauss points are needed. For the simple integration of the bending part one gets

$$\boldsymbol{K}_B = \int_0^l \boldsymbol{B}_B^T \boldsymbol{E}_B \boldsymbol{B}_B \mathrm{d}x = EI \int_0^l \boldsymbol{B}_B^T \boldsymbol{B}_B \mathrm{d}x = EI \boldsymbol{B}_B^T \boldsymbol{B}_B l = \frac{EI}{l} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix}. \quad (3.69)$$

The integration of the shear part is a bit more complex and starts with

$$\begin{aligned} \boldsymbol{K}_S &= \int_0^l \boldsymbol{B}_S^T \boldsymbol{E}_S \boldsymbol{B}_S \mathrm{d}x = GA_S \int_0^l \boldsymbol{B}_S^T \boldsymbol{B}_S \mathrm{d}x \\ &= GA_S \int_0^l \begin{bmatrix} \frac{1}{l^2} & -\frac{1}{l} + \frac{x}{l^2} & -\frac{1}{l^2} & -\frac{x}{l^2} \\ -\frac{1}{l} + \frac{x}{l^2} & \left(1 - \frac{x}{l}\right)^2 & \frac{1}{l} - \frac{x}{l^2} & \frac{x}{l} - \frac{x^2}{l^2} \\ -\frac{1}{l^2} & \frac{1}{l} - \frac{x}{l^2} & \frac{1}{l^2} & \frac{x}{l^2} \\ -\frac{x}{l^2} & \frac{x}{l} - \frac{x^2}{l^2} & \frac{x}{l^2} & \frac{x^2}{l^2} \end{bmatrix} \mathrm{d}x. \end{aligned} \quad (3.70)$$

In the following we want to perform the Gauss integration exemplarily for the coordinate (2,4). According to Equation (3.55) and Table 3.3 we get

$$\begin{aligned} K_{S_{24}} &= GA_S \int_0^l \left(\frac{x}{l} - \frac{x^2}{l^2}\right) \mathrm{d}x = \frac{GA_S}{l} \int_0^l \left(x - \frac{x^2}{l}\right) \mathrm{d}x = \frac{GA_S}{l} \int_0^l f(x) \mathrm{d}x \\ &= \frac{GA_S}{l} \left[\frac{l}{2} \left(\underbrace{1}_{w_0} \cdot f\left(\underbrace{\frac{l}{2} \cdot \frac{-1}{\sqrt{3}} + \frac{l}{2}}_{Gauss\ point\ 0} \right) + \underbrace{1}_{w_1} \cdot f\left(\underbrace{\frac{l}{2} \cdot \frac{1}{\sqrt{3}} + \frac{l}{2}}_{Gauss\ point\ 1} \right) \right) \right] \\ &= \frac{GA_S}{2} \left[\frac{l}{2} \cdot \frac{-1}{\sqrt{3}} + \frac{l}{2} - \frac{\left(\frac{l}{2} \cdot \frac{-1}{\sqrt{3}} + \frac{l}{2}\right)^2}{l} + \frac{l}{2} \cdot \frac{1}{\sqrt{3}} + \frac{l}{2} - \frac{\left(\frac{l}{2} \cdot \frac{1}{\sqrt{3}} + \frac{l}{2}\right)^2}{l} \right] \\ &= \frac{GA_S}{6} l. \end{aligned} \quad (3.71)$$

With the same procedure the other coordinates are integrated which finally yields the stiffness matrix for the shear part

$$\boldsymbol{K}_S = \frac{GA_S}{l} \begin{bmatrix} 1 & -\frac{l}{2} & -1 & -\frac{l}{2} \\ -\frac{l}{2} & \frac{l^2}{3} & \frac{l}{2} & \frac{l^2}{6} \\ -1 & \frac{l}{2} & 1 & \frac{l}{2} \\ -\frac{l}{2} & \frac{l^2}{6} & \frac{l}{2} & \frac{l^2}{3} \end{bmatrix} \quad (3.72)$$

by “full” integration.

Now we want to apply the reduced integration for the same matrix, which means that we use only a single Gauss point. Thus, we get exemplarily for coordinate (2,4)

$$
\begin{aligned}
K_{S_{24}}^{red} &= GA_S \int_0^l \left(\frac{x}{l} - \frac{x^2}{l^2}\right) \mathrm{d}x = \frac{GA_S}{l} \int_0^l \left(x - \frac{x^2}{l}\right) \mathrm{d}x = \frac{GA_S}{l} \int_0^l f(x) \mathrm{d}x \\
&= \frac{GA_S}{l} \left[\frac{l}{2} \cdot \underbrace{2}_{w_0} \cdot f\left(\underbrace{\frac{l}{2} \cdot 0 + \frac{l}{2}}_{Gauss\ point\ 0} \right) \right] \\
&= \frac{GA_S}{2} \left[2 \cdot \left(\frac{l}{2} - \frac{\left(\frac{l}{2}\right)^2}{l} \right) \right] \\
&= \frac{GA_S}{4} l
\end{aligned}
\tag{3.73}
$$

and finally for the whole matrix

$$
\boldsymbol{K}_S^{red} = \frac{GA_S}{l} \begin{bmatrix}
1 & -\frac{l}{2} & -1 & -\frac{l}{2} \\
-\frac{l}{2} & \boldsymbol{\frac{l^2}{4}} & \frac{l}{2} & \boldsymbol{\frac{l^2}{4}} \\
-1 & \frac{l}{2} & 1 & \frac{l}{2} \\
-\frac{l}{2} & \boldsymbol{\frac{l^2}{4}} & \frac{l}{2} & \boldsymbol{\frac{l^2}{4}}
\end{bmatrix}. \tag{3.74}
$$

By comparing (3.74) with (3.72) one can see that some of the integration results differ (boldfaced in Equation (3.74)), which is due to the integration error of the reduced integration. But this error is accepted because the much bigger problem of the shear locking is avoided on the other hand.

In the presented example the shear part of the stiffness matrix was reduced integrated and the bending part fully integrated. This approach is called “selectively reduced integration”. For other element formulations it is also possible to integrate both parts of the stiffness matrix reduced. A further closer look to the stiffness matrix $\boldsymbol{K}_S^{red}$ in Equation (3.74) reveals that the matrix is singular since the second and last line are identical. Such a rank deficient stiffness matrix is very typical for reduced integration. In our case of the Timoshenko beam this is not a problem because the overall stiffness matrix of the element $\boldsymbol{K}_e$ is computed as the sum of $\boldsymbol{K}_B$ and $\boldsymbol{K}_S^{red}$ and has therefore full rank. But in general, also the overall stiffness matrix may be rank deficient which leads to so-called zero-energy modes. These are displacements of the element not causing any strains in the element although rigid body motions are absent. This phenomenon is discussed in detail in Section 3.3.6.

3.3.3 Assumed Natural Strain Method

The idea of the assumed natural strain method (ANS method) is to evaluate strains in a finite element not at the integration points of the element, but at so-called collocation points. These are points where the computed strains and stresses are correct and contain no parasitic portions. For the plate bending element in Fig. 3.28 the parasitic shear strain should be evaluated somewhere on the line $\overline{AC}$. This is practically done by constructing the strain-displacement matrix $\boldsymbol{B}_e$ of the element with the help of displacements interpolated for example at points A, B, C and D and not directly with nodal displacement values. This leads to a strain field based on the collocation points and a more complex $\boldsymbol{B}_e$ matrix, but the full Gauss integration can be used for computation of the stiffness matrix.

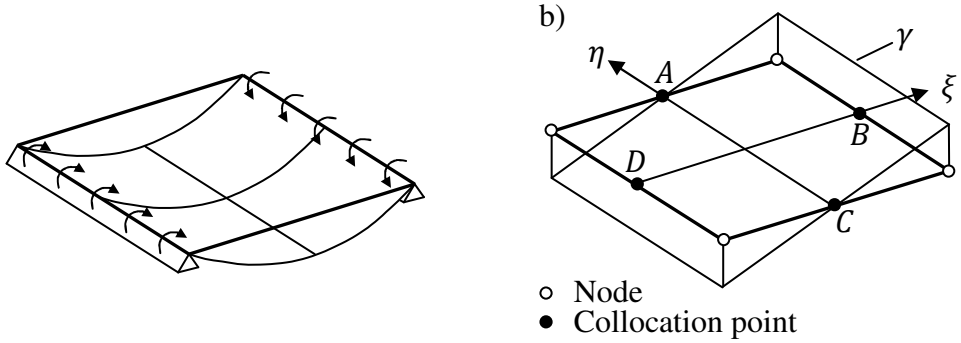


Fig. 3.28 Plate element subjected to pure bending (a) and parasitic shear strain γ with collocation points (b), from [81]

The choice of the collocation points is done with the help of suitable example problems and has to be performed separately for each element type, interpolation function order and strain portion that should be modified. Examples for ANS elements are the MITC elements described in [15,14]. A disadvantage of the ANS method is that volumetric locking cannot be avoided.

3.3.4 Discrete Strain Gap Method

The discrete strain gap method (DSG method) has some similarities to the ANS method. The method is applicable to all finite element types, but here it will be explained using the example of the Timoshenko beam. The first step is the identification of the shear deflection fraction in comparison to the overall deflection, which can be done by calculating the difference between overall and bending displacements according to

$$w_\gamma = w - w_\varphi. \tag{3.75}$$

The fraction of the pure bending displacement w_φ can be determined from the Bernoulli condition $\partial w/\partial x = -\varphi$ as

$$w_\varphi(x) = -\int \varphi(x)\mathrm{d}x. \tag{3.76}$$

The value of the integration constant does not play any role in this context because it has no contribution to the strains. Since the interpolation function for $\varphi(x)$ is linear, a quadratic

function results from Equation (3.76). Summarizing, we obtain for the so-called "shear gap"

$$w_\gamma = w - w_\varphi = w(x) + \int \varphi(x)\mathrm{d}x = \int_0^x \left[\frac{\partial w(x)}{\partial x} + \varphi(x)\right]\mathrm{d}x = \int_0^x \gamma(x)\mathrm{d}x \tag{3.77}$$

describing the difference between overall and bending displacements (see Fig. 3.29). Evaluation of the integral at the nodal positions by using the interpolation functions (3.43) and (3.44) yields

$$w_\gamma = \int_0^x \left[\frac{\partial {}^h w(x)}{\partial x} + {}^h\varphi(x)\right]\mathrm{d}x = -\frac{x}{l}w_1 + \frac{x}{l}w_2 + \left(x - \frac{x^2}{2l}\right)\varphi_1 + \frac{x^2}{2l}\varphi_2 \tag{3.78}$$

and for the nodal positions we get

$$w_\gamma(x=0) = \underbrace{0+0}_{{}^w w_\gamma(x=0)} + \underbrace{0+0}_{{}^\varphi w_\gamma(x=0)}, \tag{3.79}$$

$$w_\gamma(x=l) = \underbrace{w_2 - w_1}_{{}^w w_\gamma(x=l)} + \underbrace{\frac{l}{2}(\varphi_1 + \varphi_2)}_{{}^\varphi w_\gamma(x=l)}. \tag{3.80}$$

These discrete shear gaps have correct values at the element nodes and contain no parasitic portions, which is why they may be used for the computation of the modified shear stress. The interpolation between the nodes is done with the same functions as for the displacements, which are summarized in the matrix of shape functions

$$\boldsymbol{N} = \left[1 - \frac{x}{l} \quad 1 - \frac{x}{l} \quad \frac{x}{l} \quad \frac{x}{l}\right]. \tag{3.81}$$

The shear strain distribution can then be calculated by multiplying the derivative of the matrix of shape functions with respect to x by the vector of nodal displacements consisting of the shear gap values resulting from the overall displacements ${}^w w_\gamma$ and the rotations ${}^\varphi w_\gamma$ as

$$\begin{aligned}\gamma^{DSG}(x) &= \frac{\partial \boldsymbol{N}}{\partial x} \cdot \begin{bmatrix} {}^w w_\gamma(x=0) \\ {}^\varphi w_\gamma(x=0) \\ {}^w w_\gamma(x=l) \\ {}^\varphi w_\gamma(x=l) \end{bmatrix} = \left[-\frac{1}{l} \quad -\frac{1}{l} \quad \frac{1}{l} \quad \frac{1}{l}\right] \begin{bmatrix} 0 \\ 0 \\ w_2 - w_1 \\ \frac{l}{2}(\varphi_1 + \varphi_2) \end{bmatrix} \\ &= \frac{1}{l}(w_2 - w_1) + \frac{1}{2}(\varphi_1 + \varphi_2)\end{aligned} \tag{3.82}$$

which gives for pure bending with $w_1 = w_2 = 0$ and $\varphi_1 = -\varphi_2$ the correct shear distribution $\gamma^{DSG}(x) = 0$. The strain-displacement matrix $\boldsymbol{B}^{DSG}$ can directly be derived from (3.82).

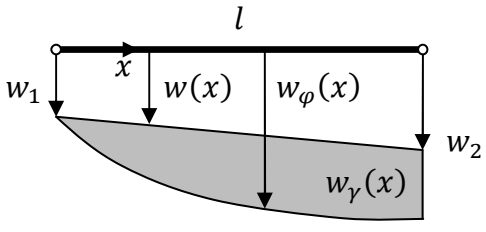


Fig. 3.29 Shear gap in beam element

An extension of the DSG method for continuum elements and other locking phenomena is possible by performing the following steps:

1. Integration of the kinematic equation and evaluation of this integral at the nodal positions of the element;
2. Interpolation of the computed discrete strain gaps to a field without parasitic portions;
3. Differentiation of the modified strain gap field to obtain the modified strain distribution and derivation of the $\boldsymbol{B}$ matrix.

As an advantage to the ANS method no collocation points have to be determined but on the other hand only geometric locking effects can be suppressed by the DSG method. Further details and explanations about this approach are given in [24,81,82].

3.3.5 Enhanced Assumed Strain Method

As described in [81], the basic idea of the enhanced assumed strain method (EAS method) is the extension of the displacement-based strains with the goal to balance the parasitic portions [128]. This is a contrast to the ANS and DSG method, where the parasitic strains are tried to be eliminated. The additional degrees of freedom resulting from the extension of strains appear only on the elemental basis and are eliminated for the overall stiffness matrix by static condensation.

A special case and famous example of EAS elements are the so-called incompatible mode elements [146,135]. The additional quadratic shape functions in these elements, used for an improved strain interpolation, need not be compatible to neighboring elements.

With the EAS method it is possible to suppress material-caused locking effects (volumetric locking, Poisson locking) as well as geometric locking effects like shear locking or membrane locking. A disadvantage is that this method is not very efficient for distorted elements and the computational effort is higher than for the other methods presented above.

3.3.6 Problems of Reduced Integration and the Hourglassing Phenomenon

As described in Section 3.3.2 the concept of underintegration is widespread within the finite element world due to its efficiency, simplicity and low computational costs. But the rank-deficient stiffness matrix typically resulting from this scheme can lead to so-called “Hourglass modes”. This name comes from the hourglass-like shape a group of elements is forming due to this effect. An example is given in Fig. 3.30 where a cantilever beam is meshed with linear

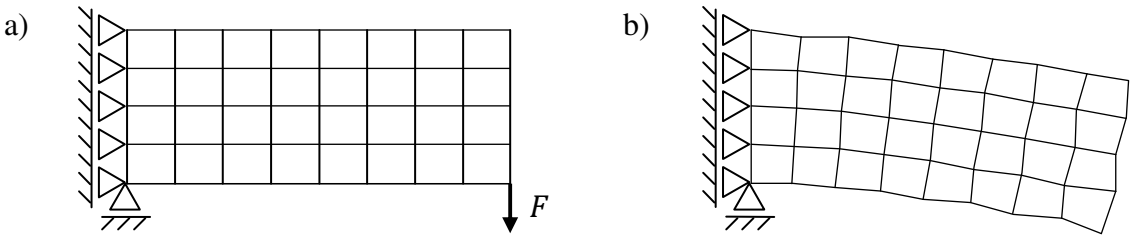


Fig. 3.30 Cantilever beam with single load (a) and hourglassing phenomenon (b)

underintegrated shell elements and subjected to a single force F. The implicit FEM computation shows typical hourglassing.

The explanation for this unwanted behavior is the concept of underintegration. Due to the rank-deficient stiffness matrix and the single integration point in the middle of a linear element the computed strains are constant for the whole element and equal zero for certain deformation states. These deformation modes are called zero-energy modes (also spurious modes or hourglass modes), because no strain energy is generated although a deformation appears. Examples of zero-energy modes, which depend on the element type, are given in Fig. 3.31 for 2D and 3D elements. Each of the 3D modes may appear three times.

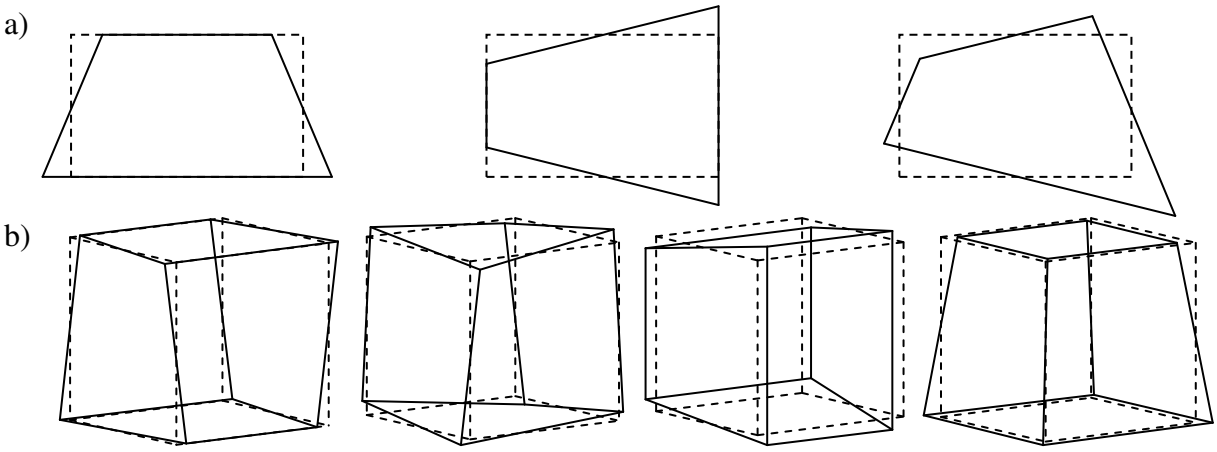


Fig. 3.31 Zero-energy modes of a linear 2D (a) and linear 3D solid element (b)

The effect of zero-energy modes can also be observed in an eigenvalue analysis and does also appear for higher order elements. Fig. 3.32a shows the first eigenmode, which is a zero-energy mode, of a single solid element with quadratic shape functions and reduced integration (material = steel). The first eigenmode with a non-zero strain energy is the 13th one. For the same cube meshed with 8 elements, the zero-energy modes do not appear because with this finer mesh and the resulting element connectivity such modes are not possible anymore (Fig. 3.32b). Thus, a first option to reduce hourglassing is to increase the mesh density. But this option is not always applicable in practical applications and does not solve all hourglassing problems. For this reason other strategies have to be developed.

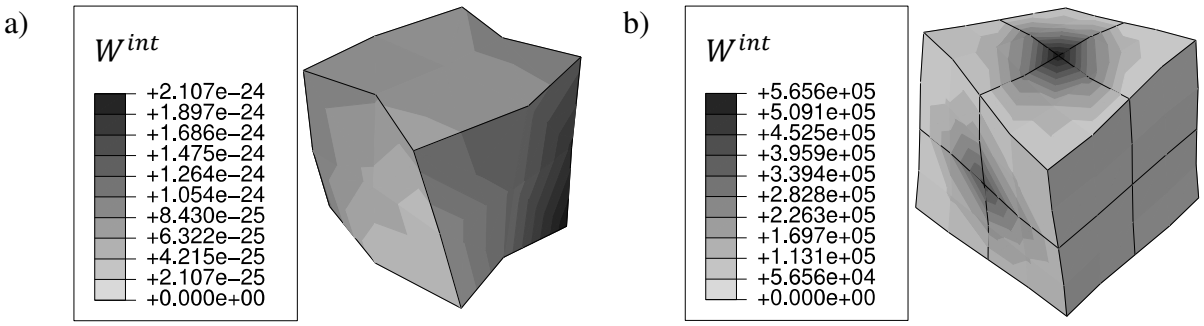


Fig. 3.32 First eigenmode and associated strain energy of a steel cube meshed with one (a) and eight (b) quadratic solid element(s) with reduced integration

The number of zero-energy modes an underintegrated element can evolve depends on the element's dimensionality, type and the used shape function as illustrated in Table 3.4. The disadvantage of Lagrange elements in comparison to Serendipity elements is the higher number of degrees of freedom due to the extra nodes as exemplarily shown for a quadratic 3D solid element in Fig. 3.33. For this reason most commercial FE codes are using Serendipity elements. The interpolation functions of both element types are constructed slightly different but are compatible at the element edges, which means that even a mix of both element types is possible within a mesh.

Table 3.4 Number of zero-energy modes for different element formulations (values in brackets remain to be established), from [102]

Dimension:	2	2	3	3
Element type:	Serendipity	Lagrange	Serendipity	Lagrange
Polynomial degree	Minimum number of zero-energy modes			
1	2	2	12	12
2	1	3	6	27
3	(6)	2	(72)	24
4	(19)	(1)	(258)	(15)

a) b)

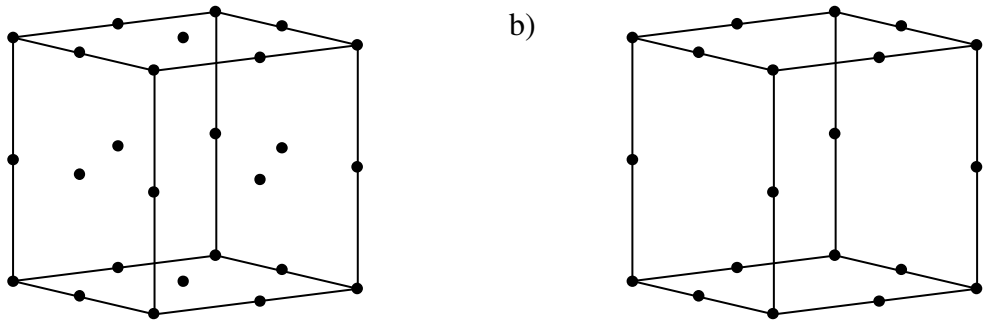
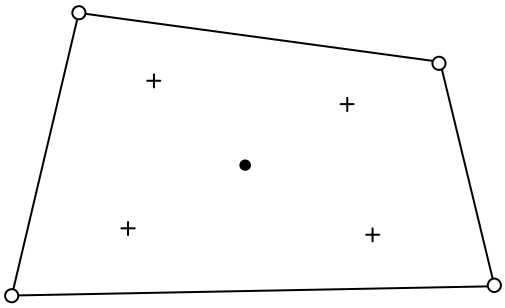


Fig. 3.33 3D quadratic Lagrange element (a) and quadratic Serendipity element (b)

3.3.7 Concepts to Avoid Hourglassing

The evaluation of all strain coordinates at just one integration point in the middle of an element can lead to zero-energy modes and the hourglassing problem described in the previous section. For this reason it is necessary to use certain strategies which yield to both hourglassing free and locking free finite elements. The first strategy discussed here is the concept of selectively underintegrated elements. Within this concept only particular strain components (e.g. shear strains) are evaluated at the Gauss point in the middle of the element as demonstrated for the Timoshenko beam in Section 3.3.2. All other strain coordinates are evaluated at the Gauss points used for the full integration. Fig. 3.34 shows a selectively integrated bilinear rectangular element for the plane strain or plane stress state. Since the shear and normal strains are not coupled, the stiffness matrices can be computed separately

and summed up afterwards. By this procedure, shear locking and a rank-deficient stiffness matrix are avoided. For plane strain elements the volumetric locking can be suppressed by evaluating the hydrostatic strain fractions in the middle of the element.

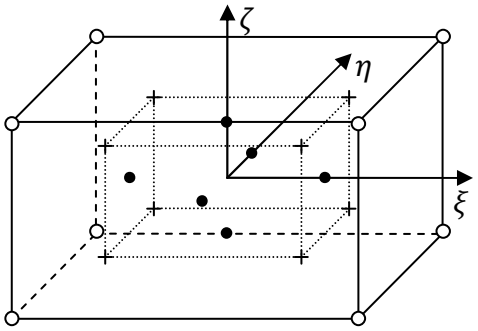


- Evaluation of γ_{xy} at $\xi = \eta = 0$

\+ Evaluation of ε_x and ε_y at $\xi, \eta = \pm 1/\sqrt{3}$

Fig. 3.34 Selectively underintegrated element

For 8-node brick elements the selective underintegration has to be modified. A single Gauss point in the middle of the element for the computation of shear strains would still lead to torsional zero-energy modes. For this reason the shear strain coordinates are evaluated at the projections of the element's midpoint onto the surfaces of the hexahedron defined by the eight Gauss points $\xi, \eta, \zeta = \pm 1/\sqrt{3}$ where the normal strains are evaluated (see Fig. 3.32).



- Evaluation of γ_{xy} at $(\xi, \eta, \zeta) = \left(0,0, \pm 1/\sqrt{3}\right)$
  γ_{xz} at $(\xi, \eta, \zeta) = \left(0, \pm 1/\sqrt{3}, 0\right)$
  γ_{yz} at $(\xi, \eta, \zeta) = \left(\pm 1/\sqrt{3}, 0,0\right)$

\+ Evaluation of ε_x, ε_y and ε_z at $\xi, \eta, \zeta = \pm 1/\sqrt{3}$

Fig. 3.35 Selectively underintegrated 8-noded brick element

Due to the additional integration points for the evaluation of shear strains the computational time for the computation of the stiffness matrix of selectively underintegrated elements is higher than for underintegrated elements, which is in particular important for non-linear computations where the stiffness-matrix is recomputed frequently. Another disadvantage of these elements is that not all of them pass the patch test [72,102,11] and trapezoidal locking is not prevented. Moreover, it can be shown that strains, computed at "reduced" Gauss points, converge faster than those computed at normal Gauss points [134,154]. The reason for this unexpected behavior is that the error of averaged strains converges faster to zero than the error at separate Gauss points. These optimal points for the strain and stress evaluation are sometimes called Barlow points [8,7].

The second strategy that should be discussed here, as a remedy for hourglassing, is the stabilization of the zero-energy modes. Typically, the stiffness matrix resulting from reduced integration is singular with respect to the zero-energy modes (shown for example in Fig. 3.31). This means, the product of the stiffness matrix and the displacement vector of an

hourglass mode equals the zero vector. In [83] it is suggested to add to the rank-deficient stiffness matrix of an underintegrated element an hourglass matrix that is singular with respect to rigid body modes and constant strains, but not singular with respect to zero-energy modes. The hourglass matrix is obtained by a kind of backward computation from the forces and displacements producing the hourglass modes. However, this technique is very expensive for non-rectilinear elements [83,50]. The foundations for a widely used technique, the perturbation hourglass stabilization, can be found in [50], where a mean strain approach is used to calculate generalized strains $\widetilde{\boldsymbol{\gamma}}$ which are added to the $\boldsymbol{B}$ matrix (strain-displacement matrix) of an element, which yields for a 4-node bilinear plane element

$$\widetilde{\boldsymbol{B}} = \begin{bmatrix} \dfrac{\partial \boldsymbol{N}}{\partial x} & \boldsymbol{0} \\ \boldsymbol{0} & \dfrac{\partial \boldsymbol{N}}{\partial y} \\ \dfrac{\partial \boldsymbol{N}}{\partial x} & \dfrac{\partial \boldsymbol{N}}{\partial y} \\ \widetilde{\boldsymbol{\gamma}}^T & \boldsymbol{0} \\ \boldsymbol{0} & \widetilde{\boldsymbol{\gamma}}^T \end{bmatrix} \tag{3.83}$$

and should restore the correct rank of the stiffness matrix, which is five for the mentioned element type (2 DOF per node – 3 rigid body modes). The $\widetilde{\boldsymbol{\gamma}}$ vector has to fulfill certain orthogonality conditions as described in [50]. Additionally, the elasticity matrix has to be modified according to

$$\widetilde{\boldsymbol{E}} = \begin{bmatrix} E_{11} & E_{12} & E_{13} & 0 & 0 \\ E_{21} & E_{22} & E_{23} & 0 & 0 \\ E_{31} & E_{32} & E_{33} & 0 & 0 \\ 0 & 0 & 0 & {}^{Q}E & 0 \\ 0 & 0 & 0 & 0 & {}^{Q}E \end{bmatrix} \tag{3.84}$$

where ${}^{Q}E$ is a fictive material parameter relating the generalized hourglass strains to the generalized hourglass stresses. A suggestion for the choice of ${}^{Q}E$ can be found in [20]. The internal force vector $\boldsymbol{f}^{int}$ is computed as

$$\boldsymbol{f}^{int} = A\widetilde{\boldsymbol{B}}\widetilde{\boldsymbol{\sigma}} = A\boldsymbol{B}^T\boldsymbol{\sigma} + A\begin{bmatrix} Q_x\widetilde{\boldsymbol{\gamma}} \\ Q_y\widetilde{\boldsymbol{\gamma}} \end{bmatrix} \tag{3.85}$$

with A as area of the element and $\widetilde{\boldsymbol{\sigma}} = [\sigma_x \quad \sigma_y \quad \tau_{xy} \quad Q_x \quad Q_y]^T$ as stress tensor in Voigt notation and can be separated into the vector of internal forces resulting from reduced integration and a stabilization force vector

$$\boldsymbol{f}^{int}_{stab} = A\begin{bmatrix} Q_x\widetilde{\boldsymbol{\gamma}} \\ Q_y\widetilde{\boldsymbol{\gamma}} \end{bmatrix}. \tag{3.86}$$

The stabilization may be of stiffness or viscous type depending on whether the stabilization forces are proportional to nodal displacements or velocities. Also a combination of both types of stabilization can be very effective [42].

Another possibility for hourglass stabilization is physical stabilization. In this approach a vector of stabilization forces, which can be determined analytically from an assumed strain field a priori, is added to the internal force vector [20,19].

3.4 Three-Point-Bending Test Problem

To demonstrate the importance of the considerations of the previous sections with regard to the choice of the finite element formulation and if necessary the anti-hourglassing approach, a three-point bending test with different finite-element types is performed. The used FEM model is illustrated in Fig. 3.36 with measures $t = 0.1mm$, $d = 15mm$, $l = 23mm$, $R = 1.5mm$ and a beam width of $5mm$. The cylinders are rigid and the beam is made of a linear elastic material with $E = 210000\,N/mm^2$ and $\nu = 0.3$. Two load cases are applied to this model. In the first load case the lower two cylinders are fixed and the upper cylinder is pressed downwards by a prescribed displacement of $0.5mm$. In the second load case the lower cylinders are fixed too but the upper cylinder is pressed downwards by a constant force $F = 0.1N$. Between beams and cylinders a surface-to-surface penalty contact formulation is used.

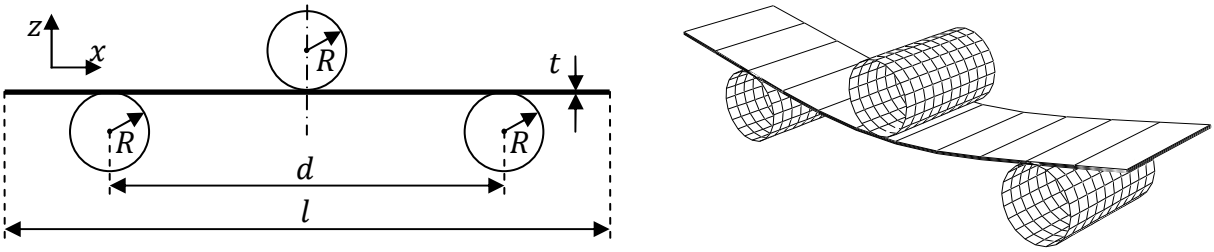


Fig. 3.36 Model for three-point bending test with different finite element types

The beam is meshed with eleven elements in the longitudinal direction and three solid elements in thickness direction. By this an aspect ratio for the solid elements of 150:63:1 is achieved. This extreme aspect ratio is used here to clearly observe the described numerical effects. Under consideration are four different finite element formulations:

- Formulation A is an underintegrated element with one integration point and therefore a constant stress/strain element;
- Element formulation B is a fully integrated element with an extra approach to avoid volumetric locking;
- Formulation C is fully integrated too but uses a heuristic approach to modify the $\boldsymbol{B}$ matrix such that the shear locking behavior is improved for poor aspect ratios;
- Formulation D is similar to C but more accurate and therefore a little bit slower.

More details about these element formulations are given in [25]. Since element formulations B-D are fully integrated, no anti-hourglassing algorithm is needed. For element formulation A three different anti-hourglassing algorithms are tested, which are

- approach 0 using no anti-hourglassing algorithm,

- approach 1 using the Flanagan-Belytschko stiffness form according to [50],
- approach 2 similar to approach 1, but using exact volume integration and
- approach 3 utilizing the Belytschko-Bindeman formulation described in [19].

Although the presented three-point bending test is a static problem, a dynamic computation with explicit time-integration is performed for being able to monitor possible problems in the dynamic behavior of the elements. To reach a quasi-static solution after an acceptable time, the load curve of Fig. 3.37a is used (for both displacement- and forced-based loading), a global velocity proportional damping constant is defined and the computation is only terminated when no changes with respect to the results are observable (see Fig. 3.37b).

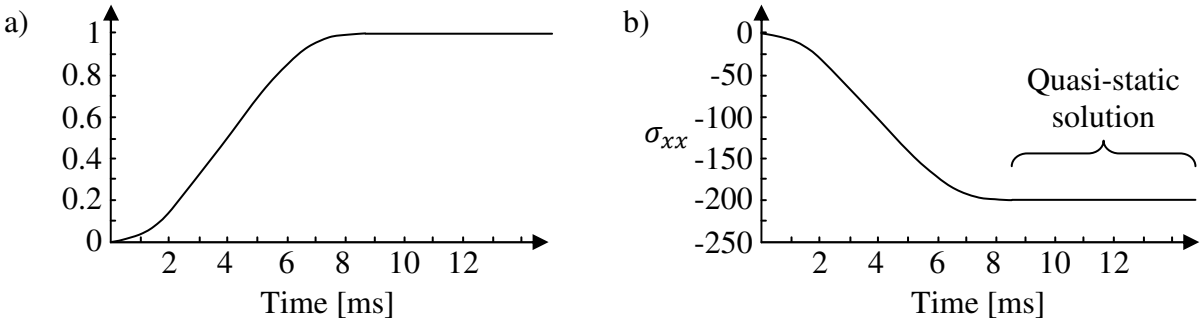


Fig. 3.37 Applied load curve for displacement- and force-based loading (a) and definition of quasi-static solution (b)

3.4.1 Results of Displacement-Based Loading

The diagram in Fig. 3.38 shows the computed bending stresses σ_{xx} in the beam evaluated at the lowest integration point(s) in the middle of the beam. Starting with the underintegrated element formulation A, one can see that the anti-hourglassing approaches 1 and 2 add too much stiffness to the element resulting in a higher bending stress. Obviously, anti-hourglassing formulation 3 works much better with element A since the stress value of this combination is very close to the result without any anti-hourglassing formulation (A0). The fully-integrated element B computes a higher maximum stress since the position of the

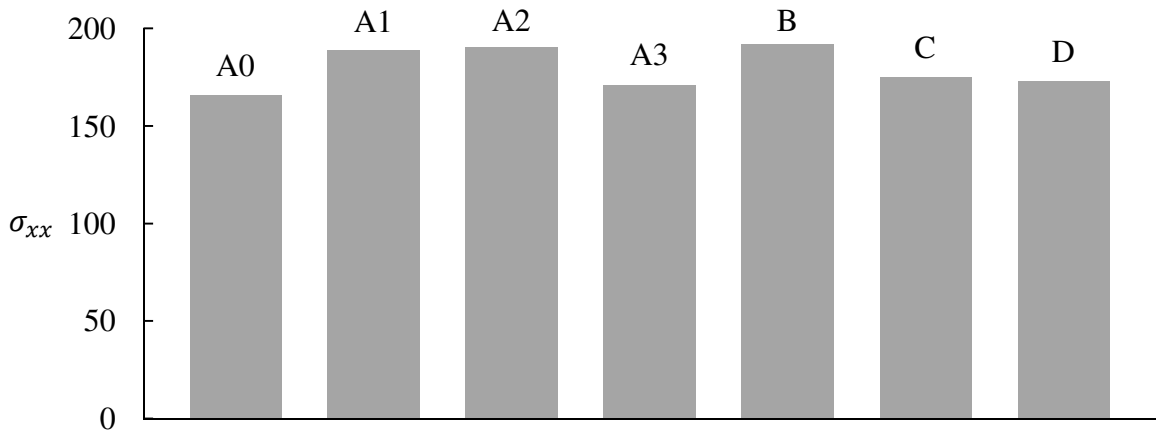


Fig. 3.38 Computed bending stress σ_{xx} for displacement-based loading

integration points is different in comparison to element A. The stresses of element formulations C and D are slightly lower than for formulation B due to the modified $\boldsymbol{B}$ matrix and small resulting errors/inaccuracies.

In meshes of real components the shape of finite elements is in most cases not like a perfect brick where all edges are perpendicular to each other. Often there are trapezoidal-like element shapes as it can be seen in Fig. 3.39a. For this reason the tests performed so far are repeated with initially deformed middle elements of the beam where the so-called taper angle α (see Fig. 3.39b) is varied between zero and twenty degrees.

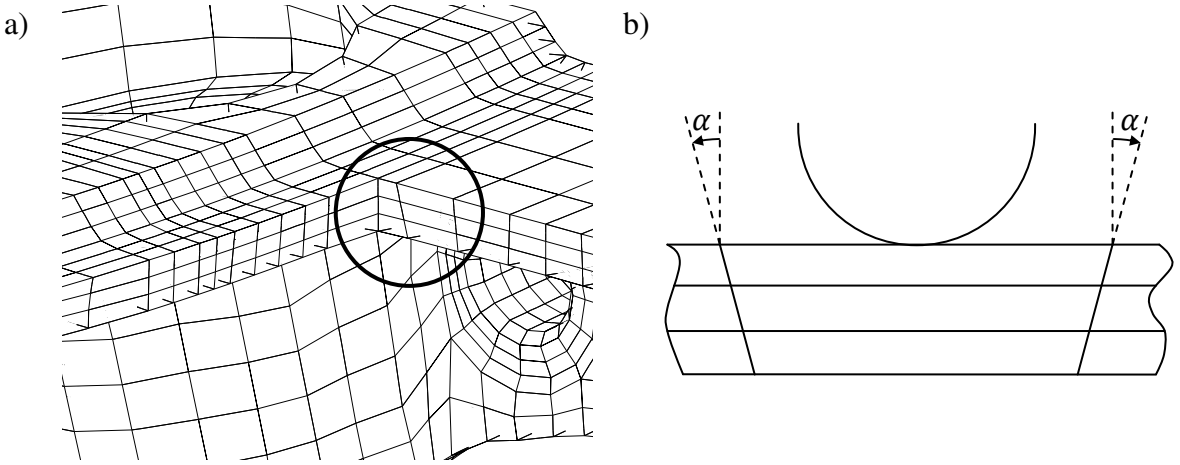


Fig. 3.39 Trapezoidal-shaped elements in mesh of a component (a) and definition of taper angle (b)

The outcome of the investigations with initially deformed elements is visible in Fig. 3.40 in combination with different anti-hourglassing formulations for the underintegrated element A. For a taper angle of zero degrees the results are identical to those of Fig. 3.38. However, for bigger taper angles some remarkable results are obtained. The underintegrated element A is very robust with respect to the taper angle if no anti-hourglassing formulation is present. Unfortunately, this will only work if no hourglassing appears, which is the case for this example but not in general and in most practical applications. Moreover, there is a tendency

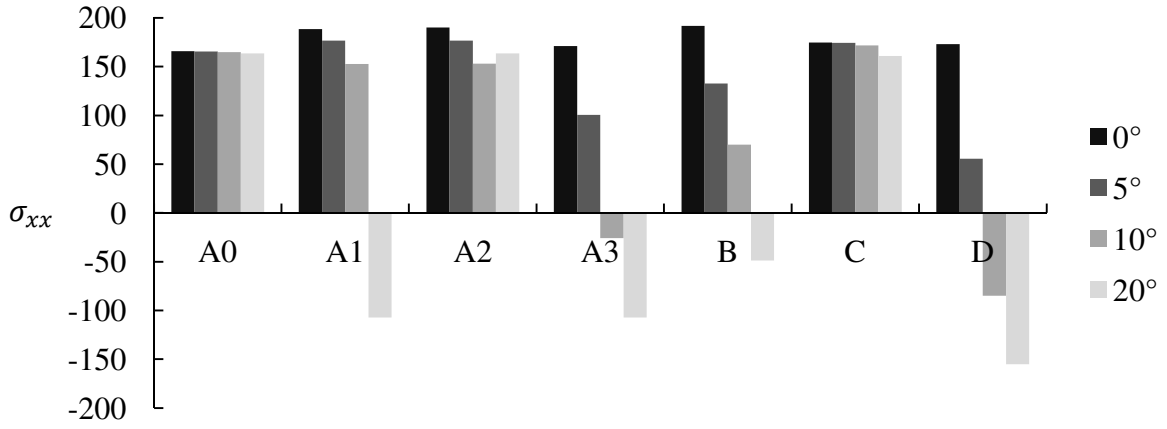


Fig. 3.40 Dependence of computed bending stress σ_{xx} on taper angle for different combinations of finite element formulations and anti-hourglassing algorithms

for all tested elements and anti-hourglassing formulations of decreasing stresses by increasing the taper angle, except for the combination A2. The anti-hourglassing algorithm 2 works very well and achieves the most accurate results compared to the other formulations. Element formulation C achieves the best results of all hourglassing-free formulations since even for bigger taper angles of $\alpha = 20°$ the difference with regard to the computed stress for $\alpha = 0°$ is small.

Another notable result is the fact that for the combinations A1, A3 and the element formulations B and D the stresses at the bottom side of the beam turn from tension into compression for bigger taper angles. An explanation for this curious behavior can be derived from Equation (3.26a) which describes the stress distribution σ_{xx} in trapezoidal shaped solid elements. For this purpose the same material parameters as for the beam model are used. The stress coordinate σ_{xx} is evaluated at $y = h/2$ with $b = 23/11mm$ and $h = 1/30\,mm$ according to the proportions of the elements in the beam model. Fig. 3.41 shows the computed stress in dependency of the taper angle α for the case $b/h = 690/11$ as in the beam model and for the case $b/h = 1$, which represents the best possible aspect ratio. As one can see, extreme aspect ratios of b/h lead to a strong dependency of the computed stress coordinate σ_{xx} on the taper angle, which may even result in a reversion of the sign. For more moderate aspect ratios the influence of the taper angle on the stresses is much smaller and almost not present for $b/h = 1$.

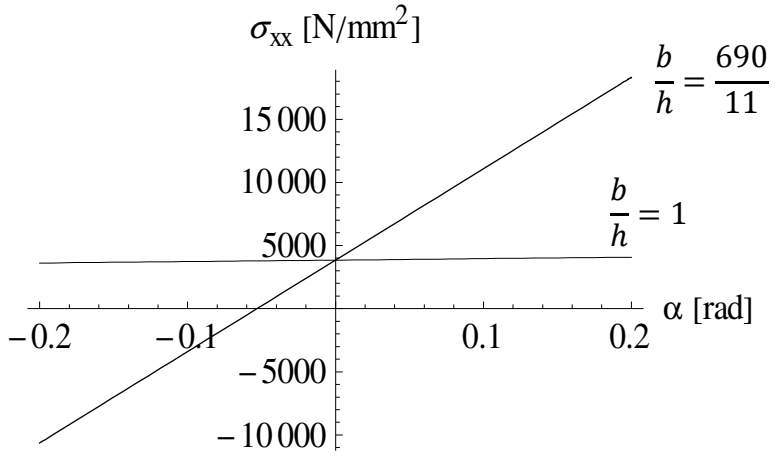


Fig. 3.41 Bending stress for fully integrated element under pure bending for different aspect ratios b/h in dependency on the taper angle α

3.4.2 Results of Forced-Based Loading

For load case two, where a single force $F = 0.1N$ is applied to the upper cylinder, the displacements of the lowest nodes in the middle of the beam (max. deflection) are evaluated in z-direction and presented in Fig. 3.42. Additionally, the solution of the beam meshed with eleven shell elements, is shown as reference solution (dashed line). The shell element formulation is based on a standard 5 parameter-model and uses reduced integration. Again, element formulation C is the most robust one and provides the most accurate results even for

bigger taper angles. Element formulation D is much more sensitive with respect to the taper angle and leads to accurate results only for a taper angle of $\alpha = 0°$, but behaves too stiff for bigger initial deformations. The underintegrated element A behaves extremely stiff in combination with the anti-hourglassing approaches 1 and 2 but gives good results in combination with anti-hourglassing algorithm 3. Without using an anti-hourglassing approach excessive hourglassing can be observed for element formulation A, leading to a dramatic stiffness decrease and huge deformations (Fig. 3.43) exceeding the range of the diagram in Fig. 3.42. Another extreme case can be observed for the fully integrated element formulation B where almost no deformation is visible even for a taper angle of $\alpha = 0°$ due to strong locking effects.

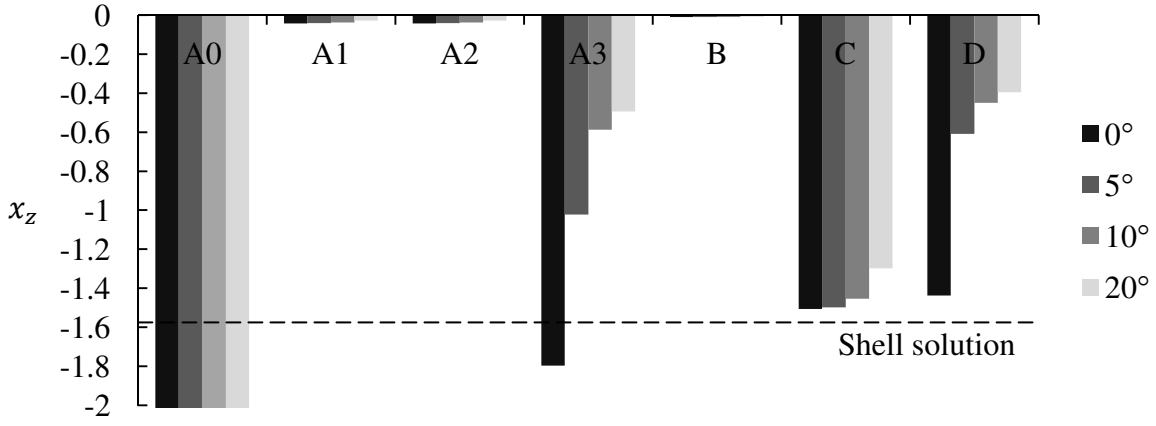


Fig. 3.42 Dependence of computed maximum deflection x_z on taper angle for different combinations of finite element formulations and anti-hourglassing algorithms

It should be finally noted that the discussed results are obtained for the example of elements with an extremely unfavorable aspect ratio which should be avoided in practical applications and real models. As already shown in Fig. 3.41 many of the used formulations will perform much better for better aspect ratios. Nevertheless, the presented tests show some tendencies and should animate the reader to implement own test computations considering particular requirements instead of using an element or anti-hourglassing formulation too carefree.

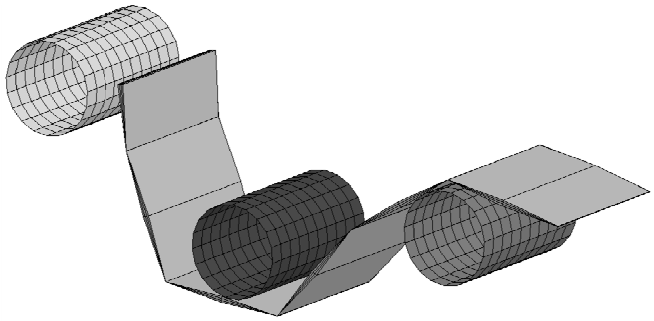


Fig. 3.43 Dramatic stiffness decrease due to excessive hourglassing with element formulation A

4 Modeling of Bearings in High Fidelity Engine Models

Typically bearings are modeled with a lot of simplifications and idealizations in ordinary finite element models. For example, modeling them by joints would be a very simple possibility. However, there are also more advanced modeling techniques which take into account the stiffness and damping characteristics of the oil film and the bearing itself [40,47,49,86,121,132,143].

In an aero-engine, there exist journal bearings as well as ball bearings. In high-fidelity models both types of bearings should be modeled without any simplifications from a structural point of view, which means for example for a ball bearing that every ball is meshed by solid elements (Fig. 1.2). This leads to the question, which mesh density is necessary for a kinematically correct behavior of the bearing model. If the mesh in the discretized bearing model is too coarse, rotation of the inner part of the bearing against the outer part is not possible due to the faceting as exemplarily shown for the pendulum in Fig. 4.1. In the following, the necessary relationships between mesh density and bearing parameters are derived for classical contact formulations as well as for so-called smooth contact formulations.

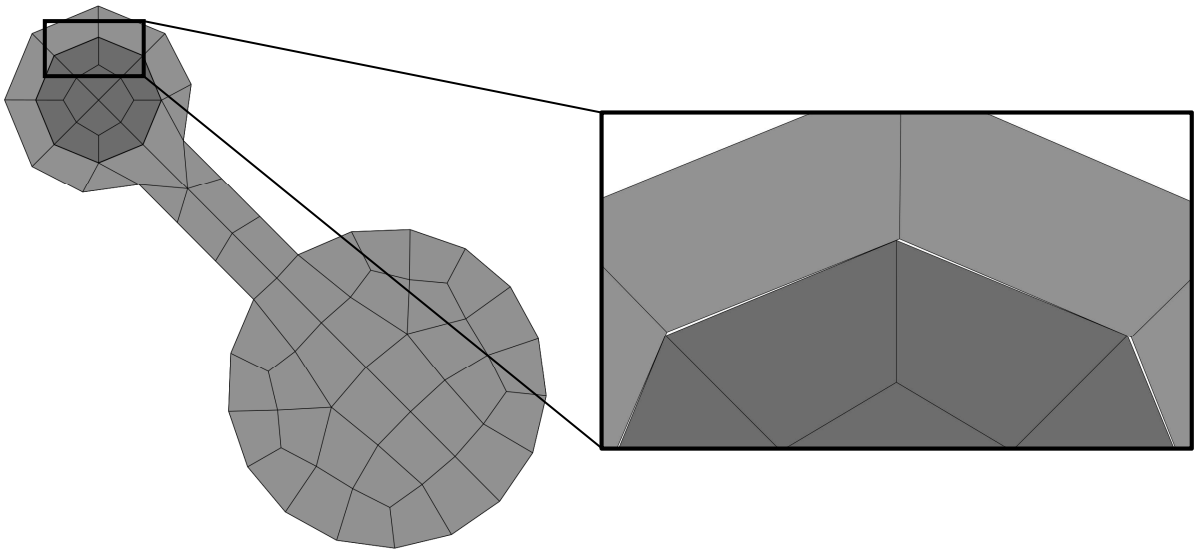


Fig. 4.1 Impossible rotation due to coarsely meshed bearing region of a pendulum

4.1 Necessary Mesh Density for Classical Contact Formulations

The following considerations are made with regard to planar problems (two-dimensional problems). Although the results are actually only applicable for such situations, the derived equations may give at least an idea about the necessary mesh density and thus are also useful for three-dimensional problems.

A simple journal bearing shown in Fig. 4.2a is determined by radius R of the inner ring and gap size t_{gap} between inner and outer ring. If such a bearing is discretized by linear finite elements, the circles become polygons with edge length c at the inner ring. For reasons of simplicity, we assume that all nodes are equally spaced around the circumference which leads to regular polygons as shown in Fig. 4.2b.

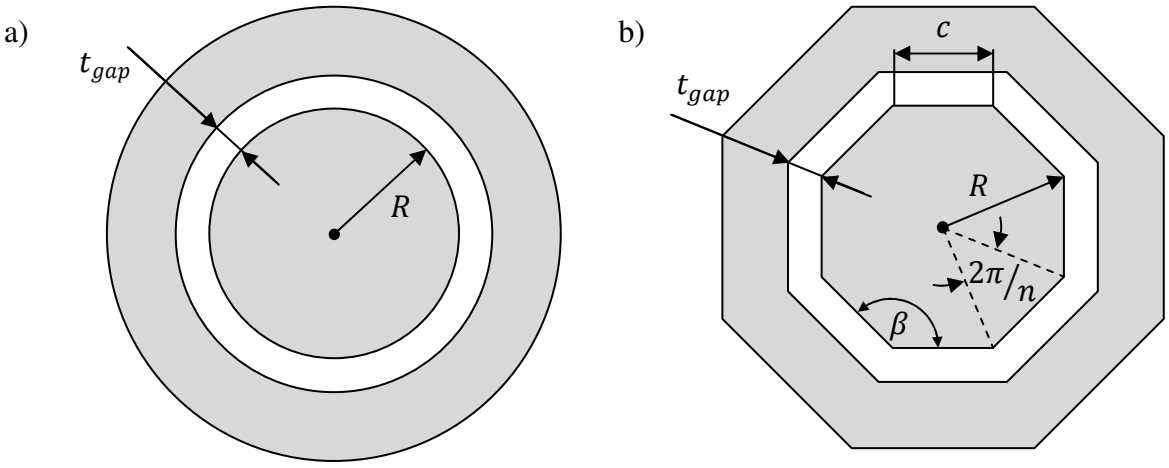


Fig. 4.2 Original journal bearing (a) and discretized bearing with faceted surface (b) [78]

The inner angle β of the polygon depends on the number of nodes n around the circumference as

$$\beta = \pi - \frac{2\pi}{n} = \frac{n-2}{n} \cdot \pi, \quad n > 2, \tag{4.1}$$

and the edge length c results from simple geometrical considerations as

$$c = 2R \cdot sin\frac{\pi}{n}. \tag{4.2}$$

According to Fig. 4.3 it is clear that the limit state between rotation or no rotation of the inner ring against the outer ring or vice versa is given for the situation where a equals b. In this situation we get from Fig. 4.3b

$$a = b = \left(R + t_{gap}\right) \cdot sin\frac{\pi}{n}. \tag{4.3}$$

In this case, the angle α becomes

$$\alpha = \frac{\pi}{2} - \frac{\beta}{2} = \frac{\pi}{2}\left(1 - \frac{n-2}{n}\right) = \frac{\pi}{n}. \tag{4.4}$$

Additionally, we know that for $a = b$

$$\cos\alpha = \frac{c/2}{b} = \frac{R}{R + t_{gap}}. \tag{4.5}$$

By using (4.4) in (4.5), we finally get the necessary relationship between radius of the bearing, gap size and mesh density in terms of the number of nodes n in circumferential direction as

$$\cos\frac{\pi}{n} = \frac{R}{R + t_{gap}} \tag{4.6}$$

describing the limit state between rotation and no rotation.

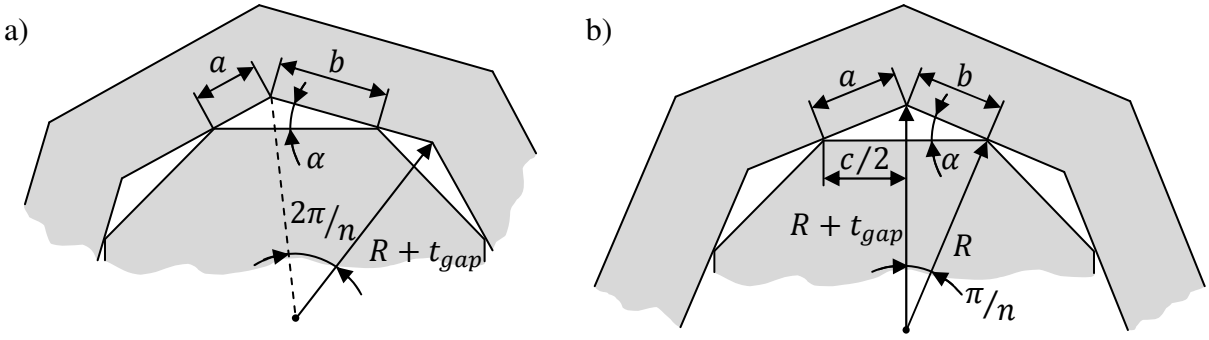


Fig. 4.3 Kinematics of bearing rotation (a) and limit situation (b)

Based on Equation (4.6), the question about the necessary mesh density for the pendulum in Fig. 4.1 for an admissible rotation can be answered rather easily. If we assume a bearing radius of $R = 10mm$ and a gap size $t_{gap} = 0.05mm$, we obtain for the minimum number of equally spaced nodes around the circumference

$$n = \frac{\pi}{\arccos\left(\frac{R}{R + t_{gap}}\right)} = \frac{\pi}{\arccos\left(\frac{10mm}{10mm + 0.05mm}\right)} \approx 31.5 \tag{4.7}$$

rounded up to $n = 32$. This result can be validated by performing a numerical FEM simulation. For this purpose the inner part of the pendulum's bearing is fixed and gravity in negative y-direction is applied as the only load. All parts are made of mild steel ($E = 210000\,N/mm^2, \nu = 0.3, \rho = 7.9 \cdot 10^{-9}\ t/mm^3$). Three different mesh densities ($n = 28$, $n = 32$, and $n = 40$) are compared, and the displacements of the node marked in Fig. 4.4 are evaluated. The initial position of the pendulum is shown in Fig. 4.4a.

The simulation results with an explicit time-integration scheme are plotted in Fig. 4.5. For 28 nodes, obviously no rotation is possible. For 32 nodes the pendulum swings as expected, but

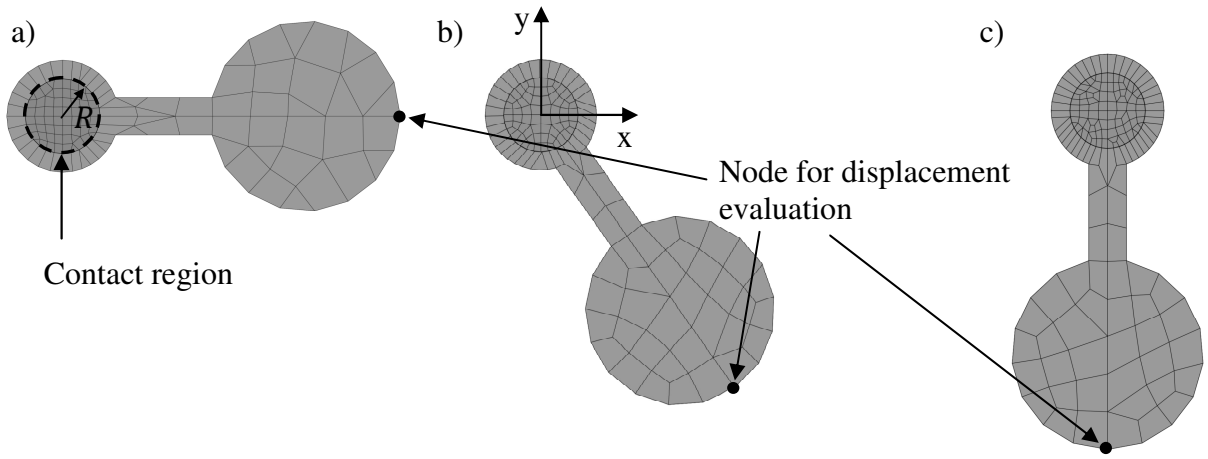


Fig. 4.4 Finite element models of pendulum with different mesh densities of bearing region: a) 28, b) 32 and c) 40 nodes in circumferential direction

apparently an amplitude decay is observable although no friction or damping was used in the computation. The explanation for this decay is the coarse mesh in the contact region. Although it allows for a rotation, the contact forces of the used penalty algorithm (in combination with a node-to-surface approach) also have a tangential component due to the coarse mesh resulting in a small deceleration of the pendulum. If the mesh density is increased (here to 40 nodes in circumferential direction), the amplitude decay becomes much smaller. But there is a second effect also described in [21], which plays an important role with respect to the amplitude decay. If two continuous bodies are impacting each other, the contact condition $\boldsymbol{g}_N = \mathbf{0}$, and therefore equality of the displacements and velocities of the impacting surfaces, is only fulfilled for the contact surfaces which have no thickness and thus no mass. In the case of non-continuous, discretized bodies, the mass is concentrated in the nodes of the finite elements. By this, more material and more mass is enforced to fulfill the contact condition than in the continuous case.

