Adaptive Isogeometric Impact Models in Flexible Multibody Systems

Vom Promotionsausschuss der
Technischen Universität Hamburg

zur Erlangung des akademischen Grades

Doktor-Ingenieur (Dr.-Ing.)

genehmigte Dissertation (Monografie)

von
Tobias Christian Rückwald

aus
Hamburg

2025

1. Gutachter: Prof. Dr.-Ing. Robert Seifried
2. Gutachter: Prof. Dr.-Ing. Alexander Düster

Tag der mündlichen Prüfung: 27. 06. 2025

MuM Notes in Mechanics and Dynamics

Editor: Prof. Dr.-Ing. Robert Seifried
Hamburg University of Technology
Institute of Mechanics and Ocean Engineering (MuM)
www.tuhh.de/mum

Volume 12
Tobias Christian Rückwald
“Adaptive Isogeometric Impact Models in Flexible Multibody Systems”
Hamburg, 2025

DOI: https://doi.org/10.15480/882.15347
ORCID: https://orcid.org/0000-0001-6317-7688

Acknowledgments

This work is the result of research conducted at the Institute of Mechanics and Ocean Engineering at Hamburg University of Technology. Many people have contributed to this work in one way or another.

First of all, I would like to thank Prof. Dr.-Ing. Robert Seifried for introducing me to this fascinating research topic and giving me the opportunity to work on it. His continuous support and guidance are the basis of this work. Moreover, I would like to thank Prof. Dr.-Ing. Alexander Düster for his interest in this work and for acting as second reviewer. I am also grateful to Prof. Dr.-Ing. Bastian Oesterle for chairing the examination committee.

I would like to thank Dr.-Ing. Alexander Held, who accompanied my work as a mentor and contributed many valuable ideas throughout the project - including the initial idea to develop an adaptive algorithm. His input has been instrumental in shaping this dissertation, and I am truly grateful for his support, both professionally and personally.

The research presented in this dissertation was made possible thanks to the supportive and collegial atmosphere at the Institute of Mechanics and Ocean Engineering. I am truly grateful to all my colleagues for the mutual support, the many coffee breaks, and the open discussions—both academic and beyond. I will especially remember the enjoyable barbecue events, which brought us together and made this time even more special. I learned a great deal during this memorable chapter, and I thank you all sincerely.

Furthermore, I am deeply thankful for the continuous support of my family, especially my parents. They have always encouraged me to pursue my own path, while standing by my side every step of the way. I truly would not be where I am today without them, and I am profoundly grateful for all they have done for me.

Most of all, I would like to thank my wife Behnaz. Her endless patience, understanding, and unwavering support have meant more to me than words can express. She has been by my side through every challenge, always believing in me and offering strength when I needed it most. Her love has been a constant source of motivation and comfort throughout this journey.

Hamburg, July 2025 Tobias Christian Rückwald

To my family.

Abstract

Detailed impact simulations in flexible multibody systems are usually based on isoparametric finite element models. However, isoparametric finite elements involve the discretization of the geometry. This work uses the isogeometric analysis (IGA) as an alternative approach in flexible multibody systems. The IGA enables the exact representation of the geometry. In the context of efficient impact simulation, model reduction and inclusion of the floating frame of reference formulation is beneficial. A precise impact simulation requires a high element resolution in the contact area. The straightforward approach for refinement in IGA is global refinement. However, global refinement not only adds elements in the contact area but also across the entire body. This work applies hierarchical refinement to IGA impact models to refine locally. The hierarchically refined IGA model is reduced with the Craig-Bampton method which includes local deformations in the contact area. However, the resulting equations of motion are numerically stiff and therefore computationally expensive to solve. A quasistatic contact model is used to reduce the numerical stiffness.

Usually, detailed impact simulation models have to be set up manually rather than being generated automatically. This is because the process requires prior knowledge of the time and location of the impact, as well as the element resolution within the contact area. If the penalty method is used to determine the occurring contact forces, the corresponding penalty factor also needs to be determined manually. This work, however, presents an adaptive procedure to simulate impacts within flexible multibody systems fully automatically. The adaptive algorithm detects impacts in the system, determines the contact locations on the bodies, refines the contact area, and determines the penalty factor, and therefore automatically simulates impacts. The work shows how to automatically simulate impacts in flexible multibody systems without user action or prior knowledge of impact location and size. This is demonstrated by application examples.

The first application example includes significant elastodynamic effects within a long flexible rod. The goal is to validate the algorithm by preserving the wave propagation and energy of the system. The second application example simulates the impacts of two flexible double pendulums. This setup is a suitable benchmark for the complete adaptive impact analysis procedure as the flexible double pendulums undergo large rigid body motions.

Contents

List of Figures

List of Tables

List of Algorithms

Abbreviations and Symbols

In the following, the most important abbreviations and symbols are defined.

Abbreviations

CAD	computer aided desgin
FEM	finite element method
IGA	isogeometric analysis
SID	standard input data
RIGA	Reduced Isogeometric Analysis
RISO	reduced isoparametric
B-splines	Basis splines
NURBS	non-uniform rational basis splines
DLL	Dynamic Link Library
MEX	MATLAB executable
RCN	reciprocal condition number

Indices

I	inertial frame
R	reference frame
C	contact body
T	target body
qs	quasistatic
lf	low frequency
hf	high frequency
con	constant
int	integrate
min	minimum
max	maximum
new	new
pre	previous
ele	element
lvl	level
intvl	interval
jac	Jacobian

Floating Frame of Reference

t	time
t_0	initial time
K	coordinate frame
P	point P
$\boldsymbol{z}_{\mathrm{I}}$	position coordinates
$\boldsymbol{z}_{\mathrm{II}}$	velocity coordinates
$\boldsymbol{r}$	position vector
$\boldsymbol{v}$	translational velocity
$\boldsymbol{\beta}$	rotational parameters
$\boldsymbol{S}$	rotation matrix
$\boldsymbol{\omega}$	angular velocity
$\boldsymbol{\Phi}$	global shape functions
$\boldsymbol{u}_{\mathrm{e}}$	elastic deformation
$\boldsymbol{q}_{\mathrm{e}}$	elastic coordinates
	number of
n_{q}	elastic coordinates
$n_{\mathrm{q}}^{\mathrm{lf}}$	low frequency modes
$n_{\mathrm{q}}^{\mathrm{hf}}$	high frequency modes
$\boldsymbol{M}$	mass matrix
m	mass of the body
$\boldsymbol{I}$	mass moment of inertia
$\boldsymbol{C}_{\mathrm{t}}$	translation coupling matrix
$\boldsymbol{C}_{\mathrm{r}}$	rotational coupling matrix
$\boldsymbol{c}$	centroid
	flexible body
$\overline{\boldsymbol{M}}_{\mathrm{e}}$	mass matrix
$\overline{\boldsymbol{K}}_{\mathrm{e}}$	stiffness matrix
$\overline{\boldsymbol{D}}_{\mathrm{e}}$	damping matrix
$\boldsymbol{h}_{\mathrm{d}}$	discrete forces
$\boldsymbol{h}_{\mathrm{b}}$	body forces
$\boldsymbol{h}_{\omega}$	generalized inertial forces
$\boldsymbol{h}_{\mathrm{e}}$	internal forces
E	energy
h	potential height
h_0	zero level of potential height

IGA

ξ, η, ζ	local coordinates
	knot vectors in
$\boldsymbol{\Xi}$	ξ-direction
$\boldsymbol{\mathcal{H}}$	η-direction
$\boldsymbol{\mathcal{Z}}$	ζ-direction
	B-splines in
N	ξ-direction
M	η-direction
L	ζ-direction
	number of B-splines in
n	ξ-direction
m	η-direction
ℓ	ζ-direction
	order of B-splines in
p	ξ-direction
q	η-direction
r	ζ-direction
	knot multiplicity in
k_ξ	ξ-direction
k_η	η-direction
k_ζ	ζ-direction
$\boldsymbol{P}$	control points
$\boldsymbol{p}$	control point
$\boldsymbol{w}$	weights
w	weight
n_{cp}	number of control points
R	NURBS basis
$\boldsymbol{S}$	NURBS solid

IGA Refinement

n_{p}	order elevation
n_{lvl}	number of levels
a	linear coefficient
$\boldsymbol{A}$	coefficient matrix

IGA Equations of Motion

$\boldsymbol{d}$	displacement field
$\boldsymbol{N}$	local shape functions
$\boldsymbol{B}$	strain displacement matrix
$\boldsymbol{C}$	strain displacement matrix
$\boldsymbol{J}$	Jacobian
$\Omega_{\mathrm{e},i}$	i-th element
$\boldsymbol{M}_{\mathrm{e},i}$	element mass matrix
$\boldsymbol{K}_{\mathrm{e},i}$	element stiffness matrix
$\boldsymbol{M}_{\mathrm{e}}$	global mass matrix
$\boldsymbol{K}_{\mathrm{e}}$	global stiffness matrix
$\boldsymbol{f}_{\mathrm{e}}$	external forces
n_{dof}	number of degrees of freedom
$\boldsymbol{\sigma}$	stresses
σ	normal stresses
τ	shear stresses
σ_{mises}	von Mises stress
$\boldsymbol{\Phi}_\sigma$	stress modes

IGA Contact

$n_{\text{points}}^{\text{eval}}$	number of contact evaluation points
g_{n}	normal gap
$\boldsymbol{n}$	normal vector
c_{p}	penalty factor
$c_{\text{p}}^{\text{mul}}$	multiplier part of c_{p}
$c_{\text{p}}^{\text{exp}}$	exponential part of c_{p}
f	contact force
$\boldsymbol{f}$	contact force
$\boldsymbol{f}_{\text{con}}$	contact equation
$\boldsymbol{J}_{\text{con}}$	Jacobian of $\boldsymbol{f}_{\text{con}}$
$\boldsymbol{f}_{\text{qs}}$	quasistatic contact equation
$\boldsymbol{J}_{\text{qs}}$	Jacobian of $\boldsymbol{f}_{\text{qs}}$
k_{jac}	Jacobian counter
$k_{\text{jac}}^{\text{max}}$	maximum Jacobian counter
k_{newton}	Newton counter
$k_{\text{newton}}^{\text{max}}$	maximum Newton counter
ϵ	tolerance for Newton's method
$\epsilon_{\text{rcn}}^{\text{min}}$	minimum reciprocal condition number

IGA Collocation

$\hat{\xi}$	collocation points in ξ-direction
$\hat{\eta}$	collocation points in η-direction
$\hat{\zeta}$	collocation points in ζ-direction
$\hat{\omega}$	collocation weight
$\hat{\boldsymbol{\omega}}$	collocation weights
$\boldsymbol{G}$	NURBS basis arranged in matrix
$\boldsymbol{r}$	integrated NURBS basis

Hertz Contact

f_{Hertz}	contact force
σ_{Hertz}	von Mises stress
δ	displacement
r^*	equivalent radius
E^*	equivalent Young's modulus
b	contact width
p_0	maximum pressure

Adaptive Procedure

f_d	distance function
n_con	number of contact steps
Δt^max	maximum time step size
f^max	maximum eigenfrequency
$n^\mathrm{ele}_\mathrm{intvl}$	number of elements in contact interval
$n^\mathrm{ele}_\mathrm{contact}$	number of elements which are in contact
w_intvl	width of contact interval
$\mathcal{U}_\mathrm{intvl}$	utilization of ω
$\mathcal{U}^\mathrm{target}_\mathrm{intvl}$	target value of $\mathcal{U}_\mathrm{intvl}$
$\mathcal{U}^\mathrm{min}_\mathrm{intvl}$	minimum value of $\mathcal{U}_\mathrm{intvl}$
$n^\mathrm{max}_\mathrm{lvl}$	maximum number of levels
n_knots	number of inserted knots
$n^\mathrm{max}_\mathrm{knots}$	maximum number of inserted knots
n^ele	number of inserted elements
$\boldsymbol{n}^\mathrm{ele}_\mathrm{lvl}$	n^ele distributed over hierarchy levels
$\boldsymbol{U}^\mathrm{coarse}_\mathrm{mat}$	elastic deformation of the control points of an element in matrix notation
$\boldsymbol{n}^\mathrm{coarse}$	NURBS basis arranged in vector
$\boldsymbol{N}^\mathrm{coarse}_\mathrm{mat}$	NURBS basis arranged in matrix

Application Examples

r	radius
ℓ	length
v_0	initial velocity
c_rod	wave speed

Overall

x, y, z	cartesian coordinate system
E	Young's modulus
ρ	density
ν	Poisson's ratio
$\boldsymbol{E}$	identity matrix
g	gravitational constant
π	pi

Operators and Other Symbols

$()^\intercal$	transpose
$\lvert() \rvert$	absolute value
$\lvert\lvert() \rvert\rvert_2$	euclidean norm
$\times$	cross product

Introduction

Impacts occur in many machines and engineering applications. Typical examples are hammer drills, valves, or pick-and-place processes. When simulating impacts, there are several modeling approaches. The simplest approach is to focus mainly on global behavior. In this way, the rigid body motion can be obtained. To simulate global behavior, it is distinguished between discontinuous and continuous methods. For a discontinuous method, the impact is infinitesimally short. The velocities after impact can be calculated using the coefficient of restitution and the balance of linear and angular momentum. For a continuous method, nonlinear force laws can be chosen, such as the Hertzian contact law [Hertz82, Johnson04]. These force laws provide not only the velocities after impact, but also the contact force over time. The Hertzian contact law does not consider damping. If damping is to be considered, extended Hertzian contact laws can be used, e.g. [LankaraniNikravesh90, GonthierEtAl04]. These contact laws are heuristically tuned such that the global impact response is adequate. In many cases, the description of the rigid body motion is sufficient. However, the distribution of the impact force is often not correct. Therefore, no correct information about the local deformations and stresses can be obtained. This shows that a correct evaluation of the impact force is necessary to obtain a correct excitation of the vibration of the bodies. To extend the impact analysis on global and local behavior, the finite element method (FEM), see e.g. [Bathe14], can be used. The impact can be simulated in detail, including stresses and displacements. However, a full FEM simulation using finite element codes such as Ansys or Abaqus can be very time consuming. The time scale of impacts is very short, mostly fractions of milliseconds. However, the time scale of rigid body motions before and after impact is in the magnitude of seconds. Here, a full FEM simulation is inefficient and takes much time, see e.g. [Seifried05, Tschigg20]. In contrast to finite element codes, the ”free flying” phases between impacts can be simulated faster using flexible multibody systems [Seifried05, Tschigg20]. Research in impact analysis with flexible multibody systems is motivated by the need to maintain the efficiency of flexible multibody simulation between impacts and to obtain correct local effects, e.g. deformations and stresses, and global behavior, e.g. rigid body motion and elastodynamic effects, such as wave propagation in the bodies.

1.1 Background and Motivation

An efficient approach to simulate flexible multibody systems is the floating frame of reference formulation [SchwertassekWallrapp14, Shabana05]. This approach is favored, since elastic deformations of bodies typically remain small if they consist of stiff material such as steel or aluminum. The large rigid body motion is represented by the motion of the floating frame, while the elastic deformation is described within this floating frame. In particular, large rigid body rotations can be efficiently simulated with the floating frame of reference formulation. The floating frame of reference formulation requires global shape functions which describe body flexibility. One possibility to obtain the global shape functions is from a linear finite element model and a subsequent linear model reduction, such as modal truncation and the Craig-Bampton method [CraigBampton68].

Previous Work

In the previous work [Tschigg20], isoparametric elements are chosen to discretize the geometry. The simplest way to determine a set of global shape functions is modal truncation. However, modal truncation does not capture precise local deformations in the contact area. Therefore, a modally truncated model is not appropriate for an accurate flexible multibody impact simulation and leads to inaccurate results, as shown in [Tschigg20]. Alternatively, the Craig-Bampton method [CraigBampton68] can be used, wherein, in addition to normal modes, constraint modes are used. Thus, the model includes low and high frequency modes. The low frequency normal modes in the range up to 100 kHz approximate the overall elastodynamic behavior. The high frequency constraint modes in the range of several hundred kHz and MHz give an accurate approximation of local deformations in the contact area. However, a system of equations consisting of low and high frequency modes is numerically stiff. Solving these equations requires small time step sizes in the numerical time integration. As an example, suppose an impact of two flexible bodies whose modes are not yet excited. The impact then excites the numerical stiff frequency band and small step sizes are required. This is computationally expensive if subsequent large rigid body motions and impacts are simulated. One solution is modally damping the high frequency modes. It increases the computational performance but computation times remain high [Tschigg20]. Alternatively, a quasistatic contact model is used in [Tschigg20]. The idea of a quasistatic contact model is to neglect the dynamics of the high frequency modes. Thereby, the numerical efficiency is increased, and the numerical stiffness is reduced. By that, it is shown in [Wielenga84, SherifEtAl12, TschiggSeifried18, Tschigg20] that the high frequency modes only have a small influence on the dynamics. The quasistatic contact model is then coupled with a penalty method for contact treatment.

IGA in Dynamics

The discretization of the geometry via isoparametric elements may be the straightforward approach. However, a detailed impact simulation depends on an accurate representation of the geometry in the contact area. This disadvantage does not hold for the isogeometric analysis (IGA) [CottrellEtAl09]. The geometry is exactly preserved with the IGA. This is because a meshed IGA model is based on its initial geometry definition [CottrellEtAl09]. In FEM, the geometry is modeled with computer aided design (CAD) software and then discretized with elements. The motivation of the IGA is to merge these two steps by using splines as local shape functions as in CAD software.

Another advantage of the IGA is that high modes are represented more accurately than with a finite element model [CottrellEtAl09]. As an example, the modes of a fixed-fixed rod represented by a 1D model can be calculated via a finite element model and an IGA model. Both models use quadratic local shape functions. If linear shape functions are chosen, both models are identical. The relative error with respect to the mode number divided by the total number of modes is shown in Fig. 1.1. The analytical solution of the eigenfrequencies of a rod can be found in [Graff91]. The finite element solution uses quadratic rod elements. Each rod element consists of two nodes at each end and one node in the middle. This means that there are two types of local shape functions. This distinction results in a branching between the lower and upper half of modes in Fig. 1.1. In the lower half, the neighboring end-nodes and mid-nodes oscillate in phase. In the upper half, the neighboring nodes oscillate out of phase. For more details on this behavior, see [CottrellEtAl09]. In the IGA, there is no distinction between degrees of freedom at the end or in the middle, and the local shape functions are identical. Therefore, no branching occurs and the accuracy of the IGA especially of the high frequency modes is higher compared to FEM. This is essential for the computation of the global shape functions, since the high frequency modes

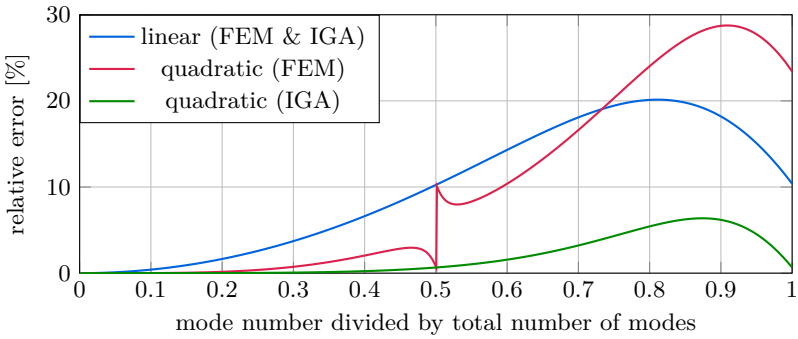


Figure 1.1: Relative mode error of a fixed-fixed rod.

represent the local deformation in the contact area.