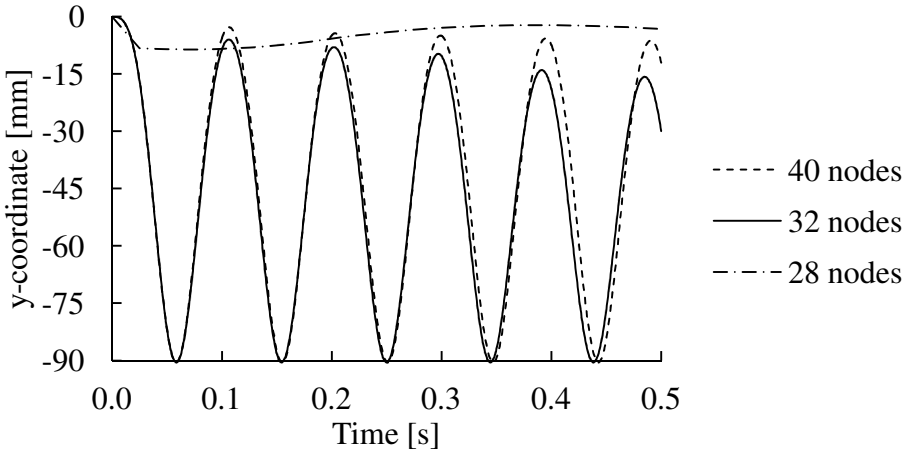


Fig. 4.5 History of y-coordinate of outermost node of pendulum for different mesh densities in the contact region

This effect is more clearly observable in an example of two cubes impacting each other (Fig. 4.6a). The lower cube is fixed at its bottom side and the upper cube may only move downwards. A gravitational load $\boldsymbol{g}$ is applied to the whole system (Fig. 4.6a). The simulation of this problem is performed with explicit and implicit time-integration schemes and different mesh densities for the cubes (1 quadrilateral element per cube and 2x2 quadrilateral elements per cube, respectively). The computed displacements evaluated at the upper left node S of the upper cube (body B) show clearly the influence of the mesh density (Fig. 4.6b). The finer the mesh, the less energy dissipation is observable. It can be also seen that the contact condition is responsible for the different solutions because up to approximately $t = 0.45s$ (time of impact) all curves are identical. These considerations demonstrate that the number of nodes calculated with Equation (4.6) should be considered only as a lower bound for the mesh density. It may be used to determine either

- the required number of nodes $n = \pi/\arccos\left(R/R + t_{gap}\right)$ for a given radius R and gap size t_{gap} or

- the required radius $R = t_{gap}\cos(\pi/n)/(1-\cos(\pi/n))$ for a given number of nodes n and gap size t_{gap} or
- the required gap size $t_{gap} = R(1-\cos(\pi/n))/\cos(\pi/n)$ for a given radius R and number of nodes n.

Especially in explicit computations, where the time-step size (and therefore the overall computational time) depends on the size of the finite elements (see Section 6.1), the second and third options are of greater interest.

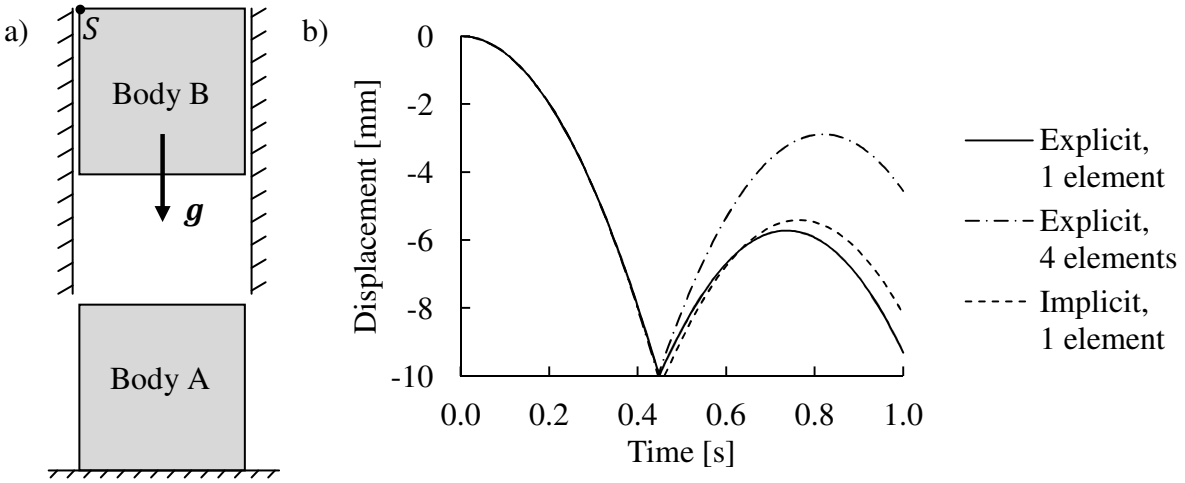


Fig. 4.6 Model of two cubes impacting each other due to gravity (a) and displacement of upper left corner of body B computed with different models (b)

Besides the rotation also the radial displacements of the bearing should be as accurate as possible. This means for an ideal bearing with zero gap size, which cannot exist in the finite element world, that there is no radial displacement. For measuring the radial displacement, a cylindrical coordinate system placed at the center of the bearing is defined. Three different cases are investigated: In the first case, a very coarse mesh with only $n = 8$ is used. By this, the necessary gap size is calculated for $R = 10mm$ as $t_{gap} = [R/\cos(\pi/n)] - R \approx 0.824mm$. In the second case, the radial displacement of the pendulum with $n = 32$ is measured, and in the third case the coarse mesh with $n = 8$ is used again, but an additional contact smoothing option is applied. By this smoothing option, an artificial contact surface defined by polynomial functions is constructed (see Fig. 4.7a).

For all three cases the pendulum swings as expected, but the differences with respect to the radial displacement are remarkable (Fig. 4.7b). The pendulum with the finest mesh (case 2) is closest to the ideal solution of zero radial displacement. The solution for the pendulum with only 8 nodes in circumferential direction and a bigger gap size (case 1) shows the biggest peaks in terms of radial displacements. But this undesired behavior can be improved by applying the smoothing option (case 3), where there is only a big radial displacement at the beginning of the simulation when the pendulum “falls” into the bearing. Afterwards, the relative displacements are similar to the model with the much finer mesh. Obviously the smoothing option is a good possibility to keep a coarse mesh without changing the gap size or

other parameters of the bearing while receiving accurate results. Of course, the derived equation for the necessary mesh density is not valid anymore if the smoothing option is activated. For this reason we have to reconsider the problem in the next section.

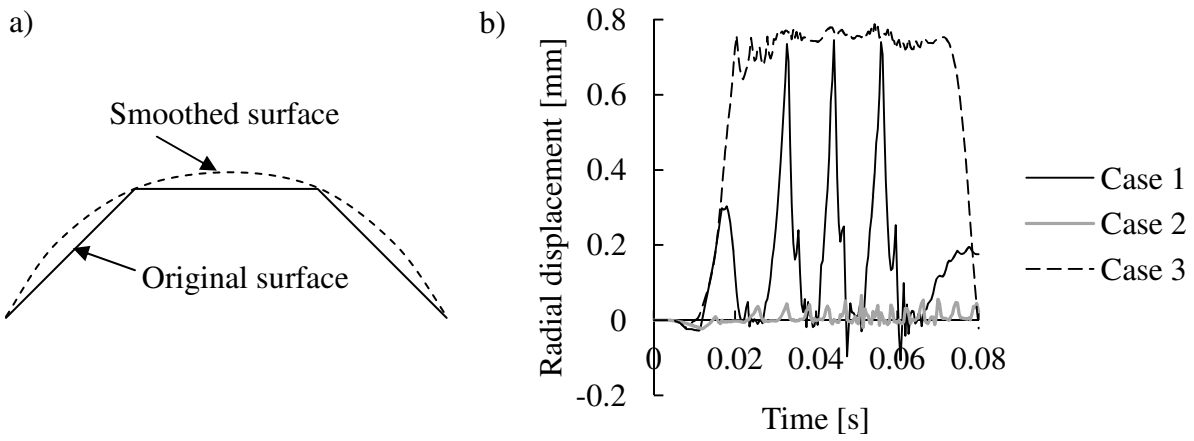


Fig. 4.7 Original surface and artificially smoothed contact surface (a) and radial displacements of simple pendulum with different mesh configurations (b) [78]

4.2 Mesh Density for Artificially Smoothed Contact Formulations

As demonstrated in the section before, an artificial smoothing of the contact surfaces can improve simulation results. Deriving the necessary relationship between radius of the bearing, gap size and mesh density is a bit more elaborate than for classical contact formulations without smoothing. As before, we restrict the considerations to the two-dimensional space and assume that the nodes of the finite elements are equally spaced in circumferential direction. All other bearing properties are defined according to Fig. 4.2. The smoothing functions can be constructed by different types of functions, for example NURBS, B-Splines or polynomials. Here, we will restrict ourselves to quadratic polynomials as smoothing functions which offer the important feature of G^1-continuity at the nodes being the transition points between two functions. Geometric G-continuity is a relaxed form of C-continuity [9,10]. The task is to construct a closed G^1-continous line with quadratic polynomials using the nodes of the polygon as grid points (Fig. 4.8). For the n edges of the polygon n quadratic polynomials are used.

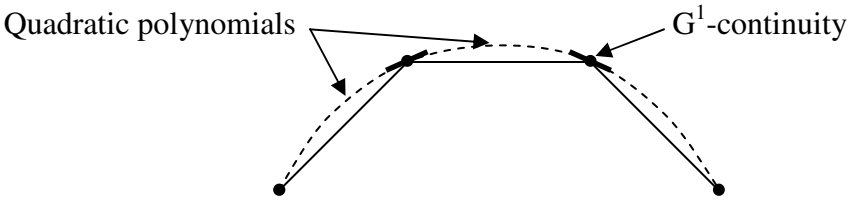


Fig. 4.8 Construction of quadratic polynomials through FE nodes [78]

4.2.1 Construction of Quadratic Polynomials

We start with an arbitrary quadratic polynomial

$$s(t) = At^2 + Bt + C \tag{4.8}$$

as visible in Fig. 4.9a. For the determination of the unknown constants A, B and C some geometrical considerations are necessary. Since the edge length of a regular n-sided polygon is given by Equation (4.2) we need to fulfill the two boundary conditions

$$s(0) = s(c) = 0 \tag{4.9}$$

where the first yields $C = 0$. The derivative of (4.8) with respect to t is

$$s'(t) = 2At + B. \tag{4.10}$$

From the G^1-continouity and symmetry condition follows $\varphi = \psi$ according to Fig. 4.9b. Since the inner angle of a regular polygon is given by Equation (4.1) as $\beta = [(n-2)/n]\pi$, it follows that

$$\varphi = \psi = \frac{1}{2}(\pi - \beta) = \frac{\pi}{2}\left(1 - \frac{n-2}{n}\right) = \frac{\pi}{n}. \tag{4.11}$$

Together with (4.10) this determines the unknown variable

$$B = s'(t=0) = \tan\varphi = \tan\frac{\pi}{n}. \tag{4.12}$$

Finally, the last unknown A can be calculated from the right boundary (4.9) as

$$s\left(t = 2R\sin\frac{\pi}{n}\right) = A \cdot \left(2R\sin\frac{\pi}{n}\right)^2 + \left(\tan\frac{\pi}{n}\right) \cdot \left(2R\sin\frac{\pi}{n}\right) = 0 \tag{4.13}$$

and yields $A = -1/[2R\cos(\pi/n)]$. The desired function (4.8) for the smoothing thus reads

$$s(t) = -\frac{1}{2R\cos\frac{\pi}{n}} \cdot t^2 + \tan\left(\frac{\pi}{n}\right) \cdot t. \tag{4.14}$$

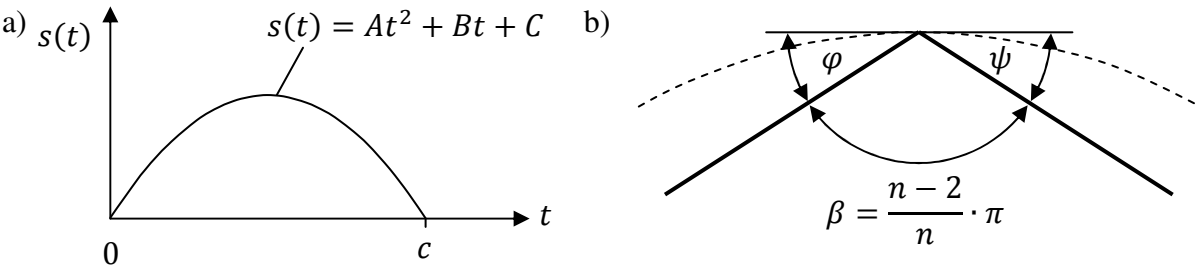


Fig. 4.9 Quadratic polynomial (a) and angles at arbitrary vertex of regular polygon (b) [78]

4.2.2 Transformation of Polynomial to the *i*-th Edge of the Polygon

The derived quadratic function has to be transferred to each edge of the polygon for covering the whole contact surface. Firstly we perform this for the inner contact surface, whose

components have the superscript "in". For this purpose it is necessary to transform the quadratic polynomial (4.14) into a parameterized form $\left[{}^{in}x(t), {}^{in}y(t)\right]$ where

$$ {}^{in}x(t) = t, \tag{4.15a} $$

$$ {}^{in}y(t) = -\frac{1}{2R\cos\frac{\pi}{n}} \cdot t^2 + \tan\left(\frac{\pi}{n}\right) \cdot t, \quad t \in [0, 2R\sin(\pi/n)]. \tag{4.15b} $$

Further, we assume without loss of generality that the polygon is constructed such that the slope of the first edge ($i = 1$) is zero and the bearing coordinate system is placed at the center of the polygon (Fig. 4.10a).

The transformation process of the smoothing function consists of a rotation and a translation, which has to be performed separately for every *i*-th edge of the polygon. Starting with the rotation, the rotation angle for the *i*-th edge is

$$ {}^{in}\gamma_i = \left(\frac{2\pi}{n}\right) \cdot (i-1). \tag{4.16} $$

The parameterized smoothing function (4.15) has to be multiplied by a corresponding rotation matrix which gives

$$ \begin{aligned} \begin{bmatrix} {}^{in}\tilde{x}_i(t) \\ {}^{in}\tilde{y}_i(t) \end{bmatrix} &= \begin{bmatrix} \cos {}^{in}\gamma_i & \sin {}^{in}\gamma_i \\ -\sin {}^{in}\gamma_i & \cos {}^{in}\gamma_i \end{bmatrix} \cdot \begin{bmatrix} {}^{in}x(t) \\ {}^{in}y(t) \end{bmatrix} \\ &= \begin{bmatrix} t \cdot \cos {}^{in}\gamma_i + {}^{in}y(t) \cdot \sin {}^{in}\gamma_i \\ -t \cdot \sin {}^{in}\gamma_i + {}^{in}y(t) \cdot \cos {}^{in}\gamma_i \end{bmatrix}. \end{aligned} \tag{4.17} $$

In a second step the rotated smoothing functions have to be translated to their polygon edges. This is done by adding a translation vector ${}^{in}\boldsymbol{u}_i$ to every smoothing function. The coordinates of the *i*-th translation vector with respect to Fig. 4.10b are

$$ {}^{in}\boldsymbol{u}_i = \begin{bmatrix} {}^{in}u_x \\ {}^{in}u_y \end{bmatrix}_i = \begin{bmatrix} R\sin\left({}^{in}\gamma_i - \frac{\pi}{n}\right) \\ R\cos\left({}^{in}\gamma_i - \frac{\pi}{n}\right) \end{bmatrix} = \begin{bmatrix} R\sin\left(\frac{\pi}{n} \cdot (2i-3)\right) \\ R\cos\left(\frac{\pi}{n} \cdot (2i-3)\right) \end{bmatrix}. \tag{4.18} $$

This finally leads to the smoothing functions of the *i*-th edge

$$ \begin{aligned} \begin{bmatrix} {}^{in}x_i(t) \\ {}^{in}y_i(t) \end{bmatrix} &= \begin{bmatrix} {}^{in}\tilde{x}_i(t) \\ {}^{in}\tilde{y}_i(t) \end{bmatrix} + \begin{bmatrix} {}^{in}u_x \\ {}^{in}u_y \end{bmatrix}_i \\ &= \begin{bmatrix} t \cdot \cos {}^{in}\gamma_i + {}^{in}y(t) \cdot \sin {}^{in}\gamma_i + R\sin\left(\frac{\pi}{n} \cdot (2i-3)\right) \\ -t \cdot \sin {}^{in}\gamma_i + {}^{in}y(t) \cdot \cos {}^{in}\gamma_i + R\cos\left(\frac{\pi}{n} \cdot (2i-3)\right) \end{bmatrix} \end{aligned} \tag{4.19} $$

with the substitutions (4.16) and (4.15b) for ${}^{in}\gamma_i$ and ${}^{in}y(t)$. With Equation (4.19) we have derived a possibility to build smoothed surfaces for arbitrary polygons as shown in Fig. 4.11.

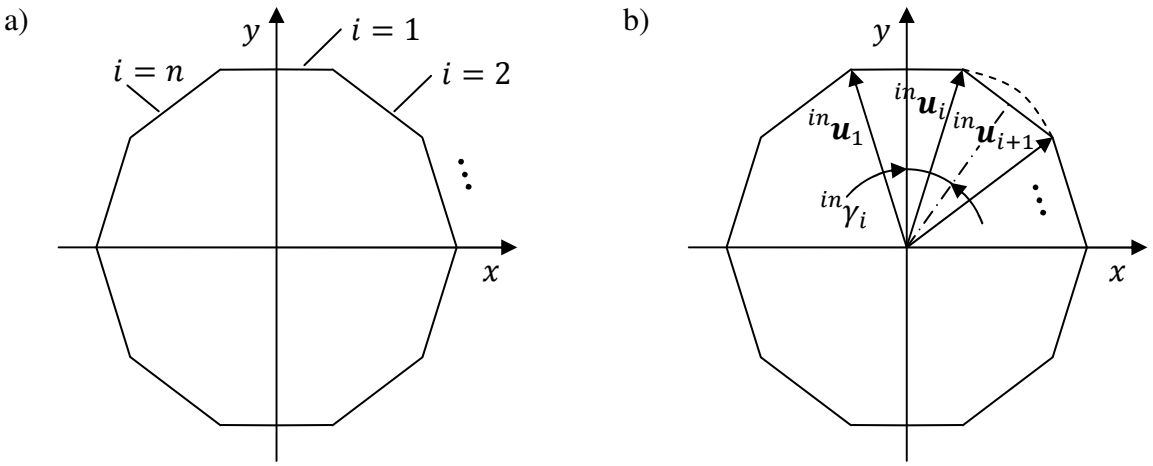


Fig. 4.10 Position of polygon and numbering of edges (a) as well as translation vectors for smoothing functions (b) [78]

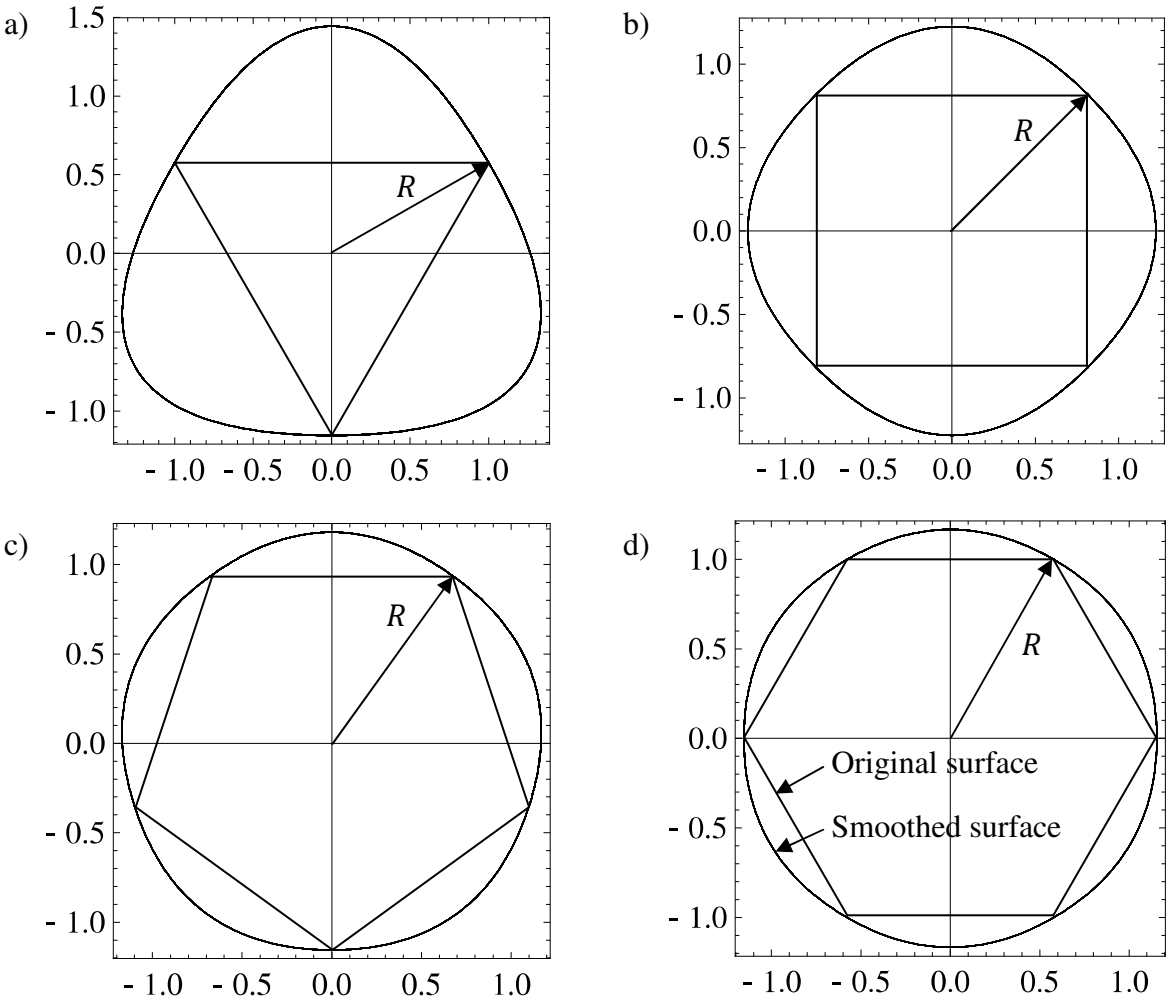


Fig. 4.11 Regular polygons with three (a), four (b), five (c) and six (d) edges with surfaces smoothed by quadratic polynomials [78]

4.2.3 Smoothing of Second Contact Surface

The construction of the smoothed contact surface for the outer bearing ring is very similar to the first one (red surface in Fig. 4.12b.). The first of the two major differences is that the radius is now $(R + t_{gap})$. The second difference is that in contrast to the first smoothed contact surface we now assume that the polygon is constructed such that the first node ${}^{out}\boldsymbol{u}_1$ of the first edge ($i = 1$) is situated at the y-axis of the bearing coordinate system (Fig. 4.12a). All components belonging to this outer contact surface have the superscript "out".

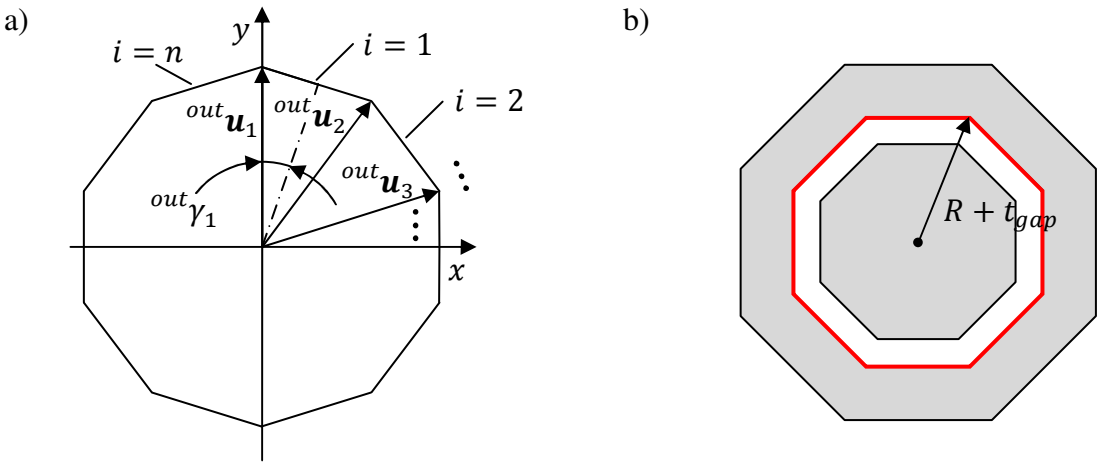


Fig. 4.12 Assumed position of outer polygon and numbering of edges and translation vectors (a) and radius of outer contact surface (b) [78]

Changing the radius R in Equation (4.15) to $(R + t_{gap})$ yields the basic parameterized quadratic polynomial

$$ {}^{out}x(t) = t, \tag{4.20a} $$

$$ {}^{out}y(t) = -\frac{1}{2(R + t_{gap})\cos\left(\frac{\pi}{n}\right)} \cdot t^2 + \tan\left(\frac{\pi}{n}\right) \cdot t, \tag{4.20b} $$

$$ t \in \left[0, 2(R + t_{gap})\sin(\pi/n)\right]. $$

The rotation angle for the i-th smoothed function of the outer contact surface is now

$$ {}^{out}\gamma_i = \left(\frac{2\pi}{n}\right) \cdot \left(i - \frac{1}{2}\right) \tag{4.21} $$

and used in the rotation matrix for the computation of the rotated functions similar to Equation (4.17) which gives

$$ \begin{bmatrix} {}^{out}\tilde{x}_i(t) \\ {}^{out}\tilde{y}_i(t) \end{bmatrix} = \begin{bmatrix} \cos {}^{out}\gamma_i & \sin {}^{out}\gamma_i \\ -\sin {}^{out}\gamma_i & \cos {}^{out}\gamma_i \end{bmatrix} \cdot \begin{bmatrix} {}^{out}x(t) \\ {}^{out}y(t) \end{bmatrix} = \begin{bmatrix} t \cdot \cos {}^{out}\gamma_i + {}^{out}y(t) \cdot \sin {}^{out}\gamma_i \\ -t \cdot \sin {}^{out}\gamma_i + {}^{out}y(t) \cdot \cos {}^{out}\gamma_i \end{bmatrix}. \tag{4.22} $$

According to Fig. 4.12, the translation vector for the outer smoothing functions reads as

$$ {}^{out}\boldsymbol{u}_i = \begin{bmatrix} {}^{out}u_x \\ {}^{out}u_y \end{bmatrix}_i = \begin{bmatrix} \left(R + t_{gap}\right) \sin\left(\frac{2\pi}{n} \cdot (i-1)\right) \\ \left(R + t_{gap}\right) \cos\left(\frac{2\pi}{n} \cdot (i-1)\right) \end{bmatrix}. \tag{4.23} $$

Superposing this translation to Equation (4.22) finally gives the smoothing function for the i-th edge

$$ \begin{aligned} \begin{bmatrix} {}^{out}x_i(t) \\ {}^{out}y_i(t) \end{bmatrix} &= \begin{bmatrix} {}^{out}\tilde{x}_i(t) \\ {}^{out}\tilde{y}_i(t) \end{bmatrix} + \begin{bmatrix} {}^{out}u_x \\ {}^{in}u_y \end{bmatrix}_i \\ &= \begin{bmatrix} t \cdot \cos {}^{out}\gamma_i + {}^{out}y(t) \cdot \sin {}^{out}\gamma_i + \left(R + t_{gap}\right) \sin\left(\frac{2\pi}{n} \cdot (i-1)\right) \\ -t \cdot \sin {}^{out}\gamma_i + {}^{out}y(t) \cdot \cos {}^{out}\gamma_i + \left(R + t_{gap}\right) \cos\left(\frac{2\pi}{n} \cdot (i-1)\right) \end{bmatrix}. \end{aligned} \tag{4.24} $$

4.2.4 Condition for Rotation of Smoothed Surfaces

The limit state between rotation and no rotation is reached if the outermost point of the inner smoothing surface ($t = c/2$) is in contact with the nodes of the outer smoothing surface as shown in Fig. 4.13a. With Equation (4.2) this situation can be expressed as

$$ {}^{in}x_i\left(t = R \sin\frac{\pi}{n}\right) = {}^{out}x_i(t = 0), \tag{4.25a} $$

$$ {}^{in}y_i\left(t = R \sin\frac{\pi}{n}\right) = {}^{out}y_i(t = 0). \tag{4.25b} $$

For further investigations it is sufficient to consider the case $i = 1$, because if Equations (4.25) are fulfilled for this case they are also fulfilled for all other i's (see Fig. 4.13a). Condition (4.25a) is automatically satisfied due to the way of constructing the smoothing functions and defining their positions. Condition (4.25b) reads for $i = 1$ with ${}^{in}\gamma_1 = 0$, ${}^{out}\gamma_1 = \pi/n$ and Equations (4.19) and (4.24)

$$ -\frac{\left(R \sin\left(\frac{\pi}{n}\right)\right)^2}{2R \cos\left(\frac{\pi}{n}\right)} + \tan\left(\frac{\pi}{n}\right) \cdot R \sin\left(\frac{\pi}{n}\right) + R \cos\left(\frac{-\pi}{n}\right) = R + t_{gap} \tag{4.26} $$

which can be simplified to

$$ \frac{2\left(R + t_{gap}\right)}{R} = \frac{1}{\cos\left(\frac{\pi}{n}\right)} + \cos\left(\frac{\pi}{n}\right). \tag{4.27} $$

Equation (4.27) finally describes the necessary relationship between radius R, gap size t_{gap}, and number of nodes n in circumferential direction for the limit state between rotation and no rotation of smoothed surfaces.

In Fig. 4.13b the advantage of using smoothed surfaces instead of a classical contact formulation for rotating surfaces is depicted for the example of a mesh with three nodes in circumferential direction. By using a classical contact formulation (bold lines) a rotation

would not be possible due to the penetration of the surfaces, but for the same mesh and geometry, a rotation is obviously possible if the surfaces are smoothed with quadratic polynomials. This means that coarser meshes may be used leading to smaller models with less degrees of freedom and bigger time-steps in explicit computations (see Section 6), both reducing the overall computational time.

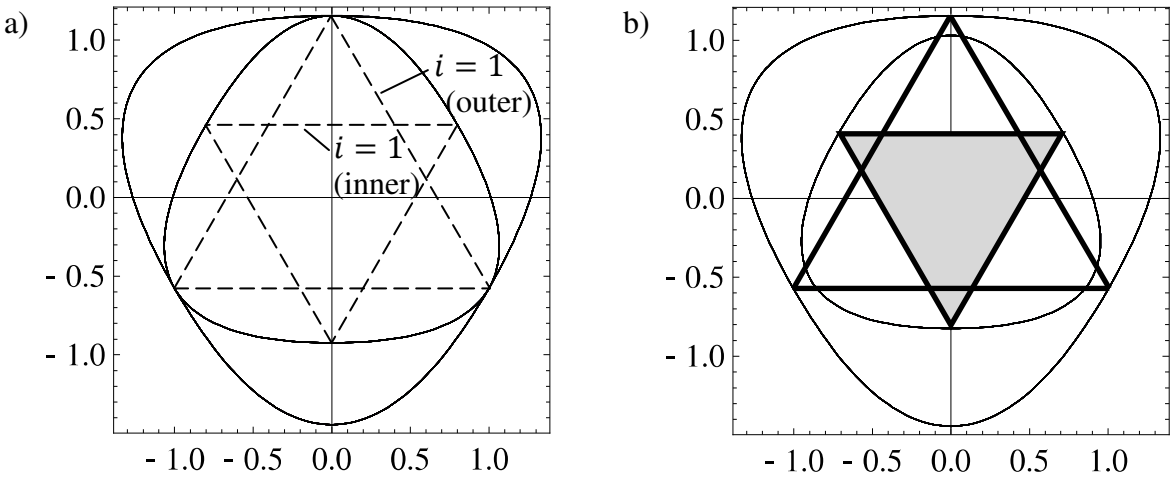


Fig. 4.13 Limit state between rotation and no rotation of smoothed surfaces for mesh with three nodes (a) [78] and possible rotation for smoothed contact surfaces (thin lines) but impossible rotation for non-smoothed contact surfaces (thick lines, b)

For demonstrating the application of Equation (4.27), we use the pendulum example in Fig. 4.4 with radius $R = 10mm$ and gap size $t_{gap} = 0.05mm$. Equation (4.27) can be solved for the number of nodes n by multiplying it with $\cos(\pi/n)$ and transforming it into the quadratic equation

$$\cos^2\frac{\pi}{n} - \frac{2\left(R + t_{gap}\right)}{R} \cdot \cos\frac{\pi}{n} + 1 = 0, \tag{4.28}$$

which has the solutions

$$\cos\frac{\pi}{n} = \frac{R + t_{gap}}{R} \pm \sqrt{\left(\frac{R + t_{gap}}{R}\right)^2 - 1}\,. \tag{4.29}$$

The plus-sign of the solution is inapplicable because the right hand side would always be greater than one, whereas the cosine function is always smaller or equal to one. For the minus-sign we obtain

$$n = \frac{\pi}{\arccos\left(\dfrac{R + t_{gap}}{R} - \sqrt{\left(\dfrac{R + t_{gap}}{R}\right)^2 - 1}\right)} \tag{4.30}$$

which here yields a value of $n \approx 7.15$ and is rounded up to the integer value $n = 8$. This means that at least eight nodes in circumferential direction are necessary in the bearing of the

pendulum model to guarantee a kinematically admissible rotation with smoothed bearing surfaces.

Appropriate implementations similar to the described smoothing technology using the smoothed surfaces for the contact computation (gap size, contact forces, etc.) are available in commercial FE tools nowadays [43,98]. For proving the correctness of the result (4.30), three different simulations for the pendulum model with mesh densities $n = 6$, $n = 7$ and $n = 8$ are performed. For these explicit computations the penalty algorithm is used and the smoothing option is activated. As visible in Fig. 4.14, the pendulums with $n = 7$ and $n = 8$ swing correctly (curves are coincident), whereas for the the coarsest mesh ($n = 6$) rotation is not possible. The pendulum correctly moves downward at the beginning, but a further displacement in this direction is inhibited and the pendulum swings back. Actually a correct rotation should not be possible for the model with $n = 7$ according to the result of Equation (4.30), but as described in Section 2.2.2 the penalty contact algorithm allows small penetrations of contact surfaces. By this, a rotation becomes possible, especially since the outcome of the calculation $n \approx 7.15$ is very close to $n = 7$.

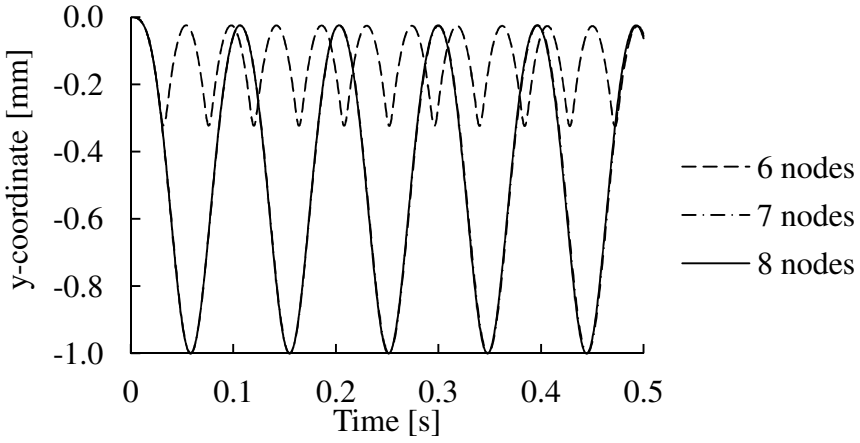


Fig. 4.14 History of y-coordinate of outermost node of pendulum for different mesh densities in the contact region with activated smoothing option [78]

Finally it should be noted that newer developments like the so-called Isogeometric Analysis [68,33] can be very useful especially for the modeling and simulation of bearings. In this approach higher-order functions like Non-Uniform Rational B-Splines (NURBS) are used as interpolation functions for finite elements, which has the advantage of an exact geometry description in the sense of Computer Aided Design (CAD). Meshing for example a circle or a ball with such elements provides also a circle or ball in the finite element world and not a polygon or polyhedron like for classical finite elements with linear geometric interpolation functions. By this, very good and realistic results are obtained also with regard to contact forces [107].

5 Time Integration

In this chapter an important aspect of nonlinear FEM, the time-integration is discussed. These procedures are necessary in transient FEM problems for solving the governing equations of motion. After a short introduction, different time-integration algorithms are presented and their effects are demonstrated using simple examples. Subsequently, very important aspects regarding stability and accuracy of the mentioned methods are considered in detail and special attention is drawn to the stability of time-integration algorithms used for the simulation of elastic rotating structures. Finally, a possibility for the optimization of parameters of the different methods is shown which gives the opportunity of adjusting the properties of a time-integration algorithm to particular applications.

5.1 Introduction

From a mechanical point of view, a dynamic system like an aero-engine can be described by an equation of motion of the form

$$\boldsymbol{M}(\boldsymbol{x})\ddot{\boldsymbol{x}} + \boldsymbol{D}(\boldsymbol{x})\dot{\boldsymbol{x}} + \boldsymbol{K}(\boldsymbol{x})\boldsymbol{x} = \boldsymbol{F}(\boldsymbol{x}, t) \tag{5.1}$$

with mass matrix $\boldsymbol{M}$, damping matrix $\boldsymbol{D}$, stiffness matrix $\boldsymbol{K}$, vector of external loads $\boldsymbol{F}(\boldsymbol{x}, t)$, and the time-dependent displacements $\boldsymbol{x}$, velocities $\dot{\boldsymbol{x}}$ and accelerations $\ddot{\boldsymbol{x}}$ (see also Section 2.1). In general, such a nonlinear system of differential equations can only be solved numerically. For this, two steps are necessary:

1. Time integration,
2. Solving the resulting system of equations (not necessary for some methods under certain circumstances).

For numerical time-integration there exist two different approaches which can be visualized with a simple example. Let us assume that we want to solve the differential equation

$$\dot{x} = f(x, t) \tag{5.2}$$

in the interval $[t_n, t_{n+1}]$ by numerical integration. This is typically done by dividing the time interval into an arbitrary number of subintervals. Starting from a given solution x_n at time point t_n, the integration between t_n and t_{n+1} can be performed as

$$x_{n+1} = x_n + \int_{t_n}^{t_{n+1}} f(x, t)\,\mathrm{d}t \approx x_n + \Delta t \cdot f(x_n, t_n) \tag{5.3}$$

with $\Delta t = t_{n+1} - t_n$ according to Fig. 5.1a and resolves straightforwardly in the solution

$$x_{n+1} \approx x_n + \Delta t \cdot f(x_n, t_n) \tag{5.4}$$

at $t = t_{n+1}$. On the right-hand side of this equation only quantities known at the current time point t_n are used. Therefore, this is an explicit approach for time-integration, especially the Euler-forward method.

A second possibility for time integration according to Fig. 5.1b is to evaluate $f(x,t)$ at $t = t_{n+1}$ resolving in

$$\int_{t_n}^{t_{n+1}} f(x,t)\,\mathrm{d}t \approx \Delta t \cdot f(x_{n+1}, t_{n+1}) \tag{5.5}$$

and therefore

$$x_{n+1} \approx x_n + \Delta t \cdot f(x_{n+1}, t_{n+1}). \tag{5.6}$$

Since there are now also unknown quantities on the right-hand side as argument of $f(x_{n+1}, t_{n+1})$, the approach is called implicit, especially Euler-backward method.

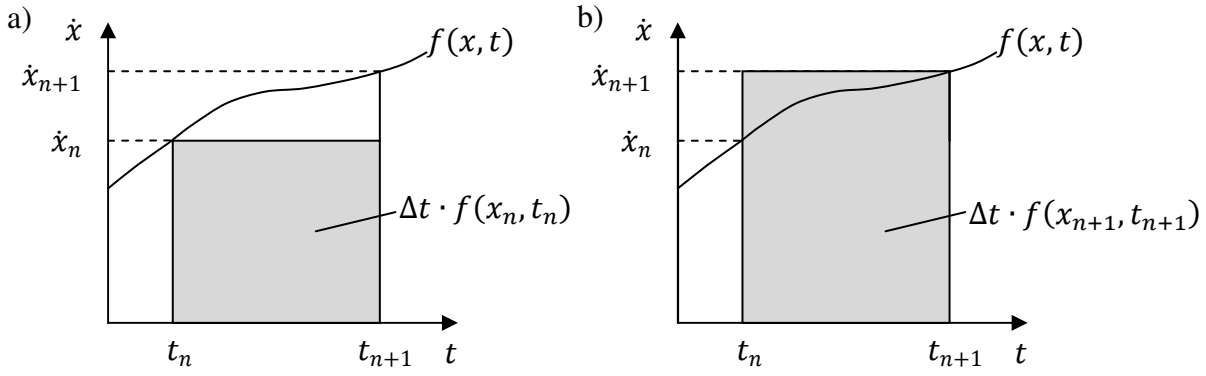


Fig. 5.1 Simple example for explicit (a) and implicit (b) time-integration

At a first glance the explicit approach seems to be more efficient, because both approaches produce a certain integration error (which decreases by using smaller time intervals Δt), but the implicit approach needs a subsequent algorithm for solving Equation (5.6) which in general is nonlinear. The solution of such a nonlinear equation (or system of equations in the general case) can be obtained by using a Newton (often also called Newton-Raphson) or Quasi-Newton algorithm. This procedure is of course very time-consuming, but as we will see later, also implicit time-integration may be very efficient and useful and may have some advantages with respect to stability and the required time-step size.

5.2 Explicit Time-Integration

Explicit time-integration procedures are typically used for highly dynamic FEM problems like impact simulations, crash simulations or simulation of explosions. The explicit Euler-Forward time-integration scheme presented in Section 5.1 is very simple, but has several disadvantages with regard to its accuracy for example, and is therefore not used in practical applications. In general, time-integration algorithms may be classified in one-step and multi-step algorithms. The main difference between both approaches is that the latter do not only use information from the current time-point t_n, but also take into account information from previous time-steps t_{n-i} with $i > 0$.

5.2.1 One-Step Methods

One-step time-integration algorithms are also known as Runge-Kutta methods and use the general approach

$$\boldsymbol{x}_{n+1} = \boldsymbol{x}_n + \Delta t \cdot \boldsymbol{\Phi}(\boldsymbol{x}_n, t_n, \Delta t). \tag{5.7}$$

With specific definitions of the function $\boldsymbol{\Phi}(\boldsymbol{x}_n, t_n, \Delta t)$ different integration algorithms can be constructed. In the simplest case $\boldsymbol{\Phi}(\boldsymbol{x}_n, t_n, \Delta t)$ equals $\boldsymbol{f}(\boldsymbol{x}_n, t_n)$ and one gets the Euler-Forward method (5.4). More accurate higher-order methods can be obtained by developing $\boldsymbol{x}(t) = \int \boldsymbol{f}(\boldsymbol{x}, t)dt$ as a Taylor series at $t = t_n$ as

$$\begin{aligned}\boldsymbol{x}_{n+1} &= \boldsymbol{x}_n + \Delta t \cdot \left[\dot{\boldsymbol{x}}_n + \frac{\Delta t}{2} \cdot \ddot{\boldsymbol{x}}_n + \frac{\Delta t^2}{6}\dddot{\boldsymbol{x}}_n + \ \ldots\right] \\ &= \boldsymbol{x}_n + \Delta t \cdot \left[\boldsymbol{f}(\boldsymbol{x}_n, t_n) + \frac{\Delta t}{2} \cdot \left(\left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}_n, t_n} \cdot \boldsymbol{f}(\boldsymbol{x}_n, t_n) + \left.\frac{\partial \boldsymbol{f}}{\partial t}\right|_{\boldsymbol{x}_n, t_n}\right)\right. \\ &\qquad \left. + \frac{\Delta t^2}{6}\dddot{\boldsymbol{x}}_n + \ \ldots\right]\end{aligned} \tag{5.8}$$

A truncation of this Taylor series after the second term $\Delta t \cdot \dot{\boldsymbol{x}}_n$ obviously leads to the already mentioned Euler-Forward method. Higher-order terms would need higher time-derivatives which, however, are not available from FEM computations. Therefore another and preferred way to obtain higher-order methods is to use the approach

$$\boldsymbol{\Phi}(\boldsymbol{x}_n, t_n, \Delta t) = a_1\boldsymbol{k}_1 + a_2\boldsymbol{k}_2 + \ \ldots + a_l\boldsymbol{k}_l \tag{5.9}$$

with

$$\boldsymbol{k}_1 = \boldsymbol{f}(\boldsymbol{x}_n, t_n) = \dot{\boldsymbol{x}}_n \tag{5.10}$$

$$\boldsymbol{k}_2 = \boldsymbol{f}(\boldsymbol{x}_n + q_{21}\boldsymbol{k}_1\Delta t, t_n + p_2\Delta t) \tag{5.11}$$

$$\vdots$$

$$\boldsymbol{k}_l = \boldsymbol{f}\big(\boldsymbol{x}_n + q_{l,1}\boldsymbol{k}_1\Delta t + q_{l,2}\boldsymbol{k}_2\Delta t + \ \ldots + q_{l,l-1}\boldsymbol{k}_{l-1}\Delta t, t_n + p_l\Delta t\big) \tag{5.12}$$

where the q's and p's are constants which have to be derived by making a Taylor series expansion of Equation (5.7) with respect to Δt and comparing it to the Taylor series expansion (5.8). Again, for $l = 1$ we get the simple Euler-Forward method. Choosing $l = 2$ leads to a second order Runge-Kutta scheme with the unknown constants a_1, a_2, q_{21} and p_2. For the determination of these constants we start with the Taylor series expansion of (5.9) and thus with (5.11). This gives for such a two-variable function

$$\begin{aligned}\boldsymbol{f}(\boldsymbol{x}_n + q_{21}\boldsymbol{k}_1\Delta t, t_n + p_2\Delta t) &= \boldsymbol{f}(\boldsymbol{x}_n, t_n) \\ &\quad + \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}_n, t_n} \cdot q_{21}\boldsymbol{k}_1\Delta t + \left.\frac{\partial \boldsymbol{f}}{\partial t}\right|_{\boldsymbol{x}_n, t_n} \cdot p_2\Delta t + \ \ldots\end{aligned} \tag{5.13}$$

which can be substituted together with (5.10) and (5.9) in (5.7) and leads to

$$\begin{aligned}\boldsymbol{x}_{n+1} &\approx \boldsymbol{x}_n + \Delta t \cdot \Bigg[a_1 \boldsymbol{f}(\boldsymbol{x}_n, t_n) \\ &\qquad + a_2 \left(\boldsymbol{f}(\boldsymbol{x}_n, t_n) + \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}_n, t_n} \cdot q_{21} \boldsymbol{k}_1 \Delta t + \left.\frac{\partial \boldsymbol{f}}{\partial t}\right|_{\boldsymbol{x}_n, t_n} \cdot p_2 \Delta t \right) \Bigg] \\ &= \boldsymbol{x}_n + \Delta t \cdot \Bigg[(a_1 + a_2) \boldsymbol{f}(\boldsymbol{x}_n, t_n) + \Delta t \left(a_2 q_{21} \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}_n, t_n} \cdot \boldsymbol{k}_1 \right. \\ &\qquad \left. + a_2 p_2 \left.\frac{\partial \boldsymbol{f}}{\partial t}\right|_{\boldsymbol{x}_n, t_n} \right) \Bigg].\end{aligned} \tag{5.14}$$

Comparing this with Equation (5.8) yields the equivalence conditions

$$a_1 + a_2 = 1,\ 2a_2 q_{21} = 1,\ 2a_2 p_2 = 1. \tag{5.15}$$

These are three equations to determine four unknowns, which means that one of the unknowns may be chosen arbitrarily. On the other hand, this leads to an infinite variety of second-order Runge-Kutta methods. By using $a_1 = 1/2$ and hence $a_2 = 1/2$, $q_{21} = 1$, $p_2 = 1$ gives the Heun method [62], for example, which is a one-step predictor-corrector scheme [30]. The midpoint method is obtained by choosing $a_1 = 0$, $a_2 = 1$, $q_{21} = p_2 = 1/2$.

The most popular Runge-Kutta method [124,90] uses a fourth-order approach. It can be derived analogously to the presented procedure and finally leads to

$$\boldsymbol{x}_{n+1} = \boldsymbol{x}_n + \frac{\Delta t}{6}(\boldsymbol{k}_1 + 2\boldsymbol{k}_2 + 2\boldsymbol{k}_3 + \boldsymbol{k}_4) \tag{5.16}$$

with

$$\begin{aligned}\boldsymbol{k}_1 &= \boldsymbol{f}(\boldsymbol{x}_n, t_n), \\ \boldsymbol{k}_2 &= \boldsymbol{f}\left(\boldsymbol{x}_n + \frac{\Delta t}{2}\boldsymbol{k}_1, t_n + \frac{\Delta t}{2}\right), \\ \boldsymbol{k}_3 &= \boldsymbol{f}\left(\boldsymbol{x}_n + \frac{\Delta t}{2}\boldsymbol{k}_2, t_n + \frac{\Delta t}{2}\right), \\ \boldsymbol{k}_4 &= \boldsymbol{f}(\boldsymbol{x}_n + \Delta t \boldsymbol{k}_3, t_n + \Delta t).\end{aligned} \tag{5.17}$$

In particular, the presented higher-order one-step methods need more than one function evaluation per time-step, which is especially in the context of FEM methods computationally costly. Lower-order methods like the Euler-Forward scheme are cheap in this sense, but not very accurate (see example in Section 5.4.1). Multi-step methods promise a better performance, because additional information about prior time-steps is taken into account.

5.2.2 Multi-Step Methods (The Central Difference Method)

As an example for a multi-step method we will discuss here only the central difference method, which is widely used in the field of FEM and provides sufficient accuracy combined with reasonable computational costs. In contrast to the procedure used for the one-step methods, we will derive expressions for the current accelerations $\ddot{\boldsymbol{x}}_n$ and velocities $\dot{\boldsymbol{x}}_n$ by using accelerations and velocities of the prior and next time-step. Using these expressions in

the equations of motion (5.1) finally leads to a system of equations which can be easily solved for the displacements of the next time-step $\boldsymbol{x}_{n+1}$. To derive the necessary expressions in the central difference method, firstly the mean slope $\bar{\dot{\boldsymbol{x}}}$ of the function $\boldsymbol{x}(t)$ between the time points t_{n-1} and t_{n+1} is calculated as

$$\bar{\dot{\boldsymbol{x}}}_n := \frac{1}{2\Delta t}(\boldsymbol{x}_{n+1} - \boldsymbol{x}_{n-1}), \tag{5.18}$$

see Fig. 5.2. Then the second derivative of $\boldsymbol{x}(t)$ is determined with the help of the mean slopes $\bar{\dot{\boldsymbol{x}}}_{n-1/2} := (\boldsymbol{x}_n - \boldsymbol{x}_{n-1})/\Delta t$ and $\bar{\dot{\boldsymbol{x}}}_{n+1/2} := (\boldsymbol{x}_{n+1} - \boldsymbol{x}_n)/\Delta t$ as

$$\begin{aligned} \bar{\ddot{\boldsymbol{x}}}_n := \frac{1}{\Delta t}\left(\bar{\dot{\boldsymbol{x}}}_{n+1/2} - \bar{\dot{\boldsymbol{x}}}_{n-1/2}\right) &= \frac{1}{\Delta t}\left(\frac{\boldsymbol{x}_{n+1} - \boldsymbol{x}_n}{\Delta t} - \frac{\boldsymbol{x}_n - \boldsymbol{x}_{n-1}}{\Delta t}\right) \\ &= \frac{1}{\Delta t^2}(\boldsymbol{x}_{n+1} - 2\boldsymbol{x}_n + \boldsymbol{x}_{n-1}). \end{aligned} \tag{5.19}$$

The derived approximations for $\dot{\boldsymbol{x}}_n$ and $\ddot{\boldsymbol{x}}_n$ can be used now in Equation (5.1) for $t = t_n$ which leads to

$$\boldsymbol{M}_n \cdot \frac{1}{\Delta t^2}(\boldsymbol{x}_{n+1} - 2\boldsymbol{x}_n + \boldsymbol{x}_{n-1}) + \boldsymbol{D}_n \cdot \frac{1}{2\Delta t}(\boldsymbol{x}_{n+1} - \boldsymbol{x}_{n-1}) + \boldsymbol{K}_n \cdot \boldsymbol{x}_n = \boldsymbol{F}_n. \tag{5.20}$$

The displacement vector at the next time step t_{n+1} can be determined then from Equation (5.20) by multiplication with Δt^2 and solving for $\boldsymbol{x}_{n+1}$ which yields

$$\begin{aligned} \boldsymbol{x}_{n+1} = &\left(\boldsymbol{M}_n + \frac{1}{2}\Delta t \boldsymbol{D}_n\right)^{-1} \\ &\cdot \left[\Delta t^2 \boldsymbol{F}_n - (\Delta t^2 \boldsymbol{K}_n - 2\boldsymbol{M}_n)\boldsymbol{x}_n - \left(\boldsymbol{M}_n - \frac{1}{2}\Delta t \boldsymbol{D}_n\right)\boldsymbol{x}_{n-1}\right]. \end{aligned} \tag{5.21}$$

With regard to the first part of Equation (5.21) it is obviously necessary to perform a matrix inversion, which in general can be computationally very costly. For this reason it is extremely useful to use a mass and damping matrix which have non-zero values only on their diagonal, in other words which are diagonal matrices. This is for example the case if the lumped mass approach [134,141] is used for modeling the mass distribution, which means that the mass of

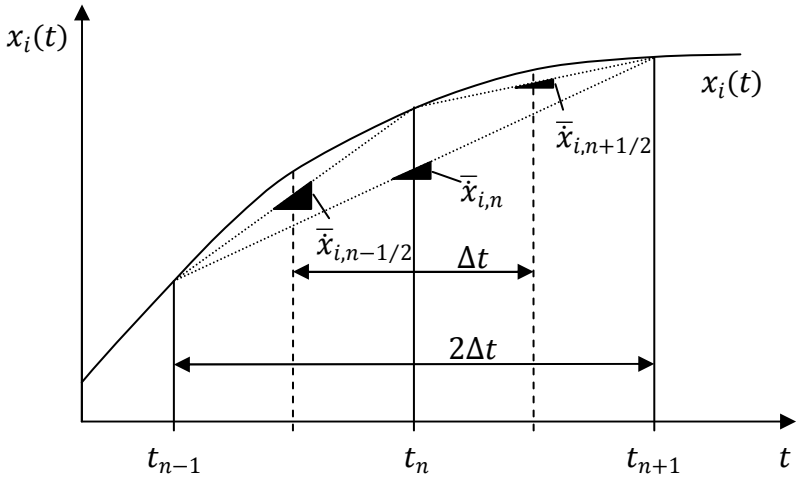


Fig. 5.2 Central difference method

an element is concentrated at its nodes. The damping matrix is a diagonal matrix if mass-proportional damping is applied. In these cases, step two of the general solution procedure for nonlinear systems of differential equations mentioned in the introduction is not necessary, because the solution is directly obtained from Equation (5.21). But for wave propagation problems it can be useful to use a consistent mass matrix, for example. In such an approach the mass matrix is not diagonal and a specific solution procedure is required for (5.21) [66].

If necessary, the velocities $\dot{\boldsymbol{x}}_n$ and accelerations $\ddot{\boldsymbol{x}}_n$ can be computed approximately from Equations (5.18) and (5.19) as $\bar{\dot{\boldsymbol{x}}}_n$ and $\bar{\ddot{\boldsymbol{x}}}_n$ which finishes the actual time-integration step. Subsequently, the procedure continues with the next time step.

As we will see later in Chapter 6, it can be necessary to change the time-step size during the computation for reasons of numerical stability. Especially in the simulation of highly dynamic processes like impact or crash simulations the time-step size may change from step to step. The derived Equations (5.18) and (5.19), however, require an equal time-step size for $\Delta t_{n-1} = t_n - t_{n-1}$ and $\Delta t_n = t_{n+1} - t_n$. If we assume different time-step sizes we get

$$\bar{\dot{\boldsymbol{x}}}_n = \frac{1}{\Delta t_{n-1} + \Delta t_n}(\boldsymbol{x}_{n+1} - \boldsymbol{x}_{n-1}) \tag{5.22}$$

for the mean velocity and

$$\bar{\ddot{\boldsymbol{x}}}_n = \frac{\Delta t_{n-1}(\boldsymbol{x}_{n+1} - \boldsymbol{x}_n) - \Delta t_n(\boldsymbol{x}_n - \boldsymbol{x}_{n-1})}{\frac{\Delta t_n + \Delta t_{n-1}}{2} \cdot \Delta t_n \Delta t_{n-1}} \tag{5.23}$$

for the mean acceleration.

5.3 Implicit Time-Integration

Implicit time-integration methods are mostly used for static or quasi-static problems. But as we will see later they can be also very useful for dynamic applications. In contrast to explicit time-integration it is necessary to solve a system of in general non-linear equations for every time-step after substituting the unknown velocities and accelerations with the respective expressions. By this, the solution procedure for one time-step is computationally far more costly than an explicit solution procedure. On the other hand, the time-step size in an implicit time-integration can be a few orders bigger than that of an explicit time-integration method, which can lead to an overall smaller computational time depending on the particular problem. In this section, some of the most common algorithms are presented which are all used in commercial finite element codes.

5.3.1 Newmark Algorithm

The Newmark time-integration algorithm has been originally developed by Nathan M. Newmark [110]. Algorithms of the Newmark family or generalizations of it are implemented in almost every commercial finite element code.

The basic concept of the Newmark algorithm is to use a linear approximation between two time-points t_n and t_{n+1}. If we firstly consider the integration of $\ddot{x}(t)$ (Fig. 5.3) the Newmark approximation is

$$\ddot{\boldsymbol{x}}^h(t) = \frac{\ddot{\boldsymbol{x}}_{n+1} - \ddot{\boldsymbol{x}}_n}{t_{n+1} - t_n} \cdot t + \frac{\ddot{\boldsymbol{x}}_n t_{n+1} - \ddot{\boldsymbol{x}}_{n+1} t_n}{t_{n+1} - t_n}. \tag{5.24}$$

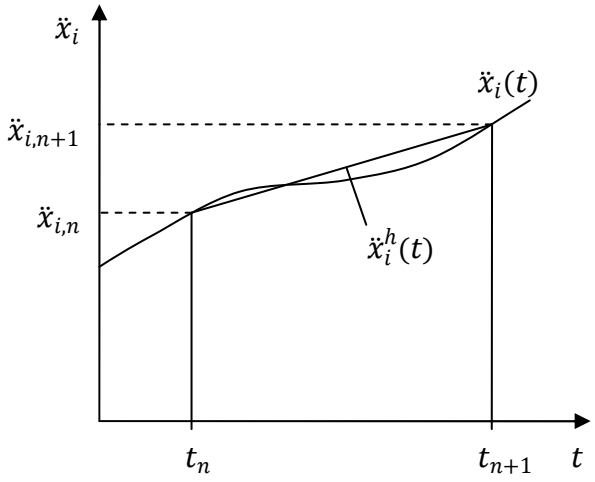


Fig. 5.3 Linear approximation of the Newmark algorithm

For the integration of (5.24) Newmark introduced the following equation:

$$\begin{aligned} \dot{\boldsymbol{x}}^h(t_{n+1}) &= \dot{\boldsymbol{x}}_n + \int_{t_n}^{t_{n+1}} \ddot{\boldsymbol{x}}^h(t)\mathrm{d}t \\ &=: \dot{\boldsymbol{x}}_n + (t_{n+1} - t_n) \cdot [(1-\gamma)\ddot{\boldsymbol{x}}_n + \gamma \ddot{\boldsymbol{x}}_{n+1}], \;\; 0 \le \gamma \le 1. \end{aligned} \tag{5.25}$$

By varying the parameter γ between 0 and 1 the accuracy of the integration for the linear approximation can be influenced. The meaning of different γ-values is illustrated in Fig. 5.4. The areas highlighted in grey show the results of the integration for different γ-values. Obviously, for the linear Newmark approximation the integration is exact for $\gamma = 1/2$. For $\gamma > 1/2$ so-called numerical damping is introduced into the system and for $\gamma < 1/2$ the system is excited.

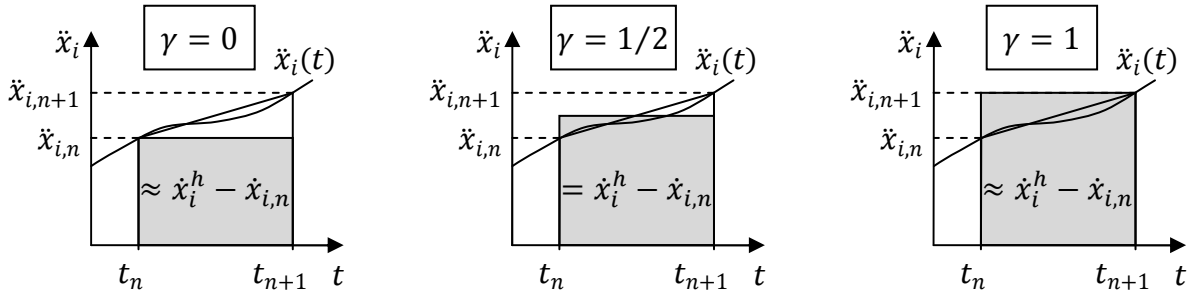


Fig. 5.4 Influence of γ-parameter on integration result

For the second integration of $\dot{\boldsymbol{x}}^h(t)$, the approximation

$$\begin{aligned}\boldsymbol{x}^h(t_{n+1}) &= \boldsymbol{x}_n + \int_{t_n}^{t_{n+1}} \dot{\boldsymbol{x}}^h(t)\mathrm{d}t \\ &=: \boldsymbol{x}_n + (t_{n+1}-t_n)\dot{\boldsymbol{x}}_n + (t_{n+1}-t_n)^2\left[\left(\frac{1}{2}-\beta\right)\ddot{\boldsymbol{x}}_n + \beta\ddot{\boldsymbol{x}}_{n+1}\right]\end{aligned} \tag{5.26}$$

is used. To understand the meaning of the parameter β, it is useful to have a closer look at the different terms in Equation (5.26). The term $(t_{n+1}-t_n)\dot{\boldsymbol{x}}_n$ just describes the area of the rectangle highlighted in Fig. 5.5 (left). The third term for $\beta = 0$ describes the area of a triangle with slope $\ddot{\boldsymbol{x}}_n$ (see Fig. 5.5, right). The parameter β is used as a kind of weighting factor to adopt the slope of this triangle. For $\beta = 0$ the slope at $t = t_n$ is used, for $\beta = 1/2$ the slope at $t = t_{n+1}$ is evaluated. For $\beta = 1/4$ the mean value of both slopes is used for the computation of the triangular area. As it can be seen from these considerations a sensible choice for β should be $0 \le \beta \le 1/2$. Nevertheless, there are some further important restrictions concerning the choice of β with respect to stability requirements as we will see later.

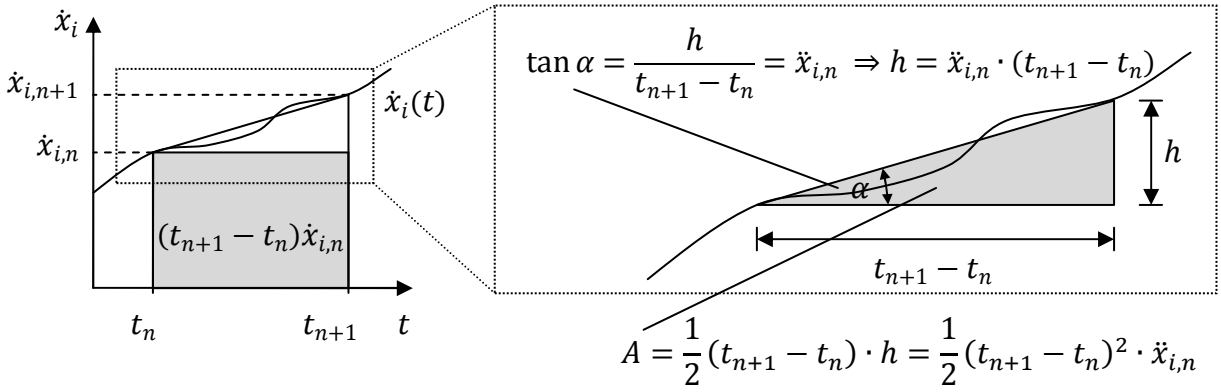


Fig. 5.5 Meaning of second Newmark equation for $\beta = 0$

If we substitute $\dot{\boldsymbol{x}}^h(t_{n+1})$ by $\dot{\boldsymbol{x}}_{n+1}$ in Equation (5.25) and $\boldsymbol{x}^h(t_{n+1})$ by $\boldsymbol{x}_{n+1}$ in Equation (5.26) in the sense of the Newmark approximation we can solve these equations for the next iterates $\dot{\boldsymbol{x}}_{n+1}$ and $\ddot{\boldsymbol{x}}_{n+1}$ to get

$$\dot{\boldsymbol{x}}_{n+1} = \frac{2\beta\Delta t(\dot{\boldsymbol{x}}_n + \Delta t\ddot{\boldsymbol{x}}_n) - \gamma(2\boldsymbol{x}_n - 2\boldsymbol{x}_{n+1} + 2\Delta t\dot{\boldsymbol{x}}_n + \Delta t^2\ddot{\boldsymbol{x}}_n)}{2\beta\Delta t} \tag{5.27}$$

and

$$\ddot{\boldsymbol{x}}_{n+1} = \frac{-2\boldsymbol{x}_n + 2\boldsymbol{x}_{n+1} + \Delta t(-2\dot{\boldsymbol{x}}_n + (2\beta-1)\Delta t\ddot{\boldsymbol{x}}_n)}{2\beta\Delta t^2}. \tag{5.28}$$

where $\Delta t = t_{n+1} - t_n$. Equations (5.27) and (5.28) can now be used in Equation (5.1) by substituting $\ddot{\boldsymbol{x}}$ by Equation (5.28), $\dot{\boldsymbol{x}}$ by Equation (5.27) and $\boldsymbol{x}$ by $\boldsymbol{x}_{n+1}$, which finally can be solved for the next displacement iterate $\boldsymbol{x}_{n+1}$ as described in Section 5.3.4.

5.3.2 Hilber-Hughes-Taylor Algorithm

The Hilber-Hughes-Taylor (HHT) algorithm is a kind of generalization of the classical Newmark method by introducing an additional variable α into the equation of motion [63] which then becomes

$$\boldsymbol{M}_{n+1}\ddot{\boldsymbol{x}}_{n+1} + (1+\alpha)\boldsymbol{D}_{n+1}\dot{\boldsymbol{x}}_{n+1} - \alpha\boldsymbol{D}_n\dot{\boldsymbol{x}}_n + (1+\alpha)\boldsymbol{K}_{n+1}\boldsymbol{x}_{n+1} - \alpha\boldsymbol{K}_n\boldsymbol{x}_n = \boldsymbol{F}_{n+1}. \tag{5.29}$$

For the substitutions of $\ddot{\boldsymbol{x}}_{n+1}$ and $\dot{\boldsymbol{x}}_{n+1}$ the same expressions (5.27) and (5.28) as in Newmark's algorithm are used. By this, the Newmark algorithm and the HHT algorithm are identical for $\alpha = 0$.

Hilber, Hughes and Taylor suggest the parameter coupling

$$\gamma = \frac{1}{2} - \alpha, \tag{5.30}$$

$$\beta = \frac{(1-\alpha)^2}{4} \tag{5.31}$$

for $-1/3 \leq \alpha \leq 0$. Choosing $\alpha < 0$ adds some numerical damping to the system as will be demonstrated later for an example problem. In contrast to the Newmark time-integration, the second order accuracy of the HHT algorithm is not affected by introducing numerical damping [56]. In addition, there exist some generalizations of this method as described in [32,41] for example.

5.3.3 Newmark-Euler Algorithm

The correct name of this integration scheme is actually Newmark-Three-Point-Euler-Backward algorithm, but for reasons of simplicity it will be termed here as Newmark-Euler algorithm. The method was originally published in [6] and used for the numerical solution of coupled partial differential equations describing the transient behavior of silicon device structures. Bathe et al. [11,13,16,127] introduced the scheme for the implicit solution of dynamic FEM problems.

The basic idea of the Newmark-Euler algorithm is to split the time step Δt into two time-steps. At the moment we may assume that the position of the split point is in the middle of Δt. The unknown solution function $\boldsymbol{x}(t)$ is approximated at the time points t_n and $t_n + \Delta t/2$ by two second-order Taylor series derived at t_{n+1} which gives according to Fig. 5.6

$$\boldsymbol{x}_{n+1/2} \approx \widetilde{\boldsymbol{x}}_{n+1/2} = \boldsymbol{x}_{n+1} - \frac{\Delta t}{2}\dot{\boldsymbol{x}}_{n+1} - \frac{\Delta t^2}{8}\ddot{\boldsymbol{x}}_{n+1} \tag{5.32}$$

(chain-dotted line) and

$$\boldsymbol{x}_n \approx \widetilde{\boldsymbol{x}}_n = \boldsymbol{x}_{n+1} - \Delta t \cdot \dot{\boldsymbol{x}}_{n+1} - \frac{\Delta t^2}{2}\ddot{\boldsymbol{x}}_{n+1} \tag{5.33}$$

(dotted line). If Equation (5.32) is multiplied by (-4) and added to Equation (5.33) we get

$$\widetilde{\boldsymbol{x}}_n - 4\widetilde{\boldsymbol{x}}_{n+1/2} = -3\boldsymbol{x}_{n+1} + \Delta t \cdot \dot{\boldsymbol{x}}_{n+1} \tag{5.34}$$

which can be solved for $\dot{\boldsymbol{x}}_{n+1}$ and results in

$$\dot{\boldsymbol{x}}_{n+1} = \frac{\widetilde{\boldsymbol{x}}_n - 4\widetilde{\boldsymbol{x}}_{n+1/2} + 3\boldsymbol{x}_{n+1}}{\Delta t} \approx \frac{\boldsymbol{x}_n - 4\boldsymbol{x}_{n+1/2} + 3\boldsymbol{x}_{n+1}}{\Delta t}. \tag{5.35}$$

The same procedure for $\dot{\boldsymbol{x}}(t)$ leads to

$$\ddot{\boldsymbol{x}}_{n+1} = \frac{\widetilde{\dot{\boldsymbol{x}}}_n - 4\widetilde{\dot{\boldsymbol{x}}}_{n+1/2} + 3\dot{\boldsymbol{x}}_{n+1}}{\Delta t} \approx \frac{\dot{\boldsymbol{x}}_n - 4\dot{\boldsymbol{x}}_{n+1/2} + 3\dot{\boldsymbol{x}}_{n+1}}{\Delta t}. \tag{5.36}$$

The unknown values for $\boldsymbol{x}_{n+1/2}$ and $\dot{\boldsymbol{x}}_{n+1/2}$ in Equations (5.35) and (5.36) are provided by a Newmark algorithm which is used for the integration from t_n to $t_n + \Delta t/2$. This means that the Newmak-Euler scheme is a two-step time-integration scheme. In the first step the classical Newmark substitutions (5.27), (5.28) with $\Delta t \to \Delta t/2$, $n+1 \to n+1/2$ are applied to the equations of motion and yield

$$\begin{aligned} &\boldsymbol{M}_{n+1/2} \cdot \frac{-4\boldsymbol{x}_n + 4\boldsymbol{x}_{n+1/2} + \Delta t\left(-2\dot{\boldsymbol{x}}_n + (2\beta - 1)\frac{\Delta t}{2}\ddot{\boldsymbol{x}}_n\right)}{\beta \Delta t^2} \\ &\quad + \boldsymbol{D}_{n+1/2} \cdot \frac{\beta \Delta t\left(\dot{\boldsymbol{x}}_n + \frac{\Delta t}{2}\ddot{\boldsymbol{x}}_n\right) - \gamma\left(2\boldsymbol{x}_n - 2\boldsymbol{x}_{n+1/2} + \Delta t\dot{\boldsymbol{x}}_n + \frac{\Delta t^2}{4}\ddot{\boldsymbol{x}}_n\right)}{\beta \Delta t} \\ &\quad + \boldsymbol{K}_{n+1/2} \cdot \boldsymbol{x}_{n+1/2} = \boldsymbol{F}_{n+1/2}. \end{aligned} \tag{5.37}$$

This generally nonlinear system of equations is solved for $\boldsymbol{x}_{n+1/2}$ by a numerical solution scheme like a Newton or Quasi-Newton algorithm. The solution for $\dot{\boldsymbol{x}}_{n+1/2}$ is obtained via Equation (5.27) by substitutions $\boldsymbol{x}_{n+1} \to \boldsymbol{x}_{n+1/2}$, $\dot{\boldsymbol{x}}_{n+1} \to \dot{\boldsymbol{x}}_{n+1/2}$, $\Delta t \to \Delta t/2$. In the second step of the Newmark-Euler time-integration scheme, the right part of Equations (5.35) and (5.36) are applied to the equations of motions for the integration over the whole time step Δt which leads to

$$\begin{aligned} \boldsymbol{M}_{n+1} \cdot \frac{\dot{\boldsymbol{x}}_n - 4\dot{\boldsymbol{x}}_{n+1/2} + 3\dot{\boldsymbol{x}}_{n+1}}{\Delta t} + \boldsymbol{D}_{n+1} \cdot \frac{\boldsymbol{x}_n - 4\boldsymbol{x}_{n+1/2} + 3\boldsymbol{x}_{n+1}}{\Delta t} \\ + \boldsymbol{K}_{n+1} \cdot \boldsymbol{x}_{n+1} = \boldsymbol{F}_{n+1}. \end{aligned} \tag{5.38}$$

Also Equation (5.38) is nonlinear in general due to the dependency of the stiffness matrix on $\boldsymbol{x}_{n+1}$ for example and has to be solved numerically for the next iterate $\boldsymbol{x}_{n+1}$.

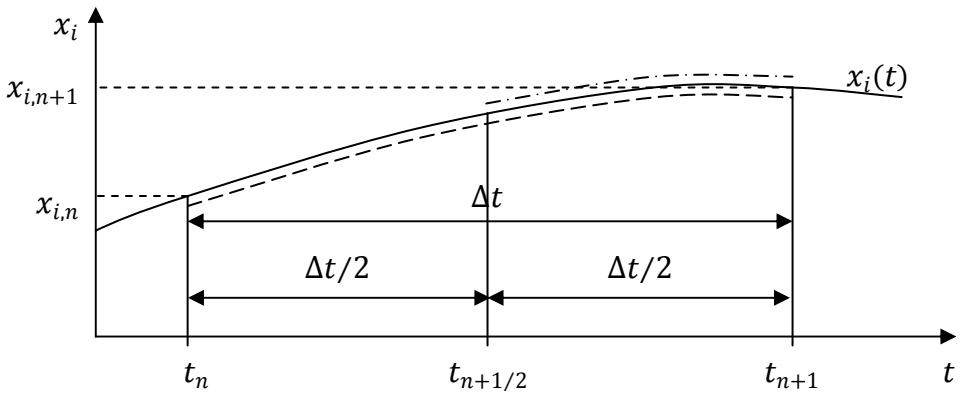


Fig. 5.6 Time-step split in Newmark-Euler algorithm

Alternatively, it is possible to split the time step Δt not at $\Delta t/2$ but at any arbitrary point between t_n and t_{n+1}. As we will see later, the time step split point has an influence on the numerical damping of the algorithm and therefore also on the stability.

A more general implementation of the Newmark-Euler scheme uses alternating Newmark and Euler-Backward steps (Composite scheme). A Newmark step is always followed by an Euler step which uses the information of the previous Newmark step and the Euler step before. The substitutions for the Euler step are

$$\ddot{\boldsymbol{x}}_{n+1} = \frac{1+\alpha}{\Delta t_n}(\dot{\boldsymbol{x}}_{n+1} - \dot{\boldsymbol{x}}_n) - \frac{\alpha}{\Delta t_{n-1}}(\dot{\boldsymbol{x}}_n - \dot{\boldsymbol{x}}_{n-1}) \tag{5.39}$$

and

$$\dot{\boldsymbol{x}}_{n+1} = \frac{1+\alpha}{\Delta t_n}(\boldsymbol{x}_{n+1} - \boldsymbol{x}_n) - \frac{\alpha}{\Delta t_{n-1}}(\boldsymbol{x}_n - \boldsymbol{x}_{n-1}). \tag{5.40}$$

If the parameter α is interpreted as split point between the previous and current time step, in other words $\alpha = (t_n - t_{n-1})/(t_{n+1} - t_{n-1})$, it can be easily demonstrated that e.g. Equation (5.40) becomes identical to Equation (5.35) for $\alpha = 0.5$: The substitution of $\alpha = (t_n - t_{n-1})/(t_{n+1} - t_{n-1}) = (\Delta t_{n-1})/(\Delta t_{n-1} + \Delta t_n)$ into Equation (5.40) leads to

$$\dot{\boldsymbol{x}}_{n+1} = \frac{1}{\Delta t_{n-1} + \Delta t_n}\left(\frac{2\Delta t_{n-1} + \Delta t_n}{\Delta t_n}(\boldsymbol{x}_{n+1} - \boldsymbol{x}_n) - \boldsymbol{x}_n + \boldsymbol{x}_{n-1}\right). \tag{5.41}$$

The choice of $\alpha = 0.5$ means $\Delta t_{n-1} = \Delta t_n$ and therefore $\Delta t_{n-1} + \Delta t_n = 2\Delta t$, which gives

$$\dot{\boldsymbol{x}}_{n+1} = \frac{(\boldsymbol{x}_{n-1} - 4\boldsymbol{x}_n + 3\boldsymbol{x}_{n+1})}{2\Delta t}. \tag{5.42}$$

and is identical to Equation (5.35) if $2\Delta t$ is interpreted as Δt and therefore $\boldsymbol{x}_{n-1}$ as $\boldsymbol{x}_n$ and $\boldsymbol{x}_n$ as $\boldsymbol{x}_{n+1/2}$. But α can also be interpreted as a further independent parameter of this scheme. The latter approach has the advantage that the size of two subsequent time steps Δt_{n-1} and Δt_n is not directly coupled via α, which can be advantageous if convergence problems appear. For a strict coupling both the Newmark and the Euler-Backward step have to be repeated if convergence is not reached in the Euler-Backward step, which is not necessary if α is not coupled to the step-sizes. In this case it would be sufficient to repeat the last time-step only.

5.3.4 Solution of Nonlinear Equations

Using implicit time-integrators typically leads to nonlinear systems of equations such as (5.37) or (5.38), which have to be solved for the unknown values of the next time-step. But also explicit time-integration schemes may produce nonlinear systems of equations, for example if a non-diagonal damping matrix is required [70,71]. For the solution in FEM applications, the classical Newton method and Quasi-Newton methods are widely used. The roots of these methods date back to a work of Isaac Newton, called "*Methodus fluxionum et serierum infinitarum*", authored between 1664 and 1671, where he introduces a new algorithm to solve a polynomial equation. In 1690, Joseph Raphson formalized and illustrated

this algorithm in his work "*Analysis Aequationum universalis*". Therefore the method is often called "Newton-Raphson" method. The abstract form of the iteration rule

$$ {}^{k+1}x = {}^{k}x - \frac{f({}^{k}x)}{f'({}^{k}x)}, \quad k = 0, 1, 2, \ldots \tag{5.43} $$

was probably derived by the English mathematician Thomas Simpson, who is also known for his work about numerical integration. Some more historical remarks with further literature references can be found in [46].

To derive Equation (5.43), we assume that we want to solve the nonlinear algebraic equation

$$ f(x) = 0. \tag{5.44} $$

In Fig. 5.7 an example of such a function is plotted. The iteration starts at the initial value $x = {}^{0}x$. At this point, the nonlinear function $f(x)$ is linearized by computing the derivative $f'({}^{0}x)$ and constructing a linear function

$$ f_0(x) = f'({}^{0}x)x + C. \tag{5.45} $$

The unknown variable C in Equation (5.45) can be determined from

$$ f({}^{0}x) = f'({}^{0}x){}^{0}x + C \tag{5.46} $$

as

$$ C = f({}^{0}x) - f'({}^{0}x){}^{0}x. \tag{5.47} $$

The idea of the Newton algorithm is to compute the zero of this linear function which should be a better approximation of the zero of the nonlinear function than the initial guess ${}^{0}x$ from

$$ f_0(x) = f'({}^{0}x)x + f({}^{0}x) - f'({}^{0}x){}^{0}x = 0 \tag{5.48} $$

resolving in

$$ x = \frac{f'({}^{0}x){}^{0}x - f({}^{0}x)}{f'({}^{0}x)} = {}^{0}x - \frac{f({}^{0}x)}{f'({}^{0}x)}. \tag{5.49} $$

Obviously this result is different from the zero of the nonlinear function $f(x)$, but the error is

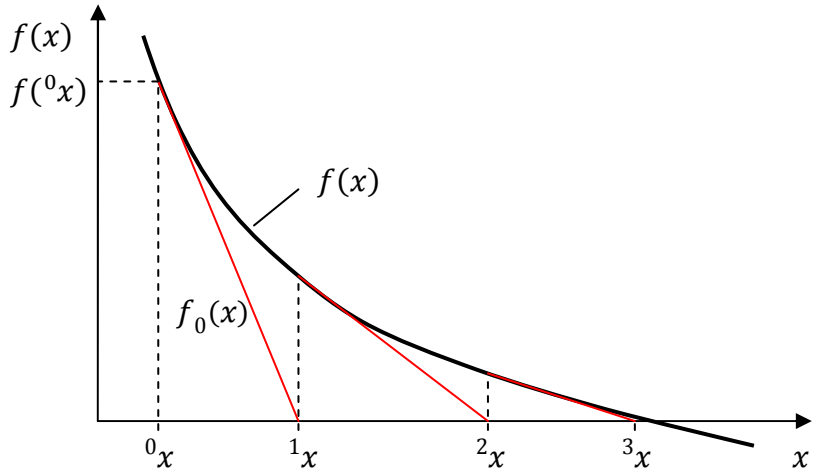


Fig. 5.7 Solving a nonlinear equation by applying a sequence of linear models (Newton algorithm)

smaller than for $x = {}^0x$ and it can be used as a starting point ${}^1x$ for the next iteration. By this, the iteration rule (5.43) is applied until the resulting error $f({}^{k+1}x)$ is smaller than some tolerance ε, i.e., $f({}^{k+1}x) < \varepsilon$.

Let us now consider the case of a mechanical problem with more than just one unknown, where Newmark time-integration is applied. Substitution of (5.27), (5.28) into Equation (5.1) yields the function $\boldsymbol{f}$ at a certain time-point t_{n+1}

$$\begin{aligned}\boldsymbol{f}(\boldsymbol{x}_{n+1}) = \boldsymbol{M}_{n+1} \cdot \frac{-2\boldsymbol{x}_n + 2\boldsymbol{x}_{n+1} + \Delta t(-2\dot{\boldsymbol{x}}_n + (2\beta - 1)\Delta t\ddot{\boldsymbol{x}}_n)}{2\beta\Delta t^2} \\ + \boldsymbol{D}_{n+1} \cdot \frac{2\beta\Delta t(\dot{\boldsymbol{x}}_n + \Delta t\ddot{\boldsymbol{x}}_n) - \gamma(2\boldsymbol{x}_n - 2\boldsymbol{x}_{n+1} + 2\Delta t\dot{\boldsymbol{x}}_n + \Delta t^2\ddot{\boldsymbol{x}}_n)}{2\beta\Delta t} \\ + \boldsymbol{K}_{n+1} \cdot \boldsymbol{x}_{n+1} - \boldsymbol{F}_{n+1}.\end{aligned} \tag{5.50}$$

Please note that also the stiffness matrix $\boldsymbol{K}_{n+1}$, the damping matrix $\boldsymbol{D}_{n+1}$ and the load vector $\boldsymbol{F}_{n+1}$ may depend on the unknown displacements $\boldsymbol{x}_{n+1}$. It should be mentioned that Equation (5.50) is often formulated in terms of internal and external loads as

$$\boldsymbol{f}(\boldsymbol{x}_{n+1}) = \boldsymbol{M}_n\ddot{\boldsymbol{x}}_{n+1} + \boldsymbol{f}_{n+1}^{int} - \boldsymbol{f}_{n+1}^{ext} \tag{5.51}$$

with $\boldsymbol{f}_{n+1}^{int} = \boldsymbol{D}_n\dot{\boldsymbol{x}}_{n+1} + \boldsymbol{K}_n\boldsymbol{x}_{n+1}$ and $\boldsymbol{f}_{n+1}^{ext} = \boldsymbol{F}_{n+1}$.

By using the iteration rule (5.43) for finding the zero of (5.50) we get

$${}^{k+1}\boldsymbol{x}_{n+1} = {}^k\boldsymbol{x}_{n+1} - \left(\left.\frac{\partial\boldsymbol{f}(\boldsymbol{x}_{n+1})}{\partial\boldsymbol{x}_{n+1}}\right|_{{}^k\boldsymbol{x}_{n+1}}\right)^{-1} \boldsymbol{f}({}^k\boldsymbol{x}_{n+1}). \tag{5.52}$$

With $\Delta\boldsymbol{x} = {}^{k+1}\boldsymbol{x}_{n+1} - {}^k\boldsymbol{x}_{n+1}$ this can also be written as

$$\left(\left.\frac{\partial\boldsymbol{f}(\boldsymbol{x}_{n+1})}{\partial\boldsymbol{x}_{n+1}}\right|_{{}^k\boldsymbol{x}_{n+1}}\right)\Delta\boldsymbol{x} = -\boldsymbol{f}({}^k\boldsymbol{x}_{n+1}) \tag{5.53}$$

which is a linear system of equations. As initial value for the Newton iteration the equilibrium solution of the previous time-step ${}^0\boldsymbol{x}_{n+1} = \boldsymbol{x}_n$ may be used. The derivative in Equation (5.53) is called the "effective tangent stiffness matrix" [21].

In many cases the effectiveness of the Newton method can be increased by combining it with a line search algorithm. The idea behind this approach is that the $\Delta\boldsymbol{x}$ found by the Newton method may be a good direction, but the step size could be too big, which is illustrated in Fig. 5.8. It would be obviously advantageous to use a smaller step size, because the residuum ε is smaller when using $\Delta x = \xi \cdot ({}^{k+1}x - {}^kx)$ with $\xi = 0.6$ instead of $\Delta x = {}^{k+1}x - {}^kx$. The difficulty is to determine an optimal value for ξ, where a lot of different strategies exist for this purpose (see for example [51,111]).

For more than one unknown, various vector norms are applicable for the convergence criterion of the Newton method [21]. One possibility is to use the Euclidean norm

$$\|\boldsymbol{a}\|_2 = \left(\sum_{i=1}^{NDOF} a_i^2\right)^{\frac{1}{2}}, \tag{5.54}$$

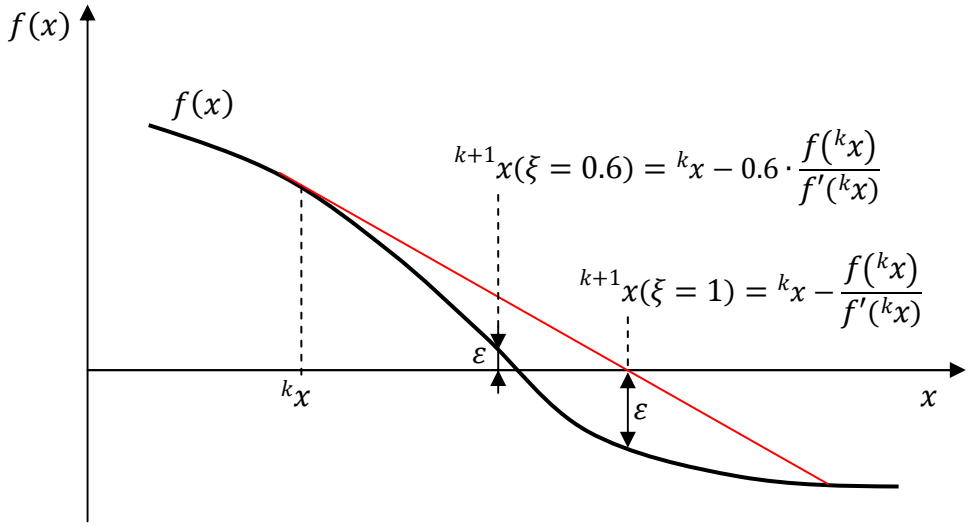


Fig. 5.8 Increasing the effectiveness of the Newton method by line search

which just describes the length of the vector $\boldsymbol{a}$. This norm can then be applied to the residual $\boldsymbol{f}({}^{k+1}\boldsymbol{x}_{n+1})$ as

$$\|\boldsymbol{f}({}^{k+1}\boldsymbol{x}_{n+1})\|_2 \leq \varepsilon \cdot \max\left\{\|\boldsymbol{f}^{ext}\|_2, \left\|\boldsymbol{f}^{int}\right\|_2, \|\boldsymbol{M}\ddot{\boldsymbol{x}}\|_2\right\} \tag{5.55}$$

or to the displacement increment $\Delta\boldsymbol{x}$ as

$$\|{}^{k+1}\Delta\boldsymbol{x}\|_2 \leq \varepsilon \cdot \|\boldsymbol{x}\|_2. \tag{5.56}$$

Another possibility would be to use the maximum norm

$$\|\boldsymbol{a}\|_\infty = \max|a_i| \tag{5.57}$$

which describes the maximum error instead of the mean error over all degrees of freedom when using the Euclidean norm. Also energy convergence criteria may be useful [21,22].

The computation of the derivative $\partial\boldsymbol{f}(\boldsymbol{x}_{n+1})/\partial\boldsymbol{x}_{n+1}$ at $\boldsymbol{x}_{n+1} = {}^{k}\boldsymbol{x}_{n+1}$, which has to be performed at every k-th step during the Newton algorithm, is computationally very expensive. Hence, so-called Quasi-Newton algorithms have been developed using approximations of the effective tangent stiffness matrix. The simplest way to construct a Quasi-Newton algorithm is just to use the derivative $f'({}^{0}x)$ of the first iteration for all following iterations. This approach is illustrated in Fig. 5.9b. Obviously this simple idea also converges but needs more iterations than a classical Newton algorithm sometimes called Full-Newton algorithm (Fig. 5.9a).

More advanced Quasi-Newton methods use secant approximations for the effective tangent stiffness matrix. This can be done by defining

$${}^{k}\delta = {}^{k}x - {}^{k-1}x \tag{5.58}$$

and

$${}^{k}\gamma = f({}^{k}x) - f({}^{k-1}x). \tag{5.59}$$

The approximation for $f'({}^{k}x)$ or $1/f'({}^{k}x)$ can then be calculated as

$$\frac{1}{f'({}^{k}x)} \approx \frac{{}^{k}\delta}{{}^{k}\gamma}. \tag{5.60}$$

As it can be seen in Fig. 5.10 the convergence of such an approach with updates of the derivative is faster than without any updates.

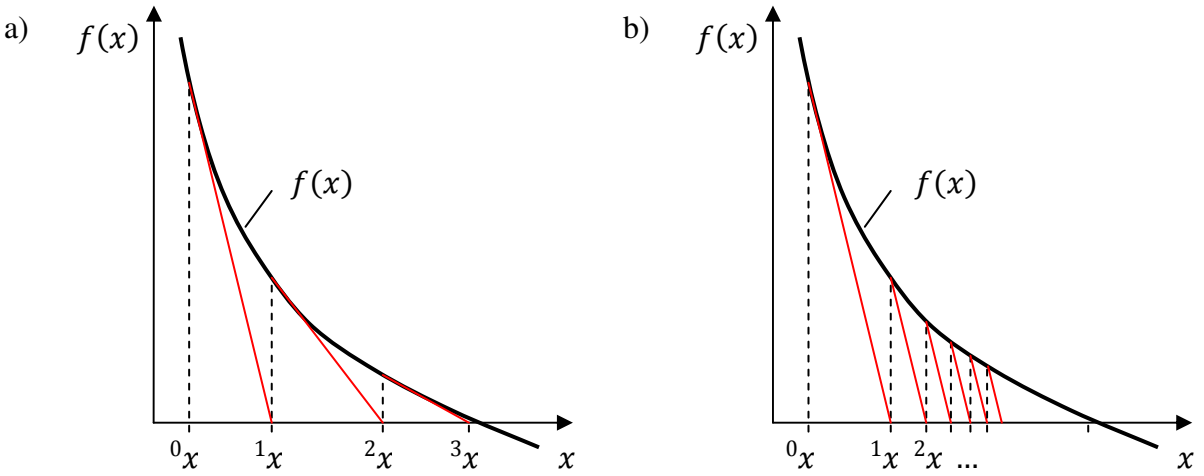


Fig. 5.9 Full-Newton algorithm (a) and Quasi-Newton algorithm (b)

An even more effective and widely used algorithm is the Broyden-Fletcher-Goldfarb-Shanno method (BFGS method), which was developed by contributions from the four mathematicians Charles Broyden, Roger Fletcher, Donald Goldfarb and David Shanno. In their approach a line search algorithm is combined with the secant approximation procedure [26,52,58,125]. Another common Quasi-Newton algorithm is the Davidon-Fletcher-Powell scheme [44,53]. It is not possible to rate the presented methods in general in terms of effectiveness or computational speed. The performance of a method is always problem dependent and may change with the size of the problem, the smoothness and the behavior of the function $f(x)$. Experience with the FEM simulation of fast rotating elastic structures shows for instance that for these kinds of problems a classical Newton algorithm converges faster and provides more stability than a Quasi-Newton approach.

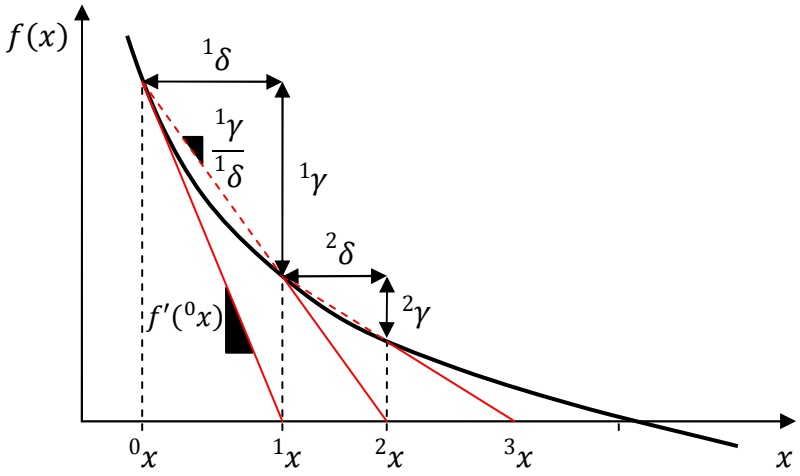


Fig. 5.10 Quasi-Newton algorithm with secant approximations of the derivative

Moreover, Quasi-Newton methods need more computer memory for the storage of additional vectors and matrices required for the approximation of the derivative. Test cases in

commercial FE tools showed that the memory allocation is about 30% higher if a Quasi-Newton method is used instead of a Full-Newton method. If this additional memory requirement leads to a situation where the available memory is insufficient to perform the whole computation "in-core", which means without using disk space for temporary swapping of data, the overall computational time may increase dramatically.

5.4 Examples

In this section, the presented algorithms and methods and especially their behavior will be demonstrated by some simple examples. By this, advantages and disadvantages of the algorithms should become more obvious.

5.4.1 Time Integration of Equation of Motion for Moving Plate

Let us first consider a very simple example which actually can be solved analytically. A rectangular plate is meshed with just one solid element. It is assumed to be rigid and the mass is $m = 4 \cdot 10^5 kg$. The plate is loaded at every node with the same constant force which sum up to $F = 8 \cdot 10^7 N$ (Fig. 5.11). At $t = 0s$ the plate is at rest and we are looking for the displacement of the plate after $1s$. For the solution of this problem we will compare the implicit Newmark time-integration and the explicit Euler-Forward method.

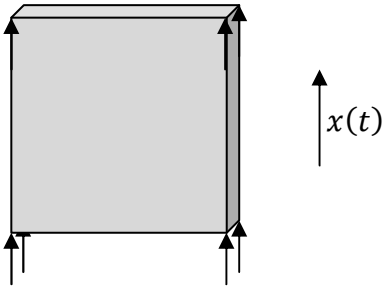


Fig. 5.11 Moving plate

The equation of motion for the described problem is just

$$m\ddot{x} = F \tag{5.61}$$

with initial conditions $x_0 = 0$, $\dot{x}_0 = 0$. If we firstly apply the Newmark time-integration (5.28), Equation (5.61) becomes

$$m \cdot \frac{-2x_n + 2x_{n+1} + \Delta t(-2\dot{x}_n + (2\beta - 1)\Delta t\ddot{x}_n)}{2\beta\Delta t^2} = F \tag{5.62}$$

which is a linear equation in x_{n+1}. Therefore, it can be directly solved without Newton iteration to get

$$x_{n+1} = \frac{F\beta\Delta t^2}{m} + x_n - \Delta t\left(-\dot{x}_n + \left(\beta - \frac{1}{2}\right)\Delta t\ddot{x}_n\right). \tag{5.63}$$

Table 5.1 shows the values for the recursive procedure from $t = 0s$ to $t = 1s$ with constant time-step size $\Delta t = 0.1s$ and standard Newmark parameters $\gamma = 0.5$ and $\beta = 0.25$. In Fig.

5.12 the resulting curves for displacement, velocity and acceleration are plotted. The values for velocity and acceleration are computed with Equations (5.27) and (5.28), respectively.

Table 5.1 Results of Newmark algorithm for $\ddot{x}_0 = 0\,m/s^2$

$t\ [s]$	n	$x_n\ [m]$	$\dot{x}_n\ [m/s]$	$\ddot{x}_n\ [m/s^2]$
0	0	0	0	0
0.1	1	0.5	10	200
0.2	2	2.5	30	200
0.3	3	6.5	50	200
0.4	4	12.5	70	200
0.5	5	20.5	90	200
0.6	6	30.5	110	200
0.7	7	42.5	130	200
0.8	8	56.5	150	200
0.9	9	72.5	170	200
1.0	10	90.5	190	200

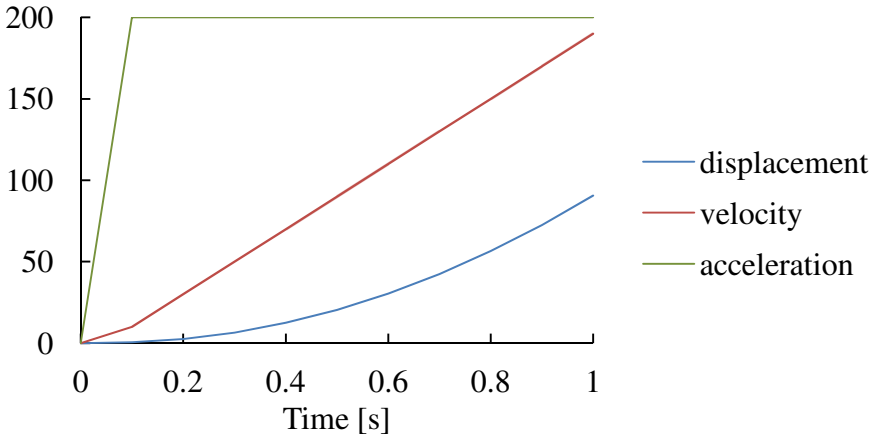


Fig. 5.12 Diagram of x, $\dot{x}$ and $\ddot{x}$ for results of Newmark algorithm for $\ddot{x}_0 = 0\,m/s^2$

If we compare the computed solution for the maximum displacement with the analytical solution

$$x(t = 1s) = \frac{\ddot{x}}{2}t^2 + \dot{x}_0 t + x_0 = \frac{Ft^2}{2m} = 100m \tag{5.64}$$

we observe a certain deviation with respect to Table 5.1. This error becomes bigger for bigger time-steps and vice versa. The deviation results from the acceleration value used for the initialization of the Newmark algorithm. For the calculation of Table 5.1 $\ddot{x}_0 = 0\,m/s^2$ is assumed as sometimes done by FE tools by default. But obviously this assumption is not correct because the force F is applied immediately at $t = 0s$ and from the equation of motion we get

$$\ddot{x}(t = 0s) = \frac{F(t = 0s)}{m} = 200\frac{m}{s^2}. \tag{5.65}$$

If this value is used for the initialization of the Newmark algorithm the correct results are

obtained as it can be seen in Table 5.2 and Fig. 5.13 since such a second-order integration scheme is able to integrate the linear velocity exactly.

Table 5.2 Results of Newmark algorithm for $\ddot{x}_0 = 200\,m/s^2$

$t\ [s]$	n	$x_n\ [m]$	$\dot{x}_n\ [m/s]$	$\ddot{x}_n\ [m/s^2]$
0	0	0	0	200
0.1	1	1	20	200
0.2	2	4	40	200
0.3	3	9	60	200
0.4	4	16	80	200
0.5	5	25	100	200
0.6	6	36	120	200
0.7	7	49	140	200
0.8	8	64	160	200
0.9	9	81	180	200
1.0	10	100	200	200

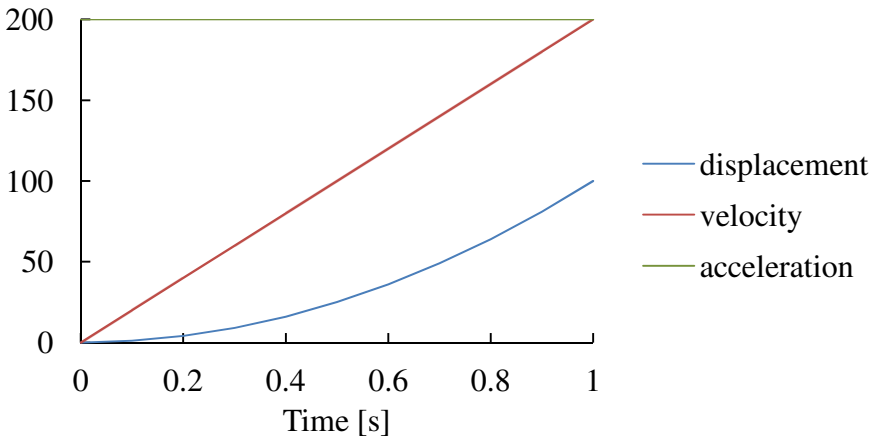


Fig. 5.13 Diagram of x, $\dot{x}$ and $\ddot{x}$ for results of Newmark algorithm for $\ddot{x}_0 = 200\,m/s^2$

In most practical applications the time-step size is small enough that initialization values for the accelerations have only a minor influence onto the result of the computation. Nevertheless, most FE tools offer the possibility of computing the initial acceleration at the beginning of a dynamic analysis. This can be done with the help of the equations of motions as demonstrated for example.

To show the behavior of an explicit time-integration scheme we now use the explicit Euler-Forward algorithm (5.4). The equations for the iteration are then

$$\dot{x}_{n+1} = \dot{x}_n + \ddot{x}_n \Delta t = \dot{x}_n + \frac{F}{m}\Delta t, \tag{5.66}$$

$$x_{n+1} = x_n + \dot{x}_n \Delta t. \tag{5.67}$$

If necessary, the acceleration can be computed from

$$\ddot{x}_{n+1} = \ddot{x}_n + \dddot{x}_n \Delta t = \frac{F}{m} + 0. \tag{5.68}$$

If we use again a time-step size of $\Delta t = 0.1s$ we get the results in Table 5.3 and Fig. 5.14. One can observe an error in the maximal displacement which decreases if the time-step size is decreased (Fig. 5.15). Obviously, the accuracy of the Euler-forward algorithm is equal to the order of the time-step since this is a first-order integration scheme.

Table 5.3 Results of Euler-Forward algorithm

$t\ [s]$	n	$x_n\ [m]$	$\dot{x}_n\ [m/s]$	$\ddot{x}_n\ [m/s^2]$
0	0	0	0	200
0.1	1	0	20	200
0.2	2	2	40	200
0.3	3	6	60	200
0.4	4	12	80	200
0.5	5	20	100	200
0.6	6	30	120	200
0.7	7	42	140	200
0.8	8	56	160	200
0.9	9	72	180	200
1.0	10	90	200	200

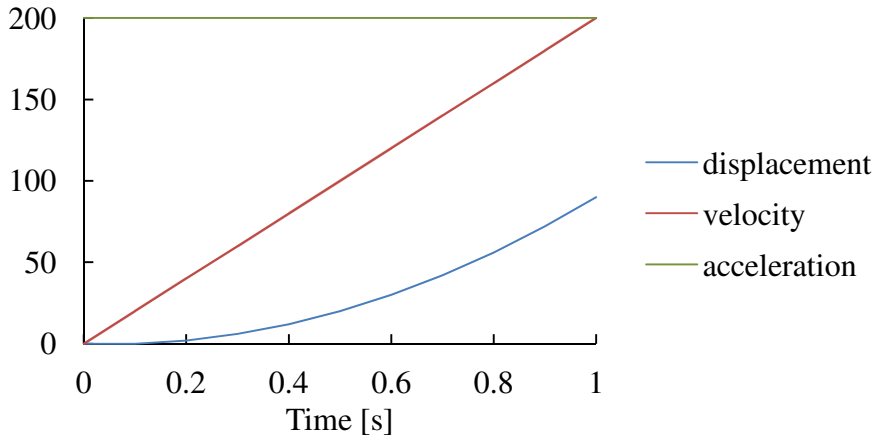


Fig. 5.14 Diagram of x, $\dot{x}$ and $\ddot{x}$ for results of Euler-Forward algorithm

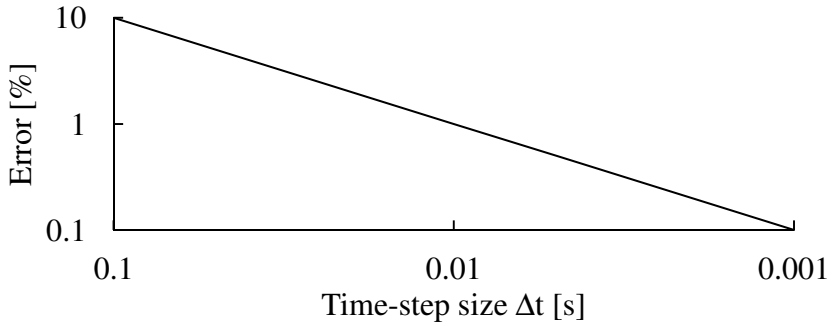


Fig. 5.15 Error of explicit Euler-Forward algorithm depending on time-step size

5.4.2 1-DOF Vibration

To study the influence of the Newmark parameters γ and β on the solution, the equation of motion of a simple 1-degree of freedom vibration problem (Fig. 5.16a) is used. Also the behavior of the explicit central difference method will be demonstrated.

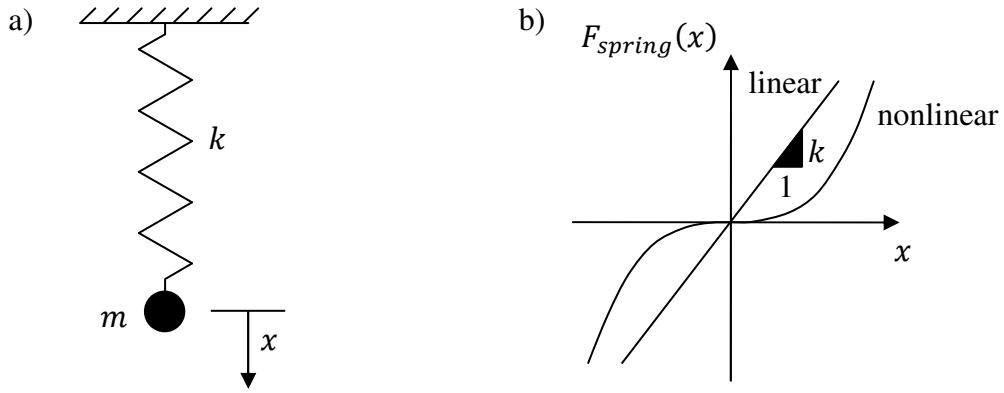


Fig. 5.16 1-degree of freedom vibration system (a) and linear and nonlinear spring characteristics (b)

The equation of motion for this free undamped system with constant spring stiffness k (Fig. 5.16b) is

$$\ddot{x} + \frac{k}{m}x = 0. \tag{5.69}$$

As initial conditions we define $x_0 = x(t = 0) = -1m$ for the displacement and $\dot{x}_0 = \dot{x}(t = 0) = 0\,m/s$ for the velocity. For convenience we assume $k/m = 1/s^2$ which simplifies Equation (5.69) to

$$\ddot{x} + x = 0. \tag{5.70}$$

From the initial displacement condition follows straight away from (5.70): $\ddot{x}_0 = \ddot{x}(t = 0) = 1\,m/s^2$. An analytical solution can be derived by using the approach

$$x(t) = x_0 sin(\omega t + \varphi_0). \tag{5.71}$$

The initial displacement condition immediately gives then $\varphi_o = \pi/2$. The first derivative of (5.71) yields

$$\dot{x}(t) = \omega x_0 cos(\omega t + \varphi_0) \tag{5.72}$$

and thus the second derivative

$$\ddot{x}(t) = -\omega^2 x_0 sin(\omega t + \varphi_0). \tag{5.73}$$

By using (5.73) and (5.71) in (5.70) the unknown eigenfrequency ω can be determined as $\omega = 1rad/s$ which finally leads to the analytical solution

$$x(t) = -sin\left(t + \frac{\pi}{2}\right) = sin\left(t - \frac{\pi}{2}\right) \tag{5.74}$$

for this simple vibration problem of Equation (5.70).

If we want so solve the problem numerically by the implicit Newmark algorithm, we have to substitute $\ddot{x}$ in Equation (5.70) by Equation (5.28) and x by x_{n+1} resulting in

$$\frac{-2x_n + 2x_{n+1} + \Delta t(-2\dot{x}_n + (2\beta - 1)\Delta t\ddot{x}_n)}{2\beta\Delta t^2} + x_{n+1} = 0 \tag{5.75}$$

which can be solved for x_{n+1} as

$$x_{n+1} = \frac{x_n + \Delta t\dot{x}_n + \Delta t^2\left(\frac{1}{2} - \beta\right)\ddot{x}_n}{\beta\Delta t^2 + 1}. \tag{5.76}$$

For a time step-size of $\Delta t = 0.2s$ and parameters $\gamma = 0.5$, $\beta = 0.25$ the solution in Fig. 5.17a is obtained. As we can observe, there is almost no difference between the analytical and the numerical solution.

To study the influence of the Newmark parameters γ and β as well as the time-step size Δt, these parameters are varied in Fig. 5.17b and Fig. 5.18. An increase of γ leads obviously to a decrease of the vibration amplitude (Fig. 5.17b), which is called numerical damping. By this, a certain stabilization of the solution is reached in some cases, since it is possible to damp higher frequencies. We will further discuss this fact in the stability analysis in the next section.

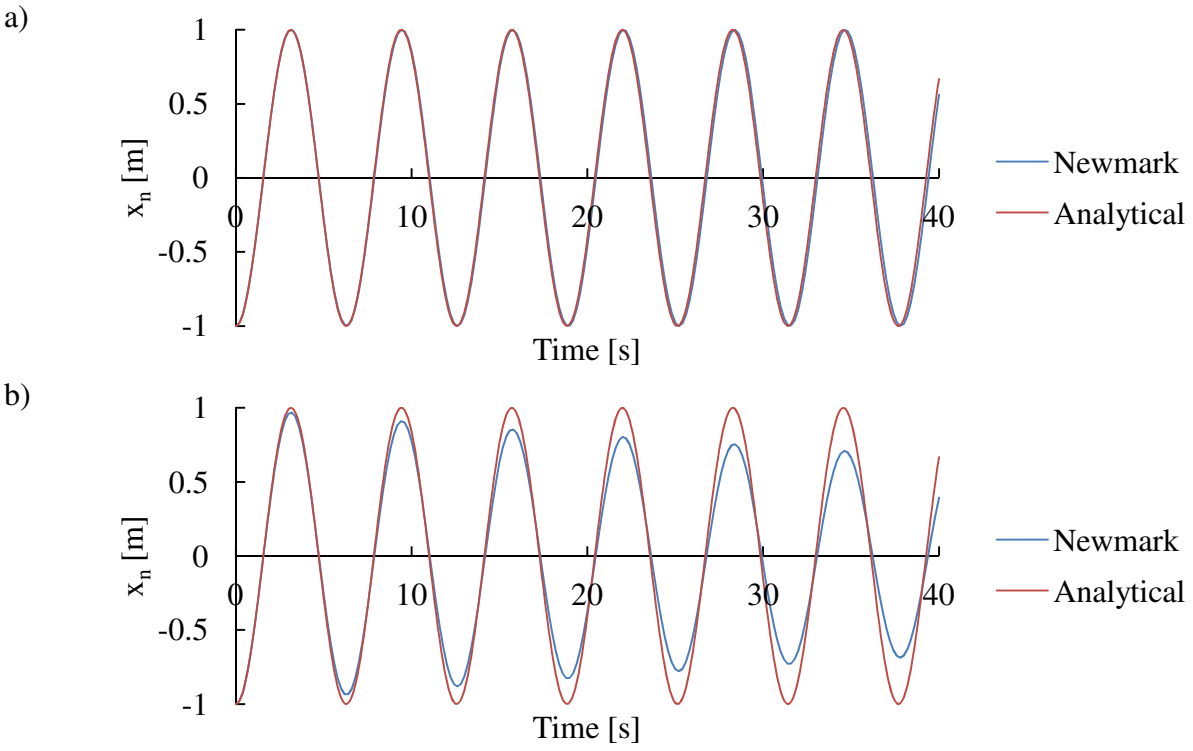


Fig. 5.17 Comparison of analytical and numerical solution obtained by Newmark algorithm for $\Delta t = 0.2s$, $\beta = 0.25$ and a) $\gamma = 0.5$, b) $\gamma = 0.6$

An increase of the time-step size still leads to a stable solution (Fig. 5.18a), however, some accuracy gets lost due to the observable period elongation. By changing the parameter β, this inaccuracy can be partially corrected (Fig. 5.18b). For this reason β is sometimes also called

accuracy parameter, but whether an increase or decrease of β improves the solution depends on the particular mechanical problem.

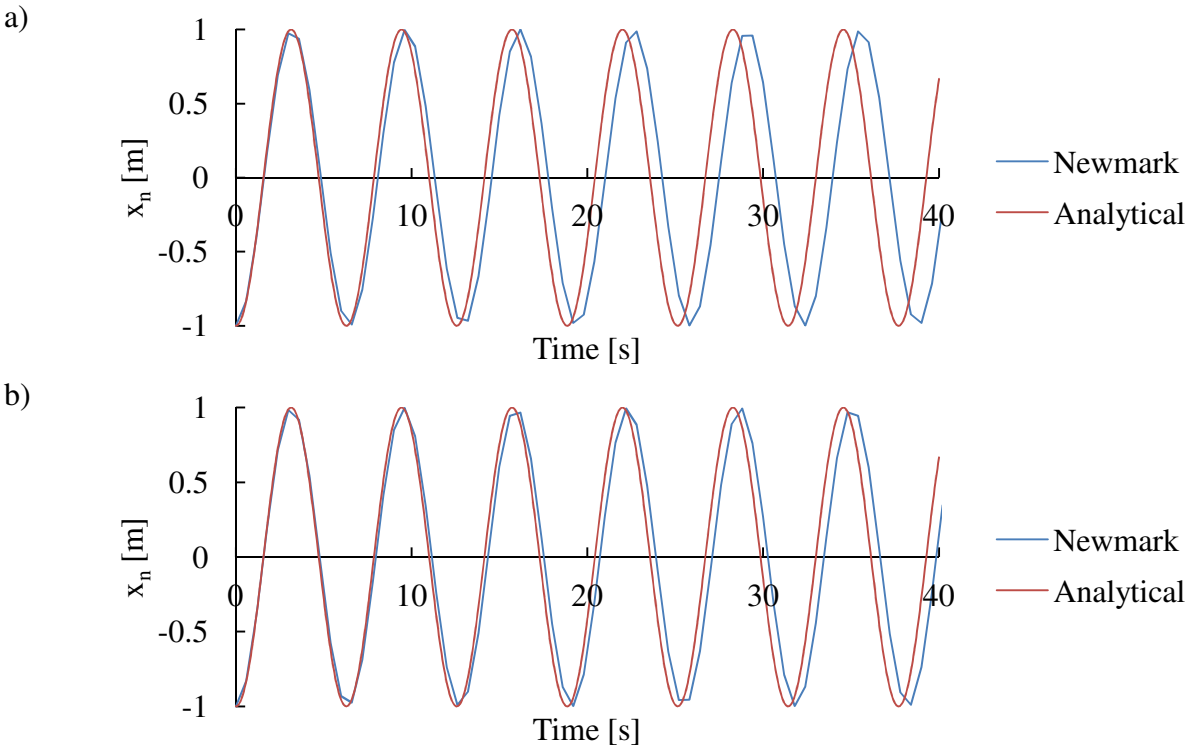


Fig. 5.18 Comparison of analytical and numerical solution obtained by Newmark algorithm for $\Delta t = 0.6s$, $\gamma = 0.5$ and a) $\beta = 0.25$, b) $\beta = 0.166$

Next, we want to use the HHT algorithm to solve the simple free vibration problem. By adding the extra terms of Equation (5.29), the equation of motion (5.70) becomes

$$\ddot{x}_{n+1} + (1+\alpha)x_{n+1} - \alpha x_n = 0. \tag{5.77}$$

Substituting $\ddot{x}_{n+1}$ by Equation (5.28) leads to

$$\frac{-2x_n + 2x_{n+1} + \Delta t(-2\dot{x}_n + (2\beta - 1)\Delta t\ddot{x}_n)}{2\beta\Delta t^2} + (1+\alpha)x_{n+1} - \alpha x_n = 0 \tag{5.78}$$

which can be solved for x_{n+1} as

$$x_{n+1} = \frac{x_n(1 + \Delta t^2\alpha\beta) + \Delta t\dot{x}_n + \Delta t^2\left(\frac{1}{2} - \beta\right)\ddot{x}_n}{1 + \Delta t^2\beta(1+\alpha)}. \tag{5.79}$$

For $\alpha = 0$ this obviously reduces to the solution (5.76) of the Newmark scheme. In Fig. 5.19 the solution of the free vibration problem obtained with the HHT algorithm for a time-step size of $\Delta t = 0.6s$ and a parameter choice of $\alpha = -0.2$ is plotted. If the parameter coupling (5.30), (5.31) is applied, strong numerical damping is introduced. But it is also possible to keep α, β and γ uncoupled. In this case the solution differs only slightly from the Newmark solution and no numerical damping is added since this is regularized by a proper value of the γ parameter. However, the advantage of using the parameter coupling is that just one

parameter has to be set by the user, which reduces the probability of choosing an unstable parameter combination by accident.