The most attractive advantage of the IGA for industrial applications is the fusion of CAD and FEM. However, despite the advantages of the IGA, it is not yet widely used in the industry. This is probably because it takes a lot of effort to change existing processes, codes, and models. There are many IGA toolboxes in research, each focused on a different application area. The only commercial software that includes the IGA is LS-DYNA. For example, this software is used to perform crash simulations with shell structures in [Leidinger22]. The use of shell structures is possible because spatial geometries are already created as surfaces in CAD software. The transformation of a spatial CAD geometry to a solid model is more challenging. This feature is under active development in LS-DYNA, see [YuEtAl22].

Manual Setup of Impact Simulations

Setting up and solving impact simulations in FEM, IGA, or flexible multibody systems requires manual setup and a lot of experience by the user. In the following, three main challenges are listed below:

First, when using the penalty method, a suited penalty factor is required. It represents virtual springs between two bodies in contact preventing penetration. In [Nour-OmidWriggers87], the penalty factor is determined optimally for the static case. However, for transient impact simulations, the penalty factor needs to be determined heuristically, see e.g. [SeifriedEtAl03]. If the penalty factor is too small, too much penetration of the bodies occurs, and thus the deformation in the contact area and the contact force are not resolved accurately. However, a too high penalty factor yields numerical difficulties.

Second, the contact model and its refinement in the contact area are usually determined heuristically. In the contact area, there must be a sufficiently high element resolution to represent the contact accurately. Refinement requires knowledge of the location and the element resolution of the contact area. This information is needed before the simulation during setup. Therefore, the effort to determine the contact location in advance increases if large rigid body translations and especially rotations occur before the impact.

Third, the time of the impact is relevant if large rigid body motions occur. Impact phases are highly dynamic and require significantly smaller step sizes compared to non-impact phases without acting contact forces. Therefore, the choice of the integrator and its step size have a great effect on the numerical performance.

1.2 Objective and Structure of the Work

The **main** objective of this work is to develop an adaptive procedure that overcomes the challenges of manual setup. The time of the impact, the location of the impact on the body, the appropriate element resolution in the contact area, and the penalty factor are automatically determined. As a result, multiple impacts in flexible multibody systems including large rigid body motions can be efficiently, accurately, and automatically simulated with the adaptive procedure developed in this thesis. The final procedure has also been published in [RückwaldEtAl24]. However, in order to achieve the main objective, three milestones are completed first, which are also published in [RückwaldEtAl21b, RückwaldEtAl21c, RückwaldEtAl21d, RückwaldEtAl22, RückwaldEtAl23].

First Milestone: The IGA in Flexible Multibody Systems

The previous work [Tschigg20] uses the flexible multibody simulation framework of the Institute of Mechanics and Ocean Engineering (MuM) at Hamburg University of Technology. First, the geometry is modeled in CAD software such as Inventor. Second, the geometry is meshed with finite elements, e.g. using Abaqus. Third, the in-house Matlab toolbox RED reads the FEM data, performs model reduction, and stores the results in a standard input data (SID) file, see [SchwertassekWallrapp14]. Fourth, the Matlab toolbox DynManto performs the simulation using the floating frame of reference formulation, see [HeldEtAl20]. The first milestone in this thesis is the development of a Matlab based IGA toolbox that allows to generate simple geometries, refine them and compute their equations of motion. Existing software cannot be used because the main objective, the adaptive impact procedure, requires deep access to the IGA software. In addition, the toolbox is required to work with the existing flexible multibody framework with only minor adjustments. This is originally achieved in [RückwaldEtAl21b], where the impact of a flexible body with a rigid surface is simulated. The contact routine is then extended in [RückwaldEtAl21d] to simulate the impact of two flexible bodies.

Second Milestone: Hierarchical Refinement

An accurate impact simulation, which captures the local deformation, requires a high element resolution in the contact area, as this is where the largest elastic deformations and stresses occur. However, the refinement methods usually used in IGA only allow global refinement. The refinement of a semicircle representing an axisymmetric sphere is exemplified in Fig. 1.2. The model on the left in Fig. 1.2 is globally refined and the model on the right uses hierarchical refinement. With global refinement, additional elements are created over the entire body. Since more elements result in more degrees of freedom, global refinement greatly increases the number of equations of motion. One method for local refinement is

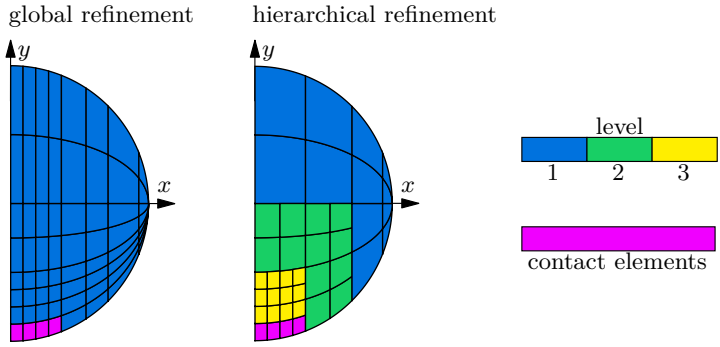


Figure 1.2: Refinement in the contact area with global and hierarchical refinement.

a hierarchical approach, where subordinate levels are introduced, as displayed in Fig. 1.2. The hierarchical refinement is widely used in literature. It is applied in elementary fluid and structural analysis [SchillingerEtAl12], heat conduction problems [D'AngellaEtAl18], topology optimization [NoëlEtAl20], and contact simulations [ZimmermannSauer17]. The aforementioned literature summarizes that the computational effort can be reduced due to the smaller number of degrees of freedom compared to global refinement. These examples are mostly from statics. In this work, hierarchical refinement is applied in a dynamic problem. Therefore, the second milestone is to include hierarchically refined models within the floating frame of reference formulation. This is presented in [RückwaldEtAl21c] for 2D models and extended in [RückwaldEtAl22] to 3D models. It is also evaluated whether the computational complexity of hierarchically refined models is reduced compared to globally refined models.

Third Milestone: Quasistatic Contact Model

The quasistatic contact was previously used in impact simulations with isoparametric elements in [TschiggSeifried18, Tschigg20]. As mentioned before, the dynamics of the high frequency modes are neglected to increase computational performance. The third milestone is to apply the quasistatic contact model to IGA contact models. This is initially presented in [RückwaldEtAl23]. As in [Tschigg20], the quasistatic contact model is paired with a penalty method to compute the contact forces. Beside the penalty method also the Lagrange multiplier method exists [Wriggers06]. It was also used in the IGA [Matzen15, Matzen-Bischoff16, SeitzEtAl16]. However, to continue the previous work [Tschigg20] on the quasistatic contact model, the penalty method is used.

Structure of the Work

This work is organized in the following way: The concepts of the floating frame of reference formulation, the IGA and the determination of the global shape functions are introduced in Chap. 2. This chapter includes the first and second milestone. Contact handling including the quasistatic contact model is detailed in Chap. 3. This corresponds to the third milestone. Details on the implementation of the developed methods are found in Chap. 4, which is the basis for the following automated contact simulations. The following Chap. 5 investigates the IGA in impact simulations and factors affecting accuracy and computation time are evaluated. Based on the findings in Chap. 5, the adaptive procedure for an automatic impact simulation is described in Chap. 6 and is the main objective of this thesis. The following Chap. 7 provides detailed discussions of two adaptive application examples. A wave propagation setup is used for validation by monitoring energy conservation. The setup of two flexible double pendulums involves multiple impacts and large rigid body motions. Therefore, this setup can be efficiently solved with the floating frame of reference formulation. Finally, the results are summarized in Chap. 8.

IGA in the Floating Frame of Reference Formulation

A well-established approach when modeling flexible multibody system is the floating frame of reference formulation [SchwertassekWallrapp14]. A key issue in using the floating frame of reference formulation is determining the global shape functions $\boldsymbol{\Phi}$. The straightforward approach is to generate a finite element model of the flexible body and apply a model reduction technique [FehrEberhard10]. In contrast, this work uses an alternative approach IGA models which are reduced by model reduction techniques. The chapter begins with a short introduction to the floating frame of reference formulation in Sect. 2.1. The basics of the IGA are briefly presented in Sect. 2.2 and Sect. 2.3. Global and hierarchical refinement are introduced in Sect. 2.4 and Sect. 2.5. The determination of the equations of motion in the IGA is described in Sect. 2.6. A more detailed introduction to the IGA can be found in [CottrellEtAl09]. Then, the equations of motion are reduced in Sect. 2.7 to determine the global shape functions. Finally in Sect. 2.8, a method for stress recovery is applied to reduced IGA models.

2.1 Floating Frame of Reference Formulation

The floating frame of reference formulation [SchwertassekWallrapp14] describes large nonlinear rigid body motions of the body reference frame $\mathrm{K_R}$ within the inertial frame $\mathrm{K_I}$. As an example, Fig. 2.1 represents the position of an arbitrary point P on a flexible body for two different points in time t. At initial time t_0, the body is undeformed. The point P on the flexible body has the position ${}^{\mathrm{R}}\boldsymbol{r}_{\mathrm{RP}_0}$ with respect to the body reference frame. The position of the body reference frame is provided by ${}^{\mathrm{R}}\boldsymbol{r}_{\mathrm{IR}}$. The indices R and I represent the reference frame $\mathrm{K_R}$ and the inertial frame $\mathrm{K_I}$. The index at the top left indicates the coordinate frame in which the variable is represented. As an example, the position ${}^{\mathrm{R}}\boldsymbol{r}_{\mathrm{IR}}$ is represented in the reference frame $\mathrm{K_R}$. This vector can be transformed to the

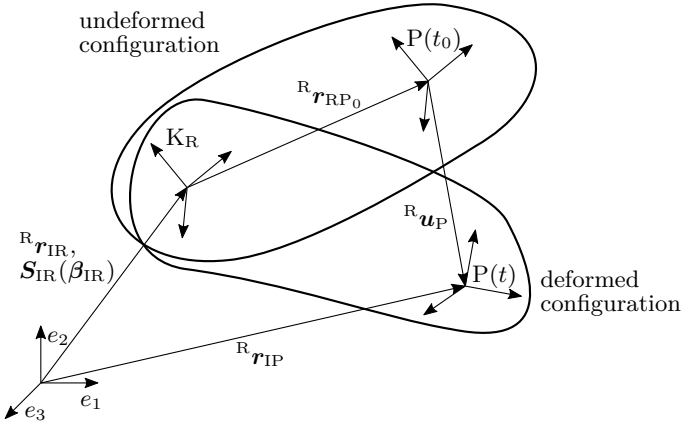


Figure 2.1: Kinematics in the floating frame of reference formulation.

inertial frame $\mathrm{K_I}$ with

$$ {}^{\mathrm{I}}\boldsymbol{r}_{\mathrm{IR}} = \boldsymbol{S}_{\mathrm{IR}}(\boldsymbol{\beta}_{\mathrm{IR}}) {}^{\mathrm{R}}\boldsymbol{r}_{\mathrm{IR}}. \tag{2.1} $$

The rotation matrix $\boldsymbol{S}_{\mathrm{IR}}(\boldsymbol{\beta}_{\mathrm{IR}})$ is parametrized by the rotational parameters $\boldsymbol{\beta}_{\mathrm{IR}}$, e.g. Cardan angles.

At time t, the body is deformed. Given that the body deformations remain small and linear elastic, they can be described conveniently in the body frame K_R. For the sake of efficiency, the elastic deformations are often approximated by n_{q} global shape functions $\boldsymbol{\Phi} = \begin{bmatrix}\boldsymbol{\Phi}_1 & \dots & \boldsymbol{\Phi}_{\mathrm{n_q}}\end{bmatrix}$ and their corresponding elastic coordinates $\boldsymbol{q}_{\mathrm{e}}$. The elastic deformation of point P can be reconstructed with

$$ {}^{\mathrm{R}}\boldsymbol{u}_{\mathrm{P}}(t, {}^{\mathrm{R}}\boldsymbol{c}_{\mathrm{RP}}) = \boldsymbol{\Phi}({}^{\mathrm{R}}\boldsymbol{c}_{\mathrm{RP}})\boldsymbol{q}_{\mathrm{e}}(t). \tag{2.2} $$

Following Fig. 2.1, the position ${}^{\mathrm{R}}\boldsymbol{r}_{\mathrm{IP}}$ of the point P within the inertial frame is expressed by

$$ {}^{\mathrm{R}}\boldsymbol{r}_{\mathrm{IP}} = {}^{\mathrm{R}}\boldsymbol{r}_{\mathrm{IR}} + {}^{\mathrm{R}}\boldsymbol{r}_{\mathrm{RP}_0} + {}^{\mathrm{R}}\boldsymbol{u}_{\mathrm{P}} \tag{2.3} $$

in the body reference frame. The position, velocity, and acceleration of any point P can be expressed in terms of the position coordinates $\boldsymbol{z}_{\mathrm{I}}$, velocity coordinates $\boldsymbol{z}_{\mathrm{II}}$ and accelerations $\dot{\boldsymbol{z}}_{\mathrm{I}}$, which are defined as

$$ \boldsymbol{z}_{\mathrm{I}} = \begin{bmatrix} {}^{\mathrm{R}}\boldsymbol{r}_{\mathrm{IR}} \\ \boldsymbol{\beta}_{\mathrm{IR}} \\ \boldsymbol{q}_{\mathrm{e}} \end{bmatrix}, \quad \boldsymbol{z}_{\mathrm{II}} = \begin{bmatrix} {}^{\mathrm{R}}\boldsymbol{v}_{\mathrm{IR}} \\ {}^{\mathrm{R}}\boldsymbol{\omega}_{\mathrm{IR}} \\ \dot{\boldsymbol{q}}_{\mathrm{e}} \end{bmatrix}, \quad \text{and} \quad \dot{\boldsymbol{z}}_{\mathrm{II}} = \begin{bmatrix} {}^{\mathrm{R}}\dot{\boldsymbol{v}}_{\mathrm{IR}} \\ {}^{\mathrm{R}}\dot{\boldsymbol{\omega}}_{\mathrm{IR}} \\ \ddot{\boldsymbol{q}}_{\mathrm{e}} \end{bmatrix}. \tag{2.4} $$

They consist of the rigid body motion of the body reference frame with the translation ${}^{\mathrm{R}}\boldsymbol{r}_{\mathrm{IR}}$ and its velocity ${}^{\mathrm{R}}\boldsymbol{v}_{\mathrm{IR}}$ from the inertial frame to the body reference

frame. The angular velocity is represented by ${}^{\mathrm{R}}\boldsymbol{\omega}_{\mathrm{IR}}$. Then, the equations of motion for a free flexible body are given by

$$\underbrace{\begin{bmatrix} m\boldsymbol{E} & m\tilde{\boldsymbol{c}}^{\intercal} & \boldsymbol{C}_{\mathrm{t}}^{\intercal} \\ m\tilde{\boldsymbol{c}} & \boldsymbol{I} & \boldsymbol{C}_{\mathrm{r}}^{\intercal} \\ \boldsymbol{C}_{\mathrm{t}} & \boldsymbol{C}_{\mathrm{r}} & \overline{\boldsymbol{M}}_{\mathrm{e}} \end{bmatrix}}_{\boldsymbol{M}} \underbrace{\begin{bmatrix} {}^{\mathrm{R}}\dot{\boldsymbol{v}}_{\mathrm{IR}} \\ {}^{\mathrm{R}}\dot{\boldsymbol{\omega}}_{\mathrm{IR}} \\ \ddot{\boldsymbol{q}}_{\mathrm{e}} \end{bmatrix}}_{\dot{\boldsymbol{z}}_{\mathrm{II}}} = \underbrace{\begin{bmatrix} \boldsymbol{h}_{\mathrm{dt}} \\ \boldsymbol{h}_{\mathrm{dr}} \\ \boldsymbol{h}_{\mathrm{de}} \end{bmatrix}}_{\boldsymbol{h}_{\mathrm{d}}} + \underbrace{\begin{bmatrix} \boldsymbol{h}_{\mathrm{bt}} \\ \boldsymbol{h}_{\mathrm{br}} \\ \boldsymbol{h}_{\mathrm{be}} \end{bmatrix}}_{\boldsymbol{h}_{\mathrm{b}}} - \underbrace{\begin{bmatrix} \boldsymbol{h}_{\omega\mathrm{t}} \\ \boldsymbol{h}_{\omega\mathrm{r}} \\ \boldsymbol{h}_{\omega\mathrm{e}} \end{bmatrix}}_{\boldsymbol{h}_{\omega}} - \underbrace{\begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \\ \overline{\boldsymbol{K}}_{\mathrm{e}}\boldsymbol{q}_{\mathrm{e}} + \overline{\boldsymbol{D}}_{\mathrm{e}}\dot{\boldsymbol{q}}_{\mathrm{e}} \end{bmatrix}}_{\boldsymbol{h}_{\mathrm{e}}}, \tag{2.5}$$

where $\boldsymbol{E}$ is the identity matrix, see [SchwertassekWallrapp14]. In Eq. (2.5), the mass matrix is denoted by $\boldsymbol{M}$, the mass of the body by m, the center of mass relative to K_{R} by $\boldsymbol{c}$, the translational and rotational coupling matrices by $\boldsymbol{C}_{\mathrm{t}}$ and $\boldsymbol{C}_{\mathrm{r}}$, the mass moment of inertia by $\boldsymbol{I}$, and the mass, stiffness, and damping matrix of the flexible body by $\overline{\boldsymbol{M}}_{\mathrm{e}}$, $\overline{\boldsymbol{K}}_{\mathrm{e}}$, and $\overline{\boldsymbol{D}}_{\mathrm{e}}$. The right-hand side of Eq. (2.5) is composed of the vector of discrete forces $\boldsymbol{h}_{\mathrm{d}}$, the body forces $\boldsymbol{h}_{\mathrm{b}}$, the generalized inertial forces $\boldsymbol{h}_{\omega}$, and the internal forces $\boldsymbol{h}_{\mathrm{e}}$. The discrete forces $\boldsymbol{h}_{\mathrm{d}}$ are surface forces and include contact forces that act on the body. This work uses Buckens-frames [SchwertassekWallrapp14] as floating frames for bodies in contact and tangent-frames [SchwertassekWallrapp14] for the remaining flexible bodies. If a Buckens-frame is used, the centroid $\boldsymbol{c}$ becomes zero, and the translational coupling matrix $\boldsymbol{C}_{\mathrm{t}}$ vanishes [SchwertassekWallrapp14]. The required data to evaluate the equations of motion of a free flexible body are provided by the SID [SchwertassekWallrapp14].