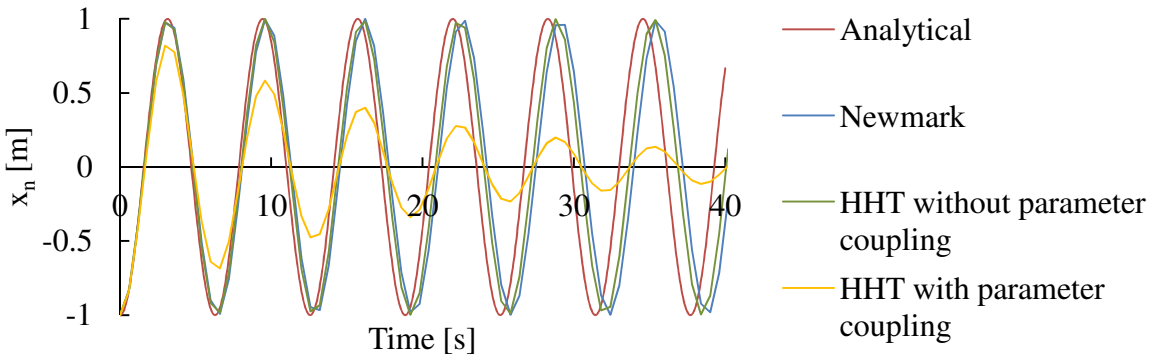


Fig. 5.19 Comparison of analytical and numerical solutions for $\Delta t = 0.6s$ obtained by Newmark algorithm ($\gamma = 0.5$, $\beta = 0.25$) and HHT algorithm with ($\alpha = -0.2 \rightarrow \gamma = 0.7$, $\beta = 0.36$) and without parameter coupling ($\alpha = -0.2$, $\gamma = 0.5$, $\beta = 0.25$)

Finally, the central difference scheme (5.21) as an explicit time-integration algorithm is applied to the free vibration problem of Equation (5.69) with $k/m = 1/s^2$, which leads to the recursion formula

$$x_{n+1} = \frac{1}{m}[-(\Delta t^2 k - 2m)x_n - m x_{n-1}] = -x_n(\Delta t^2 - 2) - x_{n-1}. \tag{5.80}$$

For a time-step size of $\Delta t = 0.1s$ we get the displacement solution in Fig. 5.20. Almost no difference is visible between the numerical and analytical results.

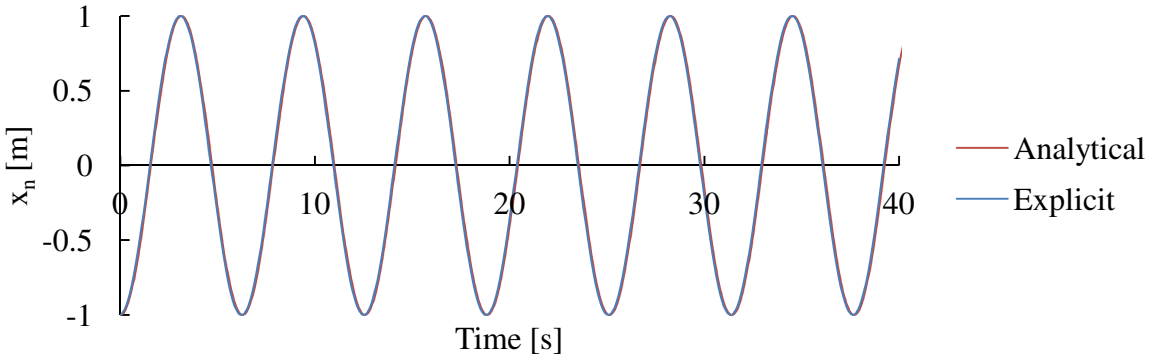


Fig. 5.20 Comparison of analytical and numerical solution obtained with the central difference method and a time-step size of $\Delta t = 0.1s$

Moreover, it is interesting to note what happens if the time-step size in the central difference scheme is increased. In Fig. 5.21 the solutions for a time-step size of $\Delta t = 1.9s$ and for $\Delta t = 2.1s$ are shown. For a bigger time-step size the accuracy is reduced as it could be expected for using $\Delta t = 1.9s$, but the solution grows unbounded if the time-step size is increased further to $\Delta t = 2.1s$. The question is why is there such a big difference with regard to the solution although there is only a small change in the time-step size? The answer to this

question is strongly related to the stability of this time-integration algorithm and will be given in Chapter 6.

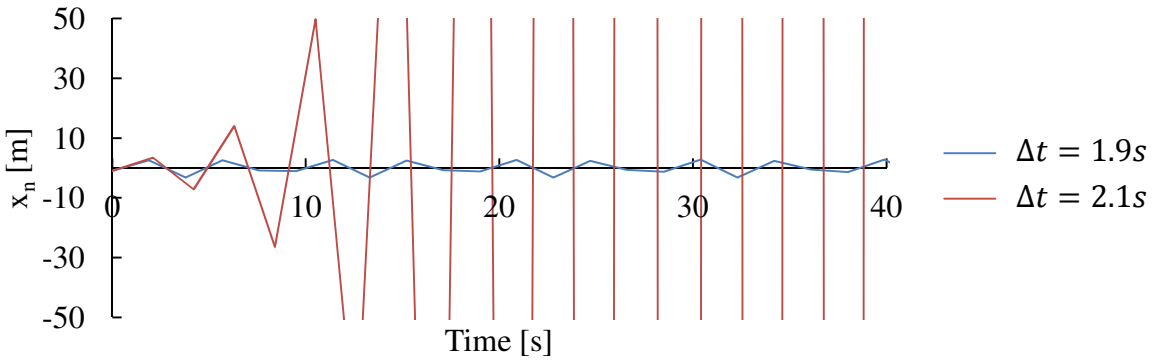


Fig. 5.21 Solutions for free vibration problem with explicit central difference method for $\Delta t = 1.9s$ and $\Delta t = 2.1s$

5.4.3 Nonlinear 1-DOF Vibration

The last example is similar to the one before, but now we assume the nonlinear spring characteristic (Fig. 5.16b)

$$F_{Spring}(x) \sim x^3. \tag{5.81}$$

The local spring stiffness can be calculated as derivative of the force-displacement curve as

$$k(x) = \frac{\partial F_{Spring}(x)}{\partial x} = \frac{\partial}{\partial x}(Cx^3) = 3Cx^2 \tag{5.82}$$

where constant C is chosen to be equal to three ($C = 3\,kg/m^2s^2$). By this, the equation of motion becomes

$$\ddot{x} + 3x^3 = 0, \tag{5.83}$$

if we assume again $m = 1kg$.

To solve (5.83) numerically, Newmark time-integration with $\gamma = 0.5$ and $\beta = 0.25$ is used, which leads to the nonlinear equation

$$f(x_{n+1}) = x_{n+1}^3 + \frac{x_{n+1}}{3\beta\Delta t^2} - \frac{2x_n - \Delta t(-2\dot{x}_n + (2\beta - 1)\Delta t\ddot{x}_n)}{6\beta\Delta t^2} = 0 \tag{5.84}$$

with respect to x_{n+1}. If we want to use a Newton or Quasi-Newton algorithm for the solution, the derivative of $f(x_{n+1})$ with respect to x_{n+1}

$$f'(x_{n+1}) = 3x_{n+1}^2 + \frac{1}{3\beta\Delta t^2} \tag{5.85}$$

is needed. In Fig. 5.22 the solutions computed by a Full-Newton and two different Quasi-Newton approaches (constant stiffness and secant stiffness) are plotted with $\Delta t = 0.6s$. For all approaches the equilibrium iterations are aborted if an accuracy of $f({}^{k+1}x_{n+1}) < 0.001$ is

reached. As initial conditions $x_0 = x(t=0) = -1m$, $\dot{x}_0 = \dot{x}(t=0) = 0\,m/s$ are used from which $\ddot{x}_0 = \ddot{x}(t=0) = -3x_0^3 = 3\,m/s$ follows.

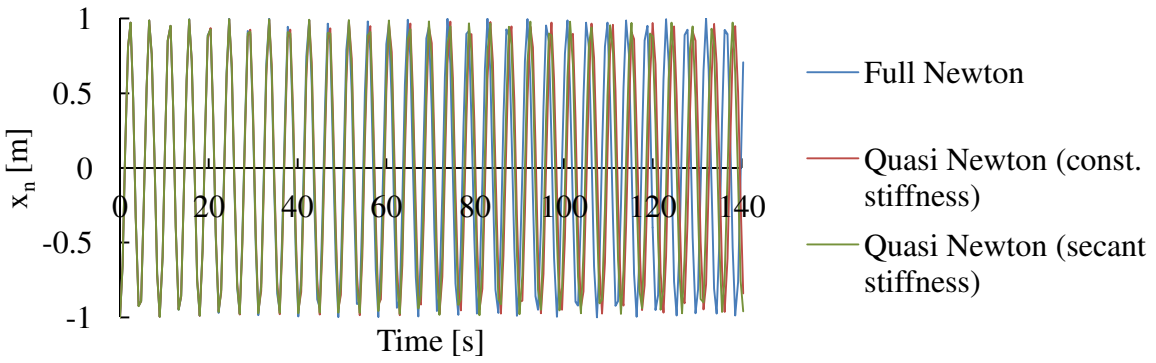


Fig. 5.22 Comparison of the numerical solutions for the nonlinear vibration problem obtained by Full-Newton and Quasi-Newton methods and a time-step size of $\Delta t = 0.6s$

The results of the Full-Newton and Quasi-Newton approaches are quite similar. However, after a longer simulation time some differences are visible. In this case, it is more interesting to have a look at the number of iterations performed by the different algorithms in every time-step (Fig. 5.23). As one can, see the number of iterations is the lowest for the Full-Newton algorithm followed by the Quasi-Newton algorithm using the secant stiffness (see also Fig. 5.10) and the Quasi-Newton algorithm using a constant stiffness (computed only once in the first iteration). On the other hand, one iteration of a Quasi-Newton algorithm needs less time than that of a Full-Newton algorithm due to the fact that there is no need for a time-consuming computation of derivatives (which corresponds to the computation of the stiffness matrix of the problem). For this reason it always depends on the particular problem (necessary time-step size, accuracy requirements, effort for computation of derivatives, nonlinearity of problem, etc.) to be solved, which solution procedure is the fastest one. The differences in the number of iterations per time-step between the three approaches are for example much smaller if a time-step size of $\Delta t = 0.1s$ is used.

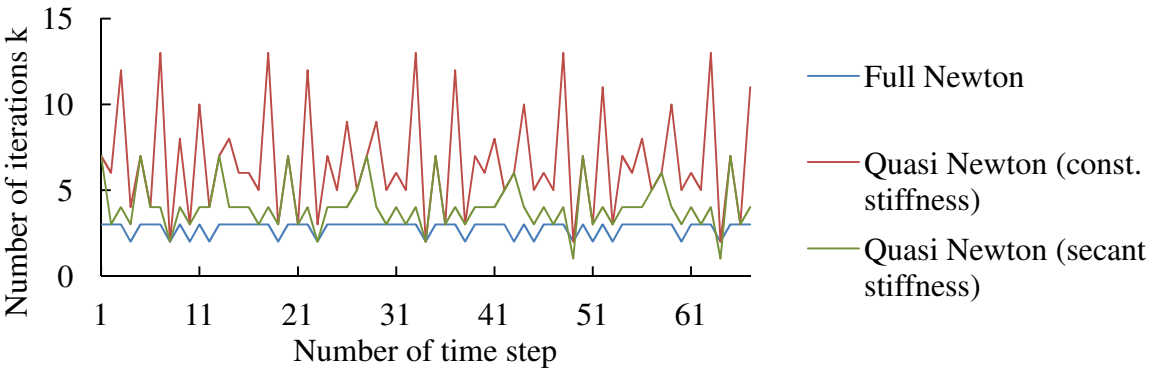


Fig. 5.23 Number of iterations needed per time-step for different strategies to solve the nonlinear equation of the free vibration problem

6 Stability and Accuracy Considerations of Time-Integration Algorithms

For every numerical algorithm or solution process, stability is a very important aspect. As we have seen in Fig. 5.21, the solution of time-integration algorithms may grow unbounded under certain circumstances. Here, we will discuss under which conditions such a phenomenon may appear and what is necessary to avoid such unwanted effects. Moreover, a stability analysis of a time-integration algorithm is always useful to show the borders of its applicability. A definition of numerical stability could be the following: A system is stable if small changes in the initial conditions (input data) lead only to small changes of the numerical solution of the system behavior (output data). In other words, the numerical solution of a stable system may not grow unbounded or become chaotic without converging to an attractor (for more details see e.g. [59,100]). It is important to stress the fact that the system under investigation is stable, otherwise a definition of numerical stability wouldn't work. This problem can be easily demonstrated for a simple example: Let us assume that we want to solve the differential equation

$$m\ddot{x} + x \cdot (\sin(\omega t) - 2) = 0 \tag{6.1}$$

with parametric excitation numerically for $\omega = 1 rad/s$, $x(0) = 0m$, $\dot{x}(0) = 1m$ and $m = 1kg$. The "stiffness" term $(\sin(\omega t) - 2)$ is always negative, but the solution $x(t)$ grows unbounded (Fig. 6.1). In such a case, a stability criterion would predict unstable behavior of the solution algorithm although the algorithm itself might be stable, but not the underlying physical problem. A deeper discussion of this topic can be found in [89].

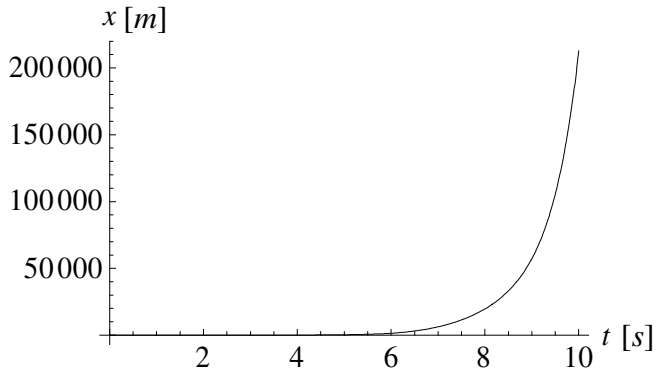


Fig. 6.1 Solution of Equation (6.1)

For stability analysis of time-integration algorithms, different strategies exist which also depend on the linearity/nonlinearity of the considered problem. From a mechanical point of view, a linear stability analysis procedure can be applied to mechanical problems which can be described by ordinary linear differential equations or systems of ordinary linear differential equations, respectively. For nonlinear problems there is no general methodology which can be used to determine the stability behavior of an algorithm. Nevertheless, there are some possibilities of showing the stability/instability of time-integration schemes for certain

nonlinear constellations. An algorithm is called unconditionally stable if no time-step size restriction is necessary for stability, and conditionally stable otherwise.

Besides the stability of an integration algorithm its accuracy is of importance. As we have seen in Section 5.4.2, there are differences between the algorithms in terms of the amplitude decay/ascent as well as in the elongation/shortening of the period. The amplitude change is strongly related to the so-called spectral radius $\rho(\boldsymbol{A})$ of an integration algorithm. The smaller the spectral radius, the bigger is the numerical damping (under the condition that $\rho(\boldsymbol{A}) \leq 1$). In this chapter we investigate the accuracy of the presented time-integration algorithms with regard to period elongation and amplitude decay again for the example of the simple free and undamped vibration problem (6.3) with initial conditions $x_0 = -1m$ and $\dot{x}_0 = 1\,m/s$ in dependency of the used time-step size. As already mentioned, the accuracy of an integration algorithm does also depend on the particular problem it is used for. Nevertheless, the investigation of such an example problem gives at least an idea about the errors that can be expected for the different schemes and allows a comparison of them.

6.1 Stability of Linear Problems

Approaches for the stability analysis of linear systems are presented in [17] and [84] for example. Especially in the field of control theory, stability criteria play an important role, which is why many approaches for stability considerations are borrowed from there. In this section, we will refer to the assessment of stability presented in [17], which uses the concept of amplification matrices and goes back to ideas presented in [93,113,144] and [34]. A detailed description of this concept, which sometimes is also called von Neumann stability analysis, can be found in [142] and [92].

As described in [17] and [11], time-integration procedures can be formulated as

$$\widetilde{\boldsymbol{x}}_{n+1} = \boldsymbol{A} \cdot \widetilde{\boldsymbol{x}}_n + \boldsymbol{L} r_{n+\nu}. \tag{6.2}$$

The matrix $\boldsymbol{A}$ describes both the integration algorithm and the mechanical problem, the vector $\boldsymbol{L}$ is the load operator, and $r_{n+\nu}$ expresses the applied load where ν depends on the particular time-integration algorithm and is e.g. equal to one for the implicit algorithms considered here and equal to zero for the explicit central difference method. Depending on the integration scheme, the vectors $\widetilde{\boldsymbol{x}}_{n+1}$ and $\widetilde{\boldsymbol{x}}_n$ contain for example displacements, velocities and accelerations (for the presented implicit schemes) or just displacements (for the central difference method). For the stability analysis and the computation of $\boldsymbol{A}$ it is sufficient to consider a simple undamped free vibration problem of the form

$$\ddot{x} + \omega^2 x = 0. \tag{6.3}$$

This has the advantage that the load operator $\boldsymbol{L}$ can be neglected because $r_{n+\nu}$ equals zero.

For the investigation of the stability of the recursive scheme (6.2), the spectral decomposition of the matrix $\boldsymbol{A}$ is used, which is given by

$$\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{J}\boldsymbol{Q}^{-1} \tag{6.4}$$

with $\boldsymbol{Q}$ as the matrix of eigenvectors of $\boldsymbol{A}$ and $\boldsymbol{J}$ as the Jordan canonical form of $\boldsymbol{A}$. If all eigenvalues are distinct, $\boldsymbol{J}$ is a diagonal matrix with eigenvalues λ_i along its diagonal [145]. For multiple eigenvalues, $\boldsymbol{J}$ is not necessarily a diagonal matrix, but consists of so-called Jordan segments for each eigenvalue, which are made up of a number of Jordan blocks containing the eigenvalues along their diagonals and the entry "1" at their superdiagonals [104].

If we want to compute the n-th recursion value starting from $n = 0$ and $\widetilde{\boldsymbol{x}}_0$ by applying (6.2) n times and using $r_{n+\nu} = 0$ for simplicity, we get

$$\widetilde{\boldsymbol{x}}_n = \boldsymbol{A} \cdot \ldots \cdot \boldsymbol{A} \cdot \boldsymbol{A} \cdot \widetilde{\boldsymbol{x}}_0 = \boldsymbol{A}^n \cdot \widetilde{\boldsymbol{x}}_0. \tag{6.5}$$

Decomposing $\boldsymbol{A}^n$ with respect to (6.4) gives

$$\boldsymbol{A}^n = (\boldsymbol{Q}\boldsymbol{J}\boldsymbol{Q}^{-1})(\boldsymbol{Q}\boldsymbol{J}\boldsymbol{Q}^{-1}) \ldots (\boldsymbol{Q}\boldsymbol{J}\boldsymbol{Q}^{-1}) = \boldsymbol{Q}\boldsymbol{J}^n\boldsymbol{Q}^{-1}. \tag{6.6}$$

If the elements of the matrix $\boldsymbol{J}$ containing the eigenvalues of $\boldsymbol{A}$ have bigger absolute values than one, the solution will grow unbounded for $n \to \infty$. For this reason the spectral radius of a matrix is defined as the eigenvalue of the matrix with the biggest absolute value as

$$\rho(\boldsymbol{A}) = \max_i |\lambda_i|. \tag{6.7}$$

The stability criterion states that an integration algorithm is unconditionally stable if the spectral radius of the matrix $\boldsymbol{A}$ is smaller or equal to one for all time-step sizes, i.e.,

$$\rho(\boldsymbol{A}) \leq 1 \quad \forall\, \Delta t \tag{6.8}$$

in case of unique eigenvalues λ_i, or

$$\rho(\boldsymbol{A}) < 1 \quad \forall\, \Delta t \tag{6.9}$$

in case of multiple eigenvalues [11,63,66]. The reason for a stricter condition in case of multiple eigenvalues can be demonstrated based on a simple example for a 2x2 matrix $\boldsymbol{A}$ with distinct eigenvalues $\lambda_1 \neq \lambda_2$. The Jordan canonical form of $\boldsymbol{A}$ is then

$$\boldsymbol{J} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \tag{6.10}$$

and therefore

$$\boldsymbol{J}^n = \begin{bmatrix} \lambda_1^n & 0 \\ 0 & \lambda_2^n \end{bmatrix}. \tag{6.11}$$

For a double eigenvalue $\lambda = \lambda_1 = \lambda_2$ the Jordan canonical form for $\boldsymbol{A}$ is

$$\boldsymbol{J} = \begin{bmatrix} \lambda & 1 \\ 0 & \lambda \end{bmatrix} \tag{6.12}$$

and consequently

$$\boldsymbol{J}^n = \begin{bmatrix} \lambda^n & n\lambda^{n-1} \\ 0 & \lambda^n \end{bmatrix}. \tag{6.13}$$

From this little example it becomes obvious that the entries of (6.11) will not grow unbounded for $n \to \infty$ in the case of distinct eigenvalues as long as $\lambda_i \leq 1$. However, for double eigenvalues, $\lambda < 1$ is necessary to guarantee that also the entry $n\lambda^{n-1}$ stays bounded in

(6.13). In Fig. 6.2 the effects of double and single eigenvalues are demonstrated for a 2x2 matrix $\boldsymbol{J}$. As it can be seen, the maximum value of all matrix elements $\max_{i,j}\left(J_{ij}^{n}\right)$ grows unbounded if the biggest eigenvalue of the matrix is greater than one for distinct eigenvalues or equal to one for a double eigenvalue. In the latter case, the matrix values are limited only for $\lambda < 1$.

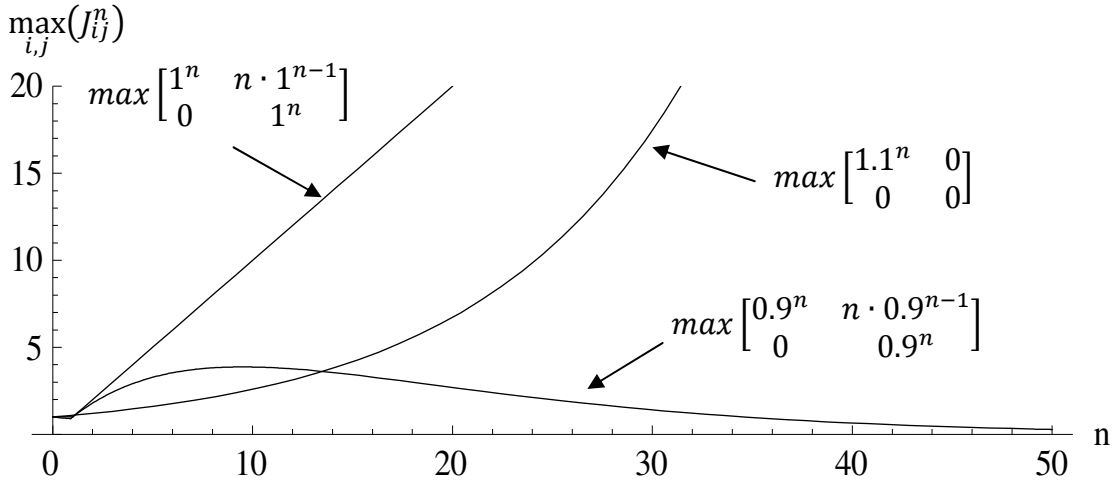


Fig. 6.2 Maximum value of all coordinates of $\boldsymbol{J}^n$ for different eigenvalue constellations

6.1.1 Newmark Algorithm

The computation of matrix $\boldsymbol{A}$ needed in the stability criterion (6.8) and (6.9), respectively, for the Newmark algorithm is possible with the help of Equations (5.27) and (5.28). Together with the equation of motion (6.3) of the underlying mechanical problem there are three equations for determining the three unknowns $\ddot{x}_{n+1}$, $\dot{x}_{n+1}$ and x_{n+1}, which finally leads after some transformations to the linear system of equations

$$\begin{bmatrix} \ddot{x}_{n+1} \\ \dot{x}_{n+1} \\ x_{n+1} \end{bmatrix} = \boldsymbol{A} \cdot \begin{bmatrix} \ddot{x}_{n} \\ \dot{x}_{n} \\ x_{n} \end{bmatrix}. \tag{6.14}$$

For the Newmark algorithm the matrix $\boldsymbol{A}$ can be derived as

$$\boldsymbol{A}_{NM} = \begin{bmatrix} \dfrac{2\beta-1}{2\beta z_1} & \dfrac{-1}{\Delta t \beta z_1} & \dfrac{-1}{\Delta t^2 \beta z_1} \\ \Delta t - \dfrac{\gamma(2\beta-1)}{\Delta t 2\beta^2\omega^2 z_1} - \dfrac{\Delta t \gamma}{2\beta} & 1 + \dfrac{\gamma}{\Delta t^2 \beta^2 \omega^2 z_1} - \dfrac{\gamma}{\beta} & \dfrac{-\gamma}{\beta \Delta t} + \dfrac{\gamma}{\Delta t^3 \beta^2 \omega^2 z_1} \\ \dfrac{-(2\beta-1)}{2\beta\omega^2 z_1} & \dfrac{1}{\Delta t \beta \omega^2 z_1} & \dfrac{1}{\Delta t^2 \beta \omega^2 z_1} \end{bmatrix} \tag{6.15}$$

with the abbreviation

$$z_1 = \left(1 + \frac{1}{\beta\omega^2\Delta t^2}\right). \tag{6.16}$$

For computing the spectral radius of $\boldsymbol{A}_{NM}$ it is necessary to determine its eigenvalues from the eigenvalue problem

$$\det(\lambda \boldsymbol{E} - \boldsymbol{A}_{NM}) = 0 \tag{6.17}$$

which yields

$$\lambda_1 = 0,$$

$$\lambda_{2,3} = \frac{1 + \frac{\Delta t^2}{T^2}\pi^2(4\beta - 2\gamma - 1) \pm \pi\sqrt{\frac{\Delta t^4}{T^4}\pi^2[(1+2\gamma)^2 - 16\beta] - 4\frac{\Delta t^2}{T^2}}}{1 + 4\beta\frac{\Delta t^2}{T^2}\pi^2}, \tag{6.18}$$

where ω is substituted by $2\pi/T$ (see Appendix A). These eigenvalues are unique if $\lambda_2 \neq \lambda_3$ and $\lambda_{2,3} \neq 0$. Equality of λ_2 and λ_3 exists if the root expression in Equation (6.18) equals zero:

$$\frac{\Delta t^4}{T^4}\pi^2[(1+2\gamma)^2 - 16\beta] - 4\frac{\Delta t^2}{T^2} = 0 \tag{6.19}$$

which is fulfilled for

$$\frac{\Delta t}{T} \in \left\{0, -\frac{2}{\pi\sqrt{(1+2\gamma)^2 - 16\beta}}, \frac{2}{\pi\sqrt{(1+2\gamma)^2 - 16\beta}}\right\}. \tag{6.20}$$

The cases $\Delta t/T = 0$ and $\Delta t/T < 0$ needn't be considered, since $\Delta t/T$ is always greater than zero. This means that the two eigenvalues are different except for the special case of

$$\frac{\Delta t}{T} = \frac{2}{\pi\sqrt{(1+2\gamma)^2 - 16\beta}} \tag{6.21}$$

where we have to apply stability criterion (6.9), else criterion (6.8) is sufficient.

By plotting the maximum absolute value of (6.18), which is the spectral radius $\rho(\boldsymbol{A}_{NM})$, it is possible to check the stability for arbitrary parameter combinations of γ, β and $\Delta t/T$. However, it is more convenient to plot the spectral radius for fixed parameters γ and β over the normalized time-step size $\Delta t/T$. As shown in Fig. 6.3, the spectral radius of the Newmark algorithm with the standard parameters ($\gamma = 0.5$, $\beta = 0.25$) is equal to one for all time-step sizes and therefore unconditionally stable (except for (6.21) which is discussed in Appendix B). A change of the γ- or β-parameter influences the spectral radius and leads in the case of $\gamma = 0.4$, $\beta = 0.25$ to conditional stability, which means that stability is only guaranteed if the normalized time-step size is smaller than a certain value. For simple problems like the free-vibration example used here, also a conditionally stable algorithm could be used if the time-step size is small enough such that $\rho(\boldsymbol{A}, \Delta t/T) \leq 1$. But for a more complex FEM problem, the highest frequency and its associated period $T = 2\pi/\omega$ is not known a priori. Therefore, the necessary time-step guaranteeing stability is also not known. For this reason, only unconditionally stable algorithms should be used in this context.

The remaining question then is, for which parameter combinations unconditional stability can be guaranteed? To derive a condition for parameters γ and β, the eigenvalues of the matrix $\boldsymbol{A}_{NM}$ given in (6.18) have to be considered again as demonstrated in Appendix B, where it is shown that choosing

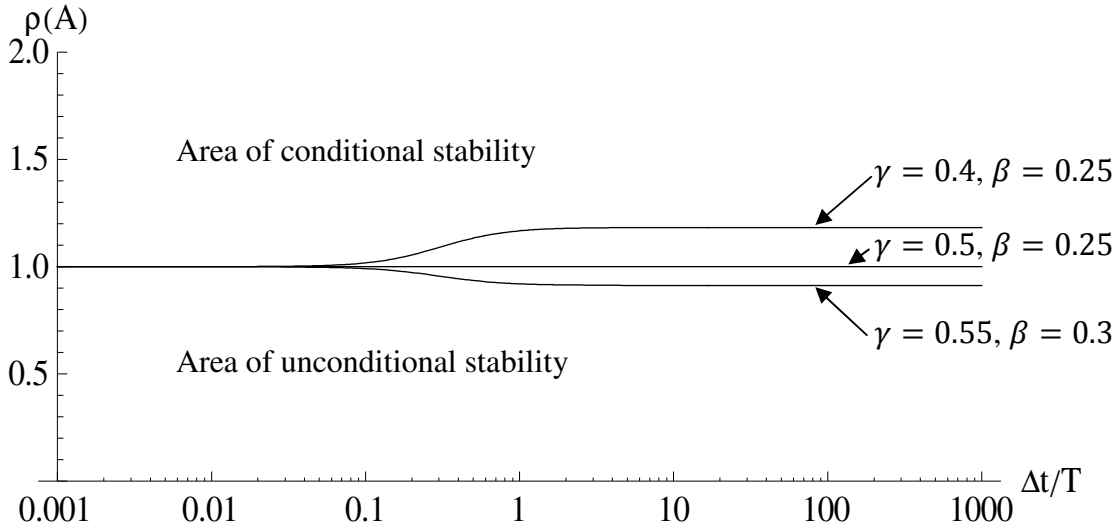


Fig. 6.3 Spectral radius of Newmark algorithm for different parameter combinations of γ and β

$$\gamma \geq \frac{1}{2} \wedge \beta \geq \frac{(2\gamma+1)^2}{16} \tag{6.22}$$

or

$$\gamma > \frac{1}{2} \wedge \frac{\gamma}{2} \leq \beta < \frac{(2\gamma+1)^2}{16} \tag{6.23}$$

leads to unconditional stability of the Newmark algorithm. Fig. 6.4 illustrates this stable parameter region of γ and β described by conditions (6.22) and (6.23).

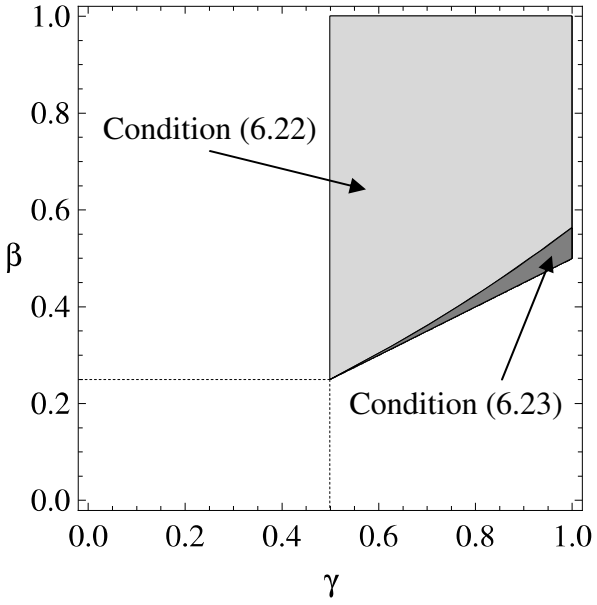


Fig. 6.4 Stable region of Newmark algorithm for all parameter combinations of γ and β

For small time steps the spectral radius is close to one for all stable parameter combinations of γ and β. But for bigger time steps the spectral radius can become quite small for certain parameter combinations, which leads to a strong numerical damping (Fig. 6.5).

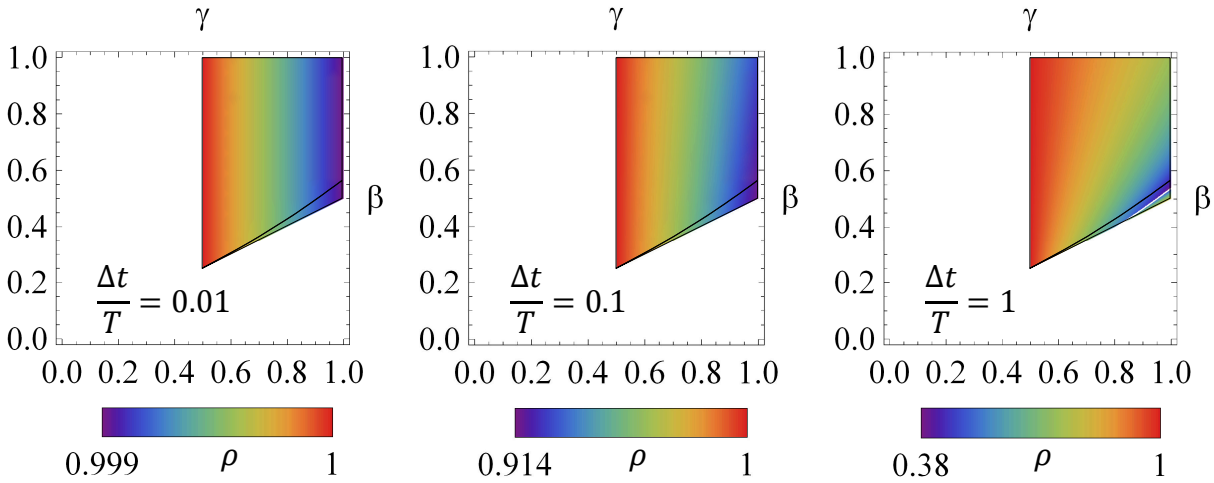


Fig. 6.5 Spectral radius ρ of Newmark algorithm for different normalized time-step sizes in stable parameter region

6.1.2 HHT Algorithm

The process for the computation of the amplification matrix $\boldsymbol{A}$ for the HHT method is similar to that for the Newmark algorithm leading also to Equation (6.14) where

$$\boldsymbol{A}_{HHT} = \frac{1}{2+2z_2\beta\Delta t^2}\begin{bmatrix} z_2(2\beta-1)\Delta t^2 & -2z_2\Delta t & -2\omega^2 \\ \Delta t[2(1-\gamma) & 2+2z_2\Delta t^2(\beta-\gamma) & -2\Delta t\gamma\omega^2 \\ +z_2\Delta t^2(2\beta-\gamma)] & & \\ (1-2\beta)\Delta t^2 & 2\Delta t & 2+2\alpha\beta\Delta t^2\omega^2 \end{bmatrix} \tag{6.24}$$

with the abbreviation

$$z_2 = (1+\alpha)\omega^2. \tag{6.25}$$

In Fig. 6.6 the spectral radii of (6.24) and therefore the HHT algorithm for different parameter combinations of α, γ and β are plotted versus $\Delta t/T$. For $\alpha = -0.05$ and $\alpha = -0.2$ the parameter coupling (5.30)/(5.31) is used. The curves show the reason for the numerical damping introduced by some parameter combinations already observed in Fig. 5.19 for $\alpha = -0.2$. Whenever the spectral radius for a given normalized time-step size is smaller than one, the corresponding frequency is damped. The damping is stronger the smaller the spectral radius is. For the solution of example problem (5.70) plotted in Fig. 5.19, a time-step size of $\Delta t = 0.6s$ is used. The eigenfrequency of the 1-DOF vibration system is $\omega = 1rad/s$ corresponding to a period of $T = 2\pi/\omega = 2\pi\ s$. By this, the normalized time-step size in this example is $\Delta t/T \approx 0.0955$ which leads to a spectral radius of $\rho(\boldsymbol{A}_{HHT}) \approx 0.99833$ for $\alpha = -0.2$.

For the determination of the stability constraints with respect to the parameters of the algorithm, it is useful to perform a numerical experiment and plot the stability condition (6.8) for $\Delta t/T \to \infty$. This is done by replacing ω by $2\pi/T$ in Equation (6.24), computing the eigenvalues of $\boldsymbol{A}_{HHT}$ and subsequently the limit values of these eigenvalues for $\Delta t/T \to \infty$, both obtained through symbolic computation. Finally, all parameter combinations with

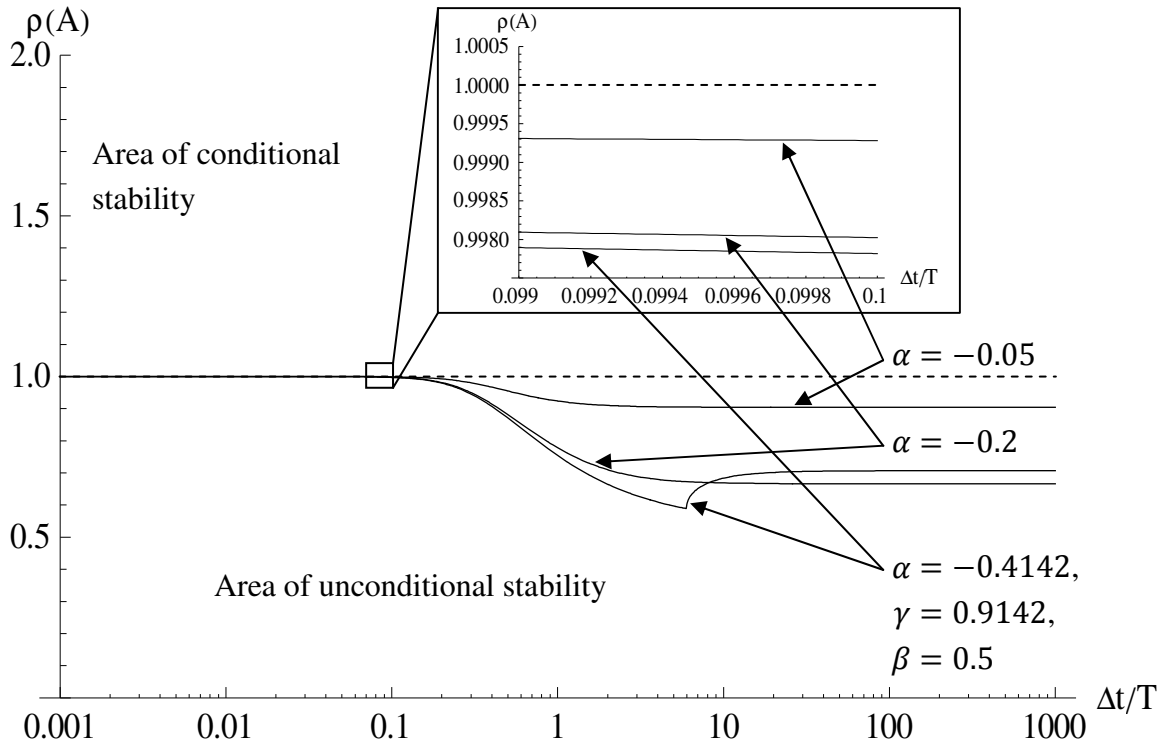


Fig. 6.6 Spectral radii of HHT algorithm for different parameter combinations of α, γ and β

$\rho(\boldsymbol{A}_{HHT}) \leq 1$ are plotted in Fig. 6.7. From this plot it becomes obvious that values $\alpha < -0.5$ do not make sense if unconditional stability is demanded.

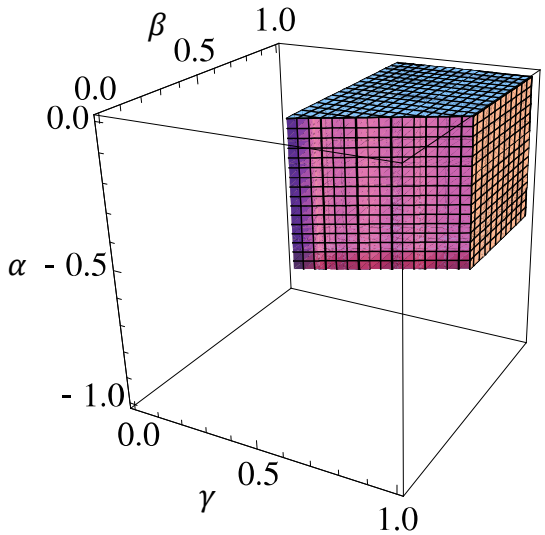


Fig. 6.7 Parameter space of HHT algorithm satisfying condition (6.8) for $\Delta t/T \to \infty$

In [63] the parameter coupling (5.30)/(5.31) is suggested, which has the advantage that only the parameter α has to be managed by the user of the algorithm. Moreover, the linear coupling between γ and α gives the possibility of directly influencing the numerical damping by changing α and keeping γ inside its stability limits. Coupling Equation (5.31) results from using (5.30) in (6.22) and demanding equality. To answer the question of worthwhile choices for parameter α, it is helpful to consider the plots in Fig. 6.8 showing the spectral radii of the

HHT algorithm depending on α and the normalized time-step size $\Delta t/T$. Starting from $\alpha = 0$ the numerical damping cannot be increased by decreasing α below $\alpha = -1/3$, obviously. For this reason a logical choice would be $-1/3 \leq \alpha \leq 0$ if the parameter coupling is applied.

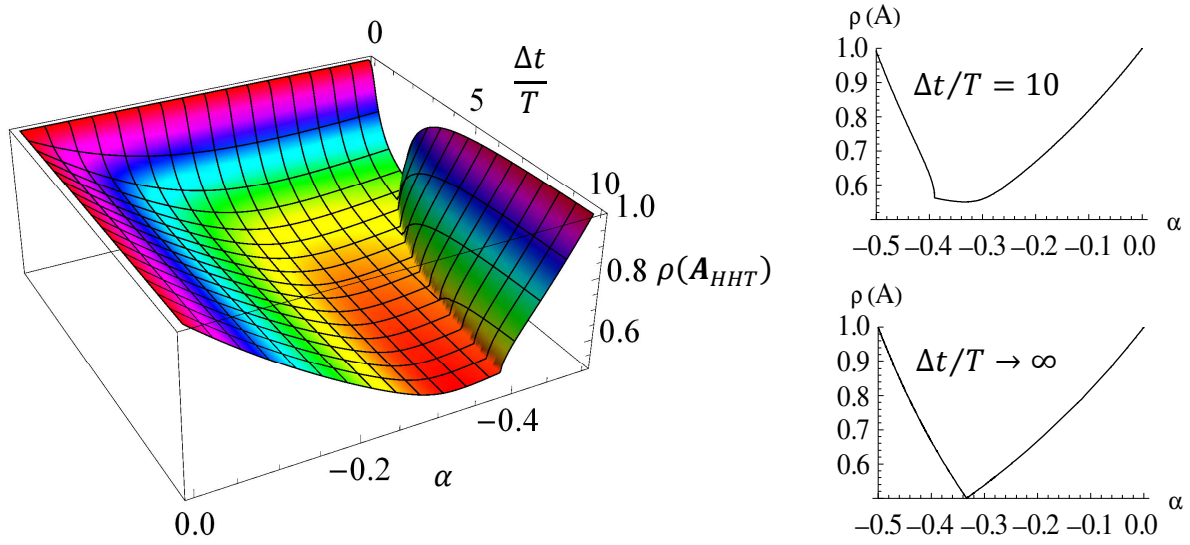


Fig. 6.8 Spectral radii of HHT algorithm depending on α (with parameter coupling of γ and β) and the normalized time-step size (left), and spectral radius depending on α for $\Delta t/T = 10$ and $\Delta t/T \to \infty$ (right)

6.1.3 Newmark-Euler Algorithm (Composite Scheme)

The derivation of the recursion matrix $\boldsymbol{A}$ for the Newmark-Euler or Composite integration scheme is more elaborate, but follows basically the same procedure as for the other schemes resulting in (6.14). Here, only the Composite scheme of Equations (5.39) and (5.40) is considered for two reasons. First, it is more general and the Newmark-Euler scheme is a special case of the Composite scheme. The second reason is that a fair comparison of the different integration schemes is only possible, if the same time-step sizes are used, whereas the time-step in the Newmark-Euler scheme is divided into a Newmark and an Euler-Backward step, where the step sizes are just as half as big. In the Composite scheme the time-step sizes of the Newmark and the Euler-Backward step can be chosen such that every time-step has the same size as the time-steps of an integration scheme the Composite scheme is compared to.

The amplification matrix in Equation (6.14) for the Composite scheme (5.39), (5.40) reads as

$$\boldsymbol{A}_{Comp} = \frac{1}{2\left((1+\alpha)^2 + \Delta t_n^2 \omega^2\right)\left(1 + \beta \Delta t_{n-1}^2 \omega^2\right)} \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} \tag{6.26}$$

where

$$A_{11} = [\Delta t_{n-1} + \alpha(\Delta t_n + \Delta t_{n-1})] \cdot \omega^2 [(1+\alpha)(2\beta - 1)\Delta t_{n-1} + 2\Delta t_n(\gamma - 1) + \Delta t_n \Delta t_{n-1}^2 (\gamma - 2\beta)\omega^2],$$

$$A_{12} = -2\omega^2\{(1+\alpha)^2(\Delta t_n + \Delta t_{n-1}) - \Delta t_n \Delta t_{n-1}[-(1+\alpha)\beta\Delta t_{n-1} + (\Delta t_{n-1} + \alpha(\Delta t_n + \Delta t_{n-1}))\gamma]\omega^2\},$$

$$A_{13} = 2\omega^2\{-(1+\alpha)^2 + \Delta t_n[\alpha(1+\alpha)\beta\Delta t_{n-1} + (\Delta t_{n-1} + \alpha(\Delta t_n + \Delta t_{n-1}))\gamma]\omega^2\},$$

$$A_{21} = [\Delta t_{n-1} + \alpha(\Delta t_n + \Delta t_{n-1})] \cdot \{-2(1+\alpha)(\gamma-1) - \Delta t_{n-1}[\Delta t_n - 2\beta\Delta t_n + (1+\alpha)\Delta t_n(\gamma - 2\beta)]\omega^2\},$$

$$A_{22} = 2\{(1+\alpha)^2 - [\alpha\Delta t_n^2 + \Delta t_n \Delta t_{n-1}(1+\alpha) - \Delta t_{n-1}^2(\beta + 2\alpha\beta + \alpha^2\beta) + (1+\alpha)\Delta t_{n-1}(\Delta t_{n-1} + \alpha(\Delta t_n + \Delta t_{n-1}))\gamma]\omega^2\},$$

$$A_{23} = 2\{-(1+\alpha)^2[\Delta t_n + (\Delta t_{n-1} + \alpha(\Delta t_n + \Delta t_{n-1}))\gamma]\omega^2 + \alpha\beta\Delta t_n^2 \Delta t_{n-1}\omega^4\},$$

$$A_{31} = [\Delta t_{n-1} + \alpha(\Delta t_n + \Delta t_{n-1})][-(1+\alpha)(2\beta-1)\Delta t_{n-1} - 2\Delta t_n(\gamma-1) + \Delta t_n \Delta t_{n-1}^2(2\beta - \gamma)\omega^2],$$

$$A_{32} = 2\{(1+\alpha)^2(\Delta t_n + \Delta t_{n-1}) - \Delta t_n \Delta t_{n-1}[-(1+\alpha)\beta\Delta t_{n-1} + (\Delta t_{n-1} + \alpha(\Delta t_n + \Delta t_{n-1}))\gamma]\omega^2\},$$

$$A_{33} = 2\big((1+\alpha)^2 - \Delta t_n(\alpha(1+\alpha)\beta\Delta t_{n-1} + (\Delta t_{n-1} + \alpha(\Delta t_n + \Delta t_{n-1}))\gamma)\omega^2\big).$$

First, we want to study the influence of the parameter α in the Composite scheme. In the Newmark-Euler scheme this parameter describes the time-split point, but in the Composite method the time-step sizes of the Newmark and Euler-Backward step may be varied independently from each other. Fig. 6.9 shows that the parameter α directly influences the spectral radius of the algorithm and therefore its stability. To find the limit for α which guarantees unconditional stability, we firstly compute the eigenvalues of (6.26) for $\gamma = 0.5$, $\beta = 0.25$ and $\Delta t_n = \Delta t_{n-1} = \Delta t$. By substituting again $\omega = 2\pi/T$ we get

$$\lambda_1 = 0,$$

$$\lambda_{2,3} = \frac{(1+\alpha)^2 - [5 + 3\alpha(4+\alpha)]\left(\frac{\Delta t\pi}{T}\right)^2 \pm 2\pi\sqrt{-\left[(1+3\alpha)\left(\frac{\Delta t}{T}\right)^3\pi^2 - 2\frac{\Delta t}{T}(1+\alpha)^2\right]^2}}{\left[1+\left(\frac{\Delta t\pi}{T}\right)^2\right]\left[(1+\alpha)^2 + 4\left(\frac{\Delta t\pi}{T}\right)^2\right]} \tag{6.27}$$

as shown in Appendix C. Unconditional stability which means stability independent of the used time step-size is reached for

$$0 \le \alpha \le \frac{1}{2} \tag{6.28}$$

if $\gamma = 0.5$ and $\beta = 0.25$ are chosen as proved in Appendix D.

An additional variation of the parameters γ and β leads to much more complex stability conditions where Fig. 6.10 gives an impression about it. Even if $\alpha = 0.5$ is chosen, the remaining parameters γ and β define a rather complex space of stable parameter combinations as revealed in Fig. 6.11. For this reason it is suggested to use e.g. the

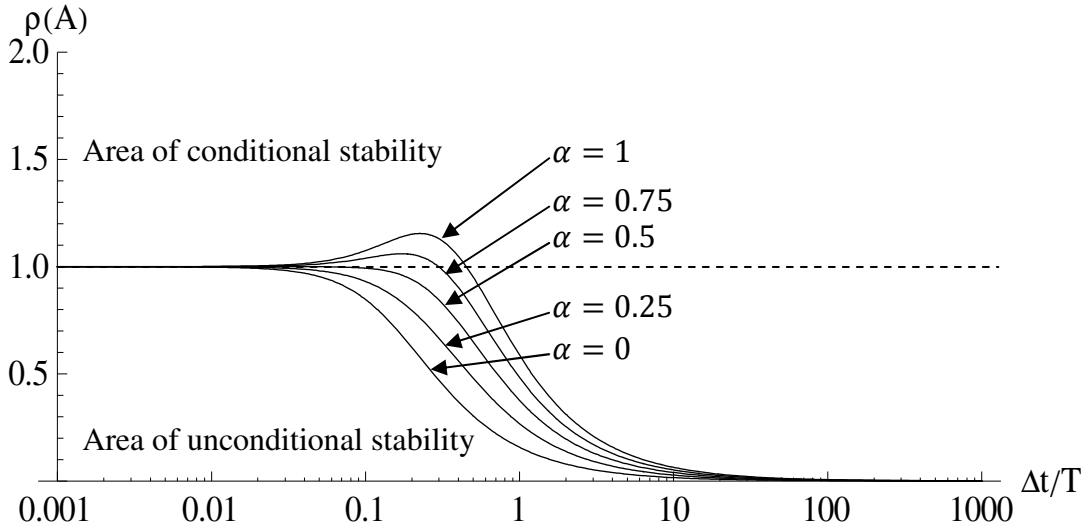


Fig. 6.9 Influence of parameter α on spectral radius of Composite time-integration algorithm for $\gamma = 0.5$, $\beta = 0.25$ and $\Delta t_n = \Delta t_{n-1} = \Delta t$

unconditionally stable parameter combination $\alpha = 0.5$, $\gamma = 0.5$ and $\beta = 0.25$. For arbitrary parameter combinations, a general statement about the stability for arbitrary time-step sizes is not possible, and even if an analytical description of the stable parameter space would exist, this description would be far too complex for practical use. Hence, a specific choice of a parameter combination should always be checked for stability by plotting the spectral radius of this combination.

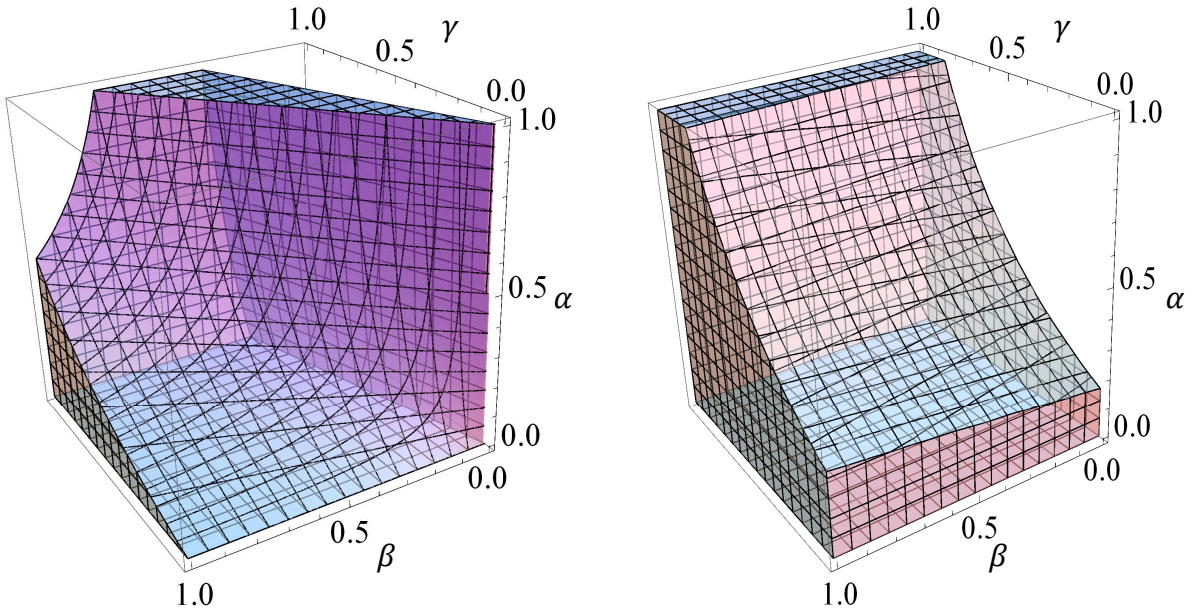


Fig. 6.10 The α, γ, β-parameter space of the Composite algorithm satisfying stability condition (6.8) with $\Delta t_n = \Delta t_{n-1} = \Delta t$ for $\Delta t/T \to \infty$ (left) and for $\Delta t/T = 0.1$ (right)

Finally, some attention should be drawn to the influence of the difference between the time-step sizes Δt_{n-1} and Δt_n on the spectral radius and thus the stability. The contour plot in Fig. 6.12 shows the spectral radius, indicated by the color, depending on all possible combinations

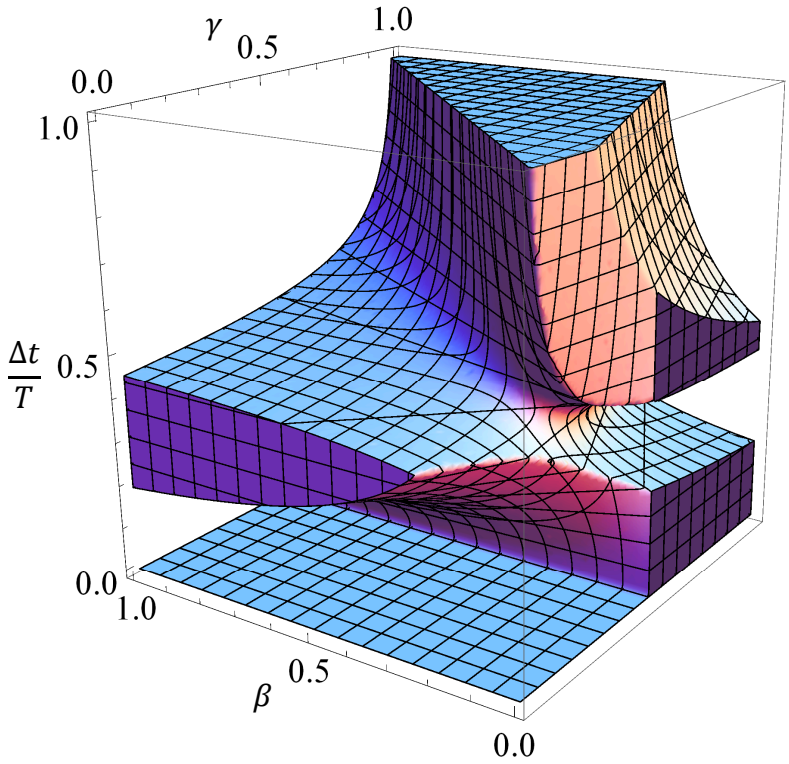


Fig. 6.11 Space of parameters γ and β which satisfy condition (6.8) for $\alpha = 0.5$ in dependence of the normalized time-step size $\Delta t/T$ with $\Delta t = \Delta t_n = \Delta t_{n-1}$

of Δt_{n-1} and Δt_n in the range $0 < \Delta t_{n-1}/T \leq 1$ and $0 < \Delta t_n/T \leq 1$. The blue line indicates combinations of $\Delta t_{n-1}/\Delta t_n$ on the stability border $\rho(\boldsymbol{A}) = 1$. As it can be seen, a strong decrease of the time-step size Δt_n compared to Δt_{n-1} results in instability due to a spectral

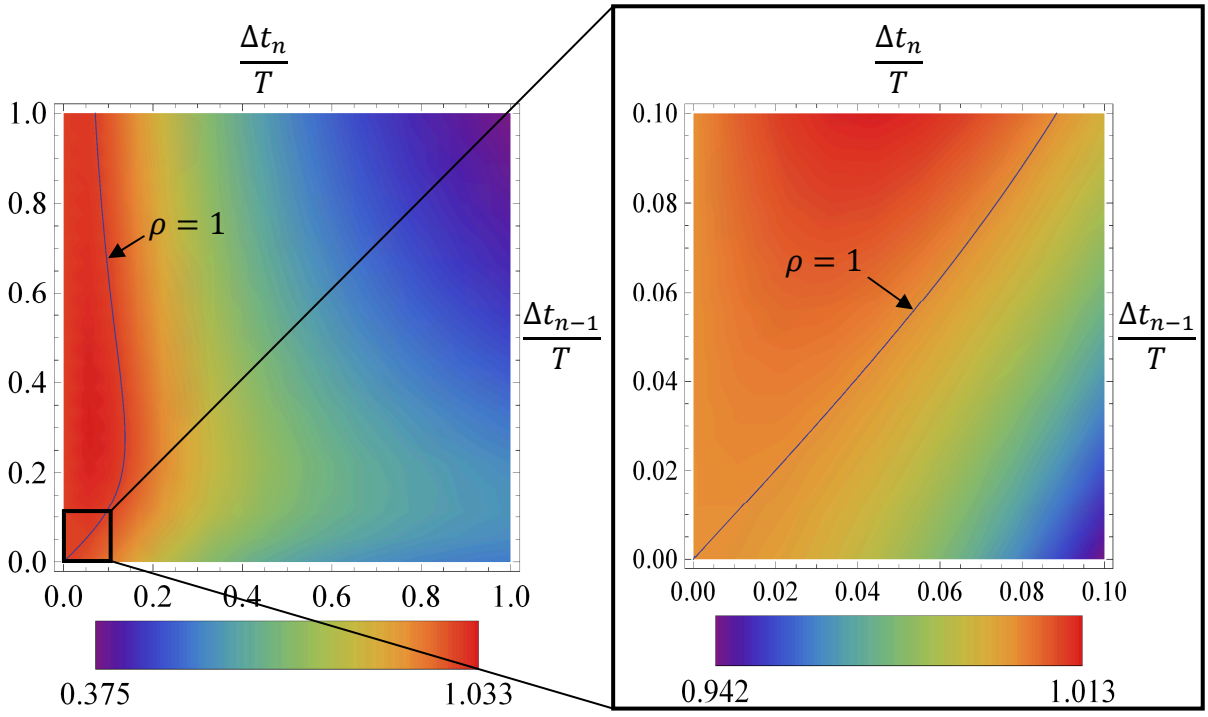


Fig. 6.12 Spectral radius of Composite scheme for $\alpha = 0.5$, $\gamma = 0.5$, $\beta = 0.25$ and varying normalized time step-sizes $\Delta t_n/T$ and $\Delta t_{n-1}/T$

radius slightly above one which, however, is not critical since such strong changes in the time-step size happen very rarely. The time-step size changes, for example by an automatic time-step control algorithm, are not very big in practical applications. Mostly, the time step-size of a following time-step is bounded like $0.5\Delta t_n \leq \Delta t_{n+1} \leq 1.5\Delta t_n$. Hence, the solution cannot become unstable. An increase of the size of the 3-Point-Euler-Backward step would cause more numerical damping and is therefore uncritical too.

6.1.4 Central Difference Method

To compute the amplification matrix for the central difference method, we substitute Equation (5.19) in (6.3) and add the trivial equation $x_n = x_n$ which yields the recursion scheme

$$\begin{bmatrix} x_{n+1} \\ x_n \end{bmatrix} = \underbrace{\begin{bmatrix} 2-\Delta t^2\omega^2 & -1 \\ 1 & 0 \end{bmatrix}}_{\boldsymbol{A}_{CDM}} \cdot \begin{bmatrix} x_n \\ x_{n-1} \end{bmatrix} \tag{6.29}$$

corresponding to Equation (6.2). The eigenvalue problem

$$det(\lambda \boldsymbol{E} - \boldsymbol{A}_{CDM}) = 0 \tag{6.30}$$

yields

$$\lambda^2 + \lambda(\Delta t^2\omega^2 - 2) + 1 = 0 \tag{6.31}$$

with the two solutions and eigenvalues of matrix $\boldsymbol{A}_{CDM}$

$$\begin{aligned} \lambda_{1,2} &= \frac{2-\Delta t^2\omega^2}{2} \pm \sqrt{\frac{(\Delta t^2\omega^2-2)^2}{4} - 1} \\ &= 1 - 2\pi^2\left(\frac{\Delta t}{T}\right)^2 \pm \sqrt{\left[1 - 2\pi^2\left(\frac{\Delta t}{T}\right)^2\right]^2 - 1}\,. \end{aligned} \tag{6.32}$$

A plot of the maximum absolute values of $\max\{|\lambda_1|, |\lambda_2|\}$ over the normalized time-step size shows an interesting effect in Fig. 6.13. The spectral radius of the matrix $\boldsymbol{A}_{CDM}$ equals one up to a certain time-step size and grows unbounded beyond this critical time-step size. As shown in Appendix E the stability conditions (6.8), (6.9) are satisfied only for

$$\Delta t \leq \frac{2}{\omega} = \frac{T}{\pi} \quad \Rightarrow \quad \frac{\Delta t}{T} \leq \frac{1}{\pi}. \tag{6.33}$$

A solution obtained by the central difference method will grow unbounded if the time-step size becomes bigger than that critical time-step size. For practical applications the maximum time-step size of the central difference method should be below this critical limit because also the amplitude error and the period elongation increase as the time-step size comes closer to the critical time-step size (see also Sections 6.3 and 6.4). With this information we are now able to explain the behavior of the numerical solution in Fig. 5.21. The critical time-step size for this example is $\Delta t_{critical} = 2/\omega = 2s$, which results in the stable behavior of the solution for $\Delta t = 1.9s$ and unstable behavior for $\Delta t = 2.1s$.

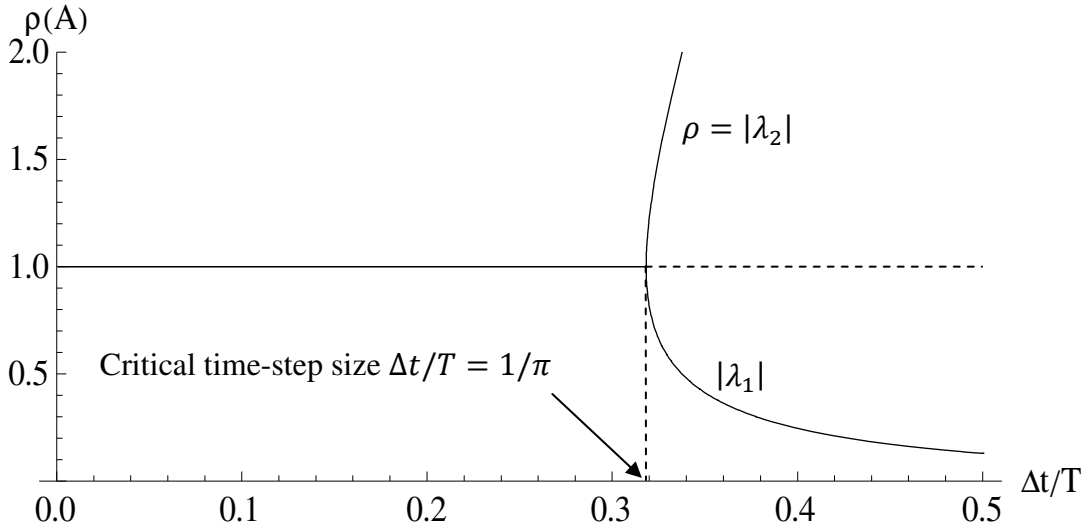


Fig. 6.13 Spectral radius of central difference method with critical time-step size

For a more complex structure, like typically used in FEM computations, the question about the highest frequency and following from that the critical time-step size may not be answered as simply as for the 1-DOF vibration system. Let us assume that in a FEM computation the highest frequency of a model is limited by the membrane mode of a rod, shell or solid element, respectively, because the membrane stiffness is usually much higher than the bending stiffness due to the much better utilization of material for membrane loading than for bending. The derivation of the critical time-step is demonstrated here for the example of a single rod element, but it is similar for other element types (shells, solids).

A rod element is defined by its length l and cross section area A as well as elastic modulus E and density ρ of the material. Typically, the mass $m = Al\rho$ of the element is distributed equally to its nodes (Fig. 6.14).

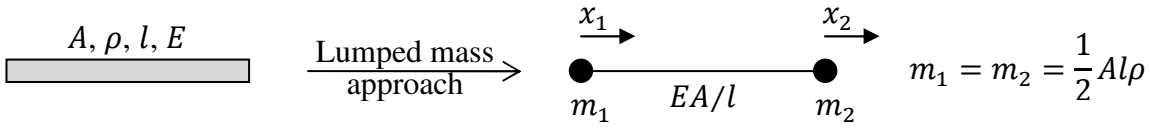


Fig. 6.14 Rod element and mass distribution with lumped mass approach

To calculate the eigenfrequency of the element we use the equation of motion

$$\boldsymbol{M}\ddot{\boldsymbol{x}} + \boldsymbol{K}\boldsymbol{x} = \boldsymbol{0} \tag{6.34}$$

with mass matrix

$$\boldsymbol{M} = \begin{bmatrix} \frac{1}{2}Al\rho & 0 \\ 0 & \frac{1}{2}Al\rho \end{bmatrix} \tag{6.35}$$

and stiffness matrix

$$\boldsymbol{K} = \frac{EA}{l}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}. \tag{6.36}$$

For the analytical solution of (6.34) we use the approach $\boldsymbol{x} = \boldsymbol{\phi} e^{i\omega t}$ resulting in $\ddot{\boldsymbol{x}} = -\omega^2 \boldsymbol{\phi} e^{i\omega t}$. Applied to Equation (6.34) this gives

$$(\boldsymbol{K} - \boldsymbol{M}\lambda)\boldsymbol{\phi} = \boldsymbol{0} \tag{6.37}$$

with $\lambda = \omega^2$. A non-trivial solution exists for

$$det(\boldsymbol{K} - \boldsymbol{M}\lambda) = \left(\frac{EA}{l} - \lambda\frac{Al\rho}{2}\right)^2 - \left(\frac{EA}{l}\right)^2 = \lambda^2\left(\frac{Al\rho}{2}\right)^2 - \lambda EA^2\rho = 0. \tag{6.38}$$

Solving (6.38) for λ leads to $\lambda_1 = 0$, which corresponds to the rigid body mode of the rod, and

$$\lambda_2 = \frac{4E}{l^2\rho}. \tag{6.39}$$

The maximum eigenfrequency of the rod element is thus

$$\omega_{max} = \sqrt{\lambda_2} = \frac{2}{l}\sqrt{\frac{E}{\rho}}. \tag{6.40}$$

With the stability condition (6.33) we get from (6.40)

$$\Delta t \leq l\sqrt{\frac{\rho}{E}}. \tag{6.41}$$

The expression $\sqrt{E/\rho}$ describes the wave propagation speed c in an elastic rod [112]. By this, we may also write

$$\Delta t \leq \frac{l}{c} \tag{6.42}$$

which is called the Courant-Friedrichs-Lewy criterion (CFL criterion) [34]. From this perspective, Δt is the time that a wave needs to propagate through a rod of length l.

For a three-dimensional elastic solid, the lateral contraction effect has to be taken into account [56] and the wave propagation speed becomes

$$c = \sqrt{\frac{E(1-\nu)}{(1+\nu)(1-2\nu)\rho}} = \sqrt{\frac{E}{\rho}}\sqrt{\frac{1-\nu}{(1+\nu)(1-2\nu)}}. \tag{6.43}$$

The influence of the Poisson's ratio ν on the wave propagation speed (and thus the critical time-step size) increases with growing ν as it can be seen in Fig. 6.15.

An overview of critical time-step sizes for different finite element formulations is given in [66]. It is also shown that in general the critical time-step size for higher-order elements is smaller than for linear elements. Also other possibilities of the mass distribution like the consistent mass approach [2] are discussed.

Another aspect of explicit time-integration, that should be mentioned, is the possibility of mass scaling [5,101,115, 140]. Since the critical time-step size is decisively determined by the material density ρ (see Equation (6.41) for instance), an increase of the density increases the critical time-step size. This speeds up the computational time, because bigger stable time

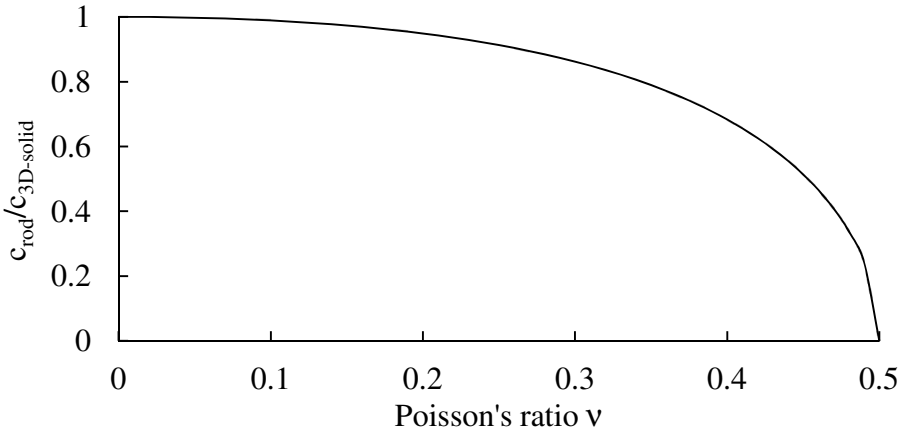


Fig. 6.15 Ratio of wave propagation speeds of rod and solid elements in dependency of Poisson's ratio ν

steps are possible, but on the other hand the mechanical system is changed by the additional mass. In practical applications, mass scaling should only be used with care and especially for parts that are not highly accelerated during the simulation. According to Equation (6.41) also the element length l influences the critical time-step size. By this, also coarser meshes will lead to bigger critical time-steps. In view of this fact, especially in explicit computations, the mesh density should not be higher than necessary.

6.2 Stability of Nonlinear Problems

As already mentioned in the introduction to this section, there is no general concept for the stability analysis of nonlinear problems like the one demonstrated for linear ones. Nevertheless, for particular integration algorithms and parameter combinations some approaches do exist. For example, in [89] a stability analysis for the explicit central difference method and the Runge-Kutta method [28] applied to strain-softening materials is performed. A stability analysis of the Newmark algorithm for structures with nonlinear damping is presented in [95] with the help of an energy approach. Also by using an energy approach, a stability analysis of the Newmark algorithm for a nonlinear material behavior is available in [67] as extension of the results from [22]. It is shown that the Newmark algorithm can suffer instability problems for $\beta \geq 1/4$ in some situations.

In [31], an interesting approach uses methods from control theory to prove the stability of the Newmark algorithm for some special cases (method with constant and linear acceleration and positive tangent stiffness). This is done by using the equation of motion of a one-degree of freedom system at a certain time-step, i.e.,

$$m\ddot{x}_{n+1} + d\dot{x}_{n+1} + kx_{n+1} = F_{n+1} \tag{6.44}$$

and reformulating it as

$$m\ddot{x}_{n+1} + d\dot{x}_{n+1} = F_{n+1} - kx_{n+1} =: L_{n+1}. \tag{6.45}$$

Modeling Equation (6.45) as a control loop results in the closed loop system in Fig. 6.16 as shown in [31].

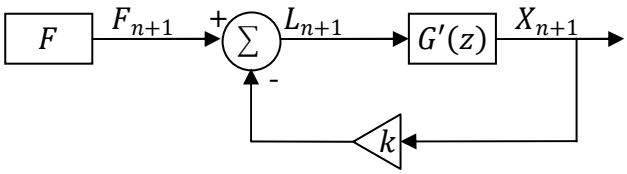


Fig. 6.16 Closed loop system representing Equation (6.45), from [31]

The discrete transfer function $G'(z)$ may be determined by a Z-transformation [54,94,114] of the Newmark equations (5.27) and (5.28) which leads to the transfer function

$$G'(z) = \frac{X(z)}{L(z)} = \frac{n_2' z^2 + n_1' z + n_0'}{d_2' z^2 + d_1' z + d_0'}. \tag{6.46}$$

The coefficients n_i' and d_i' for numerator and denominator are given in [31] and lead to

$$G'(z) = \frac{2\beta\Delta t^2 z^2 + \Delta t^2(2\gamma - 4\beta + 1)z + \Delta t^2(2\beta - 2\gamma + 1)}{(2m + 2\gamma d\Delta t)z^2 + (d\Delta t(2 - 4\gamma) - 4m)z + d\Delta t(2\gamma - 2) + 2m}. \tag{6.47}$$

For the whole closed loop system of Fig. 6.16 the discrete transfer function results from $X = G'(F - kX)$ as

$$G(z) = \frac{X}{F} = \frac{G'(z)}{1 + kG'(z)}. \tag{6.48}$$

A nonlinear stiffness behavior may be taken into account in the equation of motion (6.44) by substituting kx_{n+1} by r_{n+1} which yields

$$m\ddot{x}_{n+1} + d\dot{x}_{n+1} + r_{n+1} = F_{n+1}. \tag{6.49}$$

By writing Equation (6.49) for n instead of $n + 1$ and subtracting both equations, we find for the incremental form with $\Delta\ddot{x}_n = \ddot{x}_{n+1} - \ddot{x}_n$, $\Delta\dot{x}_n = \dot{x}_{n+1} - \dot{x}_n$, $\Delta r_n = r_{n+1} - r_n$ and $\Delta F_n = F_{n+1} - F_n$

$$m\Delta\ddot{x}_n + d\Delta\dot{x}_n + \Delta r_n = \Delta F_n. \tag{6.50}$$

If small time-steps Δt are assumed, the increment of the restoring force can be approximated as

$$\Delta r_n = r_{n+1} - r_n \approx k_t \Delta x_n \tag{6.51}$$

with k_t as the tangent stiffness. This leads together with Equation (6.50) to

$$m\Delta\ddot{x}_n + d\Delta\dot{x}_n = \Delta F_n - k_t\Delta x_n =: \Delta L_n. \tag{6.52}$$

As shown in [31] this equation can be represented by the block diagram in Fig. 6.17, where an additional block $H(z) = \Delta F(z)/F(z) = (z - 1)/z$ is introduced relating the external excitation force F_{n+1} to the incremental force ΔF_n.

From $\Delta X_n = G'(HF - k_t\Delta X_n)$ we receive the closed loop transfer function

$$\frac{\Delta X_n}{F} = H(z)\frac{G'(z)}{1 + k_t G'(z)} \tag{6.53}$$

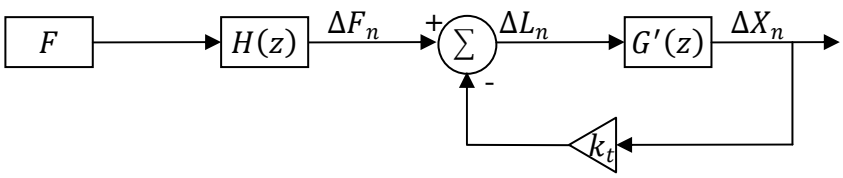


Fig. 6.17 Closed loop system representing an integration algorithm applied to a nonlinear system [31]

similar to Equation (6.48). The stability of the transfer function $H(z)$ is always given, because the pole of $H(z)$ is always zero due to its denominator z (see above). Therefore it is only necessary to consider the closed loop part with the characteristic equation

$$1 + k_t G'(z) = 0 \tag{6.54}$$

leading to

$$1 + k_t \frac{2\beta\Delta t^2 z^2 + \Delta t^2(2\gamma - 4\beta + 1)z + \Delta t^2(2\beta - 2\gamma + 1)}{(2m + 2\gamma d\Delta t)z^2 + (d\Delta t(2 - 4\gamma) - 4m)z + d\Delta t(2\gamma - 2) + 2m} = 0. \tag{6.55}$$

As explained in detail in [31], stability of the integration algorithm is reached when the absolute values of all solutions for z of Equation (6.55) are smaller or equal than 1, i.e.,

$$|z_i| \leq 1 \quad \forall\, i. \tag{6.56}$$

Solving Equation (6.55) for z gives the solution

$$z_{1,2} = \frac{2d\Delta t(2\gamma - 1) + \Delta t^2 k_t(4\beta - 2\gamma - 1) + 4m}{4(d\Delta t\gamma + \beta\Delta t^2 k_t + m)} \pm \frac{\sqrt{\Delta t^2[4d^2 + 4d\Delta t k_t(1 - 2\gamma) + k_t(\Delta t^2 k_t((1 + 2\gamma)^2 - 16\beta) - 16m)]}}{4(d\Delta t\gamma + \beta\Delta t^2 k_t + m)}. \tag{6.57}$$

It can be shown that the absolute z-values are smaller than or equal to one for $d = 0, \gamma = 1/2$ and $\beta = 1/4$ (no external and no numerical damping) for positive k_t [31], which means unconditional stability for the Newmark algorithm. It is also derived that for $d = 0, \gamma = 1/2$ and $\beta = 1/6$ the Newmark algorithm is stable under the condition

$$\frac{k_t}{m}\Delta t^2 = \omega_t^2 \Delta t^2 \leq 12 \tag{6.58}$$

or with $\omega_t = 2\pi/T$ for

$$\frac{\Delta t}{T} \leq \frac{\sqrt{3}}{\pi} \tag{6.59}$$

and positive k_t.

These results can be extended to the more general case of $\beta \geq 1/4$ or $\beta < 1/4$ by evaluating (6.57). Under the assumption of no external or numerical damping ($d = 0, \gamma = 1/2$) we find that the Newmark algorithm is unconditionally stable for $\beta \geq 1/4$ and conditionally stable for $\beta < 1/4$ if

$$\frac{k_t}{m}\Delta t^2 = \omega_t^2 \Delta t^2 \leq \frac{4}{1-4\beta} \quad \text{or} \quad \frac{\Delta t}{T} \leq \frac{1}{\pi\sqrt{1-4\beta}}, \tag{6.60}$$

still under the assumption of a positive tangent stiffness k_t. The same procedure is also performed for the HHT algorithm and leads to the result that this scheme is stable for any positive k_t when $\alpha = -1/3$ is chosen.

Unfortunately the presented methodology for the stability consideration of nonlinear problems is not able to explain all phenomena. In [149] it is shown for an example of two different nonlinear differential equations

$$\ddot{x} + S_1 x(1 + S_2 x^2) = 0 \tag{6.61}$$

and

$$\ddot{x} + S \tanh x = 0 \tag{6.62}$$

that the Newmark algorithm can produce chaotic or unstable solutions in computations of long duration responses. For $S = 100$, $x_0 = 10$ and $\dot{x}_0 = 0$, the solution of Equation (6.62) obtained with Newmark time-integration ($\gamma = 1/2$, $\beta = 1/4$) becomes unstable e.g. for $\Delta t = 0.25$ (which corresponds to $\Delta t/T \approx 0.139$). The solution of the second example problem (6.61) does not grow unbounded when solved with the Newmark algorithm, but produces chaotic results for $S_1 = -0.5$, $S_2 = -1$, $x_0 = 0.5$ and $\dot{x}_0 = 0$.

Referring to the results of [31], the described phenomena should not appear for positive tangent stiffness values $k_t > 0$. Plotting the stiffness expressions

$$k_t = \frac{\partial[S_1 x(1 + S_2 x^2)]}{\partial x} = S_1(1 + 3S_2 x^2) \tag{6.63}$$

for (6.61) and

$$k_t = \frac{\partial(S \tanh x)}{\partial x} = S(1 - \tanh^2 x) \tag{6.64}$$

for (6.62) shows that k_t is always positive for (6.62) as visible in Fig. 6.18b whose solution became unstable, whereas k_t can become negative for (6.61) as visible in Fig. 6.18a which led to chaotic but not unbounded results. This seems to be a contradiction to the stability

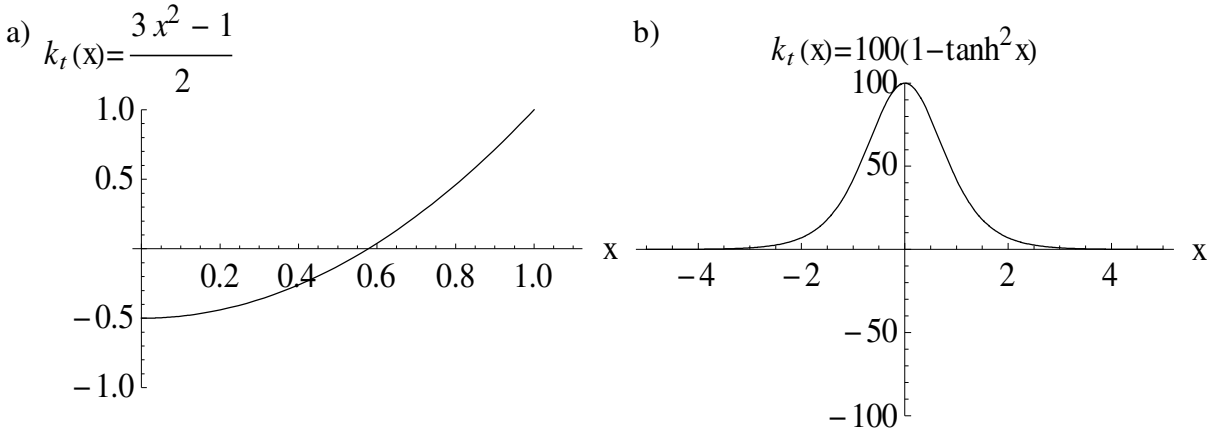


Fig. 6.18 Stiffness $k_t(x)$ of Equations (6.61) and (6.62)

conditions derived with the help of control theory and the closed loop system in Fig. 6.17. But one has to keep in mind that in Equation (6.51) "small" time-steps Δt are assumed for the computation of the tangent stiffness k_t. The instability appeared for a time-step size of $\Delta t/T \approx 0.139$ and it is questionable whether this can be considered to be a "small" time-step size. In particular, in the region of $-1 < x < 1$, the change of the stiffness $\partial k_t/\partial x$ is very big as illustrated in Fig. 6.19.

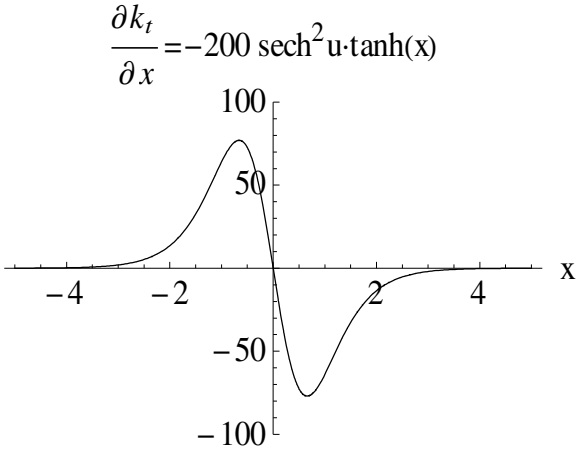


Fig. 6.19 Plot of $\partial k_t/\partial x$ for Equation (6.64)

This example demonstrates that the development of stability criteria for time-integration algorithms used for the solution of nonlinear problems is much more elaborate than for linear problems. Coming back to the differential equation (6.62), it is even more surprising that the Newmark solution becomes stable again if the time-step size is increased to $\Delta t = 0.3$ (which corresponds to $\Delta t/T \approx 0.167$) as shown in [149]. This means on the other hand that a stable solution can become unstable when the time-step size is decreased.

Consequently, the investigation of the stability of time-integration algorithms applied to nonlinear problems is still a subject of scientific research. Also for elastic rotating structures like they appear in aero-engines instability problems may appear, as shown for example in [87] for different algorithms, and will be discussed in detail in Chapter 7.

6.3 Period Elongation

Next we want to consider the effect of the period elongation caused by a time-integration algorithm. For this purpose we compute the numerical solution of Equation (6.3) for one period where $\omega = 2\pi/s$ and compare it to the analytical solution being computed at the same time-points as the numerical ones (see Fig. 6.20). For the initial conditions $x(0) = -1$, $\dot{x}(0) = 0$ the analytical solution becomes

$$x(t) = \sin\left(\omega t - \frac{\pi}{2}\right) = -\cos(\omega t) \tag{6.65}$$

as shown in Equations (5.71)-(5.74). Because every numerical scheme will cause a certain period elongation (or contraction), the numerically computed amplitude value at $t = 1s$ will

not be identical to the analytical solution. This means the numerically computed solution has the form

$$x_{num}(t) = -\cos[(\omega - \varepsilon)t] \tag{6.66}$$

where ε determines the period elongation. For $\omega = 2\pi/s$ and $t = 1s$, Equation (6.66) becomes $x_{num} = -\cos(\varepsilon)$. If we know the value for $x_{num}(t = 1s)$ from the numerical solution, we can determine ε as

$$\varepsilon = \arccos[-x_{num}(t = 1s)]. \tag{6.67}$$

The percentage of period elongation is finally computed as

$$E_p = \frac{\varepsilon}{2\pi} \cdot 100\%. \tag{6.68}$$

In Fig. 6.20 the derived method and the introduced definitions are illustrated for a better understanding.

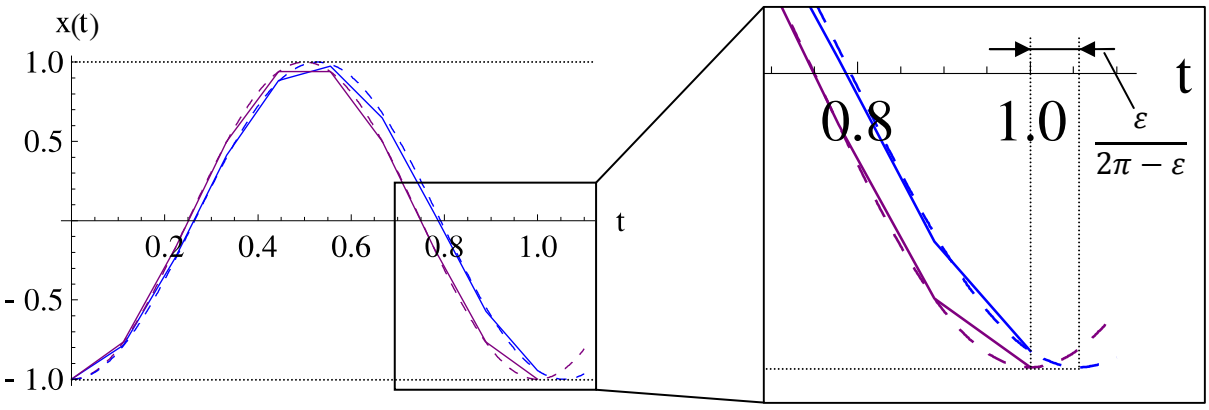


Fig. 6.20 Discrete numerical (blue) and analytical solution (purple) with adjusted cosine functions (dotted lines) and definition of period elongation (right)

Fig. 6.21 shows the percentage of period elongation E_p over the normalized time-step size $\Delta t/T$ for different time-integration algorithms. Obviously the period elongation increases with Δt for all integration schemes. The smallest errors can be expected for the Composite integration scheme (generalization of Newmark-Euler). It should be mentioned that the errors not only depend on the time-step size, but also on the algorithmic parameters α, β and γ.

6.4 Amplitude Error

For the calculation of the amplitude error, the highest positive amplitudes of the discrete numerical and analytical solutions after one period are compared. Since the discrete solution may not have a solution point at the position of its highest amplitude, a cosine function of the form $A \cdot cos(\omega t)$ is fitted to the data points. The amplitude error is then computed for the example problem (6.3) as

$$E_a = (1 - |A|) \cdot 100\%. \tag{6.69}$$

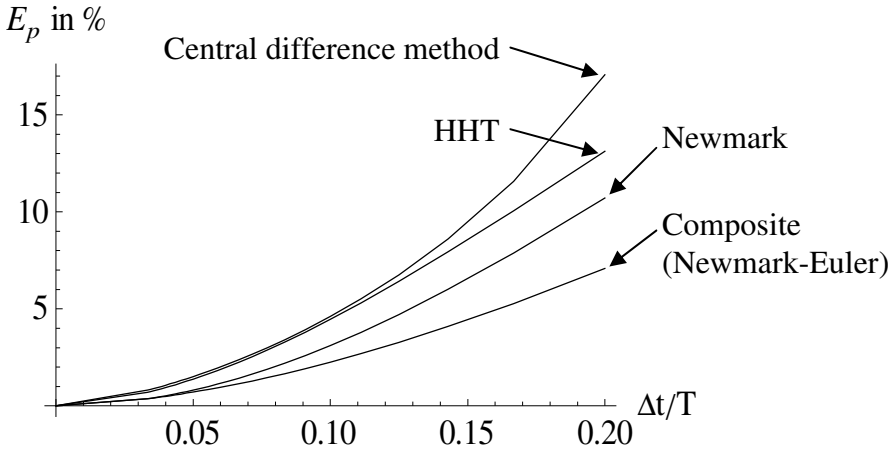


Fig. 6.21 Period elongation for free-vibration problem (6.3) integrated with different time-integration schemes: Central difference method, HHT ($\alpha = -0.1$), Newmark ($\gamma = 0.5, \beta = 0.25$), and Composite ($\alpha = 0.5, \gamma = 0.5$ and $\beta = 0.25$)

It should be noted that this procedure does not work for the central difference method when the time-steps become too big. Due to a strong amplitude increase and decrease of the period for bigger time-steps, the fitting process with the simple cosine function leads to inaccurate results. For this reason a fifth-order polynomial is used for interpolation between the discrete values of the central difference method. The maximum value of this interpolation polynomial f_{ip} is then used to compute the amplitude error E_a as

$$E_{a_{CDM}} = \left|1 - \max\left(f_{ip}\right)\right| \cdot 100\%. \tag{6.70}$$

Fig. 6.22 shows the amplitude errors for example (6.3) and the different time-integration schemes. The amplitude error for the Newmark algorithm equals zero because no numerical damping is introduced for $\gamma = 0.5$ and $\beta = 0.25$.

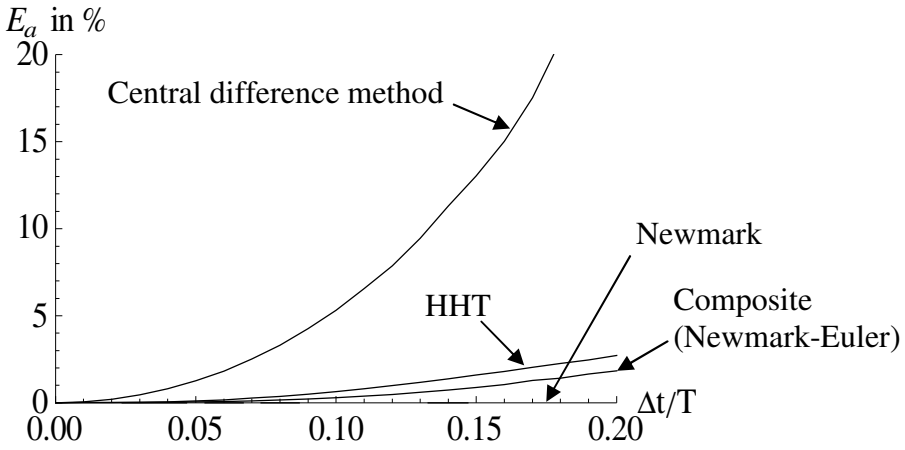


Fig. 6.22 Amplitude errors for free-vibration problem (6.3) integrated with different time-integration schemes: Central difference method, HHT ($\alpha = -0.1$), Newmark ($\gamma = 0.5, \beta = 0.25$), and Composite ($\alpha = 0.5, \gamma = 0.5$ and $\beta = 0.25$)

7 Implicit Time-Integration of Fast Rotating Structures

Especially for long-term simulations (simulation time of a few seconds) of dynamic processes, implicit time-integration algorithms may be more advantageous than explicit strategies due to their bigger time-steps and higher accuracy, and thus lower computational times and more accurate results. In the dynamic simulation of high-fidelity aero-engine models, often simulation times of a few seconds are required. If the goal of such a simulation is to study the behavior of an engine under certain flight maneuvers or conditions with respect to tip clearance behavior for example, under absence of highly dynamic events like fan blade off or bird strike, implicit time-integration promises a faster simulation process. On the other hand, the spinning of the rotor of the engine can be a highly dynamic process too, depending on the rotational velocity, which results in smaller implicit time-steps as we will see in this chapter. But even these small implicit time-steps are some orders of magnitude bigger than the necessary explicit time-steps. The mentioned scenarios fan blade off, bird strike or rubbing events cause extremely small implicit time-steps. Thus, a change of the time-integration algorithm from implicit to explicit during the simulation may make sense too in such cases.

7.1 The Instability Problem

The implicit simulation of rotating structures can cause instability phenomena which is demonstrated here by an example of a rotating plate (Fig. 7.1a). This quadratic plate has the size $a \times b \times t = 200 \times 200 \times 10mm$ and consists of linear elastic material with Young's modulus $E = 115000\,N/mm^2$, Poisson's ratio $\nu = 0.3$ and density $\rho = 4.429 \cdot 10^{-9}\,t/mm^3$ resulting in a mass of $m = 1.7716 \cdot 10^{-3}t$. The translational degrees of freedom at the center nodes of the plate are fixed allowing only rotation about the 3-axis. At its corner segments, the plate is loaded by pressure loads according to Fig. 7.1b following the rotation.

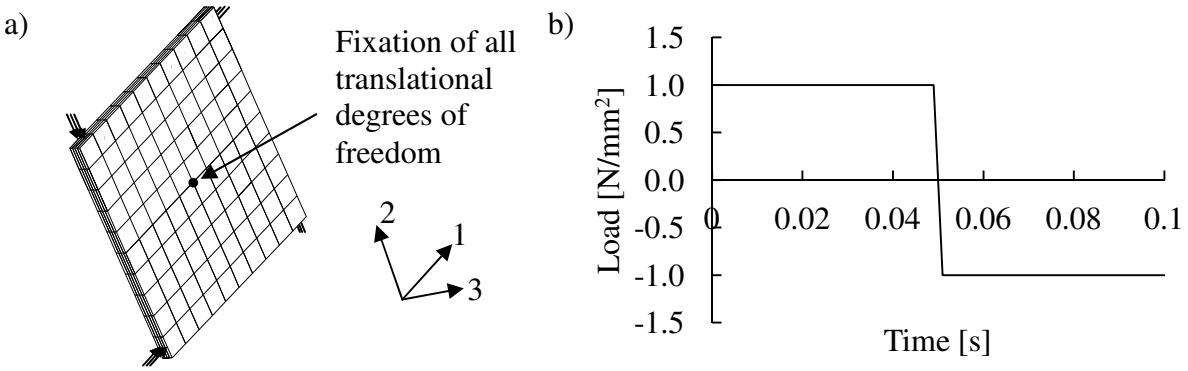


Fig. 7.1 Rotating plate (a) and load curve for rotating plate (b)

At the beginning of this nonlinear transient simulation the plate is at rest. For checking the correctness of the obtained numerical solution, an exact analytical solution is desired. With

the simplification of assuming a rigid plate, an approximate solution for the rotational velocity may be obtained from Euler's law

$$I_3 \dot{\omega}_3 = M_3 \tag{7.1}$$

resulting in

$$\omega_3(t) = \int_0^t \frac{M_3}{I_3} \mathrm{d}t. \tag{7.2}$$

According to Fig. 7.1b and Fig. 7.2a, where the forces are computed from the pressure loads on the corner segments, the torque is given as $M_3(t) = 4 \cdot F(t) \cdot r$ where

$$F(t) = \begin{cases} 200\ N & \text{for } 0 \le t \le 0.049s \\ -2 \cdot 10^5 \frac{N}{s} \cdot t + 10^4 N & \text{for } 0.049s < t < 0.051s \\ -200N & \text{for } 0.051s \le t \le 0.1s. \end{cases} \tag{7.3}$$

With $I_3 = (m/12) \cdot (a^2 + b^2)$ we get

$$\omega_3 = \frac{4r}{I_3} \cdot \begin{cases} 200N \cdot t & \text{for } 0 \le t \le 0.049s \\ -10^5 \frac{N}{s} \cdot t^2 + 10^4 N \cdot t - 240.1Ns & \text{for } 0.049s < t < 0.051s \\ -200N \cdot t + 20Ns & \text{for } 0.051s \le t \le 0.1s. \end{cases} \tag{7.4}$$

A conversion of ω_3 into the unit $rev/time$ is possible by dividing ω_3 by 2π. To obtain the commonly used unit rev/min we have to expand by $60s = 1min$ and get with

$$n = \frac{\omega_3}{2\pi} \cdot \frac{60s}{1min} = \omega_3 \cdot \frac{30\ s/min}{\pi} \tag{7.5}$$

the rotational velocity curve of the rigid plate, plotted in Fig. 7.2b.

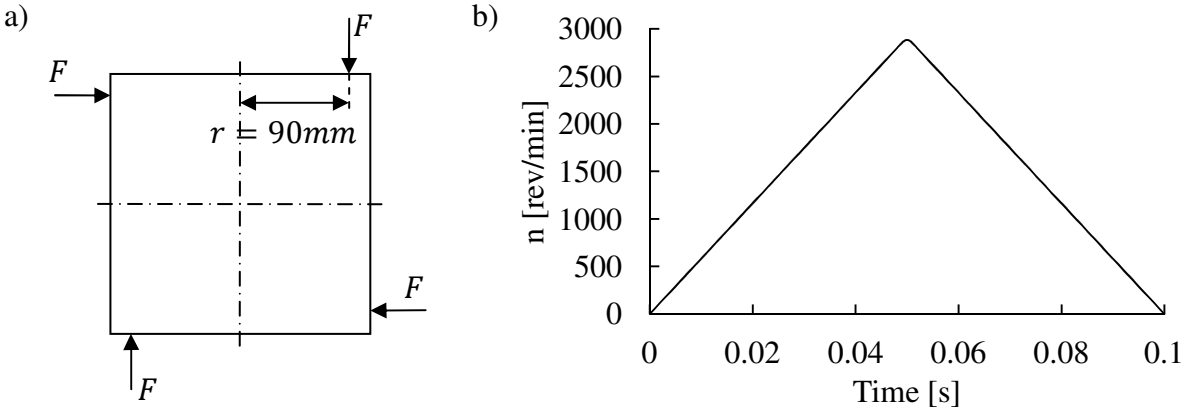


Fig. 7.2 Load application points at plate (a) and approximately computed velocity-time curve (b)

In order to perform a finite element simulation of the problem, the plate is meshed with 10x10x4 underintegrated linear solid elements (Fig. 7.1a). The simulation is carried out with the classical Newmark time-integration scheme in combination with a Newton algorithm for the solution of the resulting non-linear system of equations and a constant time-step size of $\Delta t = 0.001s$. Figure 7.3 shows the resulting velocity-time curves for different parameter

combinations of γ and β. Obviously the simulation with the standard parameters $\gamma = 0.5$ and $\beta = 0.25$ leads to an unstable behavior with extremely high velocities and finally terminates. By increasing the numerical damping ($\gamma = 0.55$, $\beta = 0.2756$) a stabilization of the computation is possible, but the computed result is unphysical and deviates from the expected solution. A further change of the Newmark parameters to $\gamma = 0.5$, $\beta = 0.2756$ leads to a stable and plausible solution. Unfortunately the problem is not solved in general by using $\gamma = 0.5$ and $\beta = 0.2756$, because for small modifications of the model instabilities also appear for this specific choice of Newmark parameters.

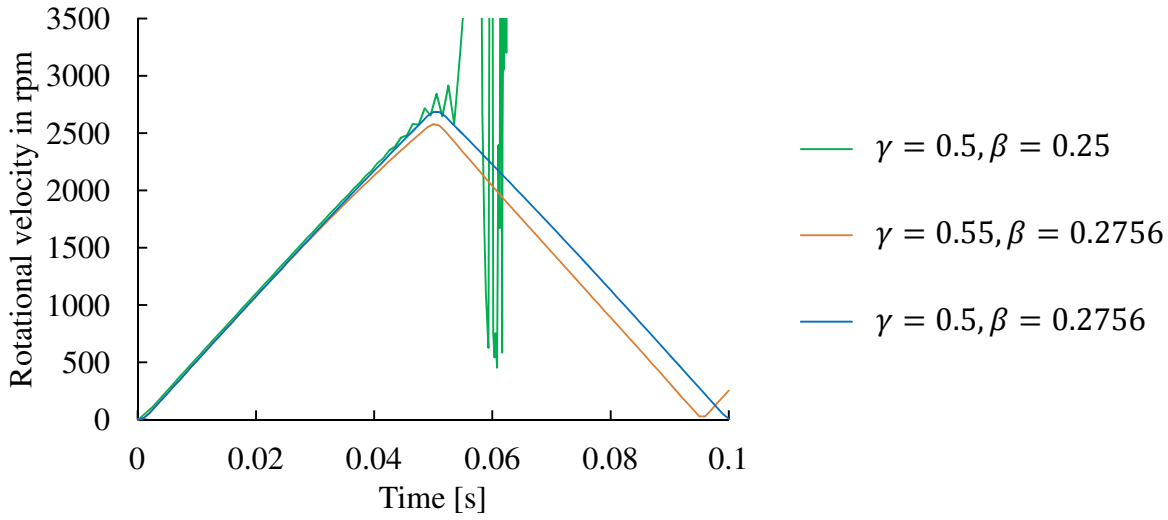


Fig. 7.3 Velocity-time curves resulting from simulation of a spinning plate with different parameters of the Newmark algorithm and constant time-step size $\Delta t = 0.001s$

For more complex structures like a simplified turbine stage of an aero-engine similar problems may appear (Fig. 7.4). The model is fixed at its outer casing region (yellow part), and in the bearing region in the center of the model friction-free contact definitions are used, which allows for a free rotation of the turbine rotor. The pressure on the turbine blades is increased linearly. By this, we would expect a quadratic behavior of the velocity curve. As

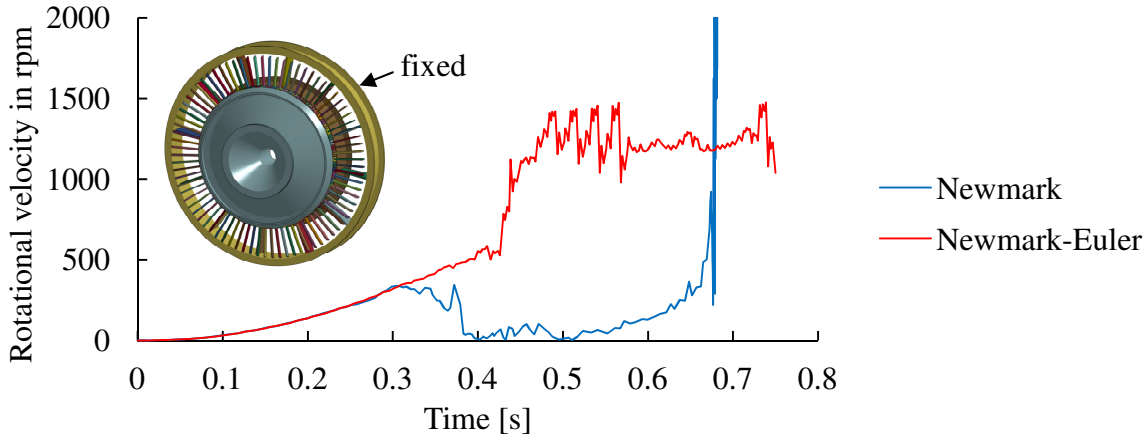


Fig. 7.4 Velocity-time curves resulting from the simulation of a simplified turbine model with different time-integration schemes and variable time-step size

visible in Fig. 7.4, instabilities appear although an automatic time-step size control is activated for both the Newmark algorithm ($\gamma = 0.5$, $\beta = 0.2756$), and the Newmark-Euler algorithm ($\alpha = 0.5$), which in this case and all following computations is the generalized Newmark-Euler scheme (Composite scheme). Actually, the latter one is intended for the simulation of rotating structures [13]. As one can observe, a bit more stability is provided by the Newmark-Euler algorithm, since the instability problem appears later. These instability problems in the implicit simulation of elastic rotating structures are also described in [13,39,55,87,129,130]. In the next section we want to explain the reasons for this unstable behavior and study in detail the numerical problems that are causing the issues based on the simple example of a rotating elastic pendulum.

7.2 Explanation of the Instability Phenomenon

A possible source of the described instability effects could be the time-integration scheme itself. However, as discussed in Chapter 6, the Newmark scheme as well as the Newmark-Euler scheme are unconditionally stable if we assume that the time-steps are small enough for considering the structure to behave linearly between two time-steps. For the spinning plate we used a constant time-step size of $\Delta t = 0.001s$. For a rotational velocity of $2500rpm = 41.67rps$, the period for one rotation is $T = 1/41.67rps = 0.024s$ resulting in a normalized time-step size of $\Delta t/T = 0.04167$. The rotation angle per time-step of the plate changes by $41.67rps \cdot 2\pi \cdot 0.001s = 0.2618rad \mathrel{\widehat{=}} 15°$. Since $\sin(0.2618rad)$ differs only by approximately 1% from its argument and $\cos(0.2618rad)$ from 1 only by approximately 3.5%, we may consider the problem to behave almost linearly between two time-steps. But even if the problem would behave nonlinearly, the Newmark scheme should remain stable with regard to Section 6.2. This means that the time-integration scheme cannot be the source of instability. For a deeper understanding of the numerical situation, it is helpful to use a more simple mechanical model which is why we start our investigations with a rigid rotating pendulum.

7.2.1 Rigid Rotating Pendulum

Usually the description of such a simple rotational problem could be done in polar coordinates with just one degree of freedom (rotation angle). But since finite element codes work with Cartesian coordinates, we also describe the kinematics of the pendulum by Cartesian coordinates (Fig. 7.6a). The equations of motion can be derived from the free-body diagram of the mass in Fig. 7.5. Summation of all forces in x- and y-direction and neglecting gravity leads to

$$m\ddot{x} = F\cos\varphi - F_c\sin\varphi, \tag{7.6a}$$

$$m\ddot{y} = F\sin\varphi + F_c\cos\varphi. \tag{7.6b}$$

With the substitution of the centripetal force $F_c = mv^2/l$ where $v^2 = (\dot{x}^2 + \dot{y}^2)$, $\cos\varphi = -y/l$ and $\sin\varphi = x/l$ we get the equations of motion

$$m\ddot{x} + F \cdot \frac{y}{l} + \frac{mx}{l^2}(\dot{x}^2 + \dot{y}^2) = 0, \tag{7.7a}$$

$$m\ddot{y} - F \cdot \frac{x}{l} + \frac{my}{l^2}(\dot{x}^2 + \dot{y}^2) = 0. \tag{7.7b}$$

Since they have to be combined with the constraint $x^2 + y^2 = l^2$, they are called differential algebraic equations (DAE's). A detailed discussion of this type of equations and solution methods may be found in [4] or [108], where also an overview about further literature is given.

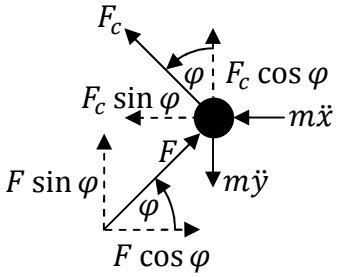


Fig. 7.5 Free-body diagram of pendulums mass point

For $m = 1kg$, $l = 1m$ and $F = 0N$, Equations (7.7) simplify to

$$\ddot{x} + x(\dot{x}^2 + \dot{y}^2) = 0, \tag{7.8a}$$

$$\ddot{y} + y(\dot{x}^2 + \dot{y}^2) = 0. \tag{7.8b}$$

Application of the Newmark time-integration to these nonlinear differential equations with substitutions (5.27) and (5.28) results in

$$\begin{aligned} f_x = {} & \frac{-2x_n + 2x_{n+1} + \Delta t(-2\dot{x}_n + (2\beta - 1)\Delta t\ddot{x}_n)}{2\beta\Delta t^2} \\ & + x_{n+1}\left[\left(\frac{2\beta\Delta t(\dot{x}_n + \Delta t\ddot{x}_n) - \gamma(2x_n - 2x_{n+1} + 2\Delta t\dot{x}_n + \Delta t^2\ddot{x}_n)}{2\beta\Delta t}\right)^2\right. \\ & \left. + \left(\frac{2\beta\Delta t(\dot{y}_n + \Delta t\ddot{y}_n) - \gamma(2y_n - 2y_{n+1} + 2\Delta t\dot{y}_n + \Delta t^2\ddot{y}_n)}{2\beta\Delta t}\right)^2\right] = 0 \end{aligned} \tag{7.9a}$$

and

$$\begin{aligned} f_y = {} & \frac{-2y_n + 2y_{n+1} + \Delta t(-2\dot{y}_n + (2\beta - 1)\Delta t\ddot{y}_n)}{2\beta\Delta t^2} \\ & + y_{n+1}\left[\left(\frac{2\beta\Delta t(\dot{x}_n + \Delta t\ddot{x}_n) - \gamma(2x_n - 2x_{n+1} + 2\Delta t\dot{x}_n + \Delta t^2\ddot{x}_n)}{2\beta\Delta t}\right)^2\right. \\ & \left. + \left(\frac{2\beta\Delta t(\dot{y}_n + \Delta t\ddot{y}_n) - \gamma(2y_n - 2y_{n+1} + 2\Delta t\dot{y}_n + \Delta t^2\ddot{y}_n)}{2\beta\Delta t}\right)^2\right] = 0. \end{aligned} \tag{7.9b}$$

For the solution of this nonlinear system of two equations, a classical Newton algorithm (5.52) is applied which uses the iterative rule

$$\begin{bmatrix} {}^{k+1}x_{n+1} \\ {}^{k+1}y_{n+1} \end{bmatrix} = \begin{bmatrix} {}^{k}x_{n+1} \\ {}^{k}y_{n+1} \end{bmatrix} - J^{-1}({}^{k}x_{n+1}, {}^{k}y_{n+1}) \cdot \begin{bmatrix} f_x({}^{k}x_{n+1}, {}^{k}y_{n+1}) \\ f_y({}^{k}x_{n+1}, {}^{k}y_{n+1}) \end{bmatrix} \tag{7.10}$$

where $J = \begin{bmatrix} \frac{\partial f_x}{\partial x_{n+1}} & \frac{\partial f_x}{\partial y_{n+1}} \\ \frac{\partial f_y}{\partial x_{n+1}} & \frac{\partial f_y}{\partial y_{n+1}} \end{bmatrix}$.

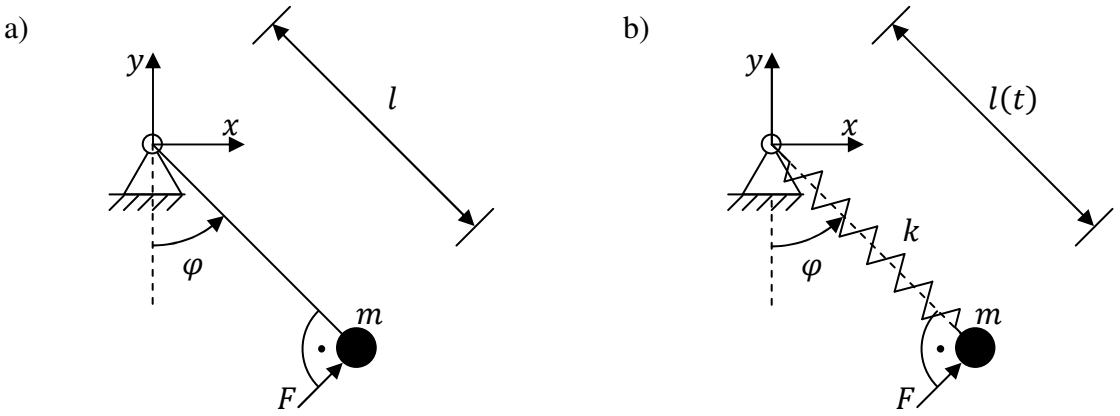


Fig. 7.6 Rotating pendulum with a) rigid and b) elastic connection of mass to ground

The integration of (7.8) is performed for the initial conditions $x(0) = 0m$, $y(0) = -l = -1m$, $\dot{x}(0) = v_0 = 10\,m/s$ and $\dot{y}(0) = 0\,m/s$ with Newmark time-integration and parameters $\gamma = 0.5$ and $\beta = 0.25$. Additionally $\ddot{x}(0) = 0\,m/s^2$, $\ddot{y}(0) = v_0^2/l$ are used as consistent accelerations for the first time-step of the Newmark algorithm. Setting the desired accuracy of the Newton algorithm to

$$|f_x({}^{k+1}x_{n+1}, {}^{k+1}y_{n+1})| < 10^{-5} \;\wedge\; |f_y({}^{k+1}x_{n+1}, {}^{k+1}y_{n+1})| < 10^{-5}, \tag{7.11a}$$

leads for a constant time-step size of $\Delta t = 0.01s$ and a simulation time of $t_{max} = 1s$ to a perfect behavior of the x- and y-coordinates of the mass point plotted in Fig. 7.7a. If the time-

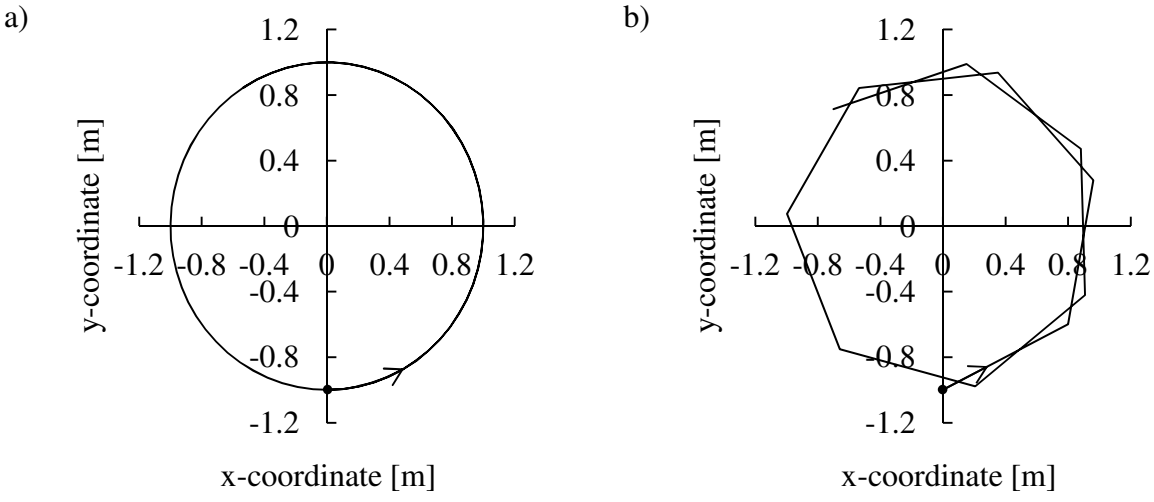


Fig. 7.7 Solutions for rigid pendulum with Newmark algorithm for a) $\Delta t = 10^{-2}s$ and b) $\Delta t = 10^{-1}s$

step size is increased to $\Delta t = 0.1s$, however, the solution loses some accuracy, but still remains stable as shown in Fig. 7.7b. Even for simulation times of more than $t_{max} = 50s$ no instability appears.