2.2 Basis Splines

The IGA consists of three spaces: the physical space, the parameter space, and the index space. The parameter space and the physical space are visualized in Figure 2.2 using a 2D model for illustration purpose. However, the description and equations also include the third dimension. The physical space consists of control points and the control net colored in red. They are introduced in the next section. The parameter space consists of the local coordinates ξ, η, and ζ. Note that ζ only appears in 3D models and is not shown in Fig. 2.2. In the planar example in Fig. 2.2, the parameter space is normalized to 1. However, all local coordinate axis can be normalized to any value. In both spaces, knots are colored in blue and elements in gray. The knot vectors $\boldsymbol{\Xi} = \begin{bmatrix} \xi_1 & \xi_2 & \dots & \xi_{\mathrm{n+p+1}} \end{bmatrix}$, $\boldsymbol{\mathcal{H}} = \begin{bmatrix} \eta_1 & \eta_2 & \dots & \eta_{\mathrm{m+q+1}} \end{bmatrix}$, and $\boldsymbol{\mathcal{Z}} = \begin{bmatrix} \zeta_1 & \zeta_2 & \dots & \zeta_{\ell+\mathrm{q+1}} \end{bmatrix}$ span up the parameter space and its elements. Additionally, the n, m, and ℓ local shape functions $N_{i,\mathrm{p}}$, $M_{j,\mathrm{q}}$, and $L_{k,\mathrm{r}}$ of order p, q, and r are defined in the parameter

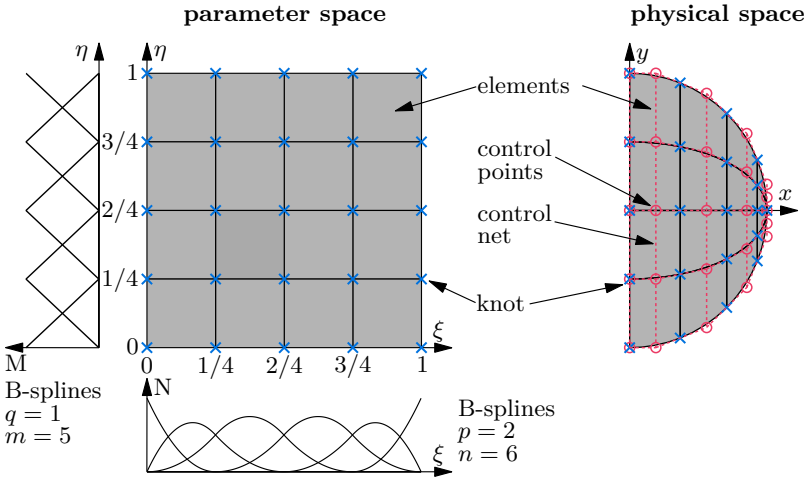


Figure 2.2: Parameter space and the physical space in the IGA.

space in the respective local coordinate direction. The local shape functions are based on Basis splines (B-splines) [CottrellEtAl09]. The knots in Fig. 2.2 are defined at the positions 0, $\frac{1}{4}$, $\frac{2}{4}$, $\frac{3}{4}$, and 1. However, the actual knot vectors are

$$\begin{aligned} \boldsymbol{\Xi} &= \begin{bmatrix} 0 & 0 & 0 & \frac{1}{4} & \frac{2}{4} & \frac{3}{4} & 1 & 1 & 1 \end{bmatrix} \in \mathbb{N}^{\mathrm{n+p+1=6+2+1}} \\ \boldsymbol{\mathcal{H}} &= \begin{bmatrix} 0 & 0 & \frac{1}{4} & \frac{2}{4} & \frac{3}{4} & 1 & 1 \end{bmatrix} \in \mathbb{N}^{\mathrm{m+q+1=5+1+1}} \end{aligned} \tag{2.6}$$

for this example. The knots are not unique and a knot can repeat itself several times. Therefore, the index space in Fig. 2.3 considers the multiplicity of knots. The multiplicity of knots is important and influences the shape of the local shape functions. Usually, open knot vectors are used, which is the case when the first and last knot of a knot vector occur $p+1$, $q+1$ or $r+1$ times. This also applies to the knots vectors in Eq. (2.6). The characteristic appearance of open knot vectors can be seen in Fig. 2.2 at the beginning and end of the B-splines.

In addition to elements, there are also patches in the IGA. Multiple elements can be part of a patch. A patch is a self-contained part of the complete geometry. Therefore, the geometry can consist of multiple patches. The boundaries of a patch are limited by knots which occur $p+1$, $q+1$, or $r+1$ times. The local shape functions become discontinuous at these knots. Since the two knot vectors from Eq. (2.6) are open knot vectors, the model only consists of one patch. Due to simplicity, IGA models in research usually consist of only one patch [CottrellEtAl09]. This is also applies to this work.

The local shape functions are based on B-splines $N_{i,\mathrm{q}}$, which are defined recursively

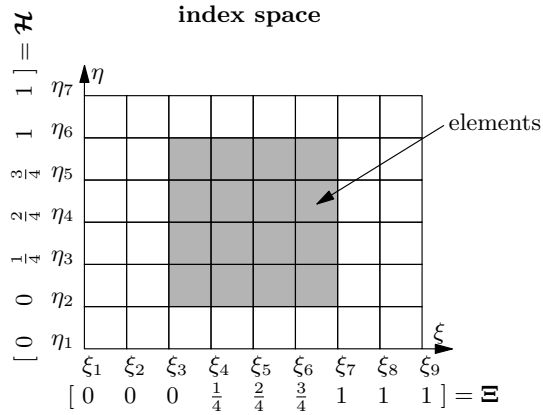


Figure 2.3: Index space in the IGA.

with the Cox-de Boor algorithm [Cox71, Boor72]

$$p = 0: \qquad N_{i,0}(\xi) = \begin{cases} 1 & \text{if } \xi_i \leq \xi < \xi_{i+1} \\ 0 & \text{otherwise} \end{cases} \tag{2.7}$$

$$p > 1: \qquad N_{i,\mathrm{p}}(\xi) = \frac{\xi - \xi_i}{\xi_{i+\mathrm{p}} - \xi_i} N_{i,\mathrm{p}-1}(\xi) + \frac{\xi_{i+\mathrm{p}+1} - \xi}{\xi_{i+\mathrm{p}+1} - \xi_{i+1}} N_{i+1,\mathrm{p}-1}(\xi). \tag{2.8}$$

The index i represents the i-th of n B-splines and also corresponds to the i-th entry of the knot vector $\boldsymbol{\Xi}$. The same equations apply to the local shape functions $M_{j,\mathrm{q}}$ and $L_{k,\mathrm{r}}$ in η- and ζ-direction. Since the same knot can occur multiple times in a knot vector, see Eq. (2.6), the denominator of one of the two fractions in Eq. (2.8) may become zero. If this is the case, the respective fraction becomes zero. Between the knots, the B-splines are C^{∞}-continuous. At the position of a knot the B-splines are $C^{\mathrm{p}-\mathrm{k}_\xi}$-, $C^{\mathrm{q}-\mathrm{k}_\eta}$-, or $C^{\mathrm{r}-\mathrm{k}_\zeta}$-continuous, where k_ξ, k_η, and k_ζ are the multiplicities of the current knot. Derivatives of B-splines are also defined recursively, see [CottrellEtAl09]. However, recursive functions are numerically inefficient. This is of minor importance in the preprocessing of IGA models, but is critical for the contact detection in each time step. Therefore, the non-recursive algorithm *DersBasisFuns* suggested in [PieglTiller97] is used in practical implementation.

2.3 Non-Uniform Rational Basis Splines

As visualized in Figure 2.2, the physical space contains a set of control points $\boldsymbol{P}$ whose task is to span the geometry in the physical space. The individual control points $\boldsymbol{p}_{i,j,k}$ are arranged by the so-called control net indicated by the indices i, j, and k. For a 3D model, the control net is defined with

$$\boldsymbol{P} = \left[\begin{array}{cccc} \boldsymbol{p}_{1,1,1} & \boldsymbol{p}_{1,2,1} & \cdots & \boldsymbol{p}_{1,\mathrm{m},1} \\ \boldsymbol{p}_{2,1,1} & \boldsymbol{p}_{2,2,1} & \cdots & \boldsymbol{p}_{2,\mathrm{m},1} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{p}_{\mathrm{n},1,1} & \boldsymbol{p}_{\mathrm{n},2,1} & \cdots & \boldsymbol{p}_{\mathrm{n},\mathrm{m},1} \end{array}\right] \left[\begin{array}{cccc} \boldsymbol{p}_{1,1,2} & \boldsymbol{p}_{1,2,2} & \cdots & \boldsymbol{p}_{1,\mathrm{m},2} \\ \boldsymbol{p}_{2,1,2} & \boldsymbol{p}_{2,2,2} & \cdots & \boldsymbol{p}_{2,\mathrm{m},2} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{P}_{\mathrm{n},1,2} & \boldsymbol{P}_{\mathrm{n},2,2} & \cdots & \boldsymbol{P}_{\mathrm{n},\mathrm{m},2} \end{array}\right] \left[\begin{array}{cccc} \boldsymbol{p}_{1,1,\ell} & \boldsymbol{p}_{1,2,\ell} & \cdots & \boldsymbol{p}_{1,\mathrm{m},\ell} \\ \boldsymbol{p}_{2,1,\ell} & \boldsymbol{p}_{2,2,\ell} & \cdots & \boldsymbol{p}_{2,\mathrm{m},\ell} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{p}_{\mathrm{n},1,\ell} & \boldsymbol{p}_{\mathrm{n},2,\ell} & \cdots & \boldsymbol{p}_{\mathrm{n},\mathrm{m},\ell} \end{array}\right] \tag{2.9}$$

representing a 3D matrix. Additionally, each individual control point

$$\boldsymbol{p}_{i,j,k} = \begin{bmatrix} p^{\mathrm{x}}_{i,j,k} \\ p^{\mathrm{y}}_{i,j,k} \\ p^{\mathrm{z}}_{i,j,k} \end{bmatrix} \tag{2.10}$$

has a x, y and z component. For easier notation, the matrix $\boldsymbol{P}$ of the control net in Eq. (2.9) can also be decomposed into the matrices $\boldsymbol{P}^{\mathrm{x}}$, $\boldsymbol{P}^{\mathrm{y}}$ and $\boldsymbol{P}^{\mathrm{z}}$, each containing only the x, y and z positions. The number of control points is identical to the number of basis functions resulting in a net of $n_{\mathrm{cp}} = n \cdot m \cdot \ell$ control points. In addition to the physical position, each control point has a weight $w_{i,j,k}$. The weights $\boldsymbol{W}$ are arranged in the control net in the same way as the control points in Eq. (2.9). The transformation from the parameter space into the physical space requires the non-uniform rational basis splines (NURBS) basis $R^{\mathrm{p,q,r}}_{i,j,k}(\xi,\eta,\zeta)$ given by

$$R^{\mathrm{p,q,r}}_{i,j,k}(\xi,\eta,\zeta) = \frac{N_{i,\mathrm{p}}(\xi) M_{j,\mathrm{q}}(\eta) L_{k,\mathrm{r}}(\zeta) w_{i,j,k}}{\sum_{\hat{i}=1}^{\mathrm{n}} \sum_{\hat{j}=1}^{\mathrm{m}} \sum_{\hat{k}=1}^{\ell} N_{\hat{i},\mathrm{p}}(\xi) M_{\hat{j},\mathrm{q}}(\eta) L_{\hat{k},\mathrm{r}}(\zeta) w_{\hat{i},\hat{j},\hat{k}}}, \tag{2.11}$$

see [CottrellEtAl09]. The first and second derivative of the NURBS basis will be required in later sections. In [CottrellEtAl09], the partial derivative is defined recursively and the second derivative is only given with respect to one variable,

e.g. $\partial^2 R^{\mathrm{p,q,r}}_{i,j,k}/\partial\xi^2$. Instead, the partial derivative can be derived with a symbolic calculator, the MATLAB SYMBOLIC MATH TOOLBOX [Mathworks23], including the partial derivative with respect to two variables, e.g. $\partial^2 R^{\mathrm{p,q,r}}_{i,j,k}/\partial\xi\partial\eta$. The first derivative results in

$$\begin{aligned}\frac{\partial R^{\mathrm{p,q,r}}_{i,j,k}}{\partial\xi} = &\frac{\frac{\partial N_{i,\mathrm{p}}}{\partial\xi} M_{j,\mathrm{q}} L_{k,\mathrm{r}} w_{i,j,k}}{\sum_{\hat{i}=1}^{\mathrm{n}}\sum_{\hat{j}=1}^{\mathrm{m}}\sum_{\hat{k}=1}^{\ell} N_{\hat{i},\mathrm{p}} M_{\hat{j},\mathrm{q}} L_{\hat{k},\mathrm{r}} w_{\hat{i},\hat{j},\hat{k}}} \\ &- \frac{N_{i,\mathrm{p}} M_{j,\mathrm{q}} L_{k,\mathrm{r}} w_{i,j,k} \sum_{\hat{i}=1}^{\mathrm{n}}\sum_{\hat{j}=1}^{\mathrm{m}}\sum_{\hat{k}=1}^{\ell} \frac{\partial N_{i,\mathrm{p}}}{\partial\xi} M_{j,\mathrm{q}} L_{k,\mathrm{r}} w_{i,j,k}}{\left(\sum_{\hat{i}=1}^{\mathrm{n}}\sum_{\hat{j}=1}^{\mathrm{m}}\sum_{\hat{k}=1}^{\ell} N_{\hat{i},\mathrm{p}} M_{\hat{j},\mathrm{q}} L_{\hat{k},\mathrm{r}} w_{\hat{i},\hat{j},\hat{k}}\right)^2},\end{aligned} \tag{2.12}$$

and the second derivative with respect to one variable results in

$$\begin{aligned}\frac{\partial^2 R^{\mathrm{p,q,r}}_{i,j,k}}{\partial\xi^2} = &\frac{\frac{\partial^2 N_{i,\mathrm{p}}}{\partial\xi^2} M_{j,\mathrm{q}} L_{k,\mathrm{r}} w_{i,j,k}}{\sum_{\hat{i}=1}^{\mathrm{n}}\sum_{\hat{j}=1}^{\mathrm{m}}\sum_{\hat{k}=1}^{\ell} N_{\hat{i},\mathrm{p}} M_{\hat{j},\mathrm{q}} L_{\hat{k},\mathrm{r}} w_{\hat{i},\hat{j},\hat{k}}} \\ &- \frac{2N_{i,\mathrm{p}} M_{j,\mathrm{q}} L_{k,\mathrm{r}} w_{i,j,k} \sum_{\hat{i}=1}^{\mathrm{n}}\sum_{\hat{j}=1}^{\mathrm{m}}\sum_{\hat{k}=1}^{\ell} \frac{\partial N_{\hat{i},\mathrm{p}}}{\partial\xi} M_{\hat{j},\mathrm{q}} L_{\hat{k},\mathrm{r}} w_{\hat{i},\hat{j},\hat{k}}}{\left(\sum_{\hat{i}=1}^{\mathrm{n}}\sum_{\hat{j}=1}^{\mathrm{m}}\sum_{\hat{k}=1}^{\ell} N_{\hat{i},\mathrm{p}} M_{\hat{j},\mathrm{q}} L_{\hat{k},\mathrm{r}} w_{\hat{i},\hat{j},\hat{k}}\right)^2} \\ &- \frac{N_{i,\mathrm{p}} M_{j,\mathrm{q}} L_{k,\mathrm{r}} w_{i,j,k} \sum_{\hat{i}=1}^{\mathrm{n}}\sum_{\hat{j}=1}^{\mathrm{m}}\sum_{\hat{k}=1}^{\ell} \frac{\partial^2 N_{\hat{i},\mathrm{p}}}{\partial\xi^2} M_{\hat{j},\mathrm{q}} L_{\hat{k},\mathrm{r}} w_{\hat{i},\hat{j},\hat{k}}}{\left(\sum_{\hat{i}=1}^{\mathrm{n}}\sum_{\hat{j}=1}^{\mathrm{m}}\sum_{\hat{k}=1}^{\ell} N_{\hat{i},\mathrm{p}} M_{\hat{j},\mathrm{q}} L_{\hat{k},\mathrm{r}} w_{\hat{i},\hat{j},\hat{k}}\right)^2} \\ &+ \frac{2N_{i,\mathrm{p}} M_{j,\mathrm{q}} L_{k,\mathrm{r}} w_{i,j,k} \left(\sum_{\hat{i}=1}^{\mathrm{n}}\sum_{\hat{j}=1}^{\mathrm{m}}\sum_{\hat{k}=1}^{\ell} \frac{\partial^2 N_{\hat{i},\mathrm{p}}}{\partial\xi^2} M_{\hat{j},\mathrm{q}} L_{\hat{k},\mathrm{r}} w_{\hat{i},\hat{j},\hat{k}}\right)^2}{\left(\sum_{\hat{i}=1}^{\mathrm{n}}\sum_{\hat{j}=1}^{\mathrm{m}}\sum_{\hat{k}=1}^{\ell} N_{\hat{i},\mathrm{p}} M_{\hat{j},\mathrm{q}} L_{\hat{k},\mathrm{r}} w_{\hat{i},\hat{j},\hat{k}}\right)^3}.\end{aligned} \tag{2.13}$$

Finally, the derivative with respect to two variables is given by

$$
\begin{aligned}
\frac{\partial^2 R^{\mathrm{p,q,r}}_{i,j,k}}{\partial\xi\partial\eta} = {} & \frac{\frac{\partial N_{i,\mathrm{p}}}{\partial\xi}\frac{\partial M_{j,\mathrm{q}}}{\partial\eta}L_{k,\mathrm{r}}w_{i,j,k}}{\sum_{\hat{i}=1}^{\mathrm{n}}\sum_{\hat{j}=1}^{\mathrm{m}}\sum_{\hat{k}=1}^{\ell}N_{\hat{i},\mathrm{p}}M_{\hat{j},\mathrm{q}}L_{\hat{k},\mathrm{r}}w_{\hat{i},\hat{j},\hat{k}}} \\
& - \frac{\frac{\partial N_{i,\mathrm{p}}}{\partial\xi}M_{j,\mathrm{q}}L_{k,\mathrm{r}}w_{i,j,k}\sum_{\hat{i}=1}^{\mathrm{n}}\sum_{\hat{j}=1}^{\mathrm{m}}\sum_{\hat{k}=1}^{\ell}N_{\hat{i},\mathrm{p}}\frac{\partial M_{\hat{j},\mathrm{q}}}{\partial\eta}L_{\hat{k},\mathrm{r}}w_{\hat{i},\hat{j},\hat{k}}}{\left(\sum_{\hat{i}=1}^{\mathrm{n}}\sum_{\hat{j}=1}^{\mathrm{m}}\sum_{\hat{k}=1}^{\ell}N_{\hat{i},\mathrm{p}}M_{\hat{j},\mathrm{q}}L_{\hat{k},\mathrm{r}}w_{\hat{i},\hat{j},\hat{k}}\right)^2} \\
& - \frac{N_{i,\mathrm{p}}\frac{\partial M_{j,\mathrm{q}}}{\partial\eta}L_{k,\mathrm{r}}w_{i,j,k}\sum_{\hat{i}=1}^{\mathrm{n}}\sum_{\hat{j}=1}^{\mathrm{m}}\sum_{\hat{k}=1}^{\ell}\frac{\partial N_{\hat{i},\mathrm{p}}}{\partial\xi}M_{\hat{j},\mathrm{q}}L_{\hat{k},\mathrm{r}}w_{\hat{i},\hat{j},\hat{k}}}{\left(\sum_{\hat{i}=1}^{\mathrm{n}}\sum_{\hat{j}=1}^{\mathrm{m}}\sum_{\hat{k}=1}^{\ell}N_{\hat{i},\mathrm{p}}M_{\hat{j},\mathrm{q}}L_{\hat{k},\mathrm{r}}w_{\hat{i},\hat{j},\hat{k}}\right)^2} \\
& - \frac{N_{i,\mathrm{p}}M_{j,\mathrm{q}}L_{k,\mathrm{r}}w_{i,j,k}\sum_{\hat{i}=1}^{\mathrm{n}}\sum_{\hat{j}=1}^{\mathrm{m}}\sum_{\hat{k}=1}^{\ell}\frac{\partial N_{\hat{i},\mathrm{p}}}{\partial\xi}\frac{\partial M_{\hat{j},\mathrm{q}}}{\partial\eta}L_{\hat{k},\mathrm{r}}w_{\hat{i},\hat{j},\hat{k}}}{\left(\sum_{\hat{i}=1}^{\mathrm{n}}\sum_{\hat{j}=1}^{\mathrm{m}}\sum_{\hat{k}=1}^{\ell}N_{\hat{i},\mathrm{p}}M_{\hat{j},\mathrm{q}}L_{\hat{k},\mathrm{r}}w_{\hat{i},\hat{j},\hat{k}}\right)^2} \\
& + \frac{2N_{i,\mathrm{p}}M_{j,\mathrm{q}}L_{k,\mathrm{r}}w_{i,j,k}\sum_{\hat{i}=1}^{\mathrm{n}}\sum_{\hat{j}=1}^{\mathrm{m}}\sum_{\hat{k}=1}^{\ell}\frac{\partial N_{\hat{i},\mathrm{p}}}{\partial\xi}M_{\hat{j},\mathrm{q}}L_{\hat{k},\mathrm{r}}w_{\hat{i},\hat{j},\hat{k}}}{\left(\sum_{\hat{i}=1}^{\mathrm{n}}\sum_{\hat{j}=1}^{\mathrm{m}}\sum_{\hat{k}=1}^{\ell}N_{\hat{i},\mathrm{p}}M_{\hat{j},\mathrm{q}}L_{\hat{k},\mathrm{r}}w_{\hat{i},\hat{j},\hat{k}}\right)^3} \\
& \cdot \sum_{\hat{i}=1}^{\mathrm{n}}\sum_{\hat{j}=1}^{\mathrm{m}}\sum_{\hat{k}=1}^{\ell}N_{\hat{i},\mathrm{p}}\frac{\partial M_{\hat{j},\mathrm{q}}}{\partial\eta}L_{\hat{k},\mathrm{r}}w_{\hat{i},\hat{j},\hat{k}}.
\end{aligned}
\tag{2.14}
$$

For an efficient computation of the partial derivatives, individual terms are pre-evaluated and then inserted into Eq. (2.11)-Eq. (2.14). To transform a point $\begin{bmatrix}\xi & \eta & \zeta\end{bmatrix}^\intercal$ in the parameter space to a point $\begin{bmatrix}x & y & z\end{bmatrix}^\intercal$ in the physical space, the computation of the NURBS solid $\boldsymbol{S}$ is required. With the NURBS basis $R^{\mathrm{p,q,r}}_{i,j,k}(\xi,\eta,\zeta)$ and the control points $\boldsymbol{p}_{i,j,k}$, the NURBS solid $\boldsymbol{S}$ can be computed with

$$
\boldsymbol{S} = \begin{bmatrix}x\\y\\z\end{bmatrix} = \sum_{i=1}^{\mathrm{n}}\sum_{j=1}^{\mathrm{m}}\sum_{k=1}^{\ell}R^{\mathrm{p,q,r}}_{i,j,k}(\xi,\eta,\zeta)\boldsymbol{p}_{i,j,k} \tag{2.15}
$$

consisting of the coordinates x, y, and z in the physical space. To compute the IGA equations of motion in Sect. 2.6, occurring stresses in Sect. 2.8, and contacts

in Sect. 3.3, the derivative of the NURBS solid $\boldsymbol{S}$ is required. The i-th partial derivative of the NURBS solid can be easily computed with

$$\frac{\partial^i \boldsymbol{S}}{\partial \xi^i} = \begin{bmatrix} \frac{\partial^i x}{\partial \xi^i} \\ \frac{\partial^i y}{\partial \xi^i} \\ \frac{\partial^i z}{\partial \xi^i} \end{bmatrix} = \sum_{i=1}^{\mathrm{n}} \sum_{j=1}^{\mathrm{m}} \sum_{k=1}^{\ell} \frac{\partial^i R_{i,j,k}^{\mathrm{p,q,r}}(\xi,\eta,\zeta)}{\partial \xi^i} \boldsymbol{p}_{i,j,k}. \tag{2.16}$$

using the partial derivative of the NURBS basis from Eq. (2.12) and Eq. (2.13). With Eq. (2.14), the partial derivative of the NURBS solid with respect to two local coordinates is computed with

$$\frac{\partial^2 \boldsymbol{S}}{\partial \xi \partial \eta} = \begin{bmatrix} \frac{\partial^2 x}{\partial \xi \partial \eta} \\ \frac{\partial^2 y}{\partial \xi \partial \eta} \\ \frac{\partial^2 z}{\partial \xi \partial \eta} \end{bmatrix} = \sum_{i=1}^{\mathrm{n}} \sum_{j=1}^{\mathrm{m}} \sum_{k=1}^{\ell} \frac{\partial^2 R_{i,j,k}^{\mathrm{p,q,r}}(\xi,\eta,\zeta)}{\partial \xi \partial \eta} \boldsymbol{p}_{i,j,k}. \tag{2.17}$$

The partial derivatives with respect to η and ζ are computed accordingly.

Although the notation of the interval limits of the sums in Eq. (2.11)-Eq. (2.17) is easy to understand, its computation is inefficient in practical implementation. In total, there are n, m, or ℓ B-splines but only $p+1$, $q+1$ or $r+1$ B-splines are nonzero, see Fig. 2.2. Therefore, the sums in Eq. (2.11)-Eq. (2.17) also include B-splines that are zero. To compute the sums in Eq. (2.11)-Eq. (2.17) more efficiently, the concept of knot spans must be understood. A knot span is an element in the IGA. Each knot span is bounded by knots. Within a knot span, only $p+1$, $q+1$, or $r+1$ B-splines are nonzero. This is called the support. The remaining B-splines are zero and can be neglected in Eq. (2.11)-Eq. (2.17). As an example, the index space from Fig. 2.3 is extended in Fig. 2.4. The supports are colored in yellow and their corresponding knot spans are colored in red. The area of the support is shifted by the multiplicity of the current knot (ξ_i, η_j, ζ_k), which is colored in blue in Fig. 2.4. In this way, the supporting control points can also be read from the control points $\boldsymbol{p}_{i,j,k}$ arranged in the control net. As a reminder, see the arrangement of control points in matrix notation in Eq. (2.9), which represents the control net. The range $\hat{i}$, $\hat{j}$, and $\hat{k}$ of the support is given by

$$\hat{i} = i - p, ..., i \tag{2.18}$$

$$\hat{j} = j - q, ..., j \tag{2.19}$$

$$\hat{k} = k - r, ..., k \tag{2.20}$$

with respect to the indices i, j, and k of the current knot (ξ_i, η_j, ζ_k). The current knot span can be determined with the algorithm *findSpan* suggested in [PieglTiller97].

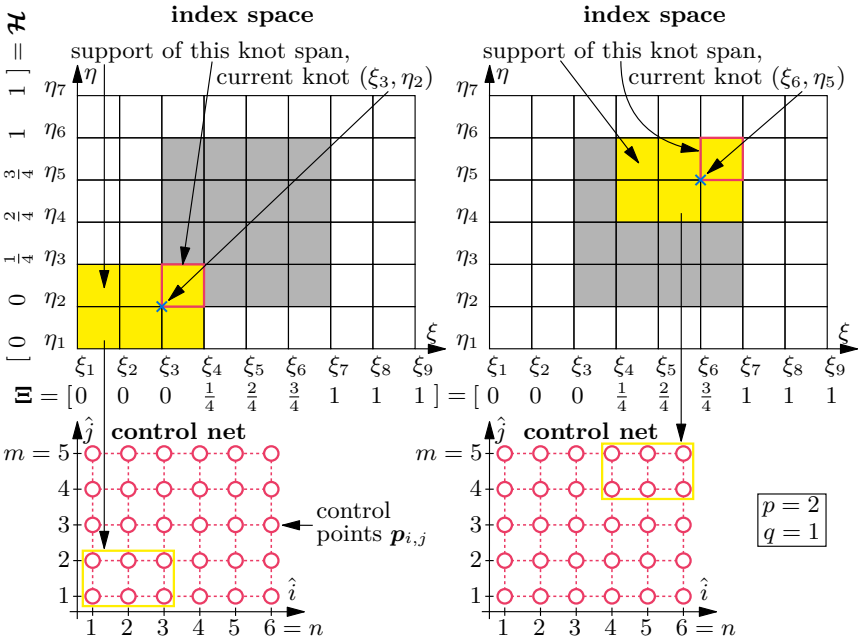


Figure 2.4: Support of a knot span in the index space and in the control net.

The displacement field $\boldsymbol{d}$ of the elastic solid can be written in matrix-vector notation as

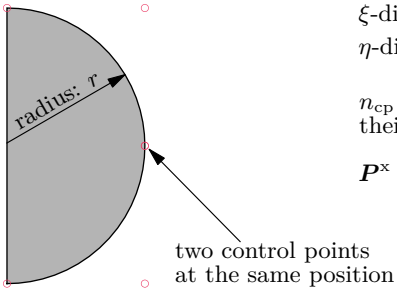
$$\boldsymbol{d} = \underbrace{\begin{bmatrix} R^{\mathrm{p,q,r}}_{1,1,1} & 0 & 0 & \dots & 0 \\ 0 & R^{\mathrm{p,q,r}}_{1,1,1} & 0 & \dots & 0 \\ 0 & 0 & R^{\mathrm{p,q,r}}_{1,1,1} & \dots & R^{\mathrm{p,q,r}}_{\mathrm{p+1,q+1,r+1}} \end{bmatrix}}_{\boldsymbol{N}} \underbrace{\begin{bmatrix} u^{\mathrm{x}}_{1,1,1} \\ u^{\mathrm{y}}_{1,1,1} \\ u^{\mathrm{z}}_{1,1,1} \\ \vdots \\ u^{\mathrm{z}}_{\mathrm{p+1,q+1,r+1}} \end{bmatrix}}_{\boldsymbol{u}}, \tag{2.21}$$

where the supporting basis functions of the corresponding knot span are summarized in the matrix $\boldsymbol{N}$ and the supporting displacements of the control points in $\boldsymbol{u}$. It is worth noting that by adding six kinematical constraints for the floating frame, all rigid body motions vanish and only the body deformations remain in $\boldsymbol{d}$.

2.4 Global Refinement

As mentioned before, an IGA model represents its geometry without discretization errors. This usually requires only a small number of degrees of freedom. However, when elastic deformations occur, an IGA model with a small number of degrees of freedom cannot accurately describe the elastic behavior. Therefore, refinement of the IGA model is required. An IGA model is usually based on an initial geometry. The semicircle in Fig. 2.2 is already refined and consists of $n_\mathrm{e} = 16$ elements. However, its initial geometry in Fig. 2.5 has only $n_\mathrm{e} = 1$ element and $n_\mathrm{cp} = 6$ control points. To describe the geometry of a semicircle, it is necessary that two control points are at the same position. Control points lying on top of each other can be constraint in the equations of motion by means of linear constraint equations. This master-slave principle can be transferred from the linear FEM [Bathe14] to the IGA.

Since the initial model in Fig. 2.5 has an insufficient number of degrees of freedom to map the overall elastic deformation of the body, the geometry needs to be refined. The straightforward approach in the IGA is global refinement. It includes the elevation of the order of the B-splines and the insertion of additional knots. The global refinement in the IGA [CottrellEtAl09] has similarities to the h- and p-refinement in the FEM [Bathe14]. The h-refinement in the FEM results in a finer mesh. Isoparametric elements are divided into smaller elements by inserting additional nodes. With p-refinement, the degree of the local shape functions is increased. Therefore, the order elevation in the IGA corresponds to the p-refinement in the FEM, and the insertion of knots corresponds to the h-refinement. When refining in FEM, manual optimization by the user is usually required [Bathe14]. This is not the case in the IGA. As an example, in contrast to the FEM, no unacceptable elements are created when additional knots are inserted. Figure 2.6 shows the overall procedure of global refinement in the IGA. The initial geometry is the semicircle from Fig. 2.5.



ξ-direction: $\boldsymbol{\Xi} = [0\ 0\ 0\ 1\ 1\ 1]\,,\ p = 2,\ n = 3$
η-direction: $\boldsymbol{\mathcal{H}} = [0\ 0\ 1\ 1]\,,\ q = 1,\ m = 2$

$n_\mathrm{cp} = 6$ control points and
their corresponding weights:

$$\boldsymbol{P}^\mathrm{x} = \begin{bmatrix} 0 & 0 \\ r & r \\ r & r \end{bmatrix},\ \boldsymbol{P}^\mathrm{y} = \begin{bmatrix} -r & r \\ -r & r \\ 0 & 0 \end{bmatrix},\ \boldsymbol{W} = \begin{bmatrix} 1 & 1 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ 1 & 1 \end{bmatrix}$$

Figure 2.5: Initial geometry of the semicircle from Fig. 2.2.

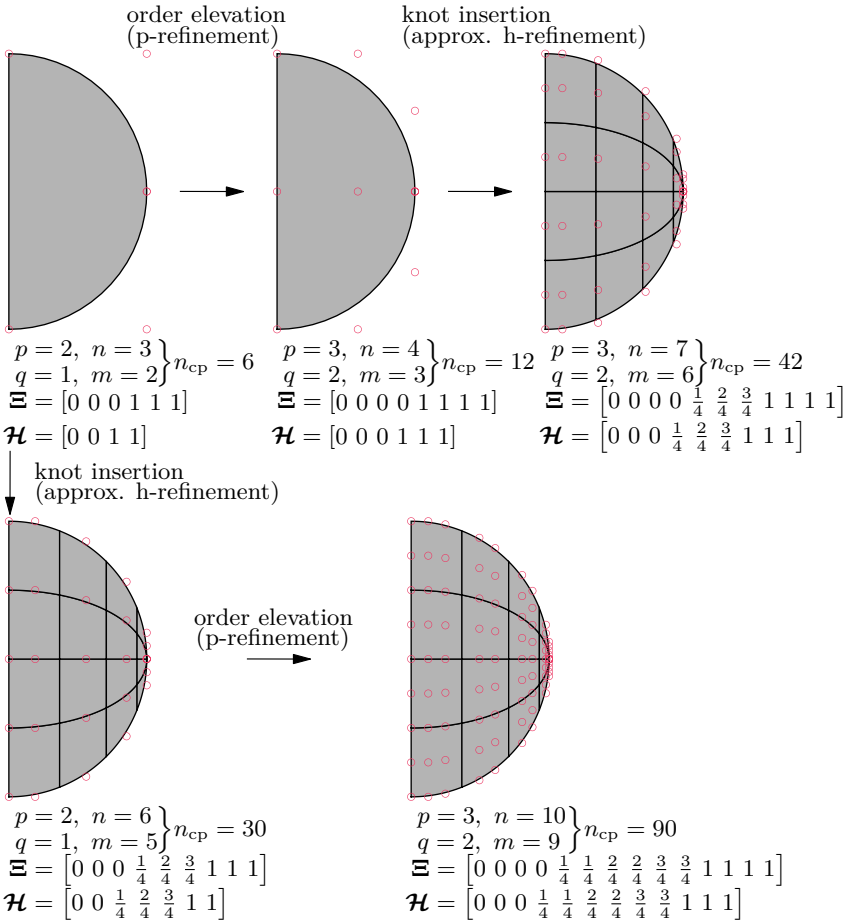
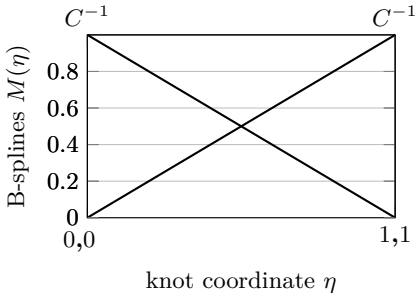


Figure 2.6: Order elevation and knot insertion in the IGA.

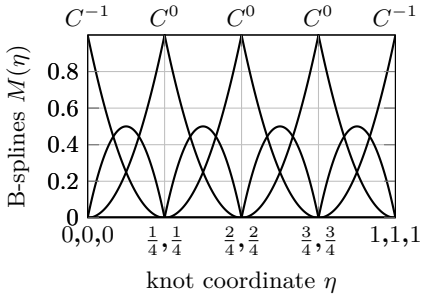
The order elevation and the insertion of knots can be performed in any arbitrary sequence. However, the sequence is relevant and influences the resulting models. As shown in the upper half of Fig. 2.6, first, the order is elevated by one. The multiplicity of all knots and the number of B-splines n and m is increased by one. The new control points and weights are computed with the algorithm described in [LeePark02]. Second, three equally spaced knots are inserted in both local coordinate directions ξ and η. Since new unique knots are added, new elements are created. The number of B-splines n and m is increased by the number of inserted knots. The new control points and weights are computed

with the algorithm described in [CottrellEtAl09]. However, the knot insertion does not fully match the h-refinement in FEM. Instead, to exactly replicate the h-refinement, the knots would have to be inserted with the multiplicity of the respective order of the B-splines. In this way, the B-splines would only be C^0-continuous, i.e. continuous but not differentiable, at the location of the inserted knots.

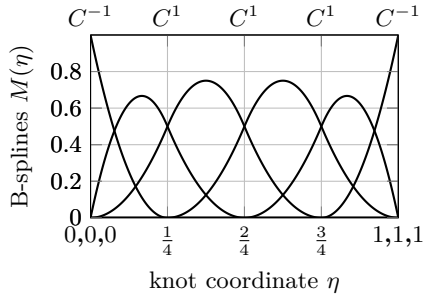
Alternatively, the sequence of refinement can be reversed, as illustrated in the lower half of Fig. 2.6. Three knots are inserted and then, the order is elevated by one. However, this sequence also increases the multiplicity of the newly inserted knots. Now, the multiplicity of the newly inserted knots is identical to the order $q = 2$, see the knot vector $\boldsymbol{\mathcal{H}}$ in the bottom right in Fig. 2.6. As described before, the B-splines are only C^0-continuous at the location of the inserted knots. The B-splines of the two different sequences of refinement are shown in Fig. 2.7. Note that in Fig. 2.7, knots appear multiple times, e.g. $[0, 0]$ or $[0, 0, 0]$, due to the multiplicity of knots, see Eq. (2.6). The B-splines of open knots vectors are



a) Initial geometry.



b) Knot insertion ⇒ order elevation.



c) Order elevation ⇒ knot insertion.

Figure 2.7: B-splines of the IGA models in Fig. 2.6.

discontinuous at the boundary of the domain. Therefore, the continuity is C^{-1} at $\eta = 0$ and $\eta = 1$ according to the notation in [CottrellEtAl09]. In Fig. 2.7a, the B-splines of the initial geometry are displayed. Since the B-splines are linear, they are identical to the local shape functions of isoparametric elements. If additional knots are inserted and then, the order is elevated, the B-splines are only C^0-continuous at the inserted knots, e.g. continuous, see Fig. 2.7b. If the sequence of refinement is reversed, the B-splines are C^1-continuous at the inserted knots, e.g. continuous and differentiable, see Fig. 2.7c. Another disadvantage of inserting knots and then elevating the order like in Fig. 2.7b is that more B-splines and thus more control points are created. The initial $n_{\mathrm{cp}} = 6$ control points then result in $n_{\mathrm{cp}} = 90$ or $n_{\mathrm{cp}} = 42$ control points. This effect becomes more relevant when not a 2D but a 3D model is refined, because the number of control points is then defined as $n_{\mathrm{cp}} = n \cdot m \cdot \ell$. For these reasons, it is preferred to increase the order first and then insert the knots in the IGA, see Fig. 2.7c. This process, which does not exist in the FEM, is called k-refinement [CottrellEtAl09]. Only k-refinement is used in this work. The example of the semicircle is continued in following section. Therefore, the refined model in the top right in Fig. 2.6 is selected. Its refined knot vectors are given by

$$\begin{aligned} \boldsymbol{\Xi} &= \left[0 \;\; 0 \;\; 0 \;\; \frac{1}{4} \;\; \frac{2}{4} \;\; \frac{3}{4} \;\; 1 \;\; 1 \;\; 1\right], \quad p = 2, \quad n = 7 \\ \boldsymbol{\mathcal{H}} &= \left[0 \;\; 0 \;\; \frac{1}{4} \;\; \frac{2}{4} \;\; \frac{3}{4} \;\; 1 \;\; 1\right], \quad q = 1, \quad m = 6. \end{aligned} \tag{2.22}$$

2.5 Hierarchical Refinement

The k-refinement from Sect. 2.4 can be used to globally refine the body in order to describe the overall elastic deformation. However, an accurate impact simulation, which also captures the local deformation, requires a high element resolution in the contact area, as this is where the largest elastic deformations and stresses occur.