7.2.2 Elastic Rotating Pendulum

Let us now consider an elastic pendulum where the mass is connected to the ground via a spring with stiffness k and undeformed length l_0 (Fig. 7.6b). In this case, the centripetal force F_c is replaced by the spring force $F_c = k \cdot (l - l_0)$ and we get for the equations of motion

$$m\ddot{x} + F \cdot \frac{y}{l} + \frac{kx}{l}(l - l_0) = 0, \tag{7.12a}$$

$$m\ddot{y} - F \cdot \frac{x}{l} + \frac{ky}{l}(l - l_0) = 0, \tag{7.12b}$$

where the current length l of the spring can be substituted by $l = \sqrt{x^2 + y^2}$ leading to

$$m\ddot{x} + F \cdot \frac{y}{\sqrt{x^2 + y^2}} + kx \cdot \frac{\sqrt{x^2 + y^2} - l_0}{\sqrt{x^2 + y^2}} = 0, \tag{7.13a}$$

$$m\ddot{y} - F \cdot \frac{x}{\sqrt{x^2 + y^2}} + ky \cdot \frac{\sqrt{x^2 + y^2} - l_0}{\sqrt{x^2 + y^2}} = 0. \tag{7.13b}$$

Like for the rigid pendulum, $m = 1kg$ and $F = 0N$ are used and these equations simplify to

$$\ddot{x} + kx \cdot \frac{\sqrt{x^2 + y^2} - l_0}{\sqrt{x^2 + y^2}} = 0, \tag{7.14a}$$

$$\ddot{y} + ky \cdot \frac{\sqrt{x^2 + y^2} - l_0}{\sqrt{x^2 + y^2}} = 0. \tag{7.14b}$$

Analogously to the numerical solution process for the rigid pendulum, Newmark time-integration is applied which leads to

$$f_x = \frac{-2x_n + 2x_{n+1} + \Delta t(-2\dot{x}_n + (2\beta - 1)\Delta t\ddot{x}_n)}{2\beta\Delta t^2} + kx_{n+1}\frac{\sqrt{x_{n+1}^2 + y_{n+1}^2} - l_0}{\sqrt{x_{n+1}^2 + y_{n+1}^2}} = 0, \tag{7.15a}$$

$$f_y = \frac{-2y_n + 2y_{n+1} + \Delta t(-2\dot{y}_n + (2\beta - 1)\Delta t\ddot{y}_n)}{2\beta\Delta t^2} + ky_{n+1}\frac{\sqrt{x_{n+1}^2 + y_{n+1}^2} - l_0}{\sqrt{x_{n+1}^2 + y_{n+1}^2}} = 0. \tag{7.15b}$$

The initial conditions and Newmark parameters are chosen as above ($x(0) = 0m$, $y(0) = -l(0)$, $\dot{x}(0) = v_0 = 10\,m/s$, $\dot{y}(0) = 0\,m/s$, $\gamma = 0.5$ and $\beta = 0.25$). The consistent accelerations are $\ddot{x}(0) = 0\,m/s^2$ and $\ddot{y}(0) = -ky \cdot (l(0) - l_0)/l = k \cdot (l(0) - l_0)$. The initial length of the spring $l = l(0)$ results from the equilibrium

$$k(l - l_0) = \frac{mv_0^2}{l}, \tag{7.16}$$

leading to the quadratic equation

$$l^2 - l_0 l - \frac{mv_0^2}{k} = 0 \tag{7.17}$$

in l, which has the solutions

$$l_{1,2} = \frac{l_0}{2} \pm \sqrt{\frac{l_0^2}{4} + \frac{mv_0^2}{k}}. \tag{7.18}$$

Since the centrifugal force will always lengthen the spring, only the positive solution remains, which for $l_0 = 1m$ and a spring stiffness of $k = 10^7\ N/m$ yields

$$l(0) = \frac{l_0}{2} + \sqrt{\frac{l_0^2}{4} + \frac{mv_0^2}{k}} = 1{,}00001m. \tag{7.19}$$

For a time-step size of $\Delta t = 0.01s$, standard Newmark parameters $\gamma = 0.5$, $\beta = 0.25$ and the same Newton algorithm as before we get the result in Fig. 7.8a. At the beginning of the simulation, the mass point moves as expected on a circular arc. However, after approximately three quarters of rotation at $t = 0.45s$ the solution "jumps" to an unexpected value and thus becomes somehow unstable. To explain this surprising behavior, it is useful to have a look at the numerical situation right before the jump appears. For this purpose, all relevant variable values at $t = 0.45s$ are listed in Table 7.1.

Table 7.1 Relevant variable values at $t = 0.45s$ before onset of instability

Variable	Value
β	0.25
Δt	$0.01\ s$
k	$10^7\ N/m$
x_n	$-0.9717077973201222\ m$
y_n	$0.23614647740490619\ m$
$\dot{x}_n$	$-2.3517560831270767\ m/s$
$\ddot{x}_n$	$-91.33130927254165\ m/s^2$

With these values it is possible to plot the function $f_x(x_{n+1}, y_{n+1})$ from Equation (7.15a) in Fig. 7.9a. The task for the Newton algorithm is to find the zero(s) of the function $\boldsymbol{f}(x_{n+1}, y_{n+1})$, which are then the equilibrium solutions at time point t_{n+1}. For a better understanding of the function $f_x(x_{n+1}, y_{n+1})$, a cut through this function at the constant value $y_{n+1} = y_n$ is plotted in Fig. 7.9b, which means that for the unknown value of y_{n+1} the value from the previous time-step as approximation is used for visualization purposes. Obviously there is more than one zero for the function $f_x(x_{n+1})$. Actually, we would expect from the Newton algorithm to converge to the solution $x_{n+1} \approx -1$ close to $x_n \approx -0.97$, but as we can see from Fig. 7.9b there are two other zeros of f_x at $x_{n+1} \approx 0$ and $x_{n+1} \approx +1$. According to

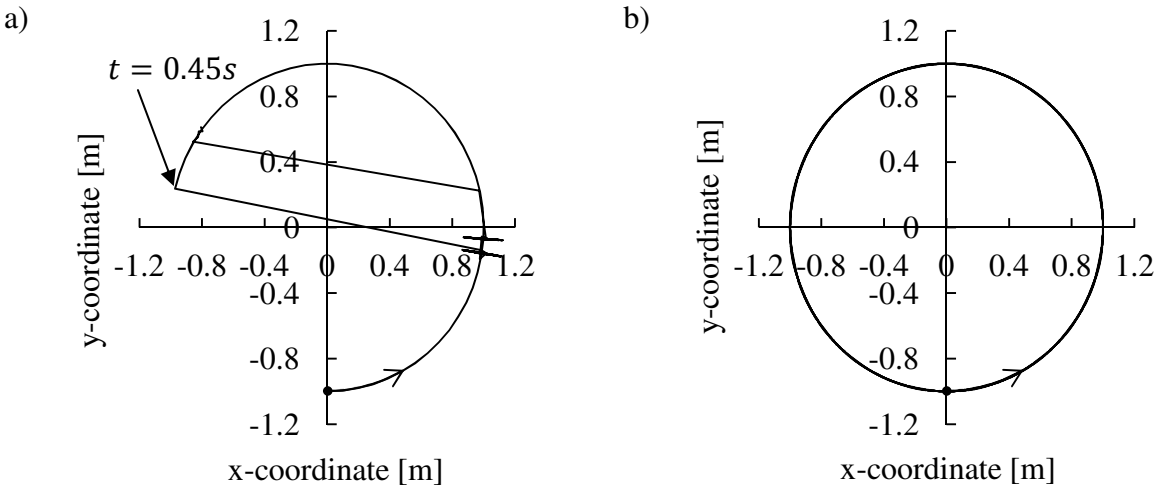


Fig. 7.8 Solutions for elastic pendulum with Newmark algorithm with appearing instability for $\Delta t = 10^{-2}s$ (a) and without instability for $\Delta t = 10^{-3}s$ (b)

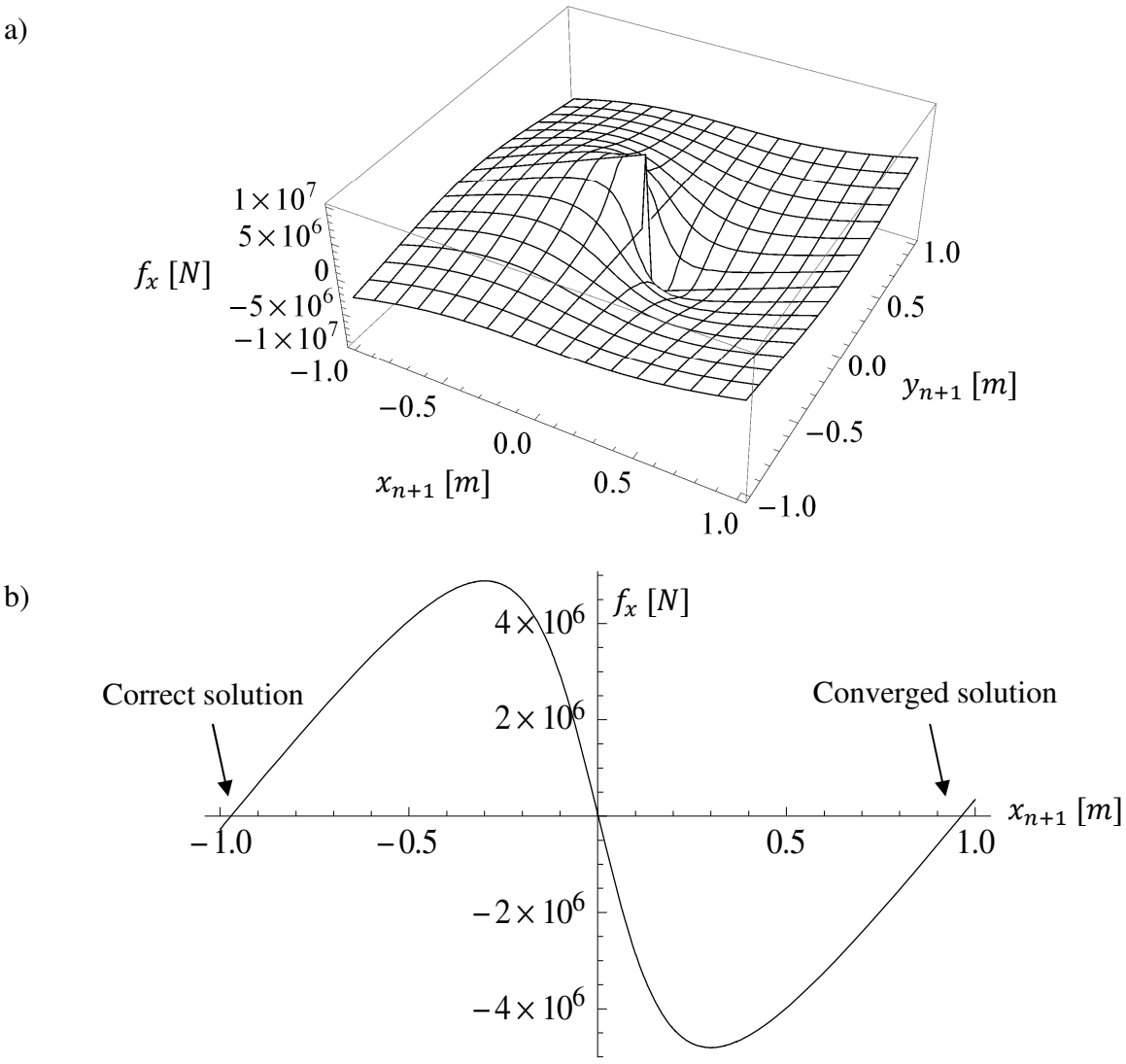


Fig. 7.9 Function $f_x(x_{n+1}, y_{n+1})$ for elastic pendulum at time point $t = 0.45s$ (a) with cut along $y_{n+1} = y_n$ (b) [77]

Fig. 7.8a, obviously the solution $x_{n+1} \approx +1$ was found which means that the Newton algorithm converges into a wrong equilibrium solution causing the "jump" of the mass point.

By this, also the velocities, computed from Equations (5.27) and (5.28) with the wrong value for x_{n+1} are not correct anymore (see Fig. 7.10). Due to the unphysical high displacement $(x_{n+1} - x_n)$ a dramatic change of the velocity is obtained, starting at $t = 0.46s$. This finally leads to the unstable behavior of the whole solution.

To answer the question why the Newton algorithm converges into the wrong equilibrium solution, it is useful to have again a closer look at Fig. 7.9b. In the region of $-1 \leq x_{n+1} \leq 1$ the function has extremely steep slopes. Small errors in the computation of the slope during the Newton iterations (errors always happen when dealing with numerics) have a big influence on the next iterate ${}^{k+1}x_{n+1}$ and finally to x_{n+1}. This also explains the fact that Quasi-Newton algorithms are much more susceptible to this kind of instabilities than Full-Newton algorithms.

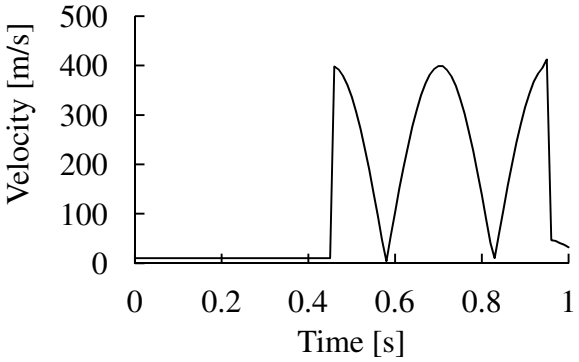


Fig. 7.10 Behavior of velocity $v_n = \sqrt{\dot{x}_n^2 + \dot{y}_n^2}$ of the elastic pendulum's mass point [77]

7.3 Influence of Stiffness and Time-Step Size

Next, we want to study the influence of spring stiffness k and time-step size Δt on the stability behavior of the pendulum. We are doing this by keeping all other values of Table 7.1 unchanged and plotting again $f_x(x_{n+1})$. It is observable in Fig. 7.11a that a change of the time-step size from $\Delta t = 0.01s$ to $\Delta t = 0.001s$ changes the function $f_x(x_{n+1})$ such that only the zero at $x_{n+1} \approx -1$ is left and therefore only one equilibrium state remains. By this, the Newton algorithm can only converge to this correct solution and no "jump" is possible anymore. The same holds for a reduction of the spring stiffness from $k = 10^7\, N/m$ to $k = 10^5\, N/m$ in Fig. 7.11b.

With this knowledge, the simulation of the elastic pendulum is repeated with changed time-step size and changed spring stiffness, respectively. Fig. 7.12 shows the resulting velocity curves $v_n = \sqrt{\dot{x}_n^2 + \dot{y}_n^2}$ for a simulation time of $t_{max} = 30s$. As one can, see there is no unsteady behavior of the curves and therefore no instability as in Fig. 7.10. Consequently, it is possible to stabilize the simulation by reducing the time-step size or decreasing the model

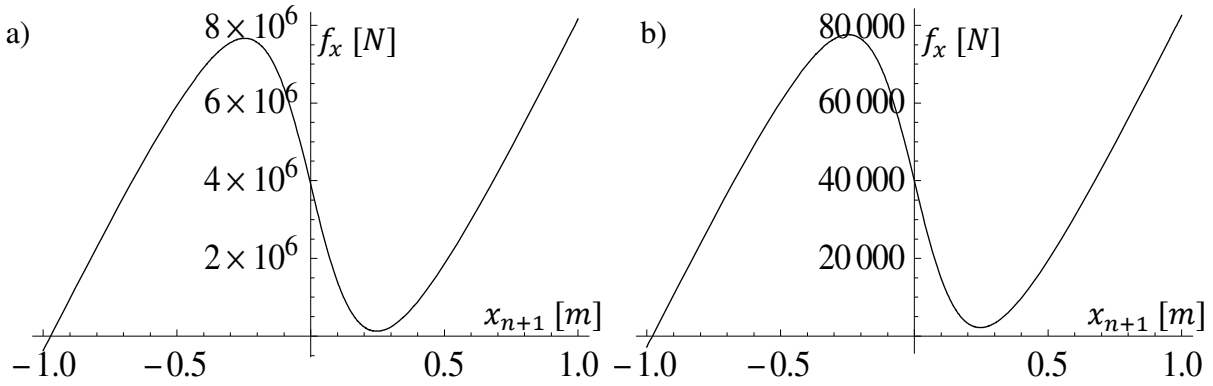


Fig. 7.11 Plot of function $f_x(x_{n+1})$ at $t = 0.45s$ for $k = 10^7\,N/m$ and $\Delta t = 0.001s$ (a) and for $k = 10^5\,N/m$ and $\Delta t = 0.01s$ (b) [77]

stiffness. The latter one shows a remarkable parallel with the mass scaling of explicit time-integration. By decreasing the stiffness of the system it is possible to use bigger time-steps. The influence of the time-step size and stiffness on the function behavior is also visible in Equations (7.15): The smaller the time-step size Δt, the more dominant becomes the first linear term of the function, which reduces the tendency of developing a second zero. The bigger the spring stiffness k becomes on the other hand, the more dominant becomes the second nonlinear term and, therefore, the danger of further zeros. It is interesting to note that for $k \to \infty$ the problem of instabilities does not exist as demonstrated by the example of the rigid pendulum.

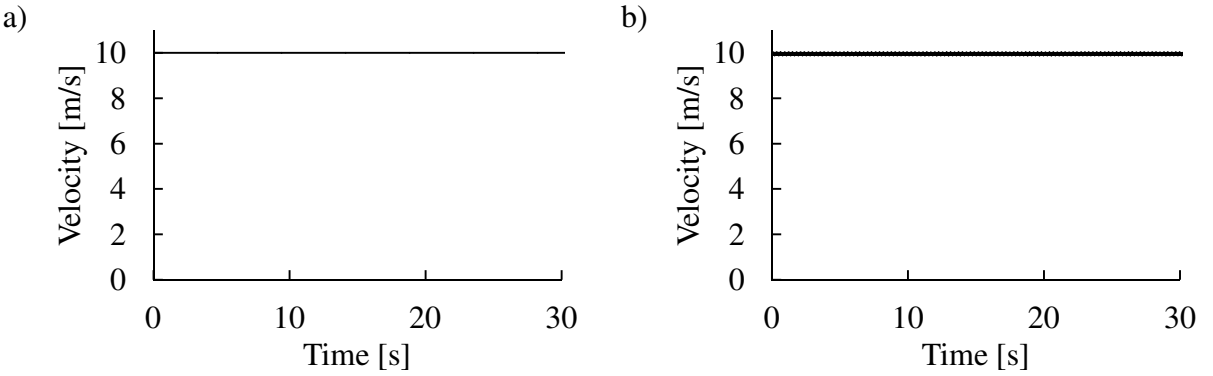


Fig. 7.12 Velocity curves $v_n(t)$ of the mass point of the elastic pendulum with $k = 10^7\,N/m$, $\Delta t = 0.001s$ (a) and $k = 10^5\,N/m$, $\Delta t = 0.01s$ (b) [77]

Other implicit time-integration algorithms such as Newmark-Euler (here the generalized Composite scheme) and HHT also suffer from the same problems. Fig. 7.13a shows the function $f_x(x_{n+1})$ for the same problem at the same time-point as used in Fig. 7.9b, but additionally for the HHT scheme with parameter coupling (5.30)/(5.31) and the Composite scheme. For the latter, the situation of the Euler step is plotted. The behavior of the HHT scheme can be influenced by varying the parameter α. A change of the spring stiffness from $k = 10^7\,N/m$ to $k = 10^5\,N/m$ changes the plotted functions for all schemes and gives better numerical conditions, see Fig. 7.13b.

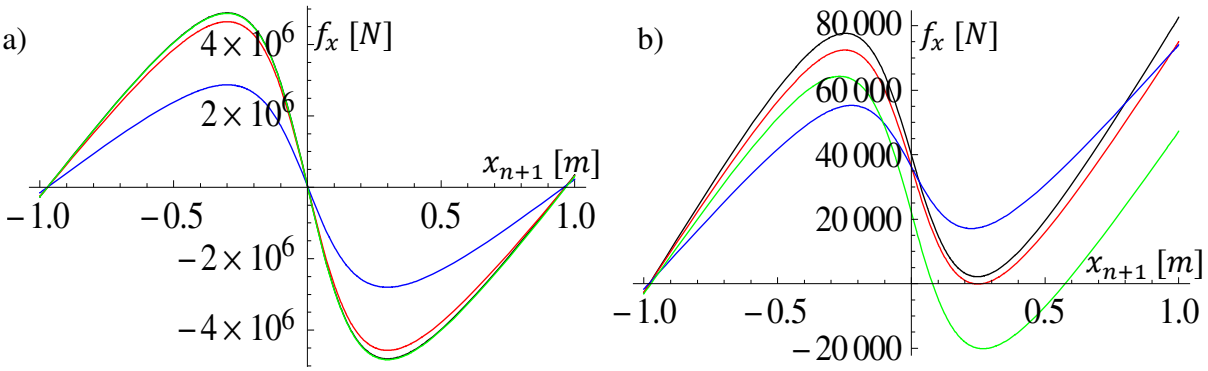


Fig. 7.13 Function $f_x(x_{n+1})$ at $t = 0.45s$ for a) $k = 10^7\,N/m$ and $\Delta t = 0.01s$ and b) for $k = 10^5\,N/m$ and $\Delta t = 0.01s$ for Newmark with $\beta = 0.25$ (black), HHT with $\alpha = -0.05$ (red), HHT with $\alpha = -0.4142$ (blue) and Newmark-Euler time-integration with $\alpha = 0.5$ (green)

7.4 Possibilities for Resolving the Instability Problem

For real structures, the reduction of the stiffness, e.g. by reducing the Young's modulus, to avoid instabilities is no option. Hence, the time-step size remains as the only parameter to influence the stability. In the following, two simple strategies are presented which may prevent the appearance of instabilities.

7.4.1 First Strategy

A hint for unfavorable numerical conditions of the nonlinear equations can be the number of iterations per time-step needed by the Newton algorithm to solve them. Typically a Newton algorithm converges very fast within a few iterations. A high number of iterations may be an indication for problems. In Fig. 7.14 the number of iterations per time-step of the Newton algorithm for the elastic pendulum problem is shown. It can be seen that the number of iterations for $k = 10^7\,N/m$ (dotted line) is much higher than for the stable solution with $k = 10^5\,N/mm$ (solid line). The number of iterations is similarly low, if instead of the pendulum stiffness the time-step size is reduced to $\Delta t = 0.001s$ (not shown).

With this knowledge we are able to construct the following simple time-step size control strategy:

- If the Newton algorithm needs more than a certain number of iterations per time-step without reaching convergence, the current equilibrium search is aborted, time-step size is reduced, e.g. according to

$$\Delta t_{new} = \Delta t_{old} - 0.5 \cdot \Delta t_{old}, \tag{7.20}$$

  and the equilibrium search is restarted with the new reduced time-step size.
- If the number of iterations per time-step is lower than a certain value, the time-step size of the next time-step is increased according to

$$\Delta t_{new} = \Delta t_{old} + 0.5 \cdot \Delta t_{old}. \tag{7.21}$$

It is very important not to use results of a converged Newton iteration with a high number of iterations, because this is a hint for numerical problems and could mean that a wrong equilibrium solution was found. Such results have to be dismissed and the equilibrium search for the actual time-step has to be repeated with the reduced time-step size.

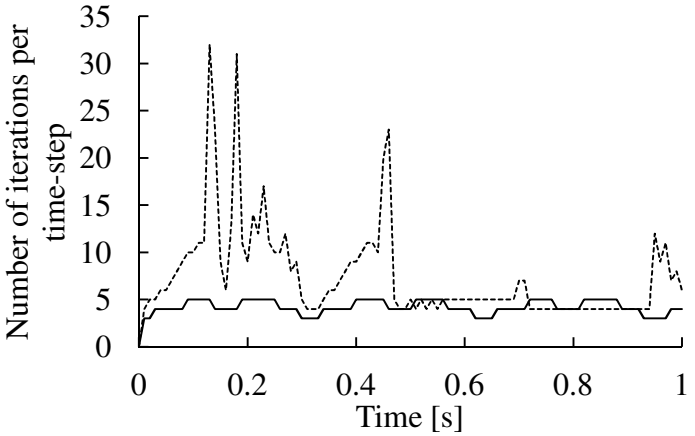


Fig. 7.14 Number of Newton iterations per time-step ($\Delta t = 10^{-2} s$) for the elastic rotating pendulum with $k = 10^7\, N/m$ (dotted line) and $k = 10^5\, N/m$ (solid line) [77]

When this simple approach is implemented for the rotating pendulum, which became unstable before, a stable solution is obtained now even for a simulation time of $t_{max} = 30s$ (Fig. 7.15a). The automatically adjusted time-step size varies in a range of $0.0021s \leq \Delta t \leq \Delta t_0 = 0.01s$ (Fig. 7.15b). The mean value of the used time-step size is $\overline{\Delta t} \approx 0.0036s$. For this particular problem, a maximum number of seven iterations per time-step turned out to be a good choice. If the Newton algorithm does not reach convergence within this limit, the equilibrium search is aborted and restarted with a smaller time-step size according to Equation (7.20). The time-step size is increased with respect to Equation (7.21) if the number of iterations falls below the limit of three.

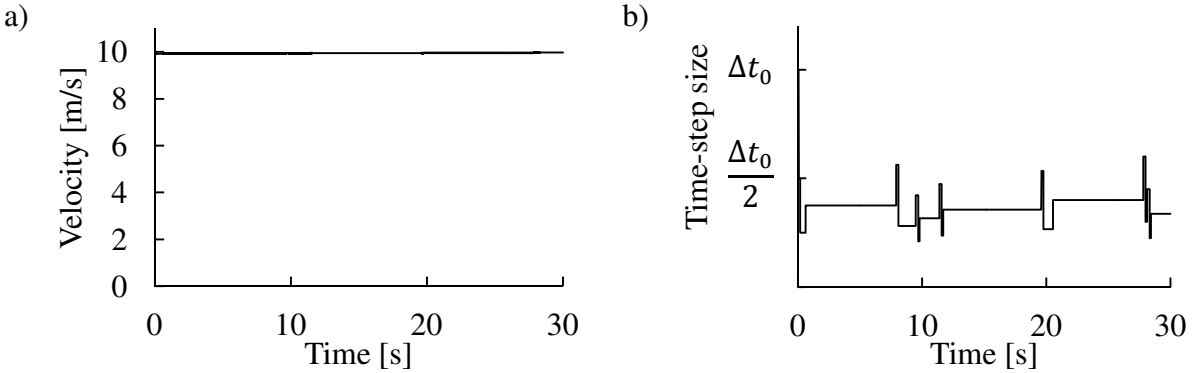


Fig. 7.15 Velocity curve $v_n(t)$ of the mass point of the elastic pendulum with $k = 10^7\, N/m$ and automatic time-step control after first strategy (a) and automatically adjusted time-step size over time (b) [77]

7.4.2 Second Strategy

A second simple strategy for the detection of possible instabilities could be to check the errors of two consecutive Newton iterations with respect to their monotonicity:

- If the error of the $(k+1)$-th Newton iteration is bigger than that of the k-th Newton iteration, this is an indication of possible numerical problems and the equilibrium search should be aborted and restarted with a reduced time-step size according to Equation (7.20).
- If the number of iterations per time-step falls below a certain value (in this case a value of three was used), the time-step size has to be increased as described by Equation (7.21).

Also this strategy leads to a stable solution for the elastic pendulum problem. But since the number of function evaluations is about nine times higher compared to the first strategy, it is slower.

7.5 Finite Element Examples

In the following, it is demonstrated that the findings for the elastic pendulum are transferable to the FEM problems already used in Section 7.1. To reach stability, it is possible to decrease the model stiffness or to use the suggested time-step control strategies. A decrease of the model stiffness is for example possible by reducing Young's modulus of the used material. If we decrease this material parameter for the simple spinning plate problem in Fig. 7.1 from $E = 115000\,N/mm^2$ by a factor of 10 to $E = 11500\,N/mm^2$, the simulation of the rotation stays stable and reasonable results are obtained in Fig. 7.16. Even better results are obtained if the presented first strategy for automatic time-step control is used. In this case, a maximum of seven Newton iterations per time-step is allowed before the equilibrium search is aborted, and the time-step size is increased if the number of iterations is smaller than three.

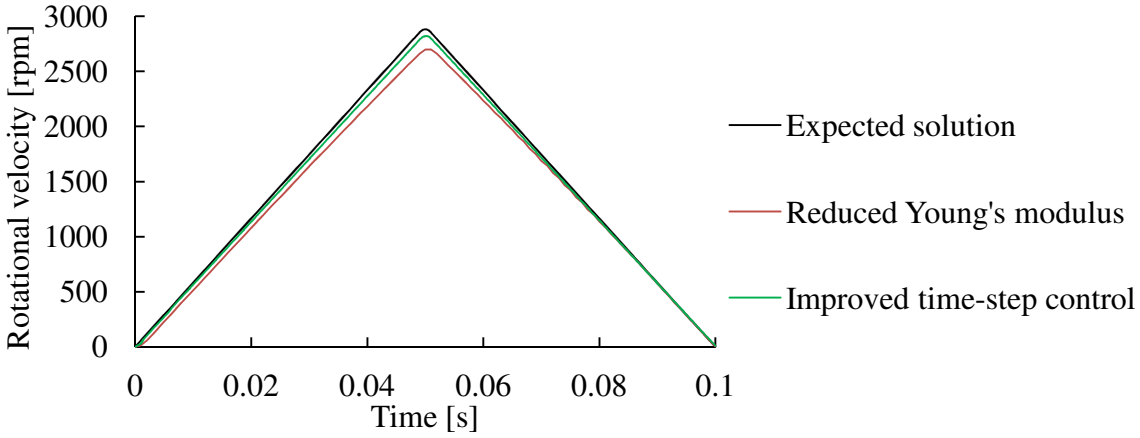


Fig. 7.16 Comparison of numerical solutions obtained with Newmark time-integration for spinning plate model with expected solution

For non-academic examples, the reduction of the model stiffness is of course no option since this influences the overall model behavior and may lead to wrong results in terms of displacements and stresses. Therefore, only the presented time-step control strategy should be used. Figure 7.17 shows that this strategy applied to the turbine model in Fig. 7.4 now leads to correct velocity-time curves for both the Newmark algorithm and the Composite scheme even for rotational velocities up to 20.000 rpm without any instability. After 0.53s the pressure load on the blades is removed and the turbine spins with a constant velocity. It can also be seen that the reached velocity is slightly smaller for the Composite scheme due to the higher numerical damping, although the same loads, boundary conditions and contact formulations are used.

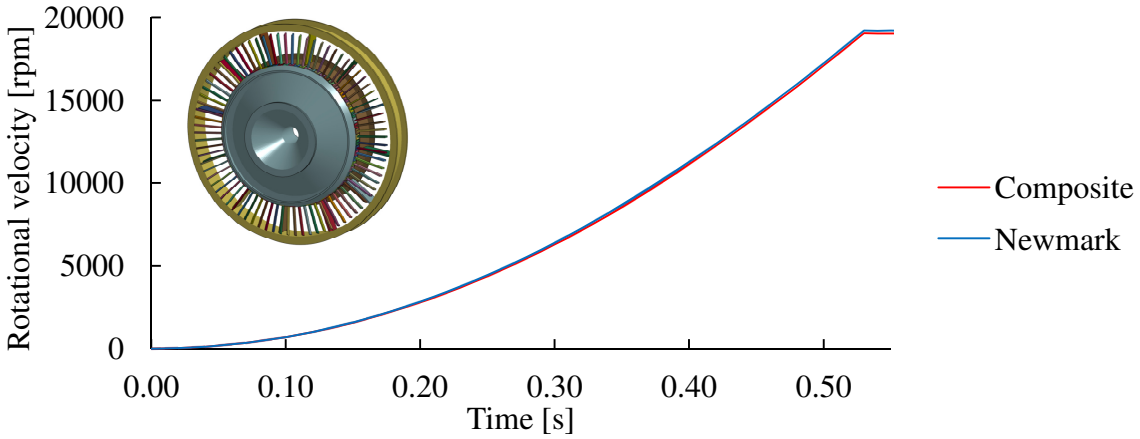


Fig. 7.17 Solutions for complex FEM example of turbine model for improved time-step control strategy and different time-integration algorithms

7.6 Optimization of Time-Integration Parameters

The goal of a FEM simulation is always to get results as fast and accurate as possible. In many cases, accuracy of a computation and reduction of computational time are conflicting objectives. This conflict appears especially for the Newmark-Euler algorithm in transient computations: On the one hand, bigger time-steps promise a reduction of computational time, but on the other hand the numerical damping of the algorithm increases with increasing time-steps (see also Fig. 6.9). With a higher numerical damping, however, an amplitude decay for periodic problems is recognizable, which is again demonstrated in Fig. 7.18 for the simple linear pendulum of Section 5.4.2 and the Newmark-Euler parameters $\alpha = 0.5$, $\gamma = 0.5$ and $\beta = 0.25$. A time step size of $\Delta t = 0.6s$ corresponds to a normalized time-step size of approximately $\Delta t/T = 0.1$. Although the numerical damping is still small for this time-step size (see Fig. 6.9), an amplitude decay is visible. If the time-step size is halved, the amplitude decay is almost not observable at the beginning but for long-term simulations the decay appears, too.

In nonlinear FEM simulations, the time-step size in combination with an automatic time-step control strategy can be influenced by setting different values for the maximum number of

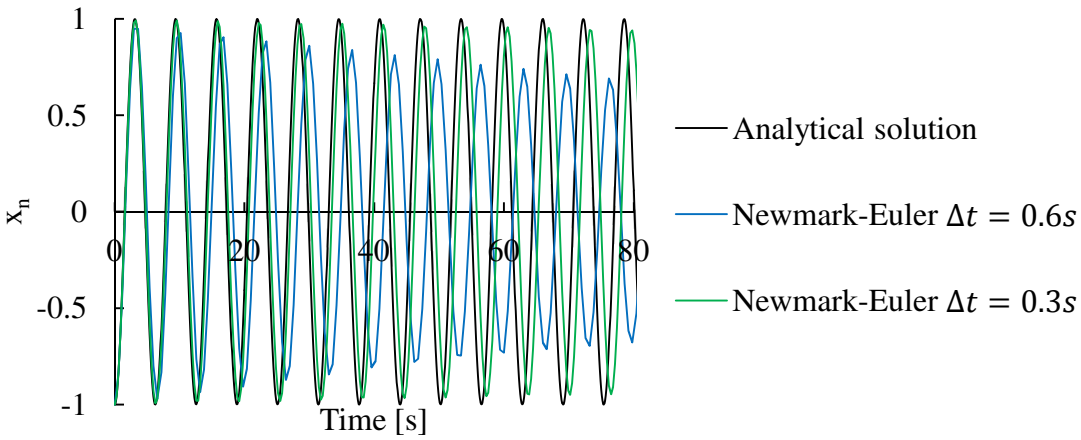


Fig. 7.18 Analytical and numerical solutions of simple linear pendulum problem obtained by Newmark-Euler algorithm for different time-step sizes

allowable Newton (or Quasi-Newton) iterations per time-step. If this value is reached in a simulation, the time-step size is reduced. If the number of iterations is smaller than another certain limit value, the time-step size is increased again. A higher number of allowable iterations typically leads to bigger time-steps, but this does not automatically mean that the computational time decreases, since for these bigger time-steps often more Newton iterations are needed for the equilibrium search. This implies that a tradeoff between time-step size and the necessary number of Newton iterations has to be found for a fast simulation. In particular, for rotating structures the time-step size should not become too big for stability reasons as explained in the sections before. A good compromise between time-step size and necessary number of Newton iterations for the turbine model is a maximum number of 9 Newton iterations. Figure 7.19 shows the time-step size following from this setting and the computed velocity curve, where the load is removed after 0.37s. Although the normalized time-step size at a rotational velocity of $n = 9300\,rpm$ is $\Delta t/T \approx 0.0211$ and thus quite small, the numerical damping observed from the spectral radius in Fig. 7.20 leads to a decrease of the rotational velocity after removal of the load (Fig. 7.19a) which is unphysical.

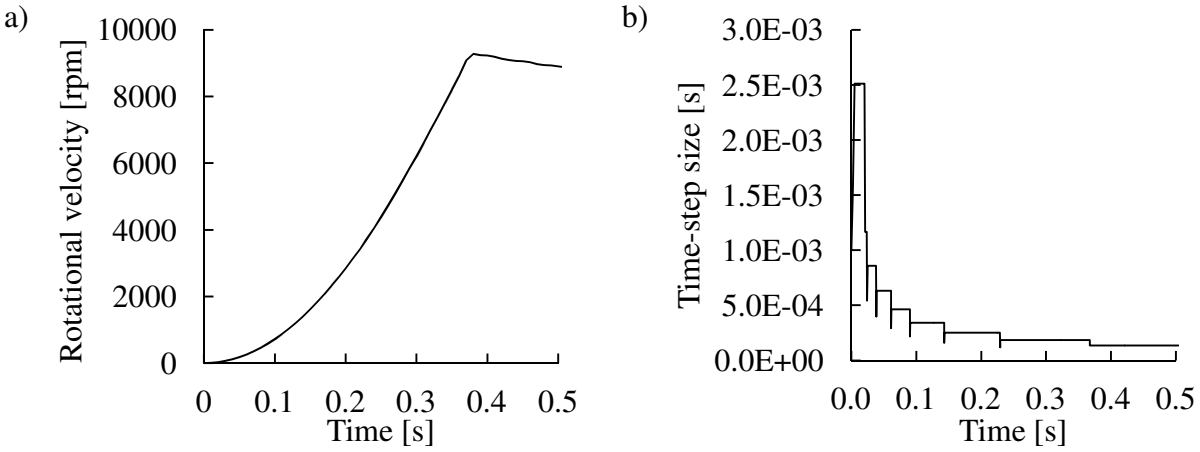


Fig. 7.19 Velocity curve for turbine model simulated with Newmark-Euler algorithm (a) and history of time-step size (b)

As already demonstrated in Section 5.4, the numerical damping of a time-integration algorithm can be influenced by its parameters. An increase of the parameter γ of the Newmark algorithm leads to higher numerical damping and vice versa. Also the parameter α of the Newmark-Euler scheme influences the damping behavior (see Fig. 6.9). Consequently, it should be possible to adjust these time-integration parameters such that the numerical damping is reduced while keeping sufficient stability of the algorithm.

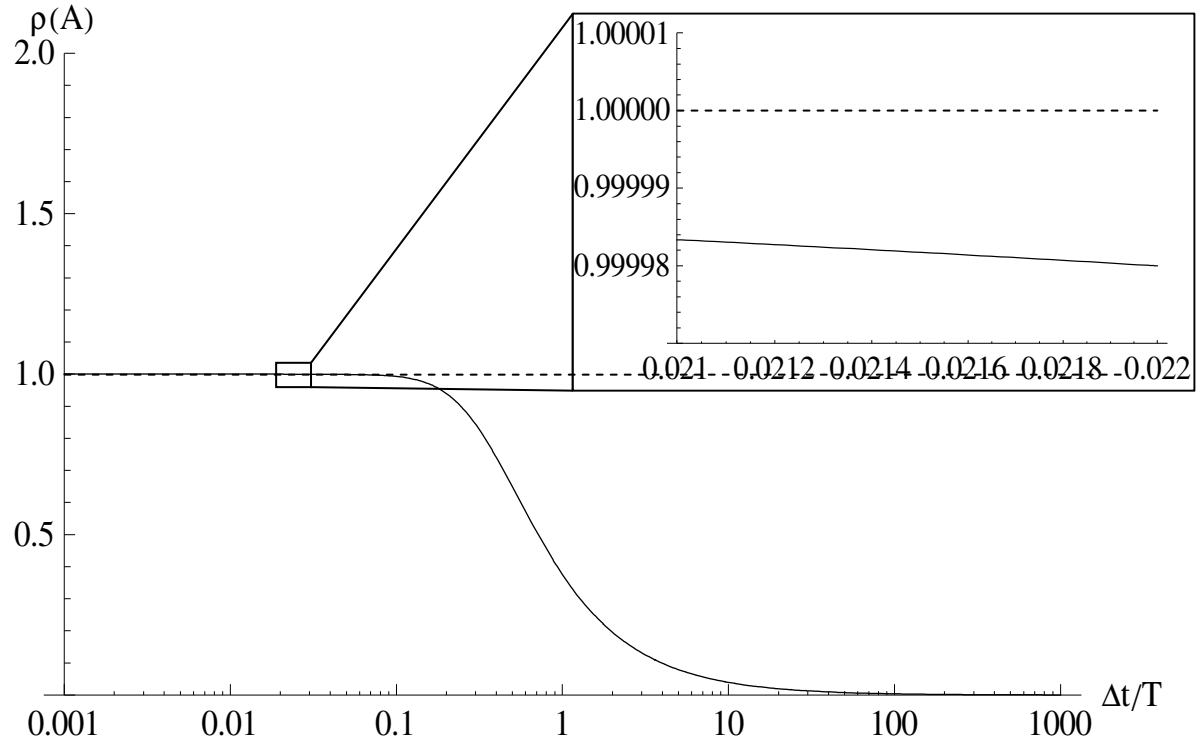


Fig. 7.20 Spectral radius $\rho(\boldsymbol{A})$ of Newmark-Euler time-integration scheme with standard parameters shows reason for numerical damping and velocity decrease of turbine model

One possibility of adjusting the parameters in the described manner is to use an optimization algorithm, because a manual parameter change satisfying all requirements is rather difficult and almost impossible as described in Section 6.1. The objective of this optimization is to maximize the integral of the spectral radius $\rho(\boldsymbol{A})$ for $0 < \Delta t/T \leq 0.1$, which reads for the Newmark-Euler scheme as

$$\max_{\alpha,\gamma,\beta} \int_0^{0.1} \rho(\boldsymbol{A}) \, \mathrm{d}\left(\frac{\Delta t}{T}\right). \tag{7.22}$$

As an additional constraint for guaranteeing stability we introduce the condition that the maximum value of the spectral radius may not exceed a value of 1.0 even for very big time-steps up to $\Delta t/T = 1000$, i.e.,

$$\max_{0 \leq \Delta t/T \leq 1000} \rho(\boldsymbol{A}) \leq 1.0. \tag{7.23}$$

The time-step sizes for the Newmark part and the Euler-Backward part are assumed to be equal. For the optimization the Downhill-Simplex method from Nelder and Mead [109] in

combination with a simple penalty strategy is used for making the algorithm applicable for constrained optimization problems. This leads to the optimized values for α, γ and β given in Table 7.2. In Fig. 7.21 the effect of this new parameter combination with respect to the spectral radius is shown. The spectral radius is always smaller or equal than one and the numerical damping in the interesting frequency spectrum is reduced.

Table 7.2 Optimized parameters of the generalized Newmark-Euler scheme

Variable	Optimized value
α	0.80767719
γ	0.73496847
β	0.33580917

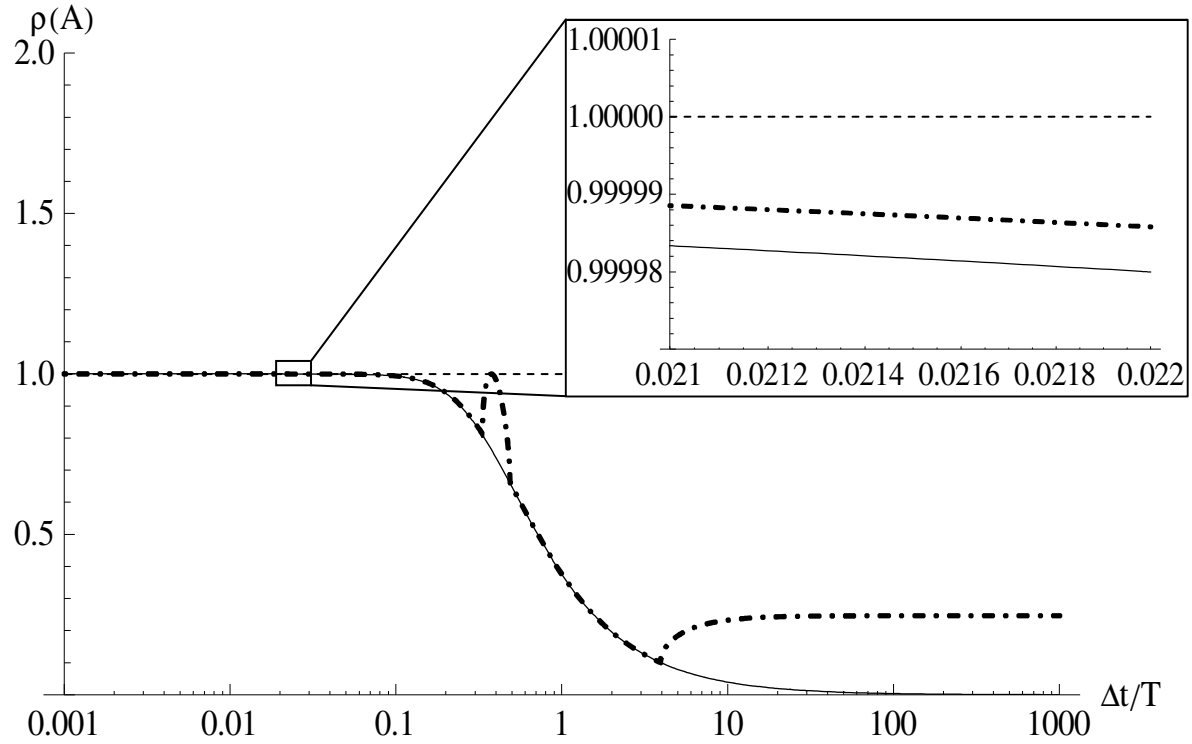


Fig. 7.21 Spectral radius $\rho(\boldsymbol{A})$ of Newmark-Euler time-integration scheme with standard (solid) and optimized parameters (dot-dashed)

As a next step, the influence of the optimized parameter combination to the turbine FEM example is investigated. For this purpose, the same computation as before is repeated, but the optimized parameter settings of Table 7.2 are used. As observable in the grey velocity curve of Fig. 7.22a, the decrease of the velocity after removing the pressure load is now much smaller and the behavior is almost ideal and no instabilities appear. Also the time-step size in Fig. 7.22b did not change dramatically in comparison to the computation with standard parameters which results in a similar computational time for both parameter settings. The presented optimized parameter values might not be the best for all possible problems and applications, but at least the used methodology shows a proper way to adjust a time-integration scheme to particular requirements.

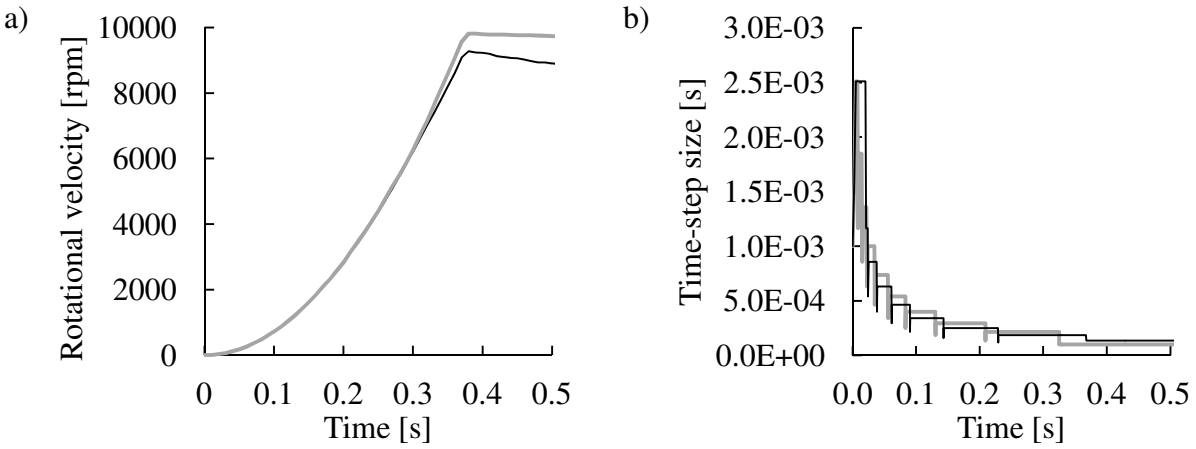


Fig. 7.22 Velocity curve of turbine model (a) and time-step size (b) for Newmark-Euler algorithm with standard (black) and optimized parameters (grey)

8 Model Reduction Strategy and Dummy Engine Model

The described high-fidelity models of aero-engines contain up to one hundred millions of degrees of freedom which results in high computational costs for the solution of the underlying boundary value problem. For this reason, model reduction strategies for reducing the number of degrees of freedom are very desirable while keeping the accuracy almost as high as in the non-reduced model. Especially for implicit computations, the benefit in terms of computational costs can be very high, since the number of degrees of freedom determines the overall computational time by a quadratic function. This relationship is only a linear one for explicit computations.

Another important aspect in developing high-fidelity models is the time needed for the meshing process. Since an automatic mesh generation for structured brick meshes is not possible yet, this part of the model generation is very time-consuming and induces a big portion of the overall costs from an economical point of view. In [75] a so-called contact meshing approach is described which simplifies the meshing process for complex structures dramatically while keeping the accuracy of the results high.

8.1 Model Reduction via Solid-by-Shell Substitution

In this section a method is presented where solid elements are substituted by shell elements in regions which are applicable for this strategy. Such regions are typically thin-walled regions like the casings of an aero engine, where no loads in thickness direction are applied and the curvature is moderate, which results in non-spatial stress states. If 3D solid elements are used, fully-integrated elements should be avoided because of the locking problems discussed in Chapter 3. Underintegrated elements do not suffer locking effects, but due to the constant strain distribution in the element at least three elements in thickness direction should be used to correctly map the linear strain distribution in thickness direction resulting from bending and keeping the integration error low (integration error is reduced if the interval becomes smaller, which means a finer mesh). Additionally, the dimensions of the element should be similar in all three directions to avoid a distortion of the element under large deformations of the whole structure. All these requirements lead to a fine mesh of 3D solid elements with many degrees of freedom.

Figure 8.1 shows such a part meshed with solid elements and the same part with the applied solid-by-shell substitution strategy. Obviously the number of degrees of freedom, and thus the computational time, can be reduced by using this strategy because shell elements are not subjected to the described restrictions for solid elements and much coarser meshes are possible without losing accuracy. For the coupling of solid and shell elements, there exist different algorithms which are implemented in modern FE tools and discussed for example in [126].

a) b)

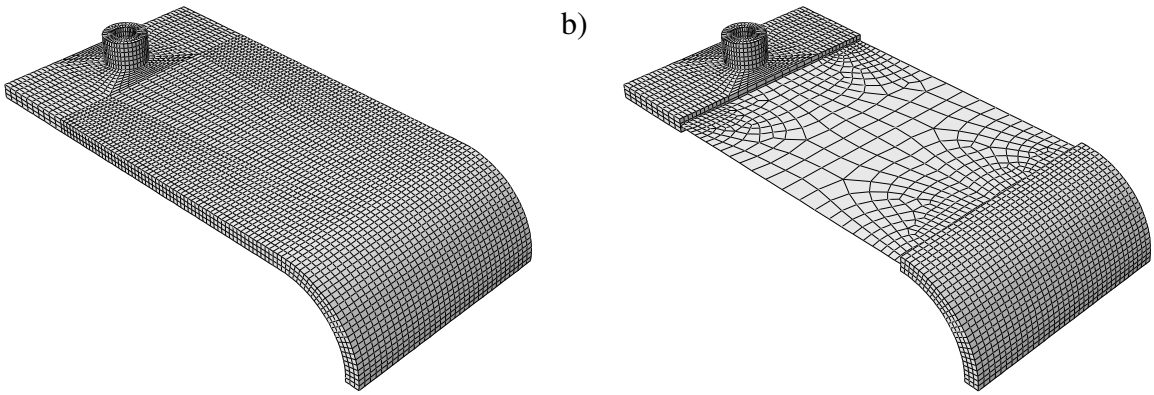


Fig. 8.1 Part meshed with solid elements (a) and applied solid-by-shell substitution strategy (b)

8.2 Considerations about Necessary Mesh Density

Before discussing the approach of model reduction of solid-by-shell substitution further, the question about the necessary mesh density of shell elements in regions with varying thickness should be addressed. Typically, the thickness within a shell element is constant. Even if the element thickness may be different at its nodal points, these values are usually averaged to a constant shell thickness. But is the arithmetic mean value a good choice for the element's thickness? And how many elements are necessary to approximate a structure with a varying thickness for keeping the displacement error for bending small? These questions will be answered for the example of a simple Euler-Bernoulli beam and should at least give an idea about the similar behavior of a plate.

8.2.1 Beam with Constant Bending Moment

According to Fig. 8.2a we consider a plane beam structure with a linearly varying thickness

$$t(x) = \frac{t_1 - t_0}{l} \cdot x + t_0. \tag{8.1}$$

The beam is subjected to a bending moment $M(x) = const.$ (Fig. 8.2b). By using the Euler-Bernoulli beam theory, the displacement $w(x)$ in z-direction can be calculated from $EI_{yy}(x)w''(x) = -M(x)$ by

$$w(x) = \frac{1}{E} \int_0^x \left[\int_0^x \frac{-M(x)}{I(x)} \mathrm{d}x \right] \mathrm{d}x + C_1 x + C_2 \tag{8.2}$$

with integration constants C_1 and C_2. For a rectangular cross section $b \times t$ with the area moment of inertia $I(x) = b \cdot t(x)^3/12$ and beam width b this leads to

$$w(x) = \frac{12M_0}{Eb} \int_0^x \int_0^x \frac{1}{\left(\frac{t_1 - t_0}{l} \cdot x + t_0\right)^3} \mathrm{d}x \, \mathrm{d}x + C_1 x + C_2. \tag{8.3}$$

Performing the integrations and determining the integration constants from the boundary conditions $w(0) = w(l) = 0$ yields

$$w(x) = \frac{6M_0 lx}{Ebt_0^2}\left(\frac{x}{lt_0 + x(t_1 - t_0)} - \frac{1}{t_1}\right) \tag{8.4}$$

with the maximum deflection

$$w_{max} = w\left(x = l \cdot \frac{t_0 - \sqrt{t_0 t_1}}{t_0 - t_1}\right) = -\frac{6M_0 l^2}{Ebt_0\left(t_1 + \sqrt{t_0 t_1}\right)^2} \tag{8.5}$$

as shown in Appendix F.

a) b)

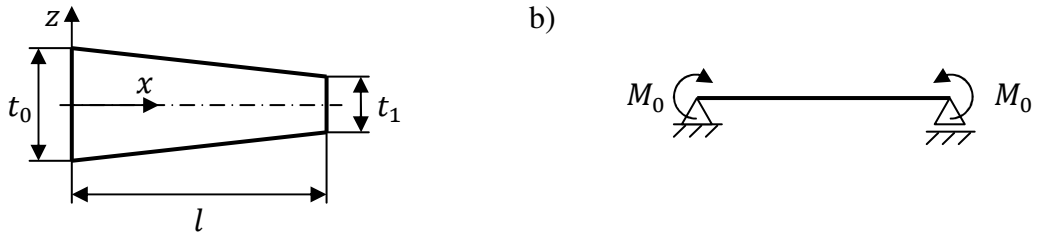


Fig. 8.2 Plane beam with linearly varying thickness (a) and bending load case (b)

For the determination of a constant beam thickness t_c, which is necessary for a beam with constant cross section of equal stiffness, different possibilities for the stiffness definition are conceivable. The first possibility is to demand equal maximum deflections $w_{max}(t_0 = t_1 = t_c) = 3M_0 l^2/(2Ebt_c^3) = w_{max}(t_0 \neq t_1)$, i.e.,

$$\frac{3M_0 l^2}{2Ebt_c^3} = \frac{6M_0 l^2}{Ebt_1\left(t_0 + \sqrt{t_0 t_1}\right)^2}, \tag{8.6}$$

which gives

$$t_c = \sqrt[3]{\frac{t_0\left(t_1 + \sqrt{t_0 t_1}\right)^2}{4}}. \tag{8.7}$$

Choosing e.g. $t_0 = 10mm$ and $t_1 = 5mm$ would result in $t_c \approx 7.142mm$. This value for t_c obviously differs from the arithmetic mean value $7.5mm$ of t_0 and t_1.

But it would also be possible to define the stiffness of the beam as the sum of all its displacements which leads to the equivalence condition

$$\int_0^l w(x, t_0 = t_1 = t_c)\mathrm{d}x = \int_0^l w(x, t_0 \neq t_1)\mathrm{d}x$$

or

$$\int_0^l \frac{6M}{Ebt_c^3}(x^2 - lx)\mathrm{d}x = \int_0^l \frac{6Mlx}{Ebt_0^2}\left(\frac{x}{lt_0 + x(t_1 - t_0)} - \frac{1}{t_1}\right)\mathrm{d}x \tag{8.8}$$

or

$$-\frac{l^3}{6t_c^3} = \frac{l^3\left[t_1^2 - t_0^2 + 2t_0t_1\ln\left(\frac{t_0}{t_1}\right)\right]}{2t_0t_1(t_0-t_1)^3}$$

and finally to

$$t_c = \sqrt[3]{\frac{t_0t_1(t_0-t_1)^3}{-3\left[t_1^2 - t_0^2 + 2t_0t_1\ln\left(\frac{t_0}{t_1}\right)\right]}}. \tag{8.9}$$

By choosing again $t_0 = 10mm$ and $t_1 = 5mm$, we obtain $t_c \approx 7.156mm$.

A third possibility for defining the beam stiffness could be the total strain energy stored in the structure as the work of stresses, which can be derived from

$$W^{int} = \frac{1}{2}\int_V \sigma_{xx}\varepsilon_{xx}\mathrm{d}V = \frac{1}{2E}\int_V^0 \sigma_{xx}^2\,\mathrm{d}V. \tag{8.10}$$

With $\sigma_{xx}(x) = M(x)\cdot z/I_{yy}(x)$ we get

$$W^{int} = \frac{1}{2E}\int_V \frac{M(x)^2}{I_{yy}(x)^2}z^2\,\mathrm{d}V \tag{8.11}$$

or with $\mathrm{d}V = \mathrm{d}A\mathrm{d}x$ and $I_{yy}(x) = \int_A z^2\mathrm{d}A$

$$W^{int} = \frac{1}{2E}\int_0^l \left[\frac{M(x)^2}{I_{yy}(x)^2}\int_A z^2\mathrm{d}A\right]\mathrm{d}x = \frac{1}{2E}\int_0^l \frac{M(x)^2}{I_{yy}(x)}\mathrm{d}x. \tag{8.12}$$

This term can also be transformed with the relation $EI_{yy}(x)w''(x) = -M(x) = M_0$ to

$$W^{int} = \frac{E}{2}\int_0^l I_{yy}(x)w''(x)^2\mathrm{d}x. \tag{8.13}$$

Evaluating (8.12) for $t_0 = t_1 = t_c$ gives

$$W^{int} = \frac{6M_0^2l}{Ebt_c^3} \tag{8.14}$$

and for $t_0 \neq t_1$

$$W^{int} = \frac{1}{2E}\int_0^l \frac{M_0^2}{\frac{b}{12}\left(\frac{t_1-t_0}{l}\cdot x + t_0\right)^3}\mathrm{d}x = \frac{3M_0^2l}{Eb}\left(\frac{t_0+t_1}{t_0^2t_1^2}\right). \tag{8.15}$$

Demanding the equality of (8.14) and (8.15) yields

$$t_c = \sqrt[3]{\frac{2t_0^2t_1^2}{t_0+t_1}}. \tag{8.16}$$

For the example $t_0 = 10mm$, $t_1 = 5mm$ a value of $t_c \approx 6.934mm$ is obtained.

Obviously, the different definitions of the equivalent beam stiffness lead to similar but not identical constant thickness values if we are looking for a constant thickness beam of the same stiffness. For our further considerations, we will choose the strain energy stored in the structure as measure for stiffness comparison.

8.2.2 Beam with Variable Bending Moment

Now we assume not only a varying thickness of the beam as given in Equation (8.1), but also a linearly varying bending moment according to

$$M(x) = \frac{M_1 - M_0}{l} \cdot x + M_0 \tag{8.17}$$

with M_0 being the bending moment at $x = 0$ and M_1 the bending moment at $x = l$ to take into account the most general case with regard to a finite beam element. For this case, the displacements are computed according to (8.2) as

$$w(x) = \frac{12}{Eb} \int_0^x \int_0^x \frac{\frac{M_1 - M_0}{l} \cdot x + M_0}{\left(\frac{t_1 - t_0}{l} \cdot x + t_0\right)^3} \mathrm{d}x\, \mathrm{d}x + C_1 x + C_2 \tag{8.18}$$

which yields after a laborious calculation to

$$w(x) = \frac{6l}{Ebt_0} \left[\frac{-\frac{(t_0 - t_1)^2 (M_1 t_0 - M_0 t_1)(l - x)x}{t_1\left(lt_0 + x(t_1 - t_0)\right)}}{(t_0 - t_1)^3} + \frac{2t_0(M_0 - M_1)[(l - x)\ln(lt_0) + x\ln(lt_1) - l\ln(lt_0 - t_0 x + t_1 x)]}{(t_0 - t_1)^3} \right]. \tag{8.19}$$

Using (8.12) with $t_0 = t_1 = t_c$ results in the strain energy

$$W^{int} = \frac{2l}{Ebt_c^3} (M_0^2 + M_0 M_1 + M_1^2) \tag{8.20}$$

and for $t_0 \neq t_1$ in

$$W^{int} = \frac{3l}{Ebt_0^2 t_1^2 (t_0 - t_1)^3} \Big[(t_0 - t_1)(M_1 t_0 - M_0 t_1) \cdot \left(M_1 t_0 (t_0 - 3t_1) + M_0 t_1 (3t_0 - t_1)\right) + 2t_0^2 t_1^2 (M_0 - M_1)^2 \ln\left(\frac{t_0}{t_1}\right) \Big]. \tag{8.21}$$

The constant beam thickness t_c we are searching for can be determined by setting (8.20) equal to (8.21) and solving for t_c which finally yields

$$t_c = \sqrt[3]{\frac{2(M_0^2 + M_0 M_1 + M_1^2) t_0^2 t_1^2 (t_0 - t_1)^3}{3(t_0 - t_1)(M_1 t_0 - M_0 t_1)\left(M_1 t_0 (t_0 - 3t_1) + M_0 t_1 (3t_0 - t_1)\right) + 2t_0^2 t_1^2 (M_0 - M_1)^2 \ln\left(\frac{t_0}{t_1}\right)}}. \tag{8.22}$$

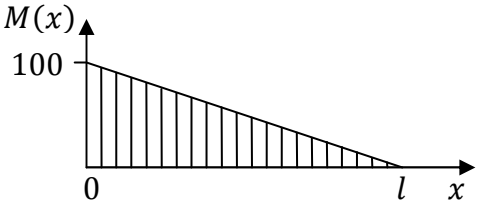
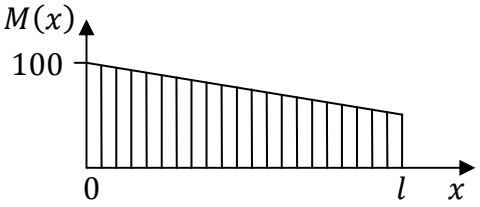
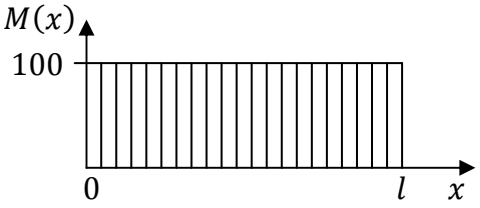
As shown in Table 8.1 for a few examples, the equivalent constant beam thickness t_c computed with Equation (8.22) does not only depend on the beam's thickness distribution, but also on the loading.

For a better visualization of the final Equation (8.22) the substitutions $a = t_0/t_1$ and $b = M_0/M_1$ are introduced. By this, (8.22) can be transformed to

$$\frac{t_c}{t_1} = \sqrt[3]{\frac{2(a-1)^3a^2(1+b+b^2)}{3(a-1)(a-b)\big(-b+a(a-3+3b)\big)+6a^2(b-1)^2\ln(a)}}. \tag{8.23}$$

The illustration of this result in Fig. 8.3 gives an impression about the influence of the varying load and thickness ratios on the constant beam thickness t_c in relation to the thickness t_1.

Table 8.1 Equivalent constant beam thickness for $t_0 = 10mm$, $t_1 = 5mm$ and different loading conditions

a) 	$M_0 = 100Nmm, M_1 = 0Nmm$ ⇩ $t_c = 8.488mm$
b) 	$M_0 = 100Nmm, M_1 = 50Nmm$ ⇩ $t_c = 7.494mm$
c) 	$M_0 = 100Nmm, M_1 = 100Nmm$ ⇩ $t_c = 6.934mm$

Since the section loads for a finite element are not known before the computation, an iterative process would be required to determine the correct thickness values. However, is such an effort really necessary? To answer this question, load case a) of Table 8.1, where the equivalent beam thickness differs most from the average thickness (see also Fig. 8.4b), is modeled with different mesh densities. Each of the used beam elements is modeled with the average thickness $t_{c,i} = \big(t_{0,i} + t_{1,i}\big)/2$ according to its position in the mesh. For the test problem the parameters $E = 210000\,N/mm^2$, $l = 200mm$, $b = 1mm$, $t(x = 0) = 10mm$, $t(x = l) = 5mm$ and the loads $M_0 = 100Nmm$ and $M_1 = 0Nmm$ (corresponds to $F = 0.5N$) are chosen. In Fig. 8.4a the computed tip deflections of the beam are plotted for different mesh densities with 1, 2, 3 or 4 Euler-Bernoulli beam elements in longitudinal direction. Additionally, the relative error of the tip deflection in comparison to the analytical solution computed with Equation (8.19) is shown. As it can be seen, the FEM solution

converges quite fast to the analytical solution with increasing mesh density, where for four elements the error is only about 2.7%

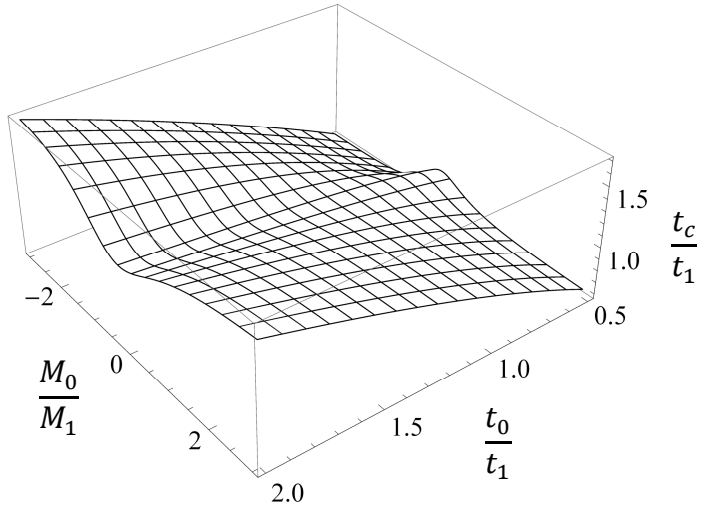


Fig. 8.3 Influence of load and thickness distribution on constant beam thickness of equal stiffness

From these considerations follows that especially in regions with varying thickness a shell mesh should not be too coarse. The necessary mesh density does not only depend on the thickness variation, but also on the applied load. Typically a refinement of the mesh leads to fast convergence of the displacement results.

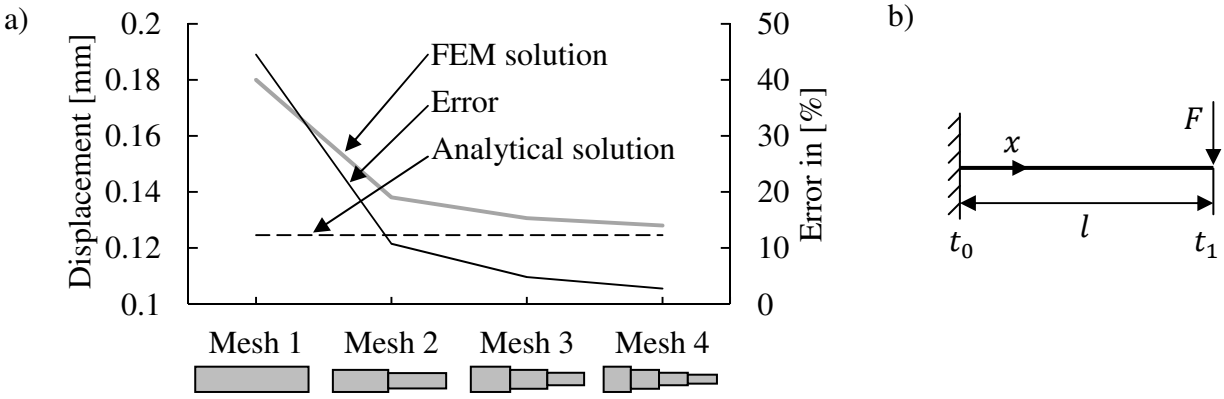


Fig. 8.4 Displacements and errors for different mesh densities of beam (a) and applied load case (b)

8.3 Tests and Examples of Solid-by-Shell Substitution

To demonstrate the capabilities of the described solid-by-shell substitution strategy, a casing part of the Dummy Engine model in Section 8.4 is chosen. This part is subjected to three different load cases as presented in Fig. 8.5. In the first load case the part is mainly subjected to tension, the second load case is a mixture of bending and shear, and the third load case consists of a thermal load.

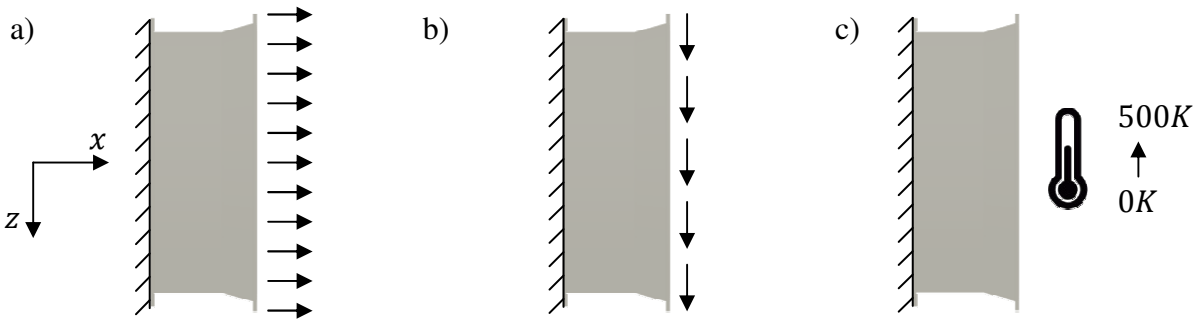


Fig. 8.5 Casing part subjected to load cases tension (a), shear (b) and temperature (c)

The described loads are applied to four different model types. In the first model, the whole structure is meshed by underintegrated solid elements (Fig. 8.6a), where three elements in thickness direction are used in all regions resulting in a very fine mesh with approximately 70000 nodes. The second model is a so-called hybrid model where the solid-by-shell substitution strategy is applied. Regions, where no spatial stress states are expected, are meshed with shell elements (Fig. 8.6b), other regions like the flange regions are meshed with solid elements. Between both regions a tie coupling is applied. The third model is also a hybrid model (Fig. 8.6c), but instead of classical shell elements so-called T-shell elements sometimes called solid shell, continuum shell or thick shell (but should not be mixed up with the classical Reissner-Mindlin shell sometimes also called thick shell) are used. This element type appears to the user as a solid element, but uses the kinematics of shell elements [69,97] and is also coupled via a tie constraint to the solid part. The fourth model is also a hybrid one with T-shell elements, but in contrast to the third model with a structured mesh in the shell region (Fig. 8.6d). The used commercial finite element code requires for the coupling of solid and classical shell elements a certain match of the meshes (Fig. 8.6b), which is not necessarily demanded for T-shells.

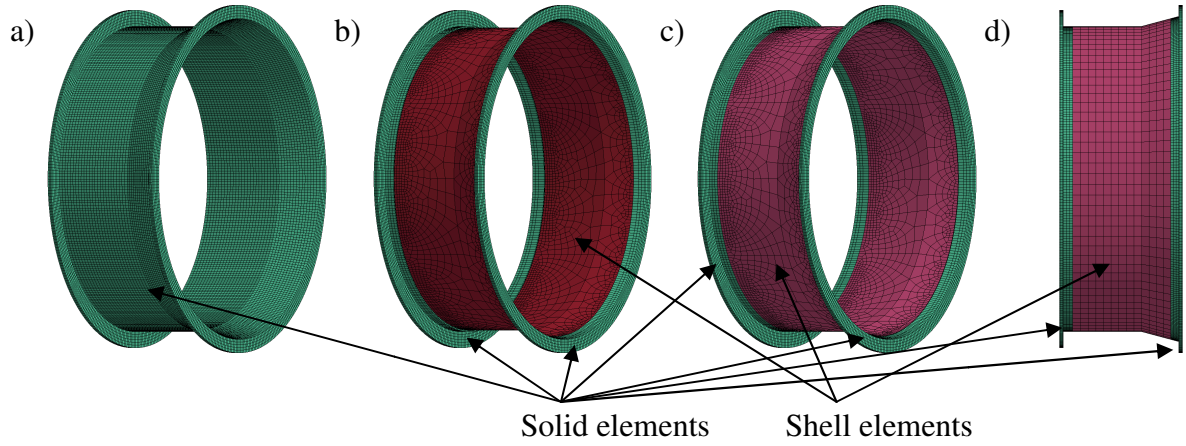


Fig. 8.6 Casing part meshed with solid elements (a), hybrid mesh of classical shells and solids (b), hybrid mesh of T-shell elements and solids (c) and hybrid mesh with structured T-shell mesh (d)

The results of these static computations are summarized in Fig. 8.7, where they are normalized with respect to the results of the solid model. The consistency in terms of displacements between the different models is very good. Only for the tension load case the

hybrid model with classical shell elements reacts too stiff. This behavior is related to the coupling algorithm between classical shells and solid elements. The algorithm impairs the Poisson effect in the coupling area which introduces an artificial stiffness into the model and results in smaller displacements. The biggest advantage of the hybrid model technology is the lower number of degrees of freedom which yields to a remarkable speed-up of the computation. The computational time of the hybrid model with a structured T-shell mesh is just about 15% of that of the solid model which corresponds well to the expected quadratic influence of the model size on the computational time.

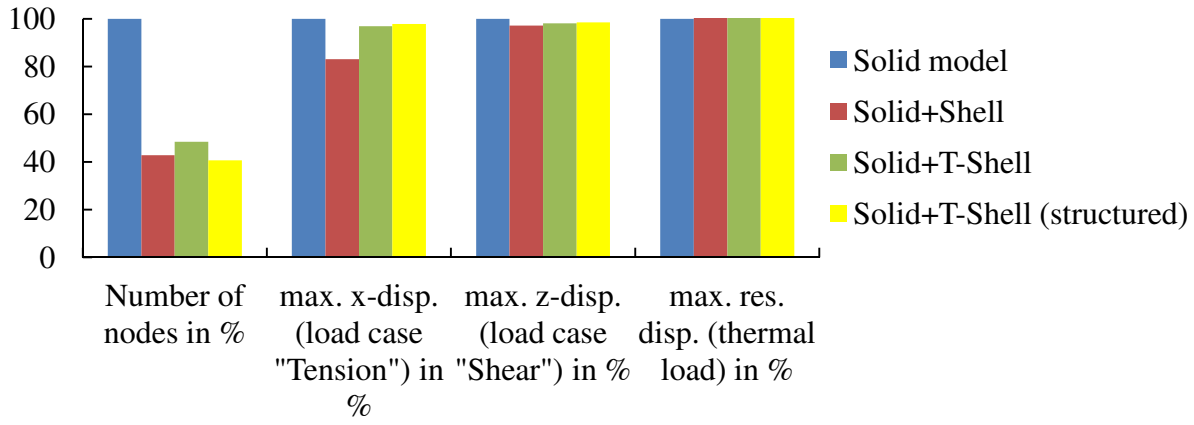


Fig. 8.7 Computational results for different casing models

8.4 Dummy Engine Model

In this section, strategies discussed and presented in the previous chapters and sections are applied to a so-called Dummy Engine model. This finite element model of an aero-engine is not related to any existing aero-engine, since it contains many simplifications, but it includes all important parts. Figure 8.8 gives an overview of the model and its most important components. Some components like different kinds of struts, guide vanes, seal fins, abradable liner and special details of the mounting system are also part of the model but not referenced in Fig. 8.8. The model contains all together 56 contact definitions in the bearing region, between the different parts of the mount system, between fan blades and fan disc, abradable liners and turbine drum, casings and blades, etc. For all rotating contact surfaces Mortar contacts are defined, all other contact definitions use the surface-to-surface approach in combination with a penalty formulation.

Figure 8.9 shows some details of the rear mount system and the region of the turbine blades, where also the used mesh is visible (Fig. 8.9b). For all those details and small components, contact is defined between the moveable parts. An elastic viscoplastic material model, which also takes into account thermal affects, is used for all parts of the engine. Since the whole model is meshed by underintegrated solid elements, an anti-hourglassing formulation according to [19] is chosen. In total, the whole structure contains more than 1100 parts and about 8.7 million finite elements with 11.3 million nodes.

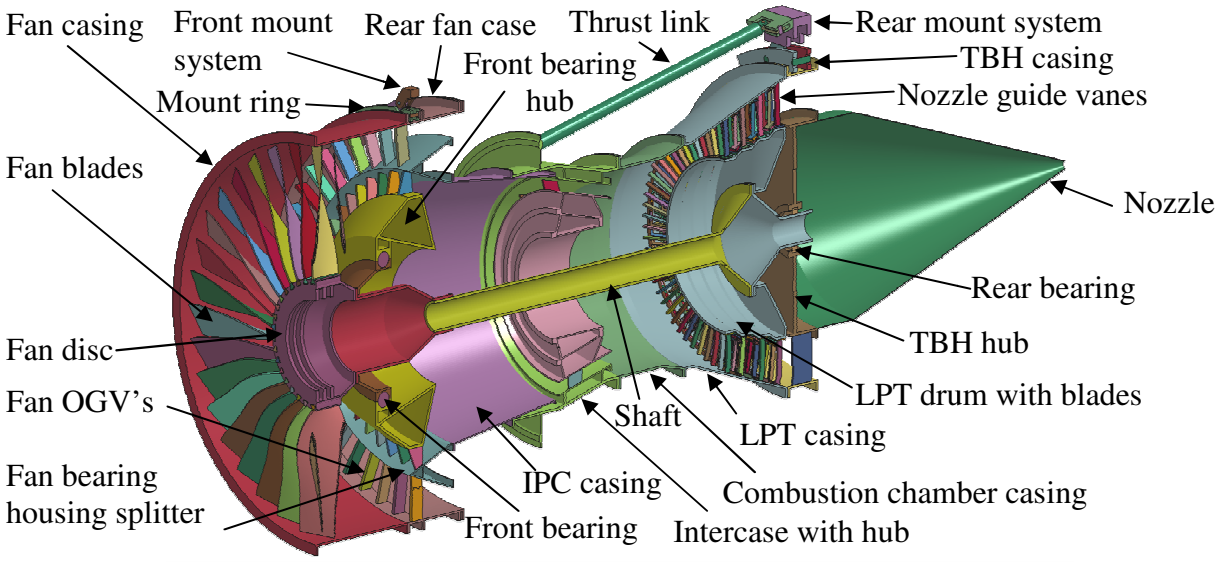


Fig. 8.8 Components of Dummy Engine model

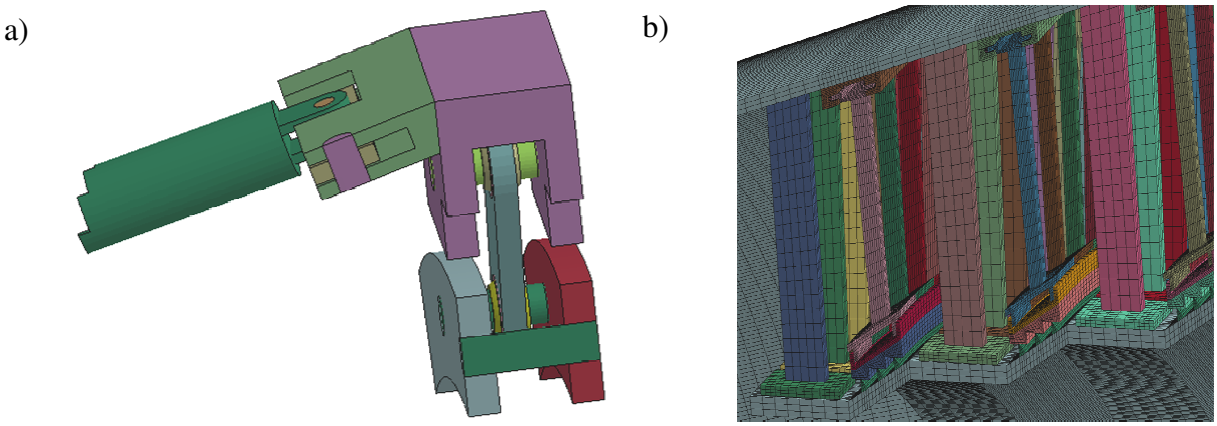


Fig. 8.9 Rear mount system (a) and mesh of turbine blades with guide vanes, liners and seal fins (b) of Dummy Engine model

8.4.1 Applied Test Load Case and Results

In the implicit solution process of a boundary value problem, as described by the FEM, which typically consists of the steps

- matrix assembly,
- symbolic factorization (ordering),
- numeric factorization and
- numeric solution,

an important aspect is the symbolic factorization or ordering. This step is responsible for conditioning the matrix such that the following steps can be performed more efficiently. In commercial FE codes, different ordering schemes are implemented. Before starting a transient computation with thousands of time-steps, it is recommendable to choose the ordering scheme carefully, because the efficiency of the ordering scheme depends on the particular model

properties and primarily on the problem size. For the Dummy Engine model three different ordering schemes are tested:

- MMD (Multiple Minimum Degree, see e.g. [139,96]),
- Metis [74] and
- LSGPart [3].

The test computations, consisting of two iterations of a single time-step, are performed on a cluster with 40 CPU cores per compute node. Figure 8.10 shows the computational time for this example problem computed with Newmark time-integration and a Full-Newton algorithm for a varied number of compute nodes together with different ordering schemes. Two aspects are important here. First, LSGPart achieves the lowest computational times for all tested numbers of compute nodes, and second, LSGPart shows the best scalability, which means that the computational time still decreases for a higher number of CPUs. For these reasons, the following implicit computation for the Dummy Engine model is performed with the LSGPart ordering scheme.

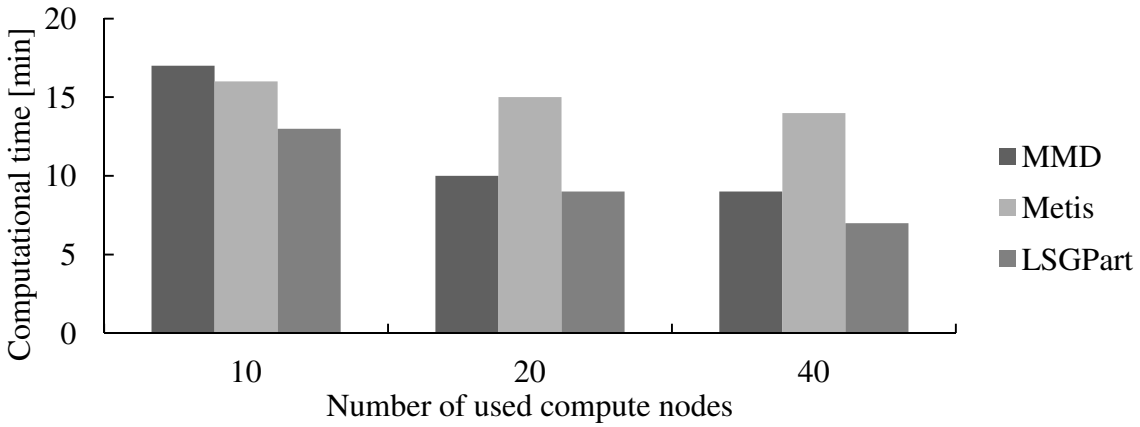


Fig. 8.10 Comparison of different ordering schemes for Dummy Engine model

The load case is simple but activates almost all contact definitions in the model: A linearly increasing pressure load is applied to the turbine blades and a linearly increasing gravity load is applied to the whole structure while the upper surfaces of the front and rear mount system are fixed. Figure 8.11 shows the computed rotational velocity of the engine's shaft for an explicit and implicit simulation of the described problem. The explicit simulation uses a slight modification of the central difference method and the implicit one generalized Newmark-Euler (Composite) time-integration in combination with a Full-Newton algorithm. As one can see, the consistency of both time-integration schemes is very good. The explicit computation is terminated after approximately 0.1s for this reason. The waviness of the velocity time curve results from a folding back effect of the fan blades, which are loosely mounted on the fan disc and reach their final position at a certain angular velocity due to the centrifugal load. In addition, the implicit computation is about 10 times faster than the explicit one for the first 0.1s of simulation time, which was another reason for the premature termination of the explicit run.

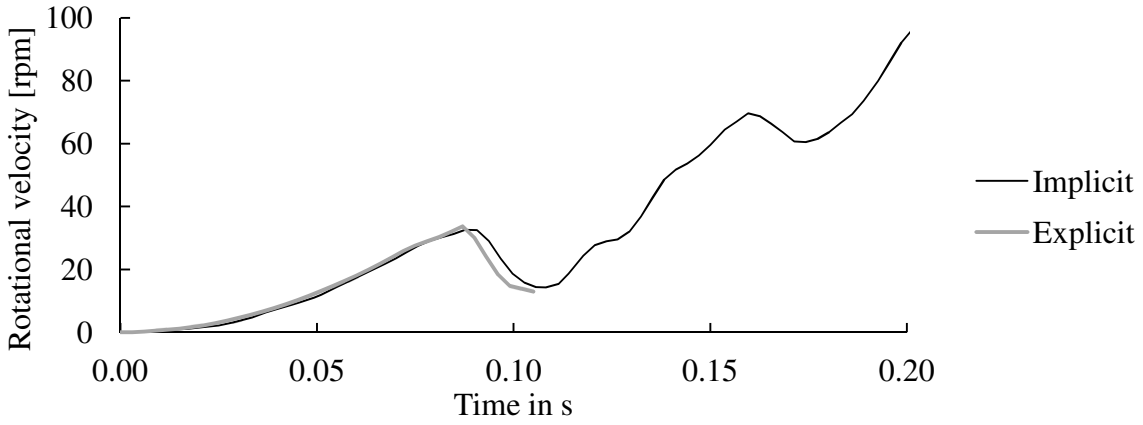


Fig. 8.11 Comparison of different time-integration schemes for Dummy Engine model

8.4.2 Test of Model Reduction Strategy

For testing the solid-by-shell reduction strategy, part regions of the Dummy Engine model, where the substitution strategy is applicable, are meshed by classical shell elements and tied to the solid area at their boundaries. Figure 8.12a shows the solid element model and Fig. 8.12b the same model with applied solid-by-shell substitutions. It should be mentioned that also other parts of the model, e.g. the blades, vanes and struts, could be meshed by shell elements but in the sense of a conservative approach we restrict ourselves to the regions shown in Fig. 8.12b.

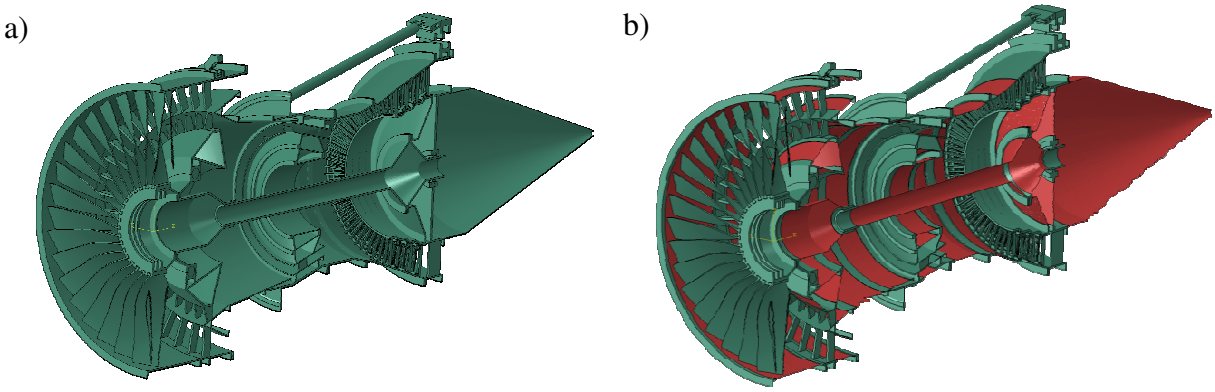


Fig. 8.12 Dummy Engine model with a) solid regions and b) with shell (red) and solid (green) regions

Since most of the shell regions belong to the static, non-rotating part of the engine, we just use the casing structure for some test computations. First, three different linear static load cases are applied. Figure 8.13 illustrates these load cases, which are tension, bending and torsion. For the torsion load case, stiff beams are used to translate the moment to the fan casing. In all three load cases the structure is fixed at the whole circumference of the rear flange of the thrust bearing housing (TBH) casing and loaded at the front flange of the fan casing. By this, most of the substitution regions are loaded. Table 8.2 summarizes the displacements resulting from the three load-cases. All results are normalized with respect to

the displacements of the solid model. For the torsion load case, the rotation of the load introduction point is evaluated. All displacements are within an aspired maximum difference of 10%.

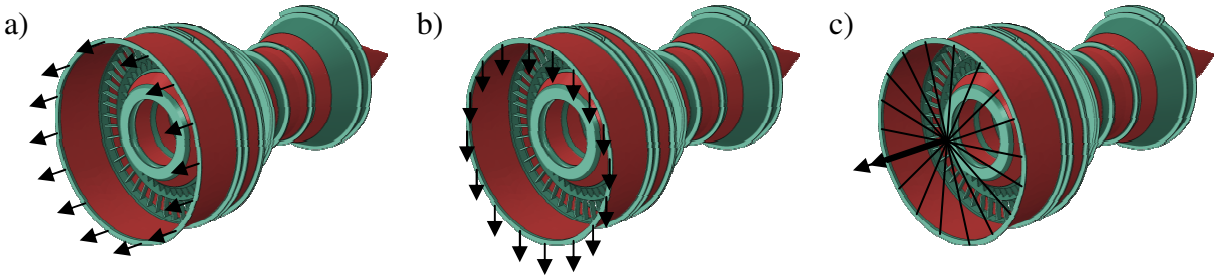


Fig. 8.13 Static structure of engine subjected to tension (a), bending (b) and torsion (c)

Table 8.2 Displacements of static structure of hybrid Dummy Engine model for different load cases

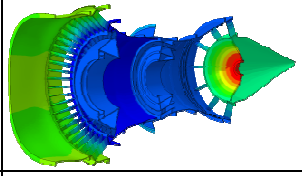
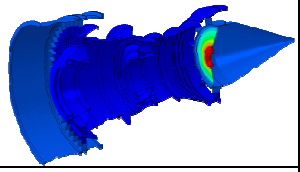
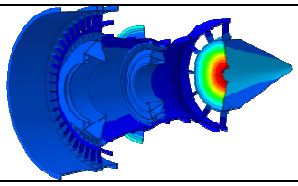
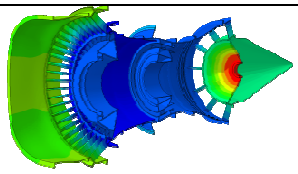
Load case	tension	bending	torsion
Max. absolute displacement	102.8%	103.9%	106%

Secondly, also the dynamic behavior should be investigated. For this purpose an eigenvalue analysis with free-free boundary conditions is performed for the static structure of Fig. 8.13. In Table 8.3 the first 20 eigenfrequencies of the solid and the hybrid structure are listed, where modes 1-6 are rigid body modes and skipped for that reason. For modes 7-14, a very good agreement is reached with a maximum frequency deviation of 3.06%. Also the mode shapes are very similar as observed by visual inspection. The 15th and 18th eigenfrequency differ quite a lot. The reason can be found in the special mode shapes shown in Table 8.3. Obviously mode shapes 15 and 18 are interchanged for both models due to a differing stiffness of the TBH hub. In the hybrid model, the shell-to-solid substitution strategy is used for this component, which is subjected to a lot of bending especially in the coupling regions between solid and shell structure. In these regions the bending stiffness differs substantially between both models due to the coupling algorithm, which explains the difference in the eigenfrequencies. For modes 16 and 17 as well as for modes 19 and 20 the agreement in terms of eigenfrequencies and mode shapes is quite good again.

The static structure of the solid model has approx. 8.9 millions of degrees of freedom whereas the hybrid model has only a size of 6.9 millions of degrees of freedom. The difference might seem smaller than expected, but especially in the turbine region of the model (Fig. 8.9b), where no substitution took place, an extremely fine mesh is used for many parts which leads to the high number of degrees of freedom in the hybrid model. Nevertheless, there is a big improvement in terms of computational time, since the hybrid model needs only 60% of the time of the solid model. Summarizing all results, we can conclude that the substitution strategy is a proper way to reduce the computational effort while keeping the accuracy of the results high. Additional savings of computational time could be gained by applying the

substitution strategy also to the rotor components of the model where parts like blades (except the blade roots) or the shaft are also well suited for a modeling by shell elements.

Table 8.3 Eigenfrequencies of static structure in Hz

Mode	Solid model		Hybrid model		Difference
	Eigen-frequency	Selected mode shapes	Eigen-frequency	Selected mode shapes	
7	59.746		58.935		1.36%
8	60.033		59.221		1.35%
9	96.721		94.32		2.48%
10	96.785		94.365		2.50%
11	101.64		99.225		2.38%
12	113.38		111.86		1.34%
13	114.1		112.58		1.33%
14	143.37		138.99		3.06%
15	155.63		139.3		25.96%
16	162.16		156.43		3.53%
17	162.22		156.45		3.56%
18	188.13		165.37		6.26%
19	201.46		194.42		3.49%
20	204.09		196.85		3.55%

9 Conclusions and Outlook

The field of finite element computations is an area which is almost impossible to cover with all its topics in a single textbook. The intention of this publication, therefore, is to draw the reader's attention to some special issues that may appear in extremely detailed high-fidelity engine models (but also in other types of models). The more components are included in a model, the more contact conditions have to be defined, monitored and managed during a nonlinear computation. The ideas behind algorithms for this purpose are described and explained for simple examples. The knowledge about these procedures is important to discover and solve problems like slow or even unstable convergence or unwanted penetrations. Both kinds of problems may also arise from the unintended stiffening or softening effects, locking and hourglassing. Choosing an appropriate finite element formulation or anti-hourglassing strategy is important for obtaining accurate and reliable results.

In this work, attention is especially drawn to time-integration procedures, which are essential in the solution process of transient, nonlinear boundary value problems. Explicit time-integration schemes are very robust and well-established for the simulation of highly dynamic processes but suffer the problem of enormous computational times when it comes to the simulation of events over a time-span of several seconds due to their extremely small time-steps. In the field of aero-engines, it is highly appreciated by many of the involved disciplines to have data of the thermo-mechanical behavior of a running engine over a bigger time-span. Typically, experiments can deliver these data only in a limited extent. One reason for these limitations are the high costs of such experiments. Furthermore, it is difficult or sometimes impossible to monitor all desired data in all regions of an engine because of the restricted accessibility to many components, extremely high temperatures and further obstacles. A solution for the described difficulties of the explicit simulation and limitations for the experiments could be the implicit simulation of aero-engines as demonstrated in this work. Although implicit time-integration requires more computational effort per time-step due to the necessary solution of in general nonlinear systems of equations in each time-step, the stable time-step sizes may be a few orders bigger than that of an explicit integration scheme. Instability problems of implicit time-integration schemes for rotating elastic structures, like the rotor of an aero-engine, are investigated and the strategies presented that allow a stable and efficient simulation.

Of course, for every simulation process it is desirable to reduce the model size while keeping the accuracy of the results high. Substituting solid by shell elements in certain model regions can be a step towards this requirement. In this context, considerations about the mesh density of shell-meshed regions are carried out and the substitution procedure is successfully implemented and applied to a fictive aero-engine model.

The mesh density is also a very important aspect in the detailed modeling of bearings. The mesh should be fine enough to allow for a kinematically correct rotation of the bearing but not

too fine, especially in explicit computations, since the stable time-step size is influenced by the finite element size. As it could be shown, also the contact description in a bearing determines the required mesh density, which can be calculated by derived equations.

As already mentioned, not all topics could be covered here and there a many further points which can improve the results and reduce the computational time in particular. The decomposition of huge models has a remarkable influence on the overall computational time on distributed systems like clusters with thousands of CPUs. Related to this, an improvement of the scalability, especially of implicit simulations, on the mentioned systems would lead to a significant speed-up of computations. For such improvements, experience has shown there needs to be close collaboration between programmers and cluster developers.

An important development that might in the future solve many problems associated with the time-consuming meshing process is isogeometric analysis. Such a direct NURBS-based computation of displacements and stresses would bypass many of the known problems during the discretization procedure and leads to a very exact representation of the geometry also in the computational domain.

Appendix

A Computation of Eigenvalues for Stability Analysis of Newmark Algorithm

Computing $\det(\lambda \boldsymbol{E} - \boldsymbol{A}_{NM})$ and setting it equal to zero yields with the substitution (6.16) the characteristic polynomial

$$\lambda^3 + \lambda^2\left(-1 + \frac{\gamma}{\beta} - \frac{1}{z_1} + \frac{1}{2\beta z_1} - \frac{1}{\beta\Delta t^2\omega^2 z_1} - \frac{\gamma}{\beta^2\Delta t^2\omega^2 z_1}\right) + \lambda\left(\frac{1}{z_1} + \frac{1}{2\beta z_1} - \frac{\gamma}{\beta z_1} + \frac{1}{\beta\Delta t^2\omega^2 z_1}\right) = 0. \tag{A.1}$$

From this, $\lambda_1 = 0$ follows directly and the quadratic equation

$$\begin{aligned}\lambda^2 + \lambda\left(-1 + \frac{\gamma}{\beta} - \frac{1}{z_1} + \frac{1}{2\beta z_1} - \frac{1}{\beta\Delta t^2\omega^2 z_1} - \frac{\gamma}{\beta^2\Delta t^2\omega^2 z_1}\right) \\ + \left(\frac{1}{z_1} + \frac{1}{2\beta z_1} - \frac{\gamma}{\beta z_1} + \frac{1}{\beta\Delta t^2\omega^2 z_1}\right) \\ = \lambda^2 + \lambda\left(\frac{\Delta t^2\omega^2\left(-\beta^2 z_1 + \gamma\beta z_1 - \beta^2 + \frac{1}{2}\beta\right) - \beta - \gamma}{\beta^2\Delta t^2\omega^2 z_1}\right) \\ + \left(\frac{\beta^2\Delta t^2\omega^2 + \frac{1}{2}\beta\Delta t^2\omega^2 - \gamma\beta\Delta t^2\omega^2 + \beta}{\beta^2\Delta t^2\omega^2 z_1}\right) = 0\end{aligned} \tag{A.2}$$

remains, which has the solutions

$$\begin{aligned}\lambda_{2,3} = \frac{\Delta t^2\omega^2\left(\beta^2 z_1 - \gamma\beta z_1 + \beta^2 - \frac{1}{2}\beta\right) + \gamma + \beta}{2\beta^2\Delta t^2\omega^2 z_1} \\ \pm\sqrt{\left[\frac{\Delta t^2\omega^2\beta\left(-\beta z_1 + \gamma z_1 - \beta + \frac{1}{2}\right) - \gamma - \beta}{2\beta^2\Delta t^2\omega^2 z_1}\right]^2 - \frac{\Delta t^2\omega^2\left[\beta + \frac{1}{2} - \gamma\right] + 1}{\beta\Delta t^2\omega^2 z_1}}.\end{aligned} \tag{A.3}$$

Reversing the substitution (6.16) results in

$$\begin{aligned}\lambda_{2,3} = \frac{4 + \Delta t^2\omega^2(4\beta - 2\gamma - 1)}{4 + 4\beta\Delta t^2\omega^2} \\ \pm\sqrt{\left[\frac{-4 - \Delta t^2\omega^2(4\beta - 2\gamma - 1)}{4 + 4\beta\Delta t^2\omega^2}\right]^2 - \frac{2\Delta t^2\omega^2(2\beta - 2\gamma + 1) + 4}{4 + 4\beta\Delta t^2\omega^2}} \\ = \frac{4 + \Delta t^2\omega^2(4\beta - 2\gamma - 1)}{4 + 4\beta\Delta t^2\omega^2}\end{aligned} \tag{A.4}$$

$$\pm\frac{\sqrt{-16\Delta t^2\omega^2+\Delta t^4\omega^4[(1+2\gamma)^2-16\beta]}}{4+4\beta\Delta t^2\omega^2}$$

and for $\omega = 2\pi/T$ finally in

$$\lambda_{2,3}=\frac{1+\frac{\Delta t^2}{T^2}\pi^2(4\beta-2\gamma-1)\pm\pi\sqrt{\frac{\Delta t^4}{T^4}\pi^2[(1+2\gamma)^2-16\beta]-4\frac{\Delta t^2}{T^2}}}{1+4\beta\frac{\Delta t^2}{T^2}\pi^2}. \quad \text{(A.5)}$$

B Derivation of Stability Conditions for Newmark Time-Integration

Unconditional stability means that the stability conditions (6.8) and (6.9) are fulfilled even for arbitrary big time steps. To evaluate (6.18) or (A.5) in terms of these conditions different cases have to be considered. Since λ_1 equals zero, only $\lambda_{2,3}$ is important. We also assume that γ and β have positive values.

Case I:

At first we consider the situation of complex eigenvalues (where $\lambda_2 \neq \lambda_3$ is always true) which appear if the root expression in (6.18) becomes negative. With the substitution $z = \Delta t/T$ this condition can be written as

$$\mathrm{z}^2\pi^2[(1+2\gamma)^2-16\beta]-4<0. \quad \text{(B.6)}$$

Since $\mathrm{z}^2\pi^2$ is always greater than zero inequality (B.6) is fulfilled for $[(1+2\gamma)^2-16\beta]\leq 0$ or $\boldsymbol{\beta\geq(1+2\gamma)^2/16}$, respectively (case Ia). But also for $\beta<(1+2\gamma)^2/16$ inequality (B.6) may be fulfilled if $\mathrm{z}^2\pi^2$ is small enough (case Ib).

Case Ia:

If $\beta\geq(1+2\gamma)^2/16$ the corresponding stability condition

$$\left|\frac{1+\mathrm{z}^2\pi^2(4\beta-2\gamma-1)\pm\mathrm{z}\pi\sqrt{\mathrm{z}^2\pi^2[(1+2\gamma)^2-16\beta]-4}}{1+4\beta\mathrm{z}^2\pi^2}\right|\leq 1 \quad \text{(B.7)}$$

can be transformed to

$$\left|\frac{1+\mathrm{z}^2\pi^2(4\beta-2\gamma-1)\pm i\mathrm{z}\pi\sqrt{-\mathrm{z}^2\pi^2[(1+2\gamma)^2-16\beta]+4}}{1+4\beta\mathrm{z}^2\pi^2}\right|\leq 1 \quad \text{(B.8)}$$

with $i=\sqrt{-1}$ being the imaginary unit which gives

$$\sqrt{[1+\mathrm{z}^2\pi^2(4\beta-2\gamma-1)]^2+\mathrm{z}^2\pi^2[-\mathrm{z}^2\pi^2[(1+2\gamma)^2-16\beta]+4]}\leq 1+4\beta\mathrm{z}^2\pi^2. \quad \text{(B.9)}$$

Squaring and simplifying (B.9) yields

$$\mathrm{z}^2(2\gamma-1)(1+4\beta\mathrm{z}^2\pi^2)\geq 0 \quad \text{(B.10)}$$

from which directly follows $\boldsymbol{\gamma\geq 1/2}$.

Case Ib:

In this case we assume $\boldsymbol{\beta < (1+2\gamma)^2/16}$ and $\boldsymbol{z^2\pi^2 < 4/[(1+2\gamma)^2 - 16\beta]}$ which also fulfills the inequality (B.6) and leads to the stability condition (B.7) resulting in $\boldsymbol{\gamma \geq 1/2}$ for conditional stability.

Case II:

Next, we consider the case of distinct eigenvalues $\lambda_2 \neq \lambda_3$ plus a positive root expression in Equation (6.18) for which

$$z^2\pi^2[(1+2\gamma)^2 - 16\beta] - 4 > 0 \tag{B.11}$$

is necessary. This condition is fulfilled for $\boldsymbol{z^2\pi^2 > 4/[(1+2\gamma)^2 - 16\beta]}$ and $(1+2\gamma)^2 - 16\beta > 0$. The latter results in the premise $\boldsymbol{\beta < (1+2\gamma)^2/16}$. Under the assumption of these two premises we define further subcases which may appear depending on the parameter choice and the time-step size as follows:

Case II(λ_2):

The stability criterion for case II(λ_2) reads

$$\left|1 + z^2\pi^2(4\beta - 2\gamma - 1) + z\pi\sqrt{z^2\pi^2[(1+2\gamma)^2 - 16\beta] - 4}\right| \leq 1 + 4\beta z^2\pi^2 \tag{B.12}$$

where we have to distinguish between subcase II(λ_2)a with

$$1 + z^2\pi^2(4\beta - 2\gamma - 1) + z\pi\sqrt{z^2\pi^2[(1+2\gamma)^2 - 16\beta] - 4} \geq 0 \tag{B.13}$$

and subcase II(λ_2)b with

$$1 + z^2\pi^2(4\beta - 2\gamma - 1) + z\pi\sqrt{z^2\pi^2[(1+2\gamma)^2 - 16\beta] - 4} < 0. \tag{B.14}$$

Case II(λ_3):

The stability criterion for case II(λ_3) reads

$$\left|1 + z^2\pi^2(4\beta - 2\gamma - 1) - z\pi\sqrt{z^2\pi^2[(1+2\gamma)^2 - 16\beta] - 4}\right| \leq 1 + 4\beta z^2\pi^2 \tag{B.15}$$

where we have to distinguish between subcase II(λ_3)a with

$$1 + z^2\pi^2(4\beta - 2\gamma - 1) - z\pi\sqrt{z^2\pi^2[(1+2\gamma)^2 - 16\beta] - 4} \geq 0 \tag{B.16}$$

and subcase II(λ_3)b with

$$1 + z^2\pi^2(4\beta - 2\gamma - 1) - z\pi\sqrt{z^2\pi^2[(1+2\gamma)^2 - 16\beta] - 4} < 0. \tag{B.17}$$

Because only the eigenvalue with the biggest absolute value is important for the stability we do not need to consider all four subcases II(λ_2)a, II(λ_2)b, II(λ_3)a and II(λ_3)b. For this reason we start with a comparison of the absolute values of the eigenvalues of the four subcases to find out whether λ_2 or λ_3 is the more critical eigenvalue.

Comparison 1 (II(λ_2)a with II(λ_3)a, which means $\lambda_2 > 0$ and $\lambda_3 > 0$):

$$\begin{aligned} &1 + \mathrm{z}^2\pi^2(4\beta - 2\gamma - 1) + \mathrm{z}\pi\sqrt{\mathrm{z}^2\pi^2[(1+2\gamma)^2 - 16\beta] - 4} \\ &\quad \gtreqless 1 + \mathrm{z}^2\pi^2(4\beta - 2\gamma - 1) - \mathrm{z}\pi\sqrt{\mathrm{z}^2\pi^2[(1+2\gamma)^2 - 16\beta] - 4} \end{aligned} \tag{B.18}$$

from which directly follows

$$\begin{aligned} &\mathrm{z}\pi\sqrt{\mathrm{z}^2\pi^2[(1+2\gamma)^2 - 16\beta] - 4} \\ &\qquad > -z\pi\sqrt{\mathrm{z}^2\pi^2[(1+2\gamma)^2 - 16\beta] - 4} \end{aligned} \tag{B.19}$$

since $\sqrt{\mathrm{z}^2\pi^2[(1+2\gamma)^2 - 16\beta] - 4} > 0$ because of premise (B.11). Therefore II(λ_2)a is more critical than II(λ_3)a.

Comparison 2 (II(λ_2)b with II(λ_3)b, which means $\lambda_2 < 0$ and $\lambda_3 < 0$):

$$\begin{aligned} &-1 - \mathrm{z}^2\pi^2(4\beta - 2\gamma - 1) - \mathrm{z}\pi\sqrt{\mathrm{z}^2\pi^2[(1+2\gamma)^2 - 16\beta] - 4} \\ &\quad \gtreqless -1 - \mathrm{z}^2\pi^2(4\beta - 2\gamma - 1) + \mathrm{z}\pi\sqrt{\mathrm{z}^2\pi^2[(1+2\gamma)^2 - 16\beta] - 4} \end{aligned} \tag{B.20}$$

from which follows

$$\begin{aligned} &-\mathrm{z}\pi\sqrt{\mathrm{z}^2\pi^2[(1+2\gamma)^2 - 16\beta] - 4} \\ &\qquad < z\pi\sqrt{\mathrm{z}^2\pi^2[(1+2\gamma)^2 - 16\beta] - 4} \end{aligned} \tag{B.21}$$

and therefore II(λ_3)b is more critical than II(λ_2)b.

Comparison 3 (II(λ_2)b with II(λ_3)a, which means $\lambda_2 < 0$ and $\lambda_3 > 0$):

This situation of $\lambda_2 < 0$ and $\lambda_3 > 0$ is not possible for the following reason. If $\lambda_3 > 0$ also $1 + \mathrm{z}^2\pi^2(4\beta - 2\gamma - 1)$ must be greater than zero. But if $1 + \mathrm{z}^2\pi^2(4\beta - 2\gamma - 1) > 0$ the eigenvalue λ_2 can never be negative.

Comparison 4 (II(λ_2)a with II(λ_3)b, which means $\lambda_2 > 0$ and $\lambda_3 < 0$):

$$\begin{aligned} &1 + \mathrm{z}^2\pi^2(4\beta - 2\gamma - 1) + \mathrm{z}\pi\sqrt{\mathrm{z}^2\pi^2[(1+2\gamma)^2 - 16\beta] - 4} \\ &\quad \gtreqless -1 - \mathrm{z}^2\pi^2(4\beta - 2\gamma - 1) + \mathrm{z}\pi\sqrt{\mathrm{z}^2\pi^2[(1+2\gamma)^2 - 16\beta] - 4} \end{aligned} \tag{B.22}$$

gives

$$1 + \mathrm{z}^2\pi^2(4\beta - 2\gamma - 1) \gtreqless 0 \tag{B.23}$$

which means at the moment we are not able to determine whether $\lambda_2 > 0$ or $\lambda_3 < 0$ is more critical.

Summarizing these comparisons, we can say that II(λ_3)a is not critical since it is less critical than II(λ_2)a and may not appear in combination with II(λ_2)b. Plus, case II(λ_2)b is not critical since it is less critical than II(λ_3)b and may not appear in combination with II(λ_3)a. Therefore cases II(λ_3)a and II(λ_2)b do not need to be considered and we continue with the remaining two cases.

Case II(λ_2)a:

In this case inequality (B.12) becomes

$$1 + \mathrm{z}^2\pi^2(4\beta - 2\gamma - 1) + \mathrm{z}\pi\sqrt{\mathrm{z}^2\pi^2[(1+2\gamma)^2 - 16\beta] - 4} \leq 1 + 4\beta\mathrm{z}^2\pi^2 \tag{B.24}$$

and can be simplified to

$$\sqrt{\mathrm{z}^2\pi^2[(1+2\gamma)^2 - 16\beta] - 4} \leq \mathrm{z}\pi(2\gamma + 1) \tag{B.25}$$

which may be squared since the left-hand side is positive because of premise (B.11) and the right-hand side is positive too, leading to

$$-\mathrm{z}^2\pi^2 16\beta \leq 4 \tag{B.26}$$

which is always true.

Case II(λ_3)b:

The stability criterion for case II(λ_3) is given in (B.15) and leads for case II(λ_3)b to the stability condition

$$-1 - \mathrm{z}^2\pi^2(4\beta - 2\gamma - 1) + \mathrm{z}\pi\sqrt{\mathrm{z}^2\pi^2[(1+2\gamma)^2 - 16\beta] - 4} \leq 1 + 4\beta\mathrm{z}^2\pi^2 \tag{B.27}$$

which can be simplified to

$$\mathrm{z}\pi\sqrt{\mathrm{z}^2\pi^2[(1+2\gamma)^2 - 16\beta] - 4} \leq 2 + \mathrm{z}^2\pi^2(8\beta - 2\gamma - 1). \tag{B.28}$$

By introducing the slack variable $s \geq 0$ with $s_{max} = 2 + \mathrm{z}^2\pi^2(8\beta - 2\gamma - 1) - \mathrm{z}\pi\sqrt{\mathrm{z}^2\pi^2[(1+2\gamma)^2 - 16\beta] - 4}$ this inequality can be transformed to the equation

$$\mathrm{z}\pi\sqrt{\mathrm{z}^2\pi^2[(1+2\gamma)^2 - 16\beta] - 4} + \mathrm{s} = 2 + \mathrm{z}^2\pi^2(8\beta - 2\gamma - 1). \tag{B.29}$$

Since premise (B.11) holds and we assume that (B.28) is true we may square Equation (B.29) and get

$$\begin{aligned} -4\pi^2z^2 + \pi^4z^4 - 16\beta\pi^4z^4 + 4\gamma\pi^4z^4 + 4\gamma^2\pi^4z^4 = \\ 4 - 4s + s^2 - 4z^2\pi^2 + 32\beta z^2\pi^2 - 8\gamma z^2\pi^2 + 2z^2\pi^2 s \\ -16\beta z^2\pi^2 s + 4\gamma z^2\pi^2 s + z^4\pi^4 - 16\beta z^4\pi^4 + 64\beta^2 z^4\pi^4 \\ +4\gamma z^4\pi^4 - 32\beta\gamma z^4\pi^4 + 4\gamma^2 z^4\pi^4 \end{aligned} \tag{B.30}$$

which can be simplified to

$$\begin{aligned} \gamma(32\beta z^2\pi^2 + 8 - 4s) \\ = \frac{4 - 4s + s^2}{z^2\pi^2} + 32\beta + 2s - 16\beta s + 64\beta^2 z^2\pi^2 \end{aligned} \tag{B.31}$$

and solved for γ yielding to

$$\gamma(s) = 2\beta + \frac{2 - s}{4\mathrm{z}^2\pi^2} + \frac{s}{4 - 2s + 16\beta\mathrm{z}^2\pi^2}. \tag{B.32}$$

Using the technique of slack variables for the solution of inequalities requires to discuss in our case the possible values of γ resulting from (B.32) for all possible values of the slack variable s. For gaining a better understanding the function $\gamma(s)$ is plotted for $\beta = 0.55$ and $z = 1.43$ in Fig. B.1. Starting from $s = 0$ we see that there is an upper boundary for γ of

$$\gamma(s) \leq \gamma(s = 0) = 2\beta + \frac{2}{4\mathrm{z}^2\pi^2} \tag{B.33}$$

which leads for arbitrary big time-steps $z \to \infty$ to the condition

$$\gamma \leq 2\beta \tag{B.34}$$

or

$$\beta \geq \frac{\gamma}{2}. \tag{B.35}$$

Obviously, there is only one local minimum for $s \geq 0$ and positive γ-values. Determining this minimum from $\partial\gamma/\partial s = 0$ yields

$$\frac{\partial\gamma}{\partial s} = -\frac{1}{4\mathrm{z}^2\pi^2} + \frac{4 + 16\beta\mathrm{z}^2\pi^2}{(4 - 2s + 16\beta\mathrm{z}^2\pi^2)^2} = 0 \tag{B.36}$$

with the solutions

$$s_{min,1,2} = 2 + 8\beta\mathrm{z}^2\pi^2 \pm 2\mathrm{z}\pi\sqrt{1 + 4\beta\mathrm{z}^2\pi^2} \tag{B.37}$$

where the first solution with the plus sign $s_{min,1} = 2 + 8\beta\mathrm{z}^2\pi^2 + 2\mathrm{z}\pi\sqrt{1 + 4\beta\mathrm{z}^2\pi^2}$ is bigger than the singularity at $s = 8\beta\mathrm{z}^2\pi^2 + 2$ and therefore corresponds to a negative γ-value which is not of interest. The second solution $s_{min,2} = 2 + 8\beta\mathrm{z}^2\pi^2 - 2\mathrm{z}\pi\sqrt{1 + 4\beta\mathrm{z}^2\pi^2}$ is really a minimum since

$$\frac{\partial^2\gamma}{\partial s^2} = \frac{16 + 64\beta\mathrm{z}^2\pi^2}{(4 - 2s + 16\beta\mathrm{z}^2\pi^2)^3} = \frac{2 + 8\beta\mathrm{z}^2\pi^2}{(2 - s + 8\beta\mathrm{z}^2\pi^2)^3} \tag{B.38}$$

and consequently

$$\begin{aligned} \left.\frac{\partial^2\gamma}{\partial s^2}\right|_{s_{min,2}} &= \frac{2 + 8\beta\mathrm{z}^2\pi^2}{\left(2 - 2 - 8\beta\mathrm{z}^2\pi^2 + 2\mathrm{z}\pi\sqrt{1 + 4\beta\mathrm{z}^2\pi^2} + 8\beta\mathrm{z}^2\pi^2\right)^3} \\ &= \frac{1 + 4\beta\mathrm{z}^2\pi^2}{4\left(\mathrm{z}\pi\sqrt{1 + 4\beta\mathrm{z}^2\pi^2}\right)^3} = \frac{1}{4\mathrm{z}^3\pi^3\sqrt{1 + 4\beta\mathrm{z}^2\pi^2}} > 0. \end{aligned} \tag{B.39}$$

Starting from $s > 0$ we can see by increasing s that $\gamma(s)$ may have all positive values with

$$\begin{aligned} \gamma(s) \geq \gamma\left(s_{min,2}\right) &= 2\beta + \frac{4\beta\mathrm{z}^2\pi^2 - \mathrm{z}\pi\sqrt{1 + 4\beta\mathrm{z}^2\pi^2}}{2\mathrm{z}^2\pi^2} \\ &\quad + \frac{1 + 4\beta\mathrm{z}^2\pi^2 - \mathrm{z}\pi\sqrt{1 + 4\beta\mathrm{z}^2\pi^2}}{2\mathrm{z}\pi\sqrt{1 + 4\beta\mathrm{z}^2\pi^2}} \\ &= -\frac{1}{2} + \frac{\sqrt{1 + 4\beta\mathrm{z}^2\pi^2}}{\mathrm{z}\pi} \end{aligned} \tag{B.40}$$

which means that if condition (B.40) is fulfilled in addition to condition (B.35) also the stability condition (B.27) for case II(λ_3)b is fulfilled. Since stability for arbitrary big time-steps should be guaranteed

$$\gamma(s) \geq \gamma\left(s_{min,2}, z\pi \to \infty\right) = -\frac{1}{2} + 2\sqrt{\beta} \tag{B.41}$$

must hold which can be transformed to the condition

$$\beta \leq \frac{(2\gamma+1)^2}{16} \tag{B.42}$$

and is already fulfilled by the premises for case II. For the smallest possible time-step according to the premises of case II we get

$$\begin{aligned} \gamma(s) &\geq \gamma\left(s_{min,2}, z\pi \to \sqrt{\frac{4}{(1+2\gamma)^2 - 16\beta}}\right) \\ &= -\frac{1}{2} + \frac{\sqrt{1 + \frac{16\beta}{(1+2\gamma)^2 - 16\beta}}}{\sqrt{\frac{4}{(1+2\gamma)^2 - 16\beta}}} \\ &= \gamma \end{aligned} \tag{B.43}$$

which is always fulfilled. Concluding from these considerations we can say that case II(λ_3)b is always stable for the premises of case II plus condition (B.35).