As described in Sect. 2.4, the refinement methods usually used in the IGA only allow global refinement [CottrellEtAl09]. The semicircle representing an axisymmetric sphere from Fig. 2.2 and Fig. 2.6 is already coarsely refined with global refinement. An impact will occur on its lower side. Therefore, the contact area needs to be refined with elements. The refinement in the contact area is exemplified in Fig. 2.8. The knot vectors from Eq. (2.22) are refined to

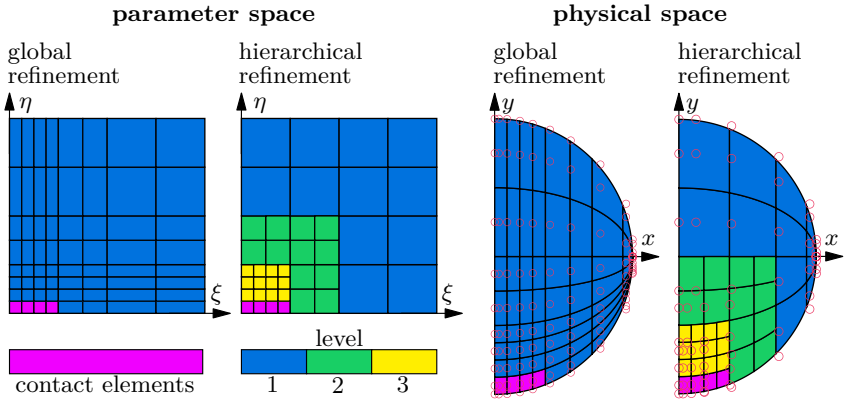


Figure 2.8: Refinement of the semicircle in the contact area with global and hierarchical refinement.

$$\begin{aligned}\boldsymbol{\Xi} &= \left[0 \;\; 0 \;\; 0 \;\; 0 \;\; \frac{1}{16} \;\; \frac{2}{16} \;\; \frac{3}{16} \;\; \frac{1}{4} \;\; \frac{3}{8} \;\; \frac{2}{4} \;\; \frac{3}{4} \;\; 1 \;\; 1 \;\; 1 \;\; 1\right], \quad p = 3, \quad n = 11 \\ \boldsymbol{\mathcal{H}} &= \left[0 \;\; 0 \;\; 0 \;\; \frac{1}{16} \;\; \frac{2}{16} \;\; \frac{3}{16} \;\; \frac{1}{4} \;\; \frac{3}{8} \;\; \frac{2}{4} \;\; \frac{3}{4} \;\; 1 \;\; 1 \;\; 1\right], \quad q = 2, \quad m = 10.\end{aligned} \tag{2.23}$$

The body to the left is refined globally and the body to the right locally. With global refinement, refining the body in the contact area creates additional elements and control points over the entire body. Therefore, the number of degrees of freedom greatly increases with global refinement. One method for local refinement is hierarchical refinement [SchillingerEtAl12], where subordinate levels are introduced. The concept of hierarchical refinement relies on the property of B-splines to be represented by a linear combination of finer B-splines defined on smaller knot intervals. For a detailed introduction to the hierarchical refinement, see [SchillingerEtAl12, Matzen15].

The concept of the hierarchical refinement relies on the property of B-splines to be represented by a linear combination of finer B-splines defined on smaller knot-intervals. With the calculation rule

$$a_j = 2^{-p} \binom{p+1}{j} = 2^{-p} \frac{(p+1)!}{j!\,(p+1-j)!} \tag{2.24}$$

linear coefficients can be determined to represent a B-spline in a higher level with B-splines of lower level. It should be noted that Eq. (2.24) is only valid for uniform knot vectors. In a uniform knot vector, all knots are equally spaced. As an example, a quadratic B-spline with the high-level knot vector

$$\boldsymbol{\Xi}_1 = \begin{bmatrix}0 & 1 & 2 & 3\end{bmatrix} \tag{2.25}$$

should be represented by a number of lower level B-splines with the corresponding low-level knot vector

$$\boldsymbol{\Xi}_2 = \begin{bmatrix} 0 & \underline{0.5} & 1 & \underline{1.5} & 2 & \underline{2.5} & 3 \end{bmatrix}. \tag{2.26}$$

The inserted knots in Eq. (2.26) are underlined. By applying Eq. (2.24) the concept of the hierarchical refinement can be visualized in Fig. 2.9. The procedure, which is implemented and used in this work, is briefly summarized in the following.

Initially, the parameter space is divided into the different hierarchy levels. Recapturing the motivation example in Fig. 2.8, the corresponding parameter space is displayed on the right hand side. This division is made up by intervals in knot coordinates. The concept of hierarchical refinement is that a finer mesh resolution can be defined section by section. Therefore, the levels are initialized using the global knot insertion procedure described in Sect. 2.4. With this procedure, the position of the control points and their weights can be determined for the finer mesh. The number of hierarchy levels is denoted as n_{lvl}.

In hierarchical refinement, B-splines of higher level are based on B-splines of lower level. For the construction of higher level B-splines, the linear combination coefficient matrix $\boldsymbol{A}$ is required. For the example in Fig. 2.9, the matrix $\boldsymbol{A}$ is given by

$$\boldsymbol{A} = \begin{bmatrix} a_1 & a_2 & a_3 & a_4 \end{bmatrix} = \begin{bmatrix} 0.25 & 0.75 & 0.75 & 0.25 \end{bmatrix}. \tag{2.27}$$

Its coefficients are determined using Eq.(2.24). However, the Eq.(2.24) is only valid if one higher level B-spline is to be represented by multiple lower level B-splines. In practice, multiple higher level B-splines must be represented. Therefore, the linear combination coefficient matrix $\boldsymbol{A}$ is determined by solving

$$\underbrace{N^{k}_{i,\mathrm{p}}}_{\substack{\text{higher level}\\ \text{B-splines}}} = \underbrace{\boldsymbol{A}^{k}_{i,j}}_{\substack{\text{linear}\\ \text{coefficients}}} \underbrace{N^{k+1}_{j,\mathrm{p}}}_{\substack{\text{lower level}\\ \text{B-splines}}} \qquad \underbrace{k}_{\substack{\text{current}\\ \text{level}}} = 1 \ldots n_{\mathrm{lvl}} - 1 \tag{2.28}$$

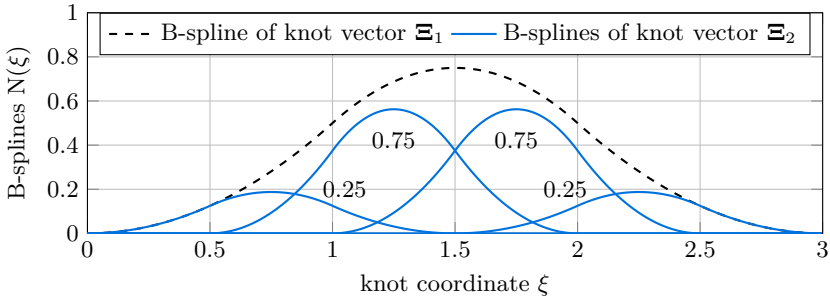


Figure 2.9: Concept of the hierarchical refinement in the IGA.

between all n_{lvl} hierarchy levels. A detailed description of this procedure can be found in [Matzen15].

In the next step, the elements are defined with the knot vectors of the different levels and the intervals, which define the hierarchy level with respect to the knot coordinates. The previous step allows certain B-splines to be identified as inactive. Thus, the supporting control points of each element can be determined. Thereby, the control points can be located in different hierarchy levels. The control points that are not part of an element are identified as inactive.

The hierarchical B-splines of the refined semicircle in Fig. 2.8 are depicted in Fig. 2.10. It can be seen that B-splines of different hierarchy levels can be active at the same local coordinate, e.g. $\xi = 0.25$. This overlap becomes relevant when calculating the NURBS basis from Eq. (2.11). The B-splines of the different dimensions of the parameter space need to be multiplied out. These intersections of the B-splines lead to the fact that for the computation of the NURBS any level can interact with any level of the other dimensions in the parameter space. The more hierarchical levels that are created, the more combinations of possible intersections of the B-splines can occur. This may increase the calculation time of the hierarchical NURBS basis compared to a globally refined model. In addition to the more complex calculation of the NURBS, the B-splines from different hierarchy levels must be constructed, which takes additional computation time.

It is noted, that the hierarchical refinement from IGA cannot be easily adapted to the FEM using isoparametric elements. As an example, h-refinement must be done with care to prevent hanging nodes. This is not an issue at the IGA. In many publications [D'AngellaEtAl18, SchillingerEtAl12, ZimmermannSauer17, GarauVázquez18, VuongEtAl11], the issue of hanging nodes or control points is not even addressed in connection with the IGA. However, the IGA is known for being watertight without gaps [CottrellEtAl09, SchillingerEtAl12]. For example in [NoëlEtAl20], it is stated that,

> "hanging nodes in hierarchical refined meshes are naturally handled by the B-spline basis."

This can be shown by a simple example. In Fig. 2.11, a 2D plate is modeled with the IGA and refined locally with hierarchical refinement. The resulting mesh is rebuild with isoparametric elements. For better comparability, both models use linear local shape functions. The hanging nodes can be easily seen in the FEM model on the left in Fig. 2.11. However, there are no control points at the equivalent positions in the IGA model. Since the IGA model uses linear B-splines, only one B-spline is nonzero at the corners of the elements. Therefore, the underlying control points have direct control of the positions of the corners of the elements and an overlap of different hierarchy levels is impossible.

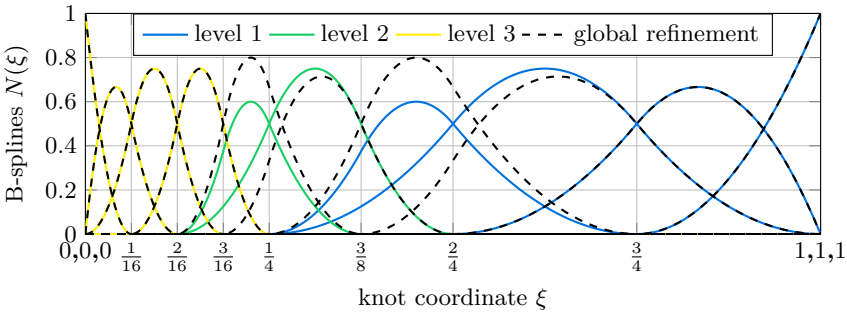


Figure 2.10: Concept of the hierarchical refinement in the IGA.

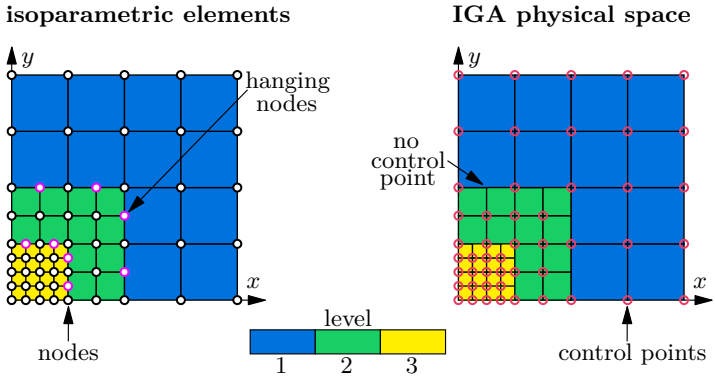


Figure 2.11: Natural handling of hanging nodes in the IGA.

2.6 IGA Equations of Motion

The procedure of setting up the equations of motion for isogeometric elements is almost identical to isoparametric elements. This is because the weak Galerkin method is applied similarly to isoparametric elements [Bathe14] as well as in IGA [CottrellEtAl09]. As with the floating frame of reference formulation, it is assumed that only small elastic deformations occur. This is mostly valid for a relatively stiff material, e.g. steel. The local mass and stiffness matrix of the i-th

element $\Omega_{\mathrm{e},i}$ is therefore given by

$$\boldsymbol{K}_{\mathrm{e},i} = \int_{\Omega_{\mathrm{e},i}} \boldsymbol{B}^{\intercal} \boldsymbol{C} \boldsymbol{B} \, \mathrm{d}\Omega_{\mathrm{e},i} \tag{2.29}$$

$$\boldsymbol{M}_{\mathrm{e},i} = \rho \int_{\Omega_{\mathrm{e},i}} \boldsymbol{N}^{\intercal} \boldsymbol{N} \, \mathrm{d}\Omega_{\mathrm{e},i} \tag{2.30}$$

where $\boldsymbol{C}$ is the material elasticity matrix. The strain displacement matrix $\boldsymbol{B}$ is obtained by differentiating the NURBS basis $R^{\mathrm{p,q,r}}_{i,j,k}$ and using the Jacobian transformation

$$\begin{bmatrix} \frac{\partial R^{\mathrm{p,q,r}}_{1,1,1}}{\partial \xi} \\ \frac{\partial R^{\mathrm{p,q,r}}_{1,1,1}}{\partial \eta} \\ \frac{\partial R^{\mathrm{p,q,r}}_{1,1,1}}{\partial \zeta} \end{bmatrix} = \underbrace{\begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \\ \frac{\partial x}{\partial \zeta} & \frac{\partial y}{\partial \zeta} & \frac{\partial z}{\partial \zeta} \end{bmatrix}}_{\boldsymbol{J}} \begin{bmatrix} \frac{\partial R^{\mathrm{p,q,r}}_{1,1,1}}{\partial x} \\ \frac{\partial R^{\mathrm{p,q,r}}_{1,1,1}}{\partial y} \\ \frac{\partial R^{\mathrm{p,q,r}}_{1,1,1}}{\partial z} \end{bmatrix} \tag{2.31}$$

$$\begin{bmatrix} \frac{\partial R^{\mathrm{p,q,r}}_{1,1,1}}{\partial x} \\ \frac{\partial R^{\mathrm{p,q,r}}_{1,1,1}}{\partial y} \\ \frac{\partial R^{\mathrm{p,q,r}}_{1,1,1}}{\partial z} \end{bmatrix} = \boldsymbol{J}^{-1} \begin{bmatrix} \frac{\partial R^{\mathrm{p,q,r}}_{1,1,1}}{\partial \xi} \\ \frac{\partial R^{\mathrm{p,q,r}}_{1,1,1}}{\partial \eta} \\ \frac{\partial R^{\mathrm{p,q,r}}_{1,1,1}}{\partial \zeta} \end{bmatrix}, \tag{2.32}$$

see [Bathe14]. The Jacobian $\boldsymbol{J}$ can be assembled using the partial derivative of the NURBS solid, see Eq. (2.16). For a 3D model, the derivatives of the NURBS basis can be arranged in the strain displacement matrix as

$$\boldsymbol{B} = \begin{bmatrix} \frac{\partial R^{\mathrm{p,q,r}}_{1,1,1}}{\partial x} & 0 & 0 & \frac{\partial R^{\mathrm{p,q,r}}_{1,1,2}}{\partial x} & \dots & 0 \\ 0 & \frac{\partial R^{\mathrm{p,q,r}}_{1,1,1}}{\partial y} & 0 & 0 & \dots & 0 \\ 0 & 0 & \frac{\partial R^{\mathrm{p,q,r}}_{1,1,1}}{\partial z} & 0 & \dots & \frac{\partial R^{\mathrm{p,q,r}}_{\mathrm{p+1,q+1,r+1}}}{\partial z} \\ \frac{\partial R^{\mathrm{p,q,r}}_{1,1,1}}{\partial y} & \frac{\partial R^{\mathrm{p,q,r}}_{1,1,1}}{\partial x} & 0 & \frac{\partial R^{\mathrm{p,q,r}}_{1,1,2}}{\partial y} & \dots & 0 \\ 0 & \frac{\partial R^{\mathrm{p,q,r}}_{1,1,1}}{\partial z} & \frac{\partial R^{\mathrm{p,q,r}}_{1,1,1}}{\partial y} & 0 & \dots & \frac{\partial R^{\mathrm{p,q,r}}_{\mathrm{p+1,q+1,r+1}}}{\partial y} \\ \frac{\partial R^{\mathrm{p,q,r}}_{1,1,1}}{\partial z} & 0 & \frac{\partial R^{\mathrm{p,q,r}}_{1,1,1}}{\partial x} & \frac{\partial R^{\mathrm{p,q,r}}_{1,1,2}}{\partial z} & \dots & \frac{\partial R^{\mathrm{p,q,r}}_{\mathrm{p+1,q+1,r+1}}}{\partial x} \end{bmatrix}. \tag{2.33}$$

For the strain displacement matrix of a 2D model with plane strain, plane stress, or axisymmetric behavior, see [Bathe14]. The integration over each element $\Omega_{\mathrm{e},i}$ is performed by Gauss quadrature in parameter space. The basis functions $\boldsymbol{N}$ and the strain displacement matrix $\boldsymbol{B}$ are evaluated at each Gauss point. As stated in [CottrellEtAl09], the same Gauss rule for a polynomial of the order p can be applied to a p-th order B-spline. Therefore, the Gauss order $p+1$, $q+1$, and $r+1$ is chosen in ξ-, η-, and ζ-direction. The global system matrices $\boldsymbol{K}_{\mathrm{e}}$ and $\boldsymbol{M}_{\mathrm{e}}$ are assembled from the corresponding element matrices $\boldsymbol{K}_{\mathrm{e},i}$ and $\boldsymbol{M}_{\mathrm{e},i}$. The equations of motion of the complete finite element model consisting of n_{e} elements are then given by

$$\boldsymbol{M}_{\mathrm{e}} \ddot{\boldsymbol{u}}_{\mathrm{e}} + \boldsymbol{K}_{\mathrm{e}} \boldsymbol{u}_{\mathrm{e}} = \boldsymbol{f}_{\mathrm{e}}, \tag{2.34}$$

where the displacements of the control points are represented by $\boldsymbol{u}_\mathrm{e}$ and external forces by $\boldsymbol{f}_\mathrm{e}$.

2.7 Model Order Reduction

For the incorporation of the isogeometric model into the equations of motion (2.5) of a flexible multibody system, the global shape functions $\boldsymbol{\Phi}$ are required. A simple and straightforward approach for reducing the full finite element model (2.34) is modal truncation. However, the low frequency eigenmodes typically are not able to precisely describe local deformation in the contact area. This leads to inaccurate results, as shown in [Tschigg20]. Alternatively, the Craig-Bampton method [CraigBampton68] can be used. The Craig-Bampton method combines fixed-interface normal modes and constraint modes. The normal modes represent the overall flexibility and the constraint modes approximate the deformation in a specific area, e.g. the contact area. Selecting predetermined control points on the exterior surface is necessary for the constrained modes. Following this procedure generates global shape functions denoted as $\boldsymbol{\Phi}$, which are orthogonalized and normalized to the mass matrix. The reduced mass and stiffness matrix are then given by

$$\overline{\boldsymbol{M}}_\mathrm{e} = \boldsymbol{\Phi}^\mathsf{T} \boldsymbol{M}_\mathrm{e} \boldsymbol{\Phi} = \boldsymbol{E} \tag{2.35}$$

$$\overline{\boldsymbol{K}}_\mathrm{e} = \boldsymbol{\Phi}^\mathsf{T} \boldsymbol{K}_\mathrm{e} \boldsymbol{\Phi} = \mathrm{diag}(\omega_i^2), \tag{2.36}$$

respectively, where ω_i are the eigenfrequencies. The elastic deformation $\boldsymbol{u}_\mathrm{e}$ can be easily recovered with

$$\boldsymbol{u}_\mathrm{e} = \boldsymbol{\Phi} \boldsymbol{q}_\mathrm{e}, \tag{2.37}$$

where $\boldsymbol{q}_\mathrm{e}$ are the elastic coordinates.

A key issue of the Craig-Bampton method is the numerical stiffness of the equations of motion (2.5). This is due to low frequency normal modes and the high frequency constrained modes. A simple method to improve the numerical performance is to modally damp the high frequency modes. This method has been already applied in IGA impact simulations [RückwaldEtAl21b, RückwaldEtAl21c, RückwaldEtAl21d, RückwaldEtAl22, RückwaldEtAl23]. However, it is still numerically burdensome and the choice of the damping parameters critical.

2.8 Stress Recovery

For component design and optimization, the occurring stresses are required. These can be extracted in post-processing of flexible multibody simulations. Since linear elasticity is assumed and the weak Galerkin method is applied as for isoparametric elements, the same formula can be used in the IGA to compute the occurring stresses $\boldsymbol{\sigma}$. Therefore, the stresses at a point (ξ, η, ζ) in the parameter space are computed with

$$\boldsymbol{\sigma}(\xi,\eta,\zeta) = \begin{bmatrix}\sigma_\mathrm{x} & \sigma_\mathrm{y} & \sigma_\mathrm{z} & \tau_\mathrm{xy} & \tau_\mathrm{xz} & \tau_\mathrm{yz}\end{bmatrix}^\mathsf{T} = \boldsymbol{CB}(\xi,\eta,\zeta)\boldsymbol{u}_\mathrm{e}, \tag{2.38}$$

where σ_x, σ_y, and σ_z are the normal stresses, and τ_xy, τ_xz, and τ_yz are the shear stresses. As a reminder, the current knot span is applied to the strain displacement matrix $\boldsymbol{B}$ and the displacements $\boldsymbol{u}$ representing only the active B-splines. Using the stresses $\boldsymbol{\sigma}$, the von Mises stress σ_mises can be computed with

$$\sigma_\mathrm{mises} = \sqrt{\sigma_\mathrm{x}^2 + \sigma_\mathrm{y}^2 + \sigma_\mathrm{z}^2 - \sigma_\mathrm{x}\sigma_\mathrm{y} - \sigma_\mathrm{x}\sigma_\mathrm{z} - \sigma_\mathrm{y}\sigma_\mathrm{z} + 3(\tau_\mathrm{xy}^2 + \tau_\mathrm{xz}^2 + \tau_\mathrm{yz}^2)}. \tag{2.39}$$

Since, the equations of motions are reduced in Sect. 2.7, the displacements $\boldsymbol{u}$ are not directly available. Therefore, the stresses

$$\boldsymbol{\sigma}(\xi,\eta,\zeta) = \boldsymbol{CB}(\xi,\eta,\zeta)\boldsymbol{\Phi}(\xi,\eta,\zeta)\boldsymbol{q}_\mathrm{e} \tag{2.40}$$

can be computed using Eq. (2.37). For an efficient stress recovery, stress modes are proposed in [TobiasEberhard11] for flexible bodies based on the FEM. This concept can be adapted to the IGA. The IGA stress modes $\boldsymbol{\Phi}_\sigma$ can be pre-evaluated at specific points in the parameter space with

$$\boldsymbol{\Phi}_\sigma(\xi,\eta,\zeta) = \boldsymbol{CB}(\xi,\eta,\zeta)\boldsymbol{\Phi}(\xi,\eta,\zeta) \tag{2.41}$$

and stored, e.g. in the SID file. Then, the computation of

$$\boldsymbol{\sigma}(\xi,\eta,\zeta) = \boldsymbol{\Phi}_\sigma(\xi,\eta,\zeta)\boldsymbol{q}_\mathrm{e} \tag{2.42}$$

allows an efficient recovery of the stresses. The stress modes are evaluated at the edges of the elements and at the Gaussian points of the elements. This concept has not yet been published in context with the IGA. However, it is used in this work for visualization of stresses in the contact region.

Contact Handling in IGA

In this work, contacts in flexible multibody systems are simulated efficiently but still accurately. The focus is on short frictionless normal contact between flexible bodies. In this case, the term "contact" can be referred to as "impact". For detailed theory on frictionless normal contact in FEM, see for example [Eberhard00, Wriggers06]. The literature is briefly summarized below.

Usually, in contact mechanics, the contact between two bodies is considered. One body is named contact body and the other target body, see Fig. 3.1. In literature, the contact body is also sometimes referred to as the "slave", and the target body as the "master". The two bodies must not penetrate each other at any time. This condition is known as the impenetrability condition. The distance between contact and target body is measured by the normal gap g_n. As visualized in Fig. 3.1, the normal gap is defined between a predetermined evaluation point on the contact body's exterior surface, which is colored red, and the green colored exterior surface of the target body. To comply with the impenetrability condition, $g_\mathrm{n} \geq 0$ must hold at every point. If the normal gap is greater than zero, there is no contact. If the normal gap g_n is equal to zero, a contact occurs. An overlapping of the bodies represents a non-physical penetration.

In case of contact, a pressure force $f \leq 0$ is applied to maintain the impenetrability condition $g_\mathrm{n} \geq 0$. Combining the condition $f \leq 0$ on the contact force and the normal gap $g_\mathrm{n} \geq 0$ results in the linear complementarity condition

$$g_\mathrm{n} \geq 0, \;\; f \leq 0, \;\; \text{and} \;\; g_\mathrm{n} f = 0. \tag{3.1}$$

This statement is also known as the Hertz–Signorini–Moreau conditions or Karush–Kuhn–Tucker conditions [Wriggers06]. In summary, a contact force occurs

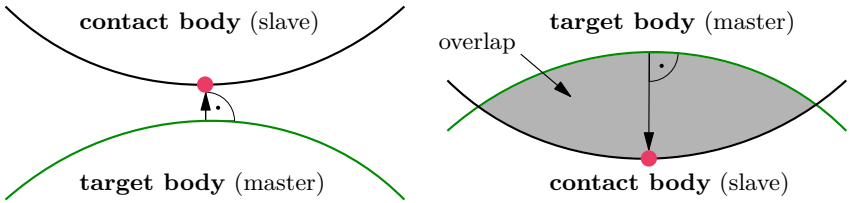


Figure 3.1: Contact between the contact and the target body.

only if a body is in contact, and a contact cannot cause a tensile force. In practice, two different methods are usually used to meet the Hertz–Signorini–Moreau conditions in Eq. (3.1). These are the Lagrange multiplier method and the penalty method [Wriggers06].

With the Lagrange multiplier method, the Hertz-Signorini-Moreau conditions are maintained almost exactly, as no penetration occurs. Algebraic constraints are set up between contact and target body to prevent the bodies from penetrating. These constraints are usually set at position level, but can also be set at velocity or acceleration level. However, unphysical penetrations may occur in the latter two cases. The Jacobian of the algebraic constraints is determined, it is multiplied by Lagrange multipliers, and introduced into the discrete forces $\boldsymbol{h}_\mathrm{d}$ of the equations of motion (2.5). If the constraints are defined on the position level, the equations of motions (2.5) are a system of differential-algebraic equations of index three. Although no penetrations occur with the Lagrange multiplier method, this method is numerically inefficient [TamarozziEtAl13]. The reason for this is the high number of variables, e.g. the additional Lagrange multipliers, and the changing structure of the resulting system of equations of motion [SimoLaursen92].

The concept of the penalty formulation is to penalize penetration with a spring force. The penalty method enforces the conditions in Eq. (3.1) by penalizing penetration. Therefore, the conditions in Eq. (3.1) are violated. However, only minor penetrations occur in practical applications. The corresponding spring constant is named the penalty factor c_p. It is a tuning factor and needs to be chosen heuristically. Thereby, the penalty factor should be chosen large enough such that the results become independent of the chosen parameter [SeifriedEtAl03]. If the penalty factor is increased beyond its converging value, the equations of motion (2.5) become numerically stiffer. This increases the computation time, or the numerical integration might even terminate unsuccessfully. If a contact occurs, the contact force $\boldsymbol{f}$ is determined with

$$\boldsymbol{f} = c_\mathrm{p} g_\mathrm{n} \boldsymbol{n}, \tag{3.2}$$

where $\boldsymbol{n}$ is a normal vector. The determination of the normal vector $\boldsymbol{n}$ is detailed in Sect. 3.4. Although small penetrations occur with the penalty method, it is used in this work. The reason for this is the simple implementation and the fact that there are no additional variables, as the Lagrange multipliers of the Lagrange Multiplier method. Additionally, the goal of this work is to continue the previous work [Tschigg20] on the quasistatic contact model for isoparametric models, which is also introduced in this chapter.

The chapter begins in Sect. 3.1 with the contact law by Hertz [Johnson04]. It will be used for validation in this work. As shown in Fig. 3.1, the normal gap is measured up to a predetermined evaluation point on the contact body's exterior surface. In Sect. 3.2, a selection of methods to determine evaluation points for discretizing IGA surfaces is described. The procedure for evaluating contacts in

the course of the floating frame of reference formulation is presented in Sect. 3.3. The determination of the normal vector $\boldsymbol{n}$ in the IGA is detailed in Sect. 3.4, as it is required for the penalty method, see Eq. (3.2). To improve efficiency of contact simulations in flexible multibody systems, the quasistatic contact model is introduced in Sect. 3.5. The quasistatic contact model will also be validated on the basis of energy conservation. Therefore, the energy of a flexible multibody system including quasistatic contact models is derived in Sect. 3.6.

3.1 Hertz Contact for Reference

The methods presented in this work are tested in simulations and need to be verified. Besides the numerical results of commercial FEM software such as Ansys, analytic contact formulations are used. In rigid body systems, it is distinguished between discontinuous and continuous contact modeling [Seifried05]. With a discontinuous model, the contact is infinitesimally short. In a continuous model, the contact occurs for a short but finite time and can also be applied for flexible systems. Since the contact force is to be validated in IGA simulations, a continuous model is required.

One of the best known continuous contact laws is that of Hertz [Hertz82, Johnson04]. The contact law by Hertz is derived from continuum mechanical consideration of the elastostatic contact of two spheres. Thus it conserves the energy, i.e. there is no damping and the coefficient of restitution is one. The Hertzian contact law also allows the computation of the contact pressure and stress distribution in the contact area. As a side note, based on the contact law by Hertz, the contact laws by Lankarani and Nikravesh [LankaraniNikravesh90], and Gonthier [GonthierEtAl04] are formulated. These contact laws include damping, whereby the coefficient of restitution is less than one. However, in contrast to the Hertzian contact, these laws do not allow a detailed impact analysis concerning occurring stresses. Since damping is not in the scope of this work, the contact law by Hertz is used for validation in this work.

A frequently used benchmark example of impact simulations with the FEM is the impact of two spheres. In this work, this example is also used to benchmark the IGA. The setup is sketched in Fig. 3.2 and is analytically solved by Hertz [Johnson04]. According to Hertz, the spheres are modeled as mass points. During the impact, the contact force

$$f_{\text{Hertz}} = \frac{4}{3} E^* \sqrt{r^*} \delta^{3/2} \tag{3.3}$$

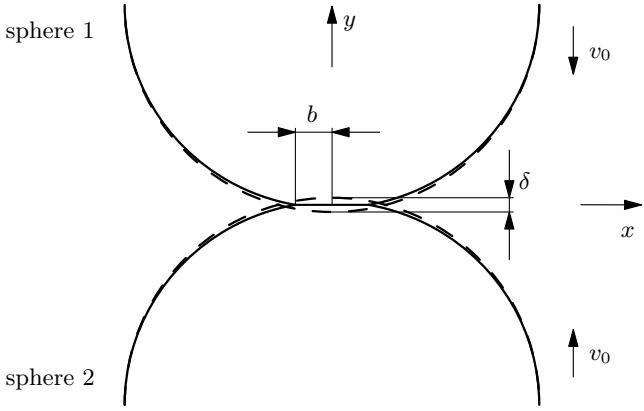


Figure 3.2: Impact of two spheres modeled with Hertz.

acts between the spheres, where δ denotes the displacement of the center of mass of the spheres. This corresponds to the overlap of the undeformed spheres, see Fig. 3.2. The equivalent Young's modulus E^* and the equivalent radius r^* are determined by

$$\frac{1}{E^*} = \frac{1-\nu_1^2}{E_1} + \frac{1-\nu_2^2}{E_2} \tag{3.4}$$

$$\frac{1}{r^*} = \frac{1}{r_1} + \frac{1}{r_2}. \tag{3.5}$$

The Young's modulus of the two spheres is denoted by E_1 and E_2, the Poisson's ratio by ν_1 and ν_2, the radius by r_1 and r_2. The contact width a and the maximum pressure p_0 are given by

$$b = \sqrt{r\delta} \tag{3.6}$$

$$p_0 = \frac{3f_{\text{Hertz}}}{2\pi b^2}. \tag{3.7}$$

Thus, the von Mises stress along the symmetry axis can be calculated as

$$\sigma_{\text{Hertz}}(y) = p_0 \left| \frac{3}{2}\frac{b^2}{b^2+y^2} - (1+\nu)\left(1 - \frac{y}{b}\arctan\left(\frac{b}{y}\right)\right)\right|. \tag{3.8}$$

3.2 Contact Discretization

To apply the penalty method, the contact law from Eq. (3.2) is applied at a selection of predetermined evaluation points. These points represent a discretization of the exterior contact surface of the body. For isoparametric elements, usually, the nodes of the elements are used [Wriggers06, Tschigg20]. Here, the degrees of freedom, the nodes, are also the boundaries of the isoparametric elements. This is not the case in the IGA. The degrees of freedom in the IGA, the control points, span up the geometry but are not physical points on it, see Fig. 2.2. The knots are on the geometry and are the boundaries of IGA elements. However, the knots do not represent the degrees of freedom. This section discusses a selection of different methods to to discretize the exterior surface of IGA bodies in contact.

The example of the refined semicircle shown in Fig. 2.8 is again revisited in Fig. 3.3 to demonstrate the contact surface. To determine the exterior surface of an IGA body, the parameter space is considered. For the example in Fig. 3.3, the exterior contact surface is defined by the constant knot coordinate η. The lower exterior surface colored in red is defined by $\eta = 0$, and the upper exterior surface in green by $\eta = 1$. The contact force is evaluated by integration via the remaining knot coordinate, in this example ξ. For a 3D model, one knot coordinate is constant and two knot coordinates are variable. The contact evaluation will be detailed in Sect. 3.3. Discrete evaluation points are now determined for the coordinates over which the contact is integrated.

In [TemizerEtAl12] four methods for the discretization of the surface are presented: Gauss points, unique knots, collocation points, and a Mortar method. The Mortar

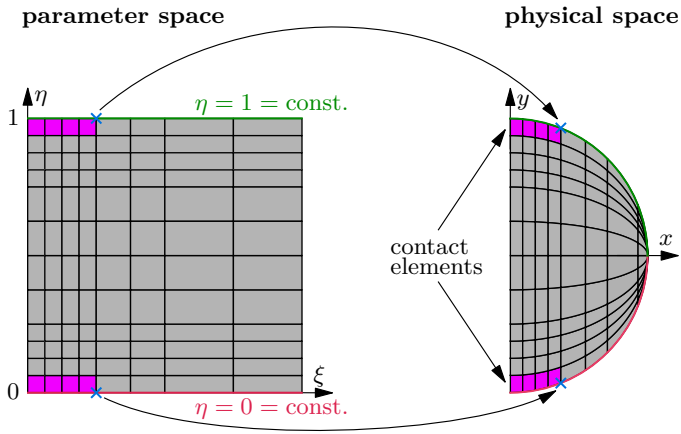
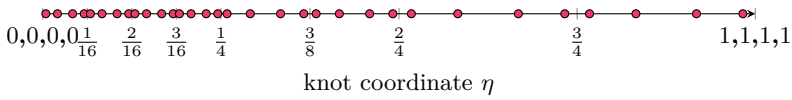
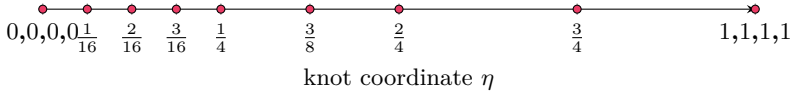


Figure 3.3: Exterior contact surface of the axisymmetric sphere.

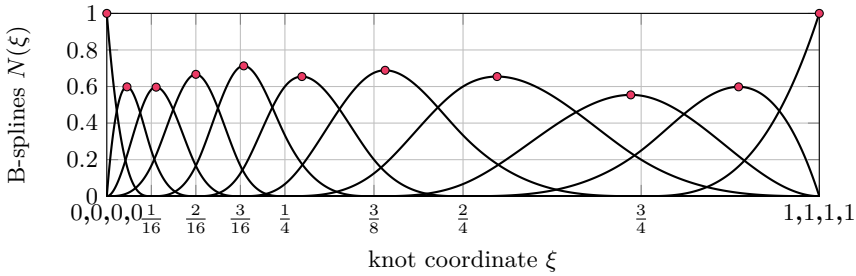
Method is not discussed in this work. The remaining three methods are presented in Fig. 3.4 using the refined semicircle in Fig. 2.8. Its knot vectors are given by Eq. (2.23).



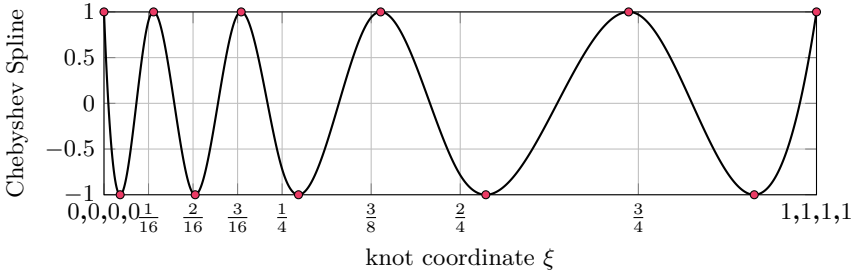
a) Gauss points.



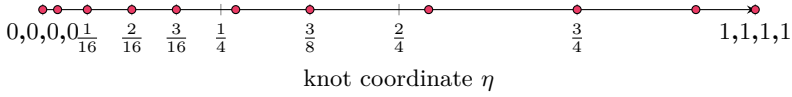
b) Unique knots.



c) Collocation: Botella points are the maxima of B-splines.



d) Collocation: Chebyshev points are the extrema of a Chebyshev spline.



e) Collocation: Greville points, see Eq. (3.9).

Figure 3.4: Methods for contact discretization.

Gauss Points

The straightforward approach is to use Gauss points as they are used to determine the element mass and stiffness matrices in Sect. 2.6. Unlike in Sect. 2.6, the Gauss points are not arranged over an entire element but only on its outer surface. This concept is called "knot-to-surface" algorithm [TemizerEtAl11, TemizerEtAl12]. However, it is also named "Gauss-point-to-segment" approach in the literature, see e.g. [MatzenBischoff16]. As stated in [CottrellEtAl09], the same Gauss rule for a polynomial of the order p can be applied to a p-th order B-spline. Therefore, each element has $p+1$ points where the contact is evaluated. Since the order of the refined semicircle in Fig. 2.8 is $p=3$, four Gauss points are placed within each element. It can be easily seen in Fig. 3.4a, that the number of evaluation points, in this case the Gauss points, is higher than the number of control points $n=11$, see Eq. (2.23). Since there are more evaluation points than control points, the contact formulation is overconstrained. This also effects the computation time, which is increased.

Unique Knots

Usually, when applying the penalty method to bodies modeled by isoparametric elements, the "node-to-surface" algorithm is used. This algorithm is also applied in the previous work [Tschigg20]. Thereby, the contact is enforced at the position of the degrees of freedom of the FEM model, the exterior nodes of the elements. This concept cannot be applied to the degrees of freedom of the IGA, which are the control points. The control points span up the splines but do not directly limit the geometry of the elements. Instead, the knots limit the elements. Therefore, [TemizerEtAl12] proposes to check the contact at the position of the unique knots as shown in Fig. 3.4b. In this way, the classic node-to-surface algorithm can be transferred from FEM to IGA. However, when applying this algorithm to isoparametric elements, the number of evaluation points is identical to the number of nodes. In the IGA, the number of unique knots is less than the number of exterior control points. As visualized in Fig. 2.10 and defined in Eq. (2.23), the number of B-splines and control points in ξ-direction is $n=11$. However, the knot vector $\boldsymbol{\Xi}$ only has nine unique knots. Therefore, this contact formulation is underconstrained and will not be discussed further in this work.