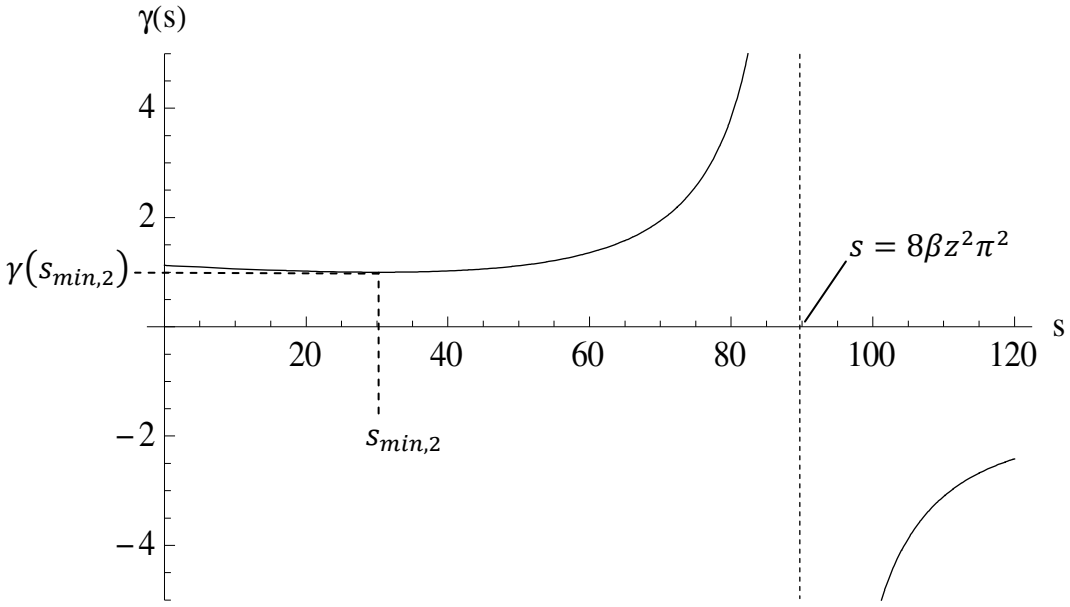


Fig. B.1 Function $\gamma(s)$ for $\beta = 0.55$ and $z = \Delta t/T = 1.43$

Case III:

The last case we have to consider is the situation of a double eigenvalue $\lambda_2 = \lambda_3$, which is true for

$$z\pi\sqrt{\mathrm{z}^2\pi^2[(1+2\gamma)^2-16\beta]-4}=0. \tag{B.44}$$

Since πz is always greater than zero

$$\mathrm{z}^2\pi^2=\frac{4}{(1+2\gamma)^2-16\beta} \tag{B.45}$$

must hold to fulfill (B.44) leading to

$$\lambda_2=\lambda_3=\frac{1+\mathrm{z}^2\pi^2(4\beta-2\gamma-1)}{1+4\beta\mathrm{z}^2\pi^2}. \tag{B.46}$$

By using (B.45) in (B.46) we get

$$\begin{aligned}\lambda_{2,3}&=\frac{1+\dfrac{4(4\beta-2\gamma-1)}{(1+2\gamma)^2-16\beta}}{1+\dfrac{16\beta}{(1+2\gamma)^2-16\beta}}\\&=\frac{(1+2\gamma)^2-16\beta+4(4\beta-2\gamma-1)}{(1+2\gamma)^2}=\frac{2\gamma-3}{2\gamma+1}.\end{aligned} \tag{B.47}$$

Stability is guaranteed for double eigenvalues according to (6.9) if

$$\left|\lambda_{2,3}\right|<1 \tag{B.48}$$

which is always true for $\boldsymbol{\gamma>1/2}$, obviously. A positive and finite value for $\mathrm{z}^2\pi^2$ requires $\boldsymbol{\beta<(1+2\gamma)^2/16}$, which directly follows from Equation (B.45) . Otherwise case C may not occur.

Summarizing all investigations above we can say that we always have complex eigenvalues for $\boldsymbol{\beta\geq(1+2\gamma)^2/16}$ and unconditional stability is guaranteed for $\boldsymbol{\gamma\geq1/2}$ following from case Ia. In case II the premise $\beta<(1+2\gamma)^2/16$ together with $\beta\geq\gamma/2$ fulfills all necessary stability conditions as long as $\mathrm{z}^2\pi^2>4/[(1+2\gamma)^2-16\beta]$ and leads to real eigenvalues. If the time-step size decreases and $\mathrm{z}^2\pi^2<4/[(1+2\gamma)^2-16\beta]$ is true case Ib (complex eigenvalues) becomes active and $\gamma\geq1/2$ is demanded additionally. For the special case of a double real eigenvalue (case III) which means $\mathrm{z}^2\pi^2=4/[(1+2\gamma)^2-16\beta]$ the stricter condition $\gamma>1/2$ is necessary. This means, unconditional stability for real eigenvalues can only be guaranteed for $\boldsymbol{\gamma/2\leq\beta<(1+2\gamma)^2/16}$ and $\boldsymbol{\gamma>1/2}$.

C Computation of Eigenvalues for Stability Analysis of Newmark-Euler Algorithm

For $\gamma=1/2$, $\beta=1/4$, $\omega=2\pi/T$, $\Delta t_n=\Delta t_{n-1}=\Delta t_n$ and with the abbreviations

$$z_3=1+\frac{\Delta t^2\pi^2}{T^2}, \tag{C.49}$$

$$z_4=(1+\alpha)^2+\frac{4\Delta t^2\pi^2}{T^2} \tag{C.50}$$

we get for (6.26)

$$\boldsymbol{A}_{Comp} = \frac{1}{z_2 z_3}\begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} \tag{C.51}$$

with

$$\begin{aligned}
A_{11} &= -(3+\alpha)(1+2\alpha)\Delta t^2\pi^2, \\
A_{12} &= 4\Delta t\pi^2\left[(1+3\alpha)\frac{\Delta t^2\pi^2}{T^2} - 2(1+\alpha)^2\right], \\
A_{13} &= 4\pi^2\left[(2+5\alpha+\alpha^2)\frac{\Delta t^2\pi^2}{T^2} - (1+\alpha)^2\right], \\
A_{21} &= \left(\frac{\Delta t + 2\alpha\Delta t}{2}\right)\left(1+\alpha - \frac{2\Delta t^2\pi^2}{T^2}\right), \\
A_{22} &= (1+\alpha)^2T^2 - (5+12\alpha+3\alpha^2)\Delta t^2\pi^2, \\
A_{23} &= -2\Delta t\pi^2\left(3+5\alpha+2\alpha^2 - \frac{2\alpha\Delta t^2\pi^2}{T^2}\right), \\
A_{31} &= \left(\frac{3+\alpha}{4}\right)(1+2\alpha)\Delta t^2T^2, \\
A_{32} &= \Delta t[2(1+\alpha)^2T^2 - (1+3\alpha)\Delta t^2\pi^2], \\
A_{33} &= (1+\alpha)^2T^2 - (2+5\alpha+\alpha^2)\Delta t^2\pi^2.
\end{aligned}$$

Computing the characteristic polynomial of $\det\left(\lambda\boldsymbol{E} - \boldsymbol{A}_{Comp}\right)$ and setting it equal to zero yields

$$\begin{aligned}
&\lambda^3 + \frac{\lambda^2}{z_3 z_4}\left(\frac{6\alpha^2\Delta t^2\pi^2}{T^2} + \frac{24\alpha\Delta t^2\pi^2}{T^2} + \frac{10\Delta t^2\pi^2}{T^2} - 2\alpha^2 - 4\alpha - 2\right) \\
&\quad + \frac{\lambda}{z_3^2 z_4^2}\left(1 + \frac{10\alpha^4\Delta t^2\pi^2}{T^2} + \frac{28\alpha^3\Delta t^2\pi^2}{T^2} + \frac{32\alpha^2\Delta t^2\pi^2}{T^2} + \frac{20\alpha\Delta t^2\pi^2}{T^2}\right. \\
&\qquad + \frac{6\Delta t^2\pi^2}{T^2} + \frac{9\alpha^4\Delta t^4\pi^4}{T^4} + \frac{24\alpha^3\Delta t^4\pi^4}{T^4} + \frac{62\alpha^2\Delta t^4\pi^4}{T^4} \\
&\qquad + \frac{40\alpha\Delta t^4\pi^4}{T^4} + \frac{9\Delta t^4\pi^4}{T^4} + \frac{36\alpha^2\Delta t^6\pi^6}{T^6} + \frac{24\alpha\Delta t^6\pi^6}{T^6} \\
&\qquad \left. + \frac{4\Delta t^6\pi^6}{T^6} + \alpha^4 + 4\alpha^3 + 6\alpha^2 + 4\alpha\right) = 0.
\end{aligned} \tag{C.52}$$

It directly follows $\lambda_1 = 0$ and the remaining quadratic equation has the solutions

$$\begin{aligned}
\lambda_{2,3} = &\frac{(1+\alpha)^2 - [5+3\alpha(4+\alpha)]\left(\frac{\Delta t\pi}{T}\right)^2}{z_3 z_4} \\
&\pm \frac{2\pi\sqrt{-\left[(1+3\alpha)\frac{\Delta t^3\pi^2}{T^3} - 2(1+\alpha)^2\frac{\Delta t}{T}\right]^2}}{z_3 z_4}.
\end{aligned} \tag{C.53}$$

Reversing the substitutions (C.49) and (C.50) finally gives

$$\lambda_{2,3} = \frac{(1+\alpha)^2 - [5+3\alpha(4+\alpha)]\left(\frac{\Delta t\pi}{T}\right)^2}{\left(1+\frac{\Delta t^2\pi^2}{T^2}\right)\left[(1+\alpha)^2+\frac{4\Delta t^2\pi^2}{T^2}\right]} \pm \frac{2\pi\sqrt{-\left[(1+3\alpha)\frac{\Delta t^3\pi^2}{T^3} - 2(1+\alpha)^2\frac{\Delta t}{T}\right]^2}}{\left(1+\frac{\Delta t^2\pi^2}{T^2}\right)\left[(1+\alpha)^2+\frac{4\Delta t^2\pi^2}{T^2}\right]}. \tag{C.54}$$

D Derivation of Stability Conditions for Newmark-Euler Time-Integration

For the derivation of the stability condition with respect to α two cases have to be considered. Since $\lambda_1 = 0$ only $\lambda_{2,3}$ is relevant. In case I we assume distinct eigenvalues $\lambda_2 \neq \lambda_3$ which automatically leads to complex eigenvalues. In case II real eigenvalues are considered which appear for a zero root expression in Equation (6.27). For both investigations the premise $\alpha \geq 0$ is used.

Case I:

According to condition (6.8) it is necessary for the stability of the Newmark-Euler time-integration scheme that $|\lambda_i| \leq 1$ is fulfilled in case of distinct eigenvalues $\lambda_2 \neq \lambda_3$ and the stability condition reads as

$$\left[\left(\frac{(1+\alpha)^2 - [5+3\alpha(4+\alpha)]\left(\frac{\Delta t\pi}{T}\right)^2}{\left(1+\frac{\Delta t^2\pi^2}{T^2}\right)\left[(1+\alpha)^2+\frac{4\Delta t^2\pi^2}{T^2}\right]}\right)^2 + \left(\frac{2\pi\sqrt{\left[(1+3\alpha)\frac{\Delta t^3\pi^2}{T^3} - 2(1+\alpha)^2\frac{\Delta t}{T}\right]^2}}{\left(1+\frac{\Delta t^2\pi^2}{T^2}\right)\left[(1+\alpha)^2+\frac{4\Delta t^2\pi^2}{T^2}\right]}\right)^2\right]^{\frac{1}{2}} \leq 1. \tag{D.55}$$

By using the abbreviation $z = \Delta t/T$ it may be transformed to

$$[((1+\alpha)^2 - [5+3\alpha(4+\alpha)]z^2\pi^2)^2 + 4\pi^2[(1+3\alpha)z^3\pi^2 - 2(1+\alpha)^2 z]^2]^{\frac{1}{2}} \leq (1+z^2\pi^2)[(1+\alpha)^2 + 4z^2\pi^2] \tag{D.56}$$

and squared and simplified leading to

$$z^2(1+\pi^2z^2)[1+\alpha-5\alpha^3-2\alpha^4+5\pi^2z^2-2\alpha\pi^2z^2+4\pi^4z^4 - \alpha^3(3+7\pi^2z^2)] \geq 0 \tag{D.57}$$

and after further simplification to

$$(1-\alpha-2\alpha^2+\pi^2z^2)[(1+\alpha)^2+4\pi^2z^2]\geq 0 \tag{D.58}$$

where the second term is always positive according to the premises $\alpha \geq 0$, $z > 0$ from which follows

$$(1-\alpha-2\alpha^2+\pi^2z^2)\geq 0. \tag{D.59}$$

Since this condition has to be fulfilled also for arbitrarily small normalized time-steps sizes $z \to 0$

$$1-\alpha-2\alpha^2\geq 0 \tag{D.60}$$

must hold which leads for positive α to the stability condition

$$\alpha \leq \frac{1}{2}. \tag{D.61}$$

Case II:

In case II we assume

$$(1+3\alpha)\frac{\Delta t^3\pi^2}{T^3}-2(1+\alpha)^2\frac{\Delta t}{T}=0 \tag{D.62}$$

leading to the double eigenvalue

$$\lambda_{2,3}=\frac{(1+\alpha)^2-[5+3\alpha(4+\alpha)]\left(\frac{\Delta t\pi}{T}\right)^2}{\left(1+\frac{\Delta t^2\pi^2}{T^2}\right)\left[(1+\alpha)^2+\frac{4\Delta t^2\pi^2}{T^2}\right]} \tag{D.63}$$

and the resulting stability condition

$$\left|\frac{(1+\alpha)^2-[5+3\alpha(4+\alpha)]\left(\frac{\Delta t\pi}{T}\right)^2}{\left(1+\frac{\Delta t^2\pi^2}{T^2}\right)\left[(1+\alpha)^2+\frac{4\Delta t^2\pi^2}{T^2}\right]}\right|<1 \tag{D.64}$$

where we have to distinguish between case IIa with

$$(1+\alpha)^2-[5+3\alpha(4+\alpha)]\left(\frac{\Delta t\pi}{T}\right)^2\geq 0 \tag{D.65}$$

and case IIb with

$$(1+\alpha)^2-[5+3\alpha(4+\alpha)]\left(\frac{\Delta t\pi}{T}\right)^2<0. \tag{D.66}$$

Case IIa:

The stability criterion for case IIa reads with $z=\Delta t/T$

$$\frac{(1+\alpha)^2-[5+3\alpha(4+\alpha)]\pi^2z^2}{(1+\pi^2z^2)[(1+\alpha)^2+4\pi^2z^2]}<1 \tag{D.67}$$

and can be transformed to

$$\frac{z^2(5+\alpha(7+2\alpha)+2\pi^2z^2)}{(1+\pi^2z^2)[(1+\alpha)^2+4\pi^2z^2]}>0 \tag{D.68}$$

which is always true for $\alpha \geq 0$.

Case IIb:

The stability criterion for case IIb reads with $z = \Delta t/T$

$$\frac{-(1+\alpha)^2+[5+3\alpha(4+\alpha)]\pi^2z^2}{(1+\pi^2z^2)[(1+\alpha)^2+4\pi^2z^2]}<1 \tag{D.69}$$

and can be transformed to

$$\alpha^2(1-\pi^2z^2)+\alpha(2-5\pi^2z^2)+1+2\pi^4z^4>0. \tag{D.70}$$

Case II only appears if Equation (D.62) is fulfilled, which can be solved for $\Delta t/T = z$ resulting in a positive and negative solution where only the positive solution is of interest and reads

$$\frac{\Delta t}{T}=z=\frac{\sqrt{2(1+\alpha)^2}}{\pi\sqrt{1+3\alpha}}. \tag{D.71}$$

Using this in (D.70) yields

$$\frac{(\alpha-1)(1+\alpha)^2(3+\alpha)(1+2\alpha)}{(1+3\alpha)^2}<0 \tag{D.72}$$

which is only fulfilled for positive α if $\alpha - 1 < 0$ or

$$\alpha<1, \tag{D.73}$$

respectively.

Summarizing the investigations of case I and case II we can conclude that $\mathbf{0 \leq \alpha \leq 1/2}$ guarantees unconditional stability for the Newmark-Euler time-integration scheme.

E Derivation of Stability Conditions for Central Difference Method

To derive a stability condition for the central difference method by using the spectral radius and therefore the eigenvalues of the amplification matrix (6.32) three different cases have to be considered.

Case I:

At first, we assume that the root expression in Equation (6.32) equals zero which means

$$\left[1-2\pi^2\left(\frac{\Delta t}{T}\right)^2\right]^2-1=0. \tag{E.74}$$

Obviously, this condition is fulfilled for $\pi\Delta t/T = \pm 1$ or $\Delta t = 0$ where only the case $\pi\Delta t/T = 1$ has practical relevance since the time-step size is always greater than zero. Hence, the stability condition (6.9) has to be applied leading to

$$1 - 2\pi^2 \left(\frac{\Delta t}{T}\right)^2 < 1 \tag{E.75}$$

which is always fulfilled for $\pi \Delta t/T = 1$.

Case II:

In case II we assume a negative root expression in Equation (6.32)

$$\left[1 - 2\pi^2 \left(\frac{\Delta t}{T}\right)^2\right]^2 - 1 < 0 \tag{E.76}$$

and therefore complex eigenvalues $\lambda_{1,2}$ which are obtained for $0 < \pi^2 z^2 < 1$ by using again the abbreviation $z = \Delta t/T$. The stability condition (6.8) becomes

$$\sqrt{(1 - 2\pi^2 z^2)^2 + \left(\sqrt{-(1 - 2\pi^2 z^2)^2 + 1}\right)^2} \leq 1 \tag{E.77}$$

and can be simplified to

$$(1 - 2\pi^2 z^2)^2 - (1 - 2\pi^2 z^2)^2 + 1 \leq 1 \tag{E.78}$$

which is always true.

Case III:

The last possibility we have to consider is a positive root expression in Equation (6.32) which holds for

$$(1 - 2\pi^2 z^2)^2 - 1 > 0 \tag{E.79}$$

or

$$\pi^2 z^2 > 1, \tag{E.80}$$

respectively and leads to real eigenvalues

$$\lambda_1 = 1 - 2\pi^2 z^2 + \sqrt{(1 - 2\pi^2 z^2)^2 - 1}, \tag{E.81}$$

$$\lambda_2 = 1 - 2\pi^2 z^2 - \sqrt{(1 - 2\pi^2 z^2)^2 - 1}. \tag{E.82}$$

The first eigenvalue λ_1 is always smaller than zero because of

$$\begin{aligned} 1 - 2\pi^2 z^2 + \sqrt{(1 - 2\pi^2 z^2)^2 - 1} &\lesseqgtr 0 \\ \left(\sqrt{(1 - 2\pi^2 z^2)^2 - 1}\right)^2 &\lesseqgtr (2\pi^2 z^2 - 1)^2 \\ -1 &< 0. \end{aligned} \tag{E.83}$$

This means for stability

$$-1 + 2\pi^2 z^2 - \sqrt{(1 - 2\pi^2 z^2)^2 - 1} \leq 1 \tag{E.84}$$

and consequently

$$2\pi^2 z^2 - 2 \leq \sqrt{(1 - 2\pi^2 z^2)^2 - 1} \tag{E.85}$$

has to be satisfied. Squaring and simplifying (E.85) yields

$$4\pi^4 z^4 - 8\pi^2 z^2 + 4 \leq -4\pi^2 z^2 + 4\pi^4 z^4 \tag{E.86}$$

and finally

$$4 \leq 4\pi^2 z^2 \tag{E.87}$$

which is fulfilled for the premise $\pi^2 z^2 > 1$.

The second real eigenvalue λ_2 is also always negative since $1 - 2\pi^2 z^2$ is negative for $\pi^2 z^2 > 1$. By this, the stability condition for λ_2 reads

$$-1 + 2\pi^2 z^2 + \sqrt{(1 - 2\pi^2 z^2)^2 - 1} \leq 1 \tag{E.88}$$

and may be transformed to

$$\sqrt{(1 - 2\pi^2 z^2)^2 - 1} \leq 2 - 2\pi^2 z^2 \tag{E.89}$$

which can never be fulfilled for $\pi^2 z^2 > 1$ since the right-hand side is negative and the left hand-side positive.

Summarizing the outcome of these investigations we can conclude that stability is guaranteed for complex eigenvalues or a double real eigenvalue which requires $\pi^2 z^2 \leq 1$ or $\Delta \boldsymbol{t} \leq \boldsymbol{T}/\boldsymbol{\pi}$. For bigger time-steps one of the resulting real eigenvalues has a magnitude smaller than one and the other one a magnitude greater than one where the latter leads to an unstable behavior of the integration scheme (see Fig. 6.13).

F Deflection Function of Thickness-Variable Beam

Starting with Equation (8.3) we get for the first integration

$$\begin{aligned} w(x) &= \frac{12M_0}{Eb} \int_0^x \left[\frac{-l^3}{2(t_1 - t_0)[lt_0 + x(t_1 - t_0)]^2} \right]_0^x \mathrm{d}x + C_1 x + C_2 \\ &= \frac{12M_0}{Eb} \int_0^x \left[\frac{-l^3}{2(t_1 - t_0)[lt_0 + x(t_1 - t_0)]^2} + \frac{l}{2(t_1 - t_0)t_0^2} \right] \mathrm{d}x + C_1 x \\ &\quad + C_2 \\ &= \frac{12M_0}{Eb} \int_0^x \left[\frac{lx[2lt_0 + x(t_1 - t_0)]}{2t_0^2[lt_0 + x(t_1 - t_0)]^2} \right] \mathrm{d}x + C_1 x + C_2 \end{aligned} \tag{F.90}$$

and for the second one

$$\begin{aligned} w(x) &= \frac{12M_0}{Eb} \left[\frac{l\left[l^2 t_0^2 + \left(lt_0 + x(t_1 - t_0)\right)^2\right]}{2(t_1 - t_0)^2 t_0^2 [lt_0 + x(t_1 - t_0)]} \right]_0^x + C_1 x + C_2 \\ &= \frac{12M_0}{Eb} \left[\frac{l\left[l^2 t_0^2 + \left(lt_0 + x(t_1 - t_0)\right)^2\right]}{2(t_1 - t_0)^2 t_0^2 [lt_0 + x(t_1 - t_0)]} - \frac{l^2}{(t_1 - t_0)^2 t_0} \right] + C_1 x + C_2 \end{aligned} \tag{F.91}$$

$$= \frac{6M_0}{Eb}\left[\frac{lx^2}{t_0^2[lt_0 + x(t_1 - t_0)]}\right] + C_1 x + C_2.$$

From the boundary condition $w(x = 0) = 0$ immediately follows $C_2 = 0$. The second boundary condition $w(x = l) = 0$ leads to

$$w(x = l) = \frac{6M_0}{Eb}\left(\frac{l^2}{t_0^2 t_1}\right) + C_1 l = 0 \tag{F.92}$$

and thus

$$C_1 = -\frac{6M_0}{Eb}\left(\frac{l}{t_0^2 t_1}\right) \tag{F.93}$$

and finally to the deflection function

$$w(x) = \frac{6M_0 lx}{Ebt_0^2}\left(\frac{x}{lt_0 + x(t_1 - t_0)} - \frac{1}{t_1}\right). \tag{F.94}$$

The point of maximum deflection is obtained by setting $w'(x) = \partial w/\partial x$ equal to zero which gives

$$w'(x) = \frac{6M_0 l[(t_1 - t_0)x^2 + 2lt_0 x - l^2 t_0]}{Ebt_o t_1[lt_0 + (t_1 - t_0)x]^2} = 0 \tag{F.95}$$

and solving it for x resulting in

$$x_{1,2} = l \cdot \frac{t_0 \pm \sqrt{t_0 t_1}}{t_0 - t_1}. \tag{F.96}$$

where only

$$x = l \cdot \frac{t_0 - \sqrt{t_0 t_1}}{t_0 - t_1} \tag{F.97}$$

fulfills $0 < x < l$. Using this solution in Equation (F.94) yields

$$w\left(x = l \cdot \frac{t_0 - \sqrt{t_0 t_1}}{t_0 - t_1}\right) = w_{max} = -\frac{6M_0 l^2}{Ebt_0\left(t_1 + \sqrt{t_0 t_1}\right)^2}. \tag{F.98}$$

References

[1] Airy, G.B. (1862): On the Strains in the Interior of Beams. Report of the 32nd meeting of the British Association for the Advancement of Science, October 1862, Cambridge, pp. 82-86, John Murray, London.

[2] Archer, J. S. (1963): Consistent Mass Matrix for Distributed Systems. Journal of the Structural Division, American Society of Civil Engineers, Volume 89, Issue ST4, pp. 161-178.

[3] Ashcraft, C. and Rouet, F.-H. (2016): A Global, Distributed Ordering Library. Proceedings of SIAM Workshop on Combinatorial Scientific Computing, October 10-12, 2016, Albuquerque, USA.

[4] Ascher, U.M. and Petzold, L.R. (1998): Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations. Society for Industrial and Applied Mathematics, Philadelphia, USA.

[5] Askes, H., Nguyen, D.C.D. and Tyas, A. (2011): Increasing the Critical Time Step: Micro-Inertia, Inertia Penalties and Mass Scaling. Computational Mechanics, Volume 47, Issue 6, pp. 657-667.

[6] Bank R.E, Coughran, W.M., Fichter W., Grosse, E.H., Rose D.J. and Smith, R.K. (1985): Transient Simulations of Silicon Devices and Circuits. IEEE Transactions on Computer-Aided Design, Volume 4, Issue 4, pp. 436-51.

[7] Barlow, J. (1989): More on Optimal Stress Points-Reduced Integration, Element Distortions and Error Estimation. International Journal for Numerical Methods in Engineering, Volume 28, Issue 7, pp. 1487-1504.

[8] Barlow, J. (1968): A Stiffness Matrix for a Curved Membrane Shell. Proceedings of Conference on Recent Advances in Stress Analysis, March 26-29, 1968, London, England.

[9] Barsky, B.A. and DeRose, T.D. (1990): Geometric Continuity of Parametric Curves: Constructions of Geometrically Continuous Splines, IEEE Computer Graphics and Applications, Volume 10, Issue 1, pp. 60-68.

[10] Barsky, B.A. and DeRose, T.D. (1989): Geometric Continuity of Parametric Curves: Three Equivalent Characterizations. IEEE Computer Graphics and Applications, Volume 9, Issue 6, pp. 60-68.

[11] Bathe, K.-J. (2014): Finite Element Procedures. 2nd edition, Klaus-Jürgen Bathe, Watertown, Massachusetts, USA.

[12] Bathe, K.-J. (2007): Conserving Energy and Momentum in Nonlinear Dynamics: A Simple Implicit Time Integration Scheme. Computers and Structures, Volume 85, Issues 7-8, pp. 437-445.

[13] Bathe, K.-J. and Baig, M.M.I. (2005): On a Composite Implicit Time Integration Procedure for Nonlinear Dynamics. Computers and Structures, Volume 83, Issues 31-32, pp. 2513-2524.

[14] Bathe, K.-J. and Dvorkin, E.N. (1986): A Formulation of General Shell Elements – the Use of Mixed Interpolation of Tensorial Components. International Journal for Numerical Methods in Engineering, Volume 22, Issue 3, pp. 697-722.

[15] Bathe, K.-J. and Dvorkin, E.N. (1985): A Four-Node Plate Bending Element Based on Mindlin/Reissner Theory and a Mixed Interpolation. International Journal for Numerical Methods in Engineering, Volume 21, Issue 2, pp. 367-383.

[16] Bathe, K.-J. and Noh, G. (2012): Insight into an Implicit Time Integration Scheme for Structural Dynamics. Computers and Structures, Volumes 98-99, pp. 1-6.

[17] Bathe, K.-J. and Wilson, E.L. (1973): Stability and Accuracy Analysis of Direct Integration Methods. Earthquake Engineering and Structural Dynamics, Volume 1, Issue 3, pp. 283-291.

[18] Belgacem, F.B., Hild, P. and Laborde, P. (1998): The Mortar Finite Element Method for Contact Problems. Mathematical and Computer Modeling, Volume 28, Issues 4-8, pp. 263-271.

[19] Belytschko, T. and Bindemann, L.P. (1993): Assumed Strain Stabilization of the Eight Node Hexahedral Element. Computer Methods in Applied Mechanics and Engineering, Volume 105, Issue 2, pp. 225-260.

[20] Belytschko, T. and Bindemann, L.P. (1991): Assumed Strain Stabilization of the 4-Node Quadrilateral with 1-Point Quadrature for Nonlinear Problems. Computer Methods in Applied Mechanics and Engineering, Volume 88, Issue 3, pp. 311-340.

[21] Belytschko, T., Liu, W.K. and Moran, B. (2000): Nonlinear Finite Elements for Continua and Structures. John Wiley & Sons, Chichester/NewYork/Brisbane/ Toronto/Singapore.

[22] Belytschko, T. and Schoeberle, D.F. (1975): On the Unconditional Stability of an Implicit Algorithm for Nonlinear Structural Dynamics. Journal of Applied Mechanics, Volume 42, Issue 4, pp. 865-869.

[23] Betsch, P., Gruttmann, F. and Stein, E. (1996): A 4-Node Finite Shell Element for the Implementation of General Hyperelastic 3D-Elasticity at Finite Strains. Computer Methods in Applied Mechanics and Engineering, Volume 130, Issues 1-2, pp. 57-79.

[24] Bletzinger, K.-U., Bischoff, M. and Ramm, E. (2000): A Unified Approach for Shear-Locking-Free Triangular and Rectangular Shell Finite Elements. Computers and Structures, Volume 75, Issue 3, pp. 321-334.

[25] Borrvall, T. (2009): A Heuristic Attempt to Reduce Transverse Shear Locking in Fully Integrated Hexahedra with Poor Aspect Ratio. Proceedings of 7th European LS-Dyna Conference, May 14-15, 2009, Salzburg, Austria.

[26] Broyden, C.G. (1970): The Convergence of a Class of Double-Rank Minimization Algorithms - 1. General Considerations. Journal of Applied Mathematics, Volume 6, Issue 1, pp. 76-90.

[27] Bucalem, M.L. and Bathe, K.-J. (2011): The Mechanics of Solids and Structures – Hierarchical Modeling and the Finite Element Solution. Springer, Berlin/Heidelberg.

[28] Butcher, J. C. (1987): The Numerical Analysis of Ordinary Differential Equations: Runge-Kutta and General Linear Methods. Wiley, Chichester/New York.

[29] Chapelle, D. and Bathe, K.-J. (2003): The Finite Element Analysis of Shells – Fundamentals. Springer, Berlin/Heidelberg.

[30] Chapra, S.C. and Canale, R.P. (2002): Numerical Methods for Engineers. 4th edition, McGraw-Hill, New York.

[31] Chen, Ch. and Ricles, J.M. (2008): Stability Analysis of Direct Integration Algorithms Applied to Nonlinear Structural Dynamics. Journal of Engineering Mechanics, ASCE, Volume 134, Issue 9, pp. 703-711.

[32] Chung, J. and Hulbert, G.M. (1993): A Time Integration Algorithm for Structural Dynamics With Improved Numerical Dissipation: The Generalized-α Method. ASME Journal of Applied Mechanics, Volume 60, Issue 2, pp. 371-375.

[33] Cottrell, J.A., Hughes, T.J.R. and Bazilevs, Y. (2009): Isogeometric Analysis: Toward Integration of CAD and FEA. John Wiley&Sons, Chichester/New York.

[34] Courant, R., Friedrichs, K. and Lewy, H. (1928): Über die partiellen Differenzengleichungen der mathematischen Physik. Mathematische Annalen, Volume 100, Issue 1, pp. 32-74.

[35] Cowper, G.R. (1973): Gaussian Quadrature Formulas for Triangles. Numerical Methods in Engineering, Volume 7, Issue 3, pp. 405-408.

[36] Cowper, G.R. (1966): The Shear Coefficient in Timoshenko's Beam Theory. Journal of Applied Mechanics, Volume 33, Issue 2, pp. 335-340.

[37] Craig, R.R. (2000): Coupling of Substructures for Dynamic Analyses: An Overview. Proceedings of the 41th AIAA Structures, Structural Dynamics, and Materials Conference and Exhibit, April 3-6, 2000, Atlanta, USA, Paper number AIAA-2000-1573.

[38] Crisfield, M.A. (1997): Non-Linear Finite Element Analysis of Solids and Structures. Volume 2: Advanced Topics, John Wiley & Sons, Chichester/New York.

[39] Crisfield, M.A. and Shi, J. (1994): A Co-Rotational Element/Time-Integration Strategy for Non-Linear Dynamics. International Journal for Numerical Methods in Engineering, Volume 37, Issue 11, pp. 1897-1913.

[40] Cui, J., Wang, H., Xiao, F. and Dong, Z. (2013): Finite Element Modeling and Analysis of Rolling Bearing based on Explicit Dynamics. Applied Mechanics and Materials, Volumes 397-400, pp. 589-592.

[41] Czekanski, A., El-Abbasi, N. and Meguid, S.A. (2001): Optimal Time Integration Parameters for Elastodynamic Contact Problems. Communications in Numerical Methods in Engineering, Volume 17, Issue 6, pp. 379-384.

[42] Daniel, W.J.T. and Belytschko, T. (2005): Suppression of Spurious Intermediate Frequency Modes in Under-Integrated Elements by Combined Stiffness/Viscous Stabilization. International Journal for Numerical Methods in Engineering, Volume 64, Issue 3, pp. 335-353.

[43] Dassault Systèmes (2014): Abaqus 6.14 Online Documentation, Dassault Systèmes Simulia Corporation, Providence, Rhode Island, USA.

[44] Davidon, W.C. (1959): Variable Metric Method for Minimization. Argonne National Laboratory Report 5990.

[45] De Lorenzis, L., Wriggers, P. and Hughes, T.J.R. (2014): Isogeometric Contact: A Review. GAMM-Mitteilungen, Volume 37, Issue 1, pp. 85-123.

[46] Deuflhard, P. (2012): A Short History of Newton's Method. Documenta Mathematica, Extra Volume: Optimization Stories, pp. 25-30.

[47] Edwin, L.-J. (2011): Numerical Model to Study of Contact Force in a Cylindrical Roller Bearing with Technical Mechanical Event Simulation. Journal of Mechanical Engineering and Automation, Volume 1, Issue 1, pp. 1-7.

[48] El-Abbasi, N. and Bathe, K.-J. (2001): Stability and Patch Test Performance of Contact Discretizations and a New Solution Algorithm. Computers & Structures, Volume 79, Issue 16, pp. 1473-1486.

[49] El-Saeidy, F.M.A. (1998): Finite Element Modeling of a Rotor Shaft Rolling Bearings System with Consideration of Bearing Nonlinearities. Journal of Vibration and Control, Volume 4, Issue 5, pp. 541-602.

[50] Flanagan, D.P. and Belytschko, T. (1981): A Uniform Strain Hexahedron and Quadrilateral with Orthogonal Hourglass Control. International Journal for Numerical Methods in Engineering, Volume 17, Issue 5, pp. 679-706.

[51] Fletcher, R. (1987): Practical Methods of Optimization. 2nd edition, John Wiley & Sons, Chichester/New York/Brisbane/Toronto/Singapore.

[52] Fletcher, R. (1970): A New Approach to Variable Metric Algorithms. The Computer Journal, Volume 13, Issue 3, pp. 317-322.

[53] Fletcher, R. and Powell, M. J. D. (1963): A Rapidly Convergent Descent Method for Minimization. The Computer Journal, Volume 6, Issue 2, pp. 163-168.

[54] Franklin, G.F., Powell, J.D. and Workman, M.L. (1990): Digital Control of Dynamic Systems. 2nd edition, Addison-Wesley, Boston, USA.

[55] Galvanetto, U. and Crisfield, M.A. (1996): An Energy-Conserving Co-Rotational Procedure for the Dynamics of Planar Beam Structures. International Journal for Numerical Methods in Engineering, Volume 39, Issue 13, pp. 2265-2282.

[56] Géradin, M. and Rixen, D.J. (2015): Mechanical Vibrations – Theory and Application to Structural Dynamics. 3rd edition, John Wiley & Sons, Chichester, UK.

[57] Girkmann, K. (1959): Flächentragwerke: Einführung in die Elastostatik der Scheiben, Platten, Schalen und Faltwerke. 5th edition, Springer, Wien.

[58] Goldfarb, D. (1970): A Family of Variable Metric Updates Derived by Variational Means. Mathematics of Computation, Volume 24, Issue 109, pp. 23-26.

[59] Hahn, W. (1967): Stability of Motion. Springer, Berlin/Heidelberg.

[60] Haufe, A., Schweizerhof, K. and DuBois, P. (2013): Properties and Limits: Review of Shell Element Formulations. Proceedings of LS-Dyna Developer Forum, September 24-25, 2013, Filderstadt, Germany.

[61] Hestenes, M.R. (1969): Multiplier and Gradient Methods. Journal of Optimization Theory and Applications, Volume 4, Issue 5, pp. 303-320.

[62] Heun, K. (1900): Neue Methoden zur approximativen Integration der Differentialgleichungen einer unabhängigen Veränderlichen. Zeitschrift für Mathematik und Physik, Volume 45, pp. 23-38.

[63] Hilber, H.M, Hughes, T.J.R. and Taylor, R.L. (1977): Improved Numerical Dissipation for Time Integration Algorithms in Structural Dynamics. Earthquake Engineering and Structural Dynamics, Volume 5, Issue 3, pp. 282-292.

[64] Hild, P. (2000): Numerical Implementation of Two Nonconforming Finite Element Methods for Unilateral Contact. Computer Methods in Applied Mechanics and Engineering, Volume 184, Issue 1, pp. 99-123.

[65] Holzapfel, G.A. (2000): Nonlinear Solid Mechanics: A Continuum Approach for Engineering. John Wiley & Sons, Chichester, UK.

[66] Hughes, T.J.R. (2000): The Finite Element Method: Linear Static and Dynamic Finite Element Analysis. Dover Publication, Mineola, New York.

[67] Hughes, T.J.R. (1977): A Note on the Stability of Newmark's Algorithm in Nonlinear Structural Dynamics. International Journal for Numerical Methods in Engineering, Volume 11, Issue 2, pp. 383-386.

[68] Hughes, T.J.R., Cottrell, J.A. and Bazilevs, Y. (2005): Isogeometric Analysis: CAD, Finite Elements, NURBS, Exact Geometry and Mesh Refinement. Computer Methods in Applied Mechanics and Engineering, Volume 194, Issues 39-41, pp. 4135-4195.

[69] Hughes, T.J.R. and Liu, W.K. (1981): Nonlinear Finite Element Analysis of Shells: Part I. Three-Dimensional Shells. Computer Methods in Applied Mechanics and Engineering, Volume 26, Issue 3, pp. 331-362.

[70] Hulbert, G. (2018): From (HHT)-Alpha to Omega? Presentation at WCCM XIII and PANACM II (World Congress on Computational Mechanics & Pan American Congress on Computational Mechanics), July 22-27, 2018, New York, USA.

[71] Hulbert, G.M. and Chung, J. (1996): Explicit Time Integration Algorithms for Structural Dynamics with Optimal Numerical Dissipation. Computer Methods in Applied Mechanics and Engineering, Volume 20, Issue 1, pp. 175-188.

[72] Irons, B.M. (1966): Numerical Integration Applied to Finite Element Methods, Proceedings of Conference on the Use of Digital Computers in Structural Engineering, July 1966, University of Newcastle, England.

[73] Karush, W. (1939): Minima of Functions of Several Variables with Inequalities as Side Constraints. M.Sc. Dissertation, Department of Mathematics, University of Chicago, Chicago, Illinois, USA.

[74] Karypis, G. and Kumar, V. (1998): A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs. SIAM Journal on Scientific Computing, Volume 20, Issue 1, pp. 359-392.

[75] Keskin, A., Kober, M., Stelldinger, E., Kühhorn, A., Böhm, H., Hornig, A., Langkamp, A. and Gude, M. (2015): On the Quantification of Errors of a Pre-Processing Effort Reducing Contact Meshing Approach. Proceedings of AAIA SciTech 2015, Kissimmee, Florida, USA, Paper 2014731.

[76] Kirchhoff, G.P. (1850): Über das Gleichgewicht und die Bewegung einer elastischen Scheibe. Journal für die reine und angewandte Mathematik, Volume 40, pp. 51-88.

[77] Kober, M. and Kühhorn, A. (2018): Stable Implicit Time-Integration of Flexible Rotating Structures - Explanation for Instabilities and Concepts for Avoidance. Applied Mathematical Modelling, Volume 60, pp. 235-243.

[78] Kober, M., Kühhorn, A., Stelldinger, E. and Keskin, A. (2014): Considerations about the Necessary Mesh Density of Bearings in Detailed Finite Element Models. Proceedings of ASME Turbo Expo, June 16-20, 2014, Düsseldorf, Germany, Paper GT2014-25624.

[79] Konyukhov, A. and Izi, R. (2015): Introduction to Computational Contact Mechanics: A Geometrical Approach. John Wiley & Sons, Chichester/New York.

[80] Konyukhov, A. and Schweizerhof, K. (2013): Computational Contact Mechanics - Geometrically Exact Theory for Arbitrary Shaped Bodies. Springer, Berlin/Heidelberg.

[81] Koschnick, F. (2004): Geometrische Locking-Effekte bei Finiten Elementen und ein allgemeines Konzept zu ihrer Vermeidung. Dissertation, Technical University München, Germany.

[82] Koschnick, F., Bischoff, M., Camprubi, N. and Bletzinger, K.-U. (2005): The Discrete Strain Gap Method and Membrane Locking. Computer Methods in Applied Mechanics and Engineering, Volume 194, Issue 21-24, pp. 2444-2463.

[83] Kosloff, D. and Frazier, G.A. (1978): Treatment of Hourglass Patterns in Low Order Finite Element Codes. International Journal for Numerical and Analytical Methods in Geomechanics, Volume 2, Issue 1, pp. 57-72.

[84] Krenk, S. (2006): Energy Conservation in Newmark based Time Integration Algorithms. Computer Methods in Applied Mechanics and Engineering, Volume 195, Issues 44-47, pp. 6110-6124.

[85] Krenk, S. (2001): Mechanics and Analysis of Beams, Columns and Cables: A Modern Introduction to the Classic Theories. 2nd edition, Springer, Berlin/Heidelberg.

[86] Krüger, A. (2014): Transient Dynamic Finite Element Modeling of Flexible Rotor Systems with Nonlinear Fluid Film Bearings and Faults. Dissertation, University of Pretoria, South Africa.

[87] Kuhl, D. and Crisfield, M.A. (1999): Energy-Conserving and Decaying Algorithms in Non-Linear Structural Dynamics. International Journal for Numerical Methods in Engineering, Volume 45, Issue 5, pp. 569-599.

[88] Kuhn, H.W. and Tucker, A.W. (1951): Nonlinear Programming. Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability, July 31 - August 12, 1950, Berkeley, California, pp. 481-492.

[89] Kulkarni, M., Belytschko, T. and Bayliss, A. (1995): Stability and Error Analysis for Time Integrators Applied to Strain-Softening Materials, Computer Methods in Applied Mechanics and Engineering, Volume 124, Issue 4, pp. 335-363.

[90] Kutta, W.M. (1901): Beitrag zur näherungsweisen Integration totaler Differentialgleichungen. Zeitschrift für Mathematik und Physik, Volume 46, pp. 435-453.

[91] Labuschagne, A., van Rensburg, N.F.J. and van der Merwe, A.J. (2009): Comparison of Linear Beam Theories. Mathematical and Computer Modelling, Volume 49, Issues 1-2, pp. 20-30.

[92] Lapidus, L. and Finder, G.F. (1982): Numerical Solution of Partial Differential Equations in Science and Engineering. John Wiley & Sons, Chichester/New York.

[93] Leech, J.W., Hsu, P.-T. and Mack, E.W. (1965): Stability of a Finite-Difference Method for Solving Matrix Equations. AIAA Journal, Volume 3, Issue 11, pp. 2172-2173.

[94] Levine, W.S. (1996): The Control Handbook. CRC Press, Boca Raton, Florida, USA.

[95] Li, Y., Wu, B. and Ou, J. (2006): Stability of Average Acceleration Method with Nonlinear Damping. Journal of Earthquake Engineering and Engineering Vibration, Volume 5, Issue 1, pp. 87-92.

[96] Liu, J.W.H. (1985): Modification of the Minimum Degree Algorithm by Multiple Elimination. ACM Transactions on Mathematical Software, Volume 11, Issue 2, pp. 141-153.

[97] Liu, W.K., Guo, Y., Tang, S. and Belytschko, T. (1998): A Multiple-Quadrature Eight-Node Hexahedral Finite Element for Large Deformation Elastoplastic Analysis. Computer Methods in Applied Mechanics and Engineering, Volume 154, Issues 1-2, pp. 69-132.

[98] Livermore Software Technology Corporation (2017): LS-Dyna Keyword User's Manual, Volume 1, LS-Dyna R10.0, Livermore, California, USA.

[99] Love, A.E.H. (1888): The Small Free Vibrations and Deformation of a Thin Elastic Shell. Philosophical Transactions of the Royal Society of London, Volume 179, pp. 491-546.

[100] Lyapunov, A.M. (1966): Stability of Motion. Academic Press, London, UK.

[101] Macek, R.W. and Aubert, B.H. (1995): A Mass Penalty Technique to Control the Critical Time Increment in Explicit Dynamic Finite Element Analyses. Earthquake Engineering and Structural Dynamics, Volume 24, Issue 10, pp. 1315-1331.

[102] MacNeal, R.H. (1994): Finite Elements: Their Design and Performance. Marcel Dekker, New York, USA.

[103] McDevitt, T. W. and Laursen, T. A. (2000): A Mortar-Finite Element Formulation for Frictional Contact Problems. International Journal for Numerical Methods in Engineering, Volume 48, Issue 10, pp. 1525-1547.

[104] Meyer, C.D. (2000): Matrix Analysis and Applied Linear Algebra. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, USA.

[105] Mindlin, R.D. (1951): Influence of Rotary Inertia and Shear on Flexural Motion of Isotropic Elastic Plates. Journal of Applied Mechanics, Volume 18, Issue 1, pp. 31-38.

[106] Naghdi, P.M. (1972): The Theory of Plates and Shells. In: Flügge, S., Handbuch der Physik, Volume 6/A2, Springer, Berlin/Heidelberg, pp. 425-640.

[107] Naveed, Z., Kühhorn, A. and Kober, M. (2019): Comparative Evaluation of Isogeometric Analysis and Classical FEM with Regard to Contact Analysis. Proceedings of 12th European LS-Dyna Conference, May 14-16, 2019, Koblenz, Germany.

[108] Negrut, D. (1998): On the Implicit Integration of Differential-Algebraic Equations of Multibody Dynamics. Ph.D. thesis, University of Iowa, USA.

[109] Nelder, J.A. and Mead, R.A. (1965): A Simplex Method for Function Minimization. Computer Journal, Volume 7, pp. 308-313.

[110] Newmark, N.M. (1959): A Method of Computation for Structural Dynamics. ASCE Journal of the Engineering Mechanics Division, Volume 85, Issue 3, pp. 67-94.

[111] Nocedal, J. and Wright, S.J. (1999): Numerical Optimization. Springer, New York.

[112] Norton, M. and Karczub, D. (2003): Fundamentals of Noise and Vibration Analysis for Engineers. 2nd edition, Cambridge University Press, Cambridge, UK.

[113] O'Brien, G.G., Hyman, M.A. and Kaplan, S. (1951): A Study of the Numerical Solution of Partial Differential Equations. Journal of Mathematics and Physics, Volume 29, Issue 1-4, pp. 223-251.

[114] Ogata, K. (1995): Discrete-Time Control Systems. 2nd edition. Prentice-Hall, Englewood Cliffs, New Jersey, USA.

[115] Olovsson, L., Simonsson, K. and Unosson, M. (2005): Selective Mass Scaling for Explicit Finite Element Analyses. International Journal for Numerical Methods in Engineering, Volume 63, Issue 10, pp. 1436-1445.

[116] Papadopoulos, P. and Taylor, R.L. (1993): A Simple Algorithm for Three-Dimensional Finite Element Analysis of Contact Problems. Computers & Structures, Volume 46, Issue 6, pp. 1107-1118.

[117] Papadopoulos, P. and Taylor, R.L. (1992): A Mixed Formulation for the Finite Element Solution of Contact Problems. Computer Methods in Applied Mechanics and Engineering, Volume 94, Issue 3, pp. 373-389.

[118] Pitkäranta, J. (1992): The Problem of Membrane Locking in Finite Element Analysis of Cylindrical Shells. Numerische Mathematik, Volume 61, Issue 1, pp. 523-542.

[119] Powell, M.J.D. (1969): A Method for Nonlinear Constraints in Minimization Problems. In: Fletcher, R., Optimization, Academic Press, New York, pp. 283-298.

[120] Puso, M.A. and Laursen, T.A. (2004): A Mortar Segment-to-Segment Contact Method for Large Deformation Solid Mechanics. Computer Methods in Applied Mechanics and Engineering, Volume 193, Issues 6-8, pp. 601-629.

[121] Raghavendra, N., Math, M.C. and Sharma, P.R. (2015): Finite Element Method Analysis of Hydrodynamic Journal Bearing. European Journal of Advances in Engineering and Technology, Volume 2, Issue 2, pp. 92-101.

[122] Ramm, E., Bischoff, M. and Braun, M. (1994): Higher Order Nonlinear Shell Formulation – A Step Back into Three Dimensions. In: Bell, K., From Finite Elements to the Troll Platform, Ivar Holland 70th Anniversary, Norwegian Institute of Technology, Trondheim, pp. 65-88.

[123] Reissner, E. (1945): The Effect of Transverse Shear Deformation on the Bending of Elastic Plates. ASME Journal of Applied Mechanics, Volume 12, Issue 2, pp. A69-A77.

[124] Runge, C. (1895): Ueber die numerische Auflösung von Differentialgleichungen. Mathematische Annalen, Volume 46, pp. 167-178.

[125] Shanno, D.F. (1970): Conditioning of Quasi-Newton Methods for Function Minimization. Mathematics of Computation, Volume 24, Issue 111, pp. 647-656.

[126] Shim, K.W., Monaghan, D.J. and Armstrong, C.G. (2002): Mixed Dimensional Coupling in Finite Element Stress Analysis. Engineering with Computers, Volume 18, pp. 241-252.

[127] Silva, W.T.M. and Bezerra, LM. (2008): Performance of Composite Implicit Time Integration Scheme for Nonlinear Dynamic Analysis. Mathematical Problems in Engineering. Volume 2008, Article ID 815029, pp. 1-16.

[128] Simo, J.C. and Rifai, S. (1990): A Class of Mixed Assumed Strain Methods and the Method of Incompatible Modes. International Journal for Numerical Methods in Engineering, Volume 29, Issue 8, pp. 1995-1683.

[129] Simo, J.C. and Tarnow, N. (1992): The Discrete Energy-Momentum Method. Conserving Algorithms for Nonlinear Elastodynamics. Zeitschrift für angewandte Mathematik und Physik, Volume 43, Issue 5, pp. 757-792.

[130] Simo, J.C., Tarnow, N. and Doblare, M. (1995): Non-Linear Dynamics of Three-Dimensional Rods: Exact Energy and Momentum Conserving Algorithms. International Journal for Numerical Methods in Engineering, Volume 38, Issue 9, pp. 1431-1473.

[131] Simo, J.C., Wriggers, P. and Taylor, R.L. (1985): A Perturbed Lagrangian Formulation for the Finite Element Solution of Contact Problems. Computer Methods in Applied Mechanics and Engineering, Volume 50, Issue 2, pp. 163-180.

[132] Starvin, M.S., Babu, S., Aithal, S., Manisekar, K. and Chellapandi, P. (2013): Finite Element Simulation of Non-Linear Deformation Behaviour in Large Diameter Angular Contact Thrust Bearing. Scientific Research and Essays, Volume 8, Issue 3, pp. 128-138.

[133] Stolarski, H. and Belytschko, T. (1982): Membrane Locking and Reduced Integration for Curved Elements. Journal of Applied Mechanics, Volume 49, Issue 1, pp. 172-176.

[134] Strang, G. and Fix, G.J. (1973): An Analysis of the Finite Element Method. Prentice-Hall, Englewood Cliffs, New Jersey, USA.

[135] Taylor, R.L., Beresford, P.J. and Wilson, E.L. (1976): A Non-Conforming Element for Stress Analysis. International Journal for Numerical Methods in Engineering, Volume 10, Issue 6, pp. 1211-1219.

[136] Taylor, R.L. and Papadopoulos, P. (1991): On a Patch Test for Contact Problems in two Dimensions. In: Wriggers, P., Wagner, W., Computational Methods in Nonlinear Mechanics, Springer, Berlin, pp. 690-702.

[137] Timoshenko, S.P. (1921): On the Correction for Shear of the Differential Equation for Transverse Vibrations of Prismatic Bars. Philosophical Magazine, Volume 41, Issue 245, pp. 744-746.

[138] Timoshenko, S.P. and Goodier, J.N. (1970): Theory of Elasticity. 3rd edition, McGraw-Hill, New York.

[139] Tinney, W.F. and Walker, J.W. (1967): Direct Solution of Sparse Network Equations by Optimally Ordered Triangular Factorization. Proceedings of the IEEE, Volume 55, Issue 11, pp. 1801-1809.

[140] Tkachuk, A. and Bischoff, M. (2013): Variational Methods for Selective Mass Scaling. Computational Mechanics, Volume 52, Issue 3, pp. 563-570.

[141] Tong, P., Pian, T.H.H. and Bucciarelli, L.L. (1971): Mode Shapes and Frequencies by the Finite Element Method Using Consistent and Lumped Masses. Computers and Structures, Volume 1, Issue 4, pp. 623-638.

[142] Trefethen, L. N. (1996): Finite Difference and Spectral Methods for Ordinary and Partial Differential Equations. Unpublished text, available at http://people.maths.ox.ac.uk/trefethen/pdetext.html

[143] Unnikrishnan, M., Janve, P.S. and Krishna, P. (2017): Development of a Finite Element Model for Spherical Roller Bearings Under Static Load. International Journal of Engineering Innovation & Research, Volume 6, Issue 3, pp. 152-157.

[144] von Neumann, J. (1961): Collected Works, Volume V – Design of Computers, Theory of Automata and Numerical Analysis, Edited by A.H. Taub. Pergamon Press, Oxford/New York/Toronto/Sydney/Paris/Frankfurt.

[145] Wilkinson, J.H. (1725): The Algebraic Eigenvalue Problem. Clarendon Press, Oxford, UK.

[146] Wilson, E.L., Taylor, R.L., Doherty, W.P. and Ghaboussi, J. (1973): Incompatible Displacement Models. In: Fenves, S.J., Numerical and Computer Models in Structural Mechanics, Academic Press, London/Oxford/Boston/New York/ San Diego, pp. 43-57.

[147] Wohlmuth, B.I. (2000): A Mortar Finite Element Method Using Dual Spaces for the Lagrange Multiplier, SIAM Journal on Numerical Analysis, Volume 38, Issue 3, pp. 989-1012.

[148] Wriggers, P. (2006): Computational Contact Mechanics. 2nd edition, Springer, Berlin/Heidelberg/New York.

[149] Xie, Y.M. and Steven, G.P. (1994): Instability, Chaos, and Growth and Decay of Energy of Time-Stepping Schemes for Non-Linear Dynamic Equations. Communications in Numerical Methods in Engineering, Volume 10, Issue 5, pp. 393-401.

[150] Yang, B., Laursen, T.A. and Meng, X. (2005): Two Dimensional Mortar Contact Methods for Large Deformation Frictional Sliding. International Journal for Numerical Methods in Engineering, Volume 62, Issue 9, pp. 1183-1225.

[151] Zavarise, G. and de Lorenzis, L. (2009): The Node-to-Segment Algorithm for 2D Frictionless Contact: Classical Formulation and Special Cases. Computer Methods in Applied Mechanics and Engineering, Volume 198, Issues 41-44, pp. 3428-3451.

[152] Zavarise, G. and de Lorenzis, L. (2009): A Modified Node-to-Segment Algorithm Passing the Contact Patch Test. International Journal for Numerical Methods in Applied Mechanics and Engineering. Volume 79, Issue 4, pp. 379-416.

[153] Zavarise, G. and Wriggers, P. (1998): A Segment-to-Segment Contact Strategy, Mathematical and Computer Modelling, Volume 28, Issues 4-8, pp. 497-515.

[154] Zlamal, M. (1978): Superconvergence and Reduced Integration in the Finite Element Method. Mathematics of Computation, Volume 32, Issue 143, pp. 683-685.