Collocation Points

A method that comes close to the "node-to-surface" method in the FEM and is neither overdetermined nor underdetermined is collocation [TemizerEtAl12, Matzen15, MatzenBischoff16]. Here, the number of evaluation points is identical to the number of control points on the exterior surface. This method is also known as the "point-to-segment" method, see [MatzenBischoff16]. In [MatzenBischoff16], three collocation methods are presented:

Botella Points
First, Botella points, see [Botella02], are suggested. Botella points are defined at the maximum points of the B-splines, see Fig. 3.4c. There is no direct calculation rule for the maximum points in the IGA. However, by setting the first derivative of the B-splines from Eq. (2.7) to zero and solving the equations using Newton's method, the Botella points can be determined. Since Newton's method is used, the second derivative of the B-splines from Eq. (2.8) is also required.

Chebyshev Points
Second, Chebyshev points are proposed as collocation points in [MatzenBischoff16]. A Chebyshev spline of order $p+1$ is shown Fig. 3.4d. Chebyshev points are extrema of the Chebyshev spline, see [Demko85]. The extrema and the corresponding spline can be computed with the MATLAB function *chbpnt*, see [Mathworks23]. A non-unique knot vector, e.g. $\boldsymbol{\Xi}$, and its corresponding order increased by one, e.g. $p+1$, are selected as the function input.

Greville Points
Third, the last collocation method proposed in [MatzenBischoff16] are Greville points. The Greville points $\hat{\xi}_i$ can be computed with

$$\hat{\xi}_i = \frac{\sum_{\hat{i}=i+1}^{\mathrm{i+p}} \xi_{\hat{i}}}{p} \qquad i = 1...n \tag{3.9}$$

resulting in n collocation points, see [Farin01]. As an example, the Greville points are shown in Fig. 3.4e. The collocation points in η- and ζ-direction are computed accordingly.

The collocation points listed above are valid for one local coordinate direction. Keeping one local coordinate constant and integrating with the remaining local coordinate corresponds to a 2D model. For a 3D model, two local coordinate directions are varied and the third is constant. In the following, it is assumed that the contact is integrated over ξ and η, and the third local coordinate ζ is kept constant at ζ_{con}. The collocation points are given by

$$\hat{\boldsymbol{\xi}} = \begin{bmatrix} \hat{\xi}_1 & \cdots \hat{\xi}_{\mathrm{n}} \end{bmatrix} \tag{3.10}$$

$$\hat{\boldsymbol{\eta}} = \begin{bmatrix} \hat{\eta}_1 & \cdots \hat{\eta}_{\mathrm{m}} \end{bmatrix}. \tag{3.11}$$

The linear combinations

$$\begin{bmatrix} \hat{\xi}_1 \\ \hat{\eta}_1 \\ \zeta_{\mathrm{con}} \end{bmatrix}, \begin{bmatrix} \hat{\xi}_1 \\ \hat{\eta}_2 \\ \zeta_{\mathrm{con}} \end{bmatrix}, \cdots, \begin{bmatrix} \hat{\xi}_1 \\ \hat{\eta}_{\mathrm{m}} \\ \zeta_{\mathrm{con}} \end{bmatrix}, \begin{bmatrix} \hat{\xi}_2 \\ \hat{\eta}_1 \\ \zeta_{\mathrm{con}} \end{bmatrix}, \cdots, \begin{bmatrix} \hat{\xi}_2 \\ \hat{\eta}_{\mathrm{m}} \\ \zeta_{\mathrm{con}} \end{bmatrix}, \cdots, \begin{bmatrix} \hat{\xi}_{\mathrm{n}} \\ \hat{\eta}_{\mathrm{m}} \\ \zeta_{\mathrm{con}} \end{bmatrix} \tag{3.12}$$

of the collocation points in Eq. (3.10) and Eq. (3.11) then results in $n_{\mathrm{points}}^{\mathrm{eval}} = n \cdot m$ contact evaluation points.

Besides the position of the collocation points, each collocation point is weighted. Weighting is required so that each point is in relation to the size of the exterior surface. This is already the case with the Gauss points, as each Gauss point has its own weight. For collocation points, the weighted collocation method is introduced in [MatzenBischoff16]. The collocation weights $\hat{\boldsymbol{w}} = \begin{bmatrix}\hat{w}_1 & \hat{w}_2 & \dots & \hat{w}_{n\cdot m}\end{bmatrix}^\intercal$ can be determined by solving

$$\boldsymbol{G}\hat{\boldsymbol{w}} = \boldsymbol{r}, \tag{3.13}$$

where $\boldsymbol{G}$ are the NURBS basis arranged in a matrix and $\boldsymbol{r}$ are the integrated NURBS basis. See Eq. (2.11) for the definition of the NURBS basis, thus the matrix $\boldsymbol{G}$ reads

$$\boldsymbol{G} = \begin{bmatrix} R^{\mathrm{p,q,r}}_{1,1,1}(\hat{\xi}_1, \hat{\eta}_1, \zeta_{\mathrm{con}}) & R^{\mathrm{p,q,r}}_{1,1,1}(\hat{\xi}_1, \hat{\eta}_2, \zeta_{\mathrm{con}}) & \cdots & R^{\mathrm{p,q,r}}_{1,1,1}(\hat{\xi}_{\mathrm{n}}, \hat{\eta}_{\mathrm{m}}, \zeta_{\mathrm{con}}) \\ R^{\mathrm{p,q,r}}_{1,1,2}(\hat{\xi}_1, \hat{\eta}_1, \zeta_{\mathrm{con}}) & R^{\mathrm{p,q,r}}_{1,1,2}(\hat{\xi}_1, \hat{\eta}_2, \zeta_{\mathrm{con}}) & \cdots & R^{\mathrm{p,q,r}}_{1,1,2}(\hat{\xi}_{\mathrm{n}}, \hat{\eta}_{\mathrm{m}}, \zeta_{\mathrm{con}}) \\ \vdots & \vdots & \ddots & \vdots \\ R^{\mathrm{p,q,r}}_{\mathrm{n,m},\ell}(\hat{\xi}_1, \hat{\eta}_1, \zeta_{\mathrm{con}}) & R^{\mathrm{p,q,r}}_{\mathrm{n,m},\ell}(\hat{\xi}_1, \hat{\eta}_2, \zeta_{\mathrm{con}}) & \cdots & R^{\mathrm{p,q,r}}_{\mathrm{n,m},\ell}(\hat{\xi}_{\mathrm{n}}, \hat{\eta}_{\mathrm{m}}, \zeta_{\mathrm{con}}) \end{bmatrix}, \tag{3.14}$$

where the NURBS basis is evaluated at all linear combinations of the collocation points, see Eq. (3.12). The right side of Eq. (3.13) is computed by integrating the NURBS basis on the exterior surface with

$$\boldsymbol{r} = \int\int \begin{bmatrix} R^{\mathrm{p,q,r}}_{1,1,1}(\xi, \eta, \zeta_{\mathrm{con}}) \\ R^{\mathrm{p,q,r}}_{1,2,1}(\xi, \eta, \zeta_{\mathrm{con}}) \\ \vdots \\ R^{\mathrm{p,q,r}}_{\mathrm{n,m},1}(\xi, \eta, \zeta_{\mathrm{con}}) \end{bmatrix} \mathrm{d}\xi\mathrm{d}\eta. \tag{3.15}$$

By solving Eq. (3.13), the collocation weights $\hat{\boldsymbol{w}}$ are obtained. As can be seen in Fig. 3.3, not all evaluation points that are on the outer surface are checked for contact. Only the points located in the contact elements are required. The contact elements, which are colored purple in Fig. 3.3, are located in the so-called contact interval. The contact interval is defined in the IGA parameter space and indicates where the contact is checked on the contact and the target body.

3.3 Contact Evaluation

This section details the contact evaluation of IGA bodies that are modeled within the floating frame of reference formulation, see Sect. 2.1. In the floating frame of

reference formulation, each body is represented in its body frame $\mathrm{K_R}$, see Eq. (2.4). However, for contact evaluation, it is desired that the bodies are represented in the inertial frame $\mathrm{K_I}$. Therefore, the position of the deformed control points ${}^{\mathrm{I}}\boldsymbol{p}_{i,j,k}$ in the inertial frame is required. As a reminder, the indices i, j, and k correspond to the arrangement of the control points in the control net, see Eq. (2.9). The positions of the deformed control points are composed of the undeformed control points ${}^{\mathrm{R}}\boldsymbol{p}^0_{i,j,k}$ and the elastic deformation $\boldsymbol{u}_{\mathrm{e},i,j,k}$ based on the global shape functions $\boldsymbol{\Phi}_{i,j,k}$ and the elastic coordinates $\boldsymbol{q}_{\mathrm{e}}$ as

$$ {}^{\mathrm{R}}\boldsymbol{p}_{i,j,k} = {}^{\mathrm{R}}\boldsymbol{p}^0_{i,j,k} + \boldsymbol{u}_{\mathrm{e},i,j,k} = {}^{\mathrm{R}}\boldsymbol{p}^0_{i,j,k} + \boldsymbol{\Phi}_{i,j,k}\boldsymbol{q}_{\mathrm{e}}. \tag{3.16} $$

The absolute position of the control points is given by

$$ {}^{\mathrm{I}}\boldsymbol{p}_{i,j,k} = \boldsymbol{S}_{\mathrm{IR}}(\boldsymbol{\beta}_{\mathrm{IR}}) \left({}^{\mathrm{R}}\boldsymbol{r}_{\mathrm{IR}} + {}^{\mathrm{R}}\boldsymbol{p}_{i,j,k} \right) \tag{3.17} $$

requiring the reference frame position ${}^{\mathrm{R}}\boldsymbol{r}_{\mathrm{IR}}$, the rotational parameters $\boldsymbol{\beta}_{\mathrm{IR}}$, and the corresponding rotation matrix $\boldsymbol{S}_{\mathrm{IR}}(\boldsymbol{\beta}_{\mathrm{IR}})$. In the course of contact evaluation, one body is defined as the contact body denoted by the index C and the other as the target body denoted by T, see Fig. 3.5. The contact evaluation points $\boldsymbol{x}_{\mathrm{C},i}$, e.g. collocation or Gauss points, are located on the exterior surface of the contact body within the contact interval and are tested for contact with the exterior surface of the target body. As an example, the contact force is determined by integrating a surface with the local coordinates ξ_{T} and η_{T}, keeping ζ_{T} constant. The contact of two bodies is checked by solving the nonlinear equation

$$ \begin{aligned} \boldsymbol{f}_{\mathrm{con}}(\xi_{\mathrm{T}}, \eta_{\mathrm{T}}) &= \left[\begin{pmatrix} \frac{\partial \boldsymbol{x}_{\mathrm{T}}(\xi_{\mathrm{T}}, \eta_{\mathrm{T}}, \zeta_{\mathrm{T}})}{\partial \xi_{\mathrm{T}}} \end{pmatrix}^{\mathsf{T}} \begin{pmatrix} \frac{\partial \boldsymbol{x}_{\mathrm{T}}(\xi_{\mathrm{T}}, \eta_{\mathrm{T}}, \zeta_{\mathrm{T}})}{\partial \eta_{\mathrm{T}}} \end{pmatrix}^{\mathsf{T}} \right] \left(\boldsymbol{x}_{\mathrm{C},i}(\hat{\xi}_{\mathrm{C},i}, \hat{\eta}_{\mathrm{C},i}, \zeta_{\mathrm{con}}) - \boldsymbol{x}_{\mathrm{T}}(\xi_{\mathrm{T}}, \eta_{\mathrm{T}}, \zeta_{\mathrm{T}}) \right) \\ &= \boldsymbol{0} \end{aligned} \tag{3.18} $$

with the Newton's method

$$ \begin{bmatrix} \xi_{\mathrm{T}}^{\mathrm{new}} \\ \eta_{\mathrm{T}}^{\mathrm{new}} \end{bmatrix} = \begin{bmatrix} \xi_{\mathrm{T}}^{\mathrm{pre}} \\ \eta_{\mathrm{T}}^{\mathrm{pre}} \end{bmatrix} (\boldsymbol{J}_{\mathrm{con}})^{-1} \boldsymbol{f}_{\mathrm{con}}(\xi_{\mathrm{T}}^{\mathrm{pre}}, \eta_{\mathrm{T}}^{\mathrm{pre}}) \tag{3.19} $$

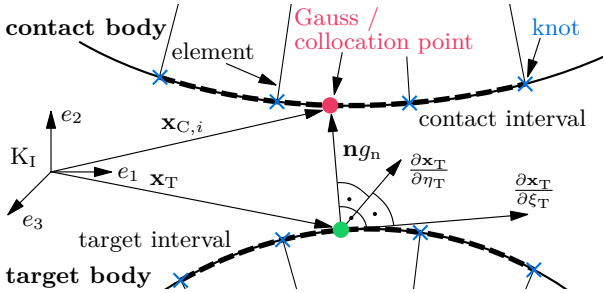


Figure 3.5: Contact detection between the contact and target body.

for the respective knot coordinates. Using Newton's method, the new iterative solution $\begin{bmatrix}\xi_\mathrm{T}^\mathrm{new} & \eta_\mathrm{T}^\mathrm{new}\end{bmatrix}^\intercal$ is updated with the previous solution $\begin{bmatrix}\xi_\mathrm{T}^\mathrm{pre} & \eta_\mathrm{T}^\mathrm{pre}\end{bmatrix}^\intercal$. The Jacobian $\boldsymbol{J}_\mathrm{con}$ of Eq. (3.18) reads

$$\boldsymbol{J}_\mathrm{con} = \begin{bmatrix} \left(\frac{\partial^2 \boldsymbol{x}_\mathrm{T}}{\partial \xi_\mathrm{T}^2}\right)^\intercal (\boldsymbol{x}_{\mathrm{C},i} - \boldsymbol{x}_\mathrm{T}) - \left(\frac{\partial \boldsymbol{x}_\mathrm{T}}{\partial \xi_\mathrm{T}}\right)^\intercal \frac{\partial \boldsymbol{x}_\mathrm{T}}{\partial \xi_\mathrm{T}} \\ \left(\frac{\partial^2 \boldsymbol{x}_\mathrm{T}}{\partial \xi_\mathrm{T} \partial \eta_\mathrm{T}}\right)^\intercal (\boldsymbol{x}_{\mathrm{C},i} - \boldsymbol{x}_\mathrm{T}) - \left(\frac{\partial \boldsymbol{x}_\mathrm{T}}{\partial \xi_\mathrm{T}}\right)^\intercal \frac{\partial \boldsymbol{x}_\mathrm{T}}{\partial \eta_\mathrm{T}} \end{bmatrix} \begin{bmatrix} \left(\frac{\partial^2 \boldsymbol{x}_\mathrm{T}}{\partial \xi_\mathrm{T} \partial \eta_\mathrm{T}}\right)^\intercal (\boldsymbol{x}_{\mathrm{C},i} - \boldsymbol{x}_\mathrm{T}) - \left(\frac{\partial \boldsymbol{x}_\mathrm{T}}{\partial \xi_\mathrm{T}}\right)^\intercal \frac{\partial \boldsymbol{x}_\mathrm{T}}{\partial \eta_\mathrm{T}} \\ \left(\frac{\partial^2 \boldsymbol{x}_\mathrm{T}}{\partial \eta_\mathrm{T}^2}\right)^\intercal (\boldsymbol{x}_{\mathrm{C},i} - \boldsymbol{x}_\mathrm{T}) - \left(\frac{\partial \boldsymbol{x}_\mathrm{T}}{\partial \eta_\mathrm{T}}\right)^\intercal \frac{\partial \boldsymbol{x}_\mathrm{T}}{\partial \eta_\mathrm{T}} \end{bmatrix}. \tag{3.20}$$

By solving Eq. (3.18) for the knot coordinates ξ_T and η_T, the target point $\boldsymbol{x}_\mathrm{T}$ closest to the current evaluation point $\boldsymbol{x}_{\mathrm{C},i}$ is found. The distance g_n between the contact and target point is determined by

$$g_\mathrm{n} = \boldsymbol{n}^\intercal (\boldsymbol{x}_{\mathrm{C},i}(\hat{\xi}_{\mathrm{C},i}, \hat{\eta}_{\mathrm{C},i}, \zeta_\mathrm{con}) - \boldsymbol{x}_\mathrm{T}(\xi_\mathrm{T}, \eta_\mathrm{T}, \zeta_\mathrm{T})), \tag{3.21}$$

where the normal vector $\boldsymbol{n}$ is orthogonal to the surface of the target body. A positive normal gap g_n indicates a non-active contact. If the normal gap g_n is negative, the contact forces $\boldsymbol{f}_{\mathrm{C},i}$ and $\boldsymbol{f}_{\mathrm{T},i}$ of the contact and target body for the current collocation point are determined by the penalty approach as

$$\boldsymbol{f}_{\mathrm{C},i} = -c_\mathrm{p} g_\mathrm{n} \boldsymbol{N}_\mathrm{C}^\intercal \boldsymbol{n} \hat{w}_i \det(\boldsymbol{J}) \tag{3.22}$$

$$\boldsymbol{f}_{\mathrm{T},i} = +c_\mathrm{p} g_\mathrm{n} \boldsymbol{N}_\mathrm{T}^\intercal \boldsymbol{n} \hat{w}_i \det(\boldsymbol{J}). \tag{3.23}$$

Here, c_p is the penalty factor, $\boldsymbol{N}_\mathrm{C}$ and $\boldsymbol{N}_\mathrm{T}$ are the local shape functions according to Eq. (2.21), $\det(\boldsymbol{J})$ the determinant of the Jacobian from Eq. (2.31), and $\hat{w}_i$ is the collocation weight of the current collocation point, see Sect. 3.2. If the flexible body is axisymmetric, Eq. (3.22) and Eq. (3.23) are multiplied by 2π and the perpendicular distance between the current evaluation point and its axis of rotation. To eliminate the distinction between contact and target body, the roles are switched and the resulting contact forces are averaged. The resulting contact forces at each collocation point are assembled to the corresponding degree of freedom. The assembled vector ${}^\mathrm{I}\boldsymbol{f}_{i,j,k}$ represents the contact force at each individual control point in the inertial frame $\mathrm{K_I}$. Including the contact forces in the equations of motion (2.5) requires the determination of the discrete forces $\boldsymbol{h}_\mathrm{d}$. For this, the contact forces are transformed to the body frame by

$${}^\mathrm{R}\boldsymbol{f}_{i,j,k} = \boldsymbol{S}_\mathrm{IR}^\intercal \, {}^\mathrm{I}\boldsymbol{f}_{i,j,k}. \tag{3.24}$$

According to [SchwertassekWallrapp14], the discrete forces $\boldsymbol{h}_\mathrm{d}$ are finally computed by

$$\boldsymbol{h}_\mathrm{d} = \sum_{i=1}^{\mathrm{n}} \sum_{j=1}^{\mathrm{m}} \sum_{k=1}^{\ell} \begin{bmatrix} {}^\mathrm{R}\boldsymbol{f}_{i,j,k} \\ {}^\mathrm{R}\boldsymbol{p}_{i,j,k} \times {}^\mathrm{R}\boldsymbol{f}_{i,j,k} \\ \boldsymbol{\Phi}_{i,j,k}^\intercal \, {}^\mathrm{R}\boldsymbol{f}_{i,j,k} \end{bmatrix}. \tag{3.25}$$

Regarding the implementation, it is noted that the contact search can be parallelized with respect to the evaluation points $\boldsymbol{x}_{\mathrm{C},i}(\hat{\xi}_{\mathrm{C},i}, \hat{\eta}_{\mathrm{C},i}, \zeta_\mathrm{con})$.

3.4 Determination of the Normal Vector

In Sect. 3.3, the procedure to compute the contact force is described. However, the normal vector $\boldsymbol{n}$ is required for the described contact detection and to determine the contact force. As visualized in Fig. 3.5, the direction of the normal vector must fulfill two criteria: First, the normal vector $\boldsymbol{n}$ is orthogonal to the exterior surface of the target body. Second, the vector points to the outside of the target body. While the first condition is easy to meet, the second condition is more challenging.

In this section, a 2D axisymmetric example is used to show the determination of the normal vector $\boldsymbol{n}$. It is assumed that the contact is integrated with the local coordinate ξ while η remains constant. As an example, the gradients with respect to the local coordinates ξ and η of the axisymmetric sphere are visualized in Fig. 3.6. Here, the local coordinates $\eta_{\min}$ and $\eta_{\max}$ are the minimum and maximum values in η direction, and ξ_{int} is the current knot at which the contact is integrated. Two points at the top and the bottom of the sphere are monitored in Fig. 3.6. The points $\begin{bmatrix}\xi_{\text{int}} & \eta_{\min}\end{bmatrix}^{\mathsf{T}}$ and $\begin{bmatrix}\xi_{\text{int}} & \eta_{\max}\end{bmatrix}^{\mathsf{T}}$ in the parameter space correspond to the positions $\boldsymbol{x}(\xi_{\text{int}}, \eta_{\min})$ and $\boldsymbol{x}(\xi_{\text{int}}, \eta_{\max})$ in the physical space. The gradients $\frac{\partial \boldsymbol{x}(\xi,\eta)}{\partial \xi}$ with respect to the integrating local coordinate ξ are orthogonal to the exterior surface and point towards the tapered end of the semicircle. The gradients $\frac{\partial \boldsymbol{x}(\xi,\eta)}{\partial \eta}$ with respect to the constant local coordinate η point vertically in the positive y direction of the physical space. The direction of the gradients can be understood by the direction of the element's boundaries

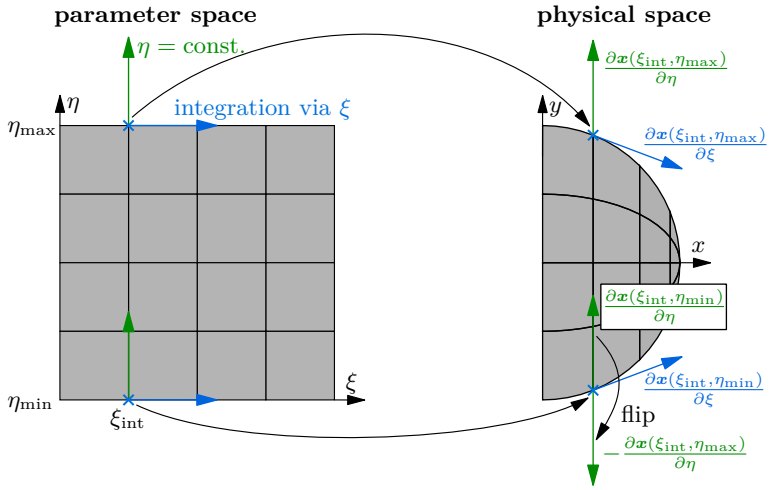


Figure 3.6: Gradients with respect to the local coordinates of the axisymmetric sphere.

which are also vertical. However, the gradients with respect to the constant local coordinate, e.g. η, are not necessarily orthogonal to the exterior surface. Therefore, the gradient with respect to the constant local coordinate cannot be used as the normal vector $\boldsymbol{n}$. This work proposes a method to determine the normal vector $\boldsymbol{n}$ independent of the IGA model.

To determine the normal vector $\boldsymbol{n}$ pointing outside of the body, a property of the gradient with respect to the constant local coordinate is used. The constant local coordinate η can only accept values between $\eta_{\min}$ and $\eta_{\max}$. If the constant local coordinate η is set to its maximum value $\eta_{\max}$, the gradient points towards the outside of the body, see Fig. 3.6. If the constant local coordinate η is set to its minimum value $\eta_{\min}$, the gradient points towards the inside of the body. This property can be explained by the direction of the local coordinate η axis in the parameter space in Fig. 3.6. Therefore, the gradient with respect to the constant local coordinate η is flipped in Fig. 3.6 if the local coordinate η takes on its minimum value $\eta_{\min}$. In this way, the gradient with respect to the constant local coordinate points outside the body.

The first condition for the normal vector $\boldsymbol{n}$ is fulfilled since it points outside the body. Therefore, the second condition remains so that the normal vector $\boldsymbol{n}$ is orthogonal to the exterior surface. This condition can be satisfied based on the gradient with respect to the integrating local coordinate ξ. This gradient is tangential to the exterior surface and is colored blue in the example in Fig. 3.6. For a 2D IGA body, the position vector $\boldsymbol{x}(\xi, \eta)$ can be divided into its x and y components so that the partial derivative results in

$$\frac{\partial \boldsymbol{x}(\xi,\eta)}{\partial \xi} = \begin{bmatrix} \frac{\partial x(\xi,\eta)}{\partial \xi} \\ \frac{\partial y(\xi,\eta)}{\partial \xi} \end{bmatrix}. \tag{3.26}$$

Then, there are two vectors

$$\boldsymbol{n}_{\text{candidate1}} = \begin{bmatrix} +\frac{\partial x(\xi,\eta)}{\partial \xi} \\ -\frac{\partial y(\xi,\eta)}{\partial \xi} \end{bmatrix} \tag{3.27}$$

$$\boldsymbol{n}_{\text{candidate2}} = \begin{bmatrix} -\frac{\partial x(\xi,\eta)}{\partial \xi} \\ +\frac{\partial y(\xi,\eta)}{\partial \xi} \end{bmatrix} \tag{3.28}$$

which are orthogonal to the gradient in Eq. (3.26). Therefore, the vectors $\boldsymbol{n}_{\text{candidate1}}$ and $\boldsymbol{n}_{\text{candidate2}}$ in Eq. (3.27) and Eq. (3.28) are the candidates for the normal vector $\boldsymbol{n}$. Both candidates are shown in Fig. 3.7 and are colored in red. One candidate is pointing to the inside of the body and one to the outside. To select the candidate which points outside the body, it can be compared with the gradient with respect to the constant local coordinate pointing outside the body. The angle between the candidates in Eq. (3.27) and Eq. (3.28), and the gradient with respect to the constant local coordinate η can be determined. The candidate with the smaller angle is orthogonal to the outer surface, points outside the body,

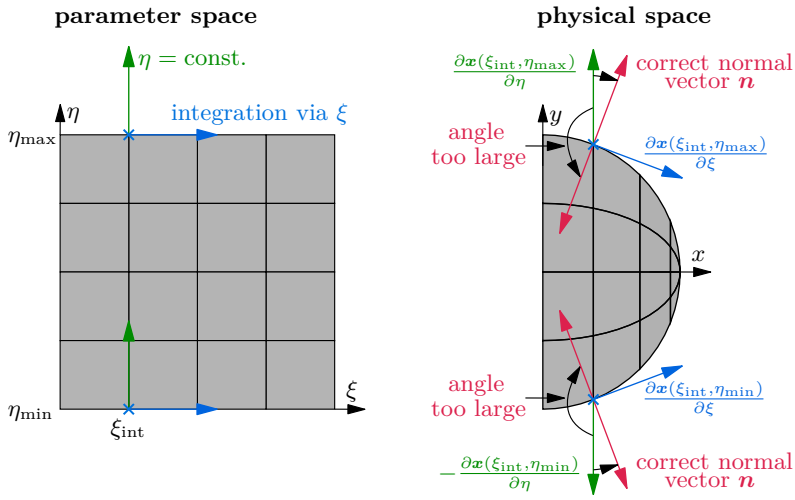


Figure 3.7: Comparison of the angle to the gradient of the constant coordinate, e.g. η.

and can therefore be used as the searched normal vector $\boldsymbol{n}$. Then, the normal vector $\boldsymbol{n}$ only needs to be normalized to length one.

For a 3D IGA body, the two candidates for the normal vector

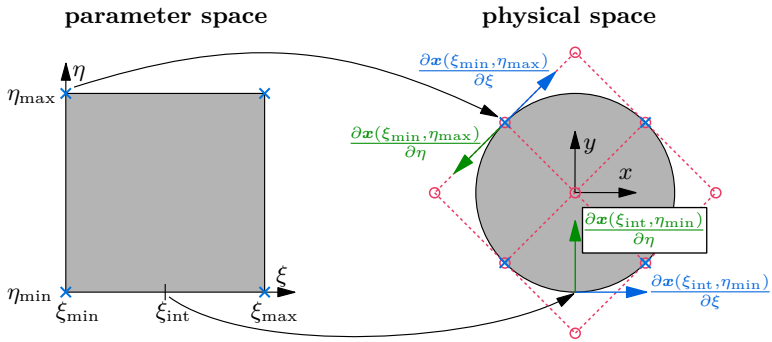
$$\boldsymbol{n}_{\text{candidate1}} = +\frac{\partial \boldsymbol{x}(\xi,\eta,\zeta)}{\partial \xi} \times \frac{\partial \boldsymbol{x}(\xi,\eta,\zeta)}{\partial \eta} \tag{3.29}$$

$$\boldsymbol{n}_{\text{candidate2}} = -\frac{\partial \boldsymbol{x}(\xi,\eta,\zeta)}{\partial \xi} \times \frac{\partial \boldsymbol{x}(\xi,\eta,\zeta)}{\partial \eta} \tag{3.30}$$

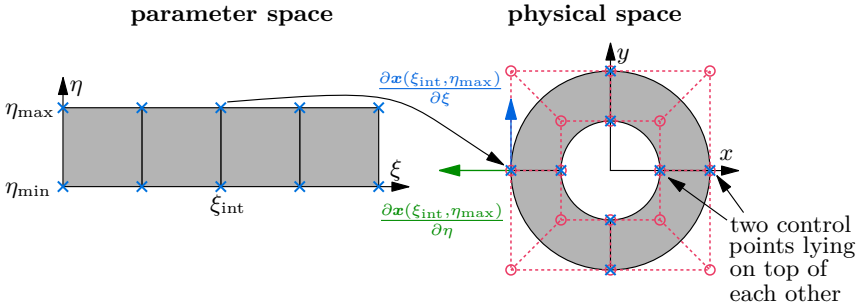
are determined with the cross product. In this case, it is assumed that the contact is integrated with the local coordinates ξ and η while the local coordinate ζ remains constant. If the contact is integrated via other local coordinates, the procedure described above still remains the same and the corresponding equations can be easily adjusted.

The described comparison of the angles works for most IGA models used in practice. However, there are IGA models for which the angle comparison does not work at certain locations. As an example, a 2D circle can be represented in two different ways in the IGA, see Fig. 3.8. The first geometry in Fig. 3.8a consists of only one element and the B-splines are of order $p = q = 2$ in both local coordinate directions. The second geometry in Fig. 3.8b is a pipe shaped IGA model where the B-splines are of order $p = 2$ and $q = 1$. By setting the inner radius of the pipe to zero, a circle can be modeled.

First, the single element model in Fig. 3.8a is discussed. The minimum and maximum values of the parameter space are denoted by $\xi_{\min}$, $\xi_{\max}$, $\eta_{\min}$, and $\eta_{\max}$.



a) IGA model of a circle based on one element.



b) Setting the inner radius of the IGA pipe model to zero results in a circle.

Figure 3.8: Two different approaches to model a circle in the IGA.

If the contact occurs between the boundaries of the parameter space, e.g. at the bottom of the circle in Fig. 3.8a, it is always possible to have the gradient with respect to the constant local coordinate point outside the body. However, if the contact is evaluated at one of the four corners of the parameter space, both gradients are tangential to the exterior surface, see Fig. 3.8a. Therefore, the inside and outside cannot be clearly identified. As a side note, this issue is of minor interest. The proposed IGA model in Fig. 3.8a contains elements with unequal aspect ratios, resulting in element locking. This occurs when the model is refined at one of the four corners of the parameter space, see Fig. 3.9. If an impact occurs at these corners, the results would be inaccurate.

Second, the pipe model in Fig. 3.8b is discussed. The model consists of four elements. Each element has the shape of a 90° arc. The pipe is closed by placing the first and the last control points at the same position [CottrellEtAl09]. As

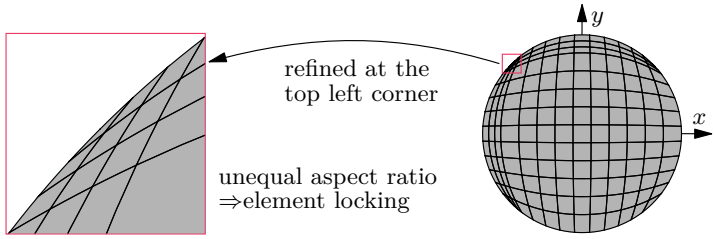


Figure 3.9: Element locking due to unequally shaped elements.

mentioned before in Sect. 2.4, control points lying on top of each other can be constraint in the equations of motion by means of linear constraint equations. Otherwise, the loop is not closed and the model behaves like a circlip. In this model, the contact is integrated via the local coordinate ξ and η remains constant. The gradient with respect to the constant local coordinate η always points outside the body and it is always orthogonal to the exterior surface.

3.5 Quasistatic Contact Model

This work uses the floating frame of reference formulation, see Sect. 2.1, to set up flexible multibody impact simulations. Therefore, global shape functions $\mathbf{\Phi}$ are required to model the elastic deformations. The IGA is used to generate a model which is reduced using the Craig-Bampton method, see Sect. 2.7. The resulting global shape functions are partitioned into low frequency and high frequency modes. The low frequency modes in the range up to 100 kHz represent the overall body flexibility and the high frequency modes represent the local deformation in the contact area. However, a combination of low and high frequency modes result in numerically stiff equations of motion (2.5). In the previous work [Tschigg20], modal damping is introduced to critically damp the high frequency modes. Alternatively to modal damping, a quasistatic contact model is used to improve the computational performance. In [Tschigg20], the quasistatic contact model is applied to isoparametric models. This work applies the quasistatic contact model in combination with the IGA.

In the quasistatic contact model, the equations of motion describing the elasticity are partitioned into two parts: the low frequency (lf) modes and the high

frequency (hf) modes. The partitioning can be applied to Eq. (2.5) resulting in

$$\begin{bmatrix} m\boldsymbol{E} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{C}_{\mathrm{r}}^{\mathrm{lf}\intercal} & \boldsymbol{C}_{\mathrm{r}}^{\mathrm{hf}\intercal} \\ \boldsymbol{0} & \boldsymbol{C}_{\mathrm{r}}^{\mathrm{lf}} & \overline{\boldsymbol{M}}_{\mathrm{e}}^{\mathrm{lf}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{C}_{\mathrm{r}}^{\mathrm{hf}} & \boldsymbol{0} & \overline{\boldsymbol{M}}_{\mathrm{e}}^{\mathrm{hf}} \end{bmatrix} \begin{bmatrix} {}^{\mathrm{R}}\dot{\boldsymbol{v}}_{\mathrm{IR}} \\ {}^{\mathrm{R}}\dot{\boldsymbol{\omega}}_{\mathrm{IR}} \\ \ddot{\boldsymbol{q}}_{\mathrm{e}}^{\mathrm{lf}} \\ \ddot{\boldsymbol{q}}_{\mathrm{e}}^{\mathrm{hf}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{h}_{\mathrm{dt}} \\ \boldsymbol{h}_{\mathrm{dr}} \\ \boldsymbol{h}_{\mathrm{d}}^{\mathrm{lf}} \\ \boldsymbol{h}_{\mathrm{d}}^{\mathrm{hf}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{h}_{\mathrm{bt}} \\ \boldsymbol{h}_{\mathrm{br}} \\ \boldsymbol{h}_{\mathrm{be}}^{\mathrm{lf}} \\ \boldsymbol{h}_{\mathrm{be}}^{\mathrm{hf}} \end{bmatrix} - \begin{bmatrix} \boldsymbol{h}_{\omega\mathrm{t}} \\ \boldsymbol{h}_{\omega\mathrm{r}} \\ \boldsymbol{h}_{\omega\mathrm{e}}^{\mathrm{lf}} \\ \boldsymbol{h}_{\omega\mathrm{e}}^{\mathrm{hf}} \end{bmatrix} - \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \\ \overline{\boldsymbol{K}}_{\mathrm{e}}^{\mathrm{lf}} \boldsymbol{q}_{\mathrm{e}}^{\mathrm{lf}} \\ \overline{\boldsymbol{K}}_{\mathrm{e}}^{\mathrm{hf}} \boldsymbol{q}_{\mathrm{e}}^{\mathrm{hf}} \end{bmatrix}, \tag{3.31}$$

where a Buckens-frame [SchwertassekWallrapp14] is used and standard damping is neglected. If a Buckens-frame is used, the centroid $\boldsymbol{c}$ becomes zero, and the translational coupling matrix $\boldsymbol{C}_{\mathrm{t}}$ vanishes [SchwertassekWallrapp14]. The equations of motion (3.31) include $n_{\mathrm{q}}^{\mathrm{lf}}$ low frequency elastic coordinates $\boldsymbol{q}_{\mathrm{e}}^{\mathrm{lf}}$ and $n_{\mathrm{q}}^{\mathrm{hf}}$ high frequency elastic coordinates $\boldsymbol{q}_{\mathrm{e}}^{\mathrm{hf}}$. The high frequency modes are only coupled with the rest of the system via the rigid body rotation through the matrix $\mathbf{C}_{\mathrm{r}}^{\mathrm{hf}}$. In [Koppens89], it is proposed to neglect the rotational coupling $\mathbf{C}_{\mathrm{r}}$ for small elastic deformations, which includes the low and high frequency part $\mathbf{C}_{\mathrm{r}}^{\mathrm{lf}}$ and $\mathbf{C}_{\mathrm{r}}^{\mathrm{hf}}$. The same is proposed in [Wielenga84] provided that the elastic deformations and the elastic coordinates remain small. Finally, it is shown in [SherifEtAl12, TschiggSeifried18] that the high frequency coupling $\mathbf{C}_{\mathrm{r}}^{\mathrm{hf}}$ is significantly smaller than the low frequency coupling $\mathbf{C}_{\mathrm{r}}^{\mathrm{lf}}$. Thus, their contribution can be neglected in Eq. (3.31). With these assumptions, the differential-algebraic system of equations of motion for a single body in contact follows as

$$\begin{bmatrix} m\boldsymbol{E} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{C}_{\mathrm{r}}^{\mathrm{lf}\intercal} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{C}_{\mathrm{r}}^{\mathrm{lf}} & \overline{\boldsymbol{M}}_{\mathrm{e}}^{\mathrm{lf}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} {}^{\mathrm{R}}\dot{\boldsymbol{v}}_{\mathrm{IR}} \\ {}^{\mathrm{R}}\dot{\boldsymbol{\omega}}_{\mathrm{IR}} \\ \ddot{\boldsymbol{q}}_{\mathrm{e}}^{\mathrm{lf}} \\ \boldsymbol{0} \end{bmatrix} = \begin{bmatrix} \boldsymbol{h}_{\mathrm{dt}} \\ \boldsymbol{h}_{\mathrm{dr}} \\ \boldsymbol{h}_{\mathrm{d}}^{\mathrm{lf}} \\ \boldsymbol{h}_{\mathrm{d}}^{\mathrm{hf}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{h}_{\mathrm{bt}} \\ \boldsymbol{h}_{\mathrm{br}} \\ \boldsymbol{h}_{\mathrm{be}}^{\mathrm{lf}} \\ \boldsymbol{0} \end{bmatrix} - \begin{bmatrix} \boldsymbol{h}_{\omega\mathrm{t}} \\ \boldsymbol{h}_{\omega\mathrm{r}} \\ \boldsymbol{h}_{\omega\mathrm{e}}^{\mathrm{lf}} \\ \boldsymbol{0} \end{bmatrix} - \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \\ \overline{\boldsymbol{K}}_{\mathrm{e}}^{\mathrm{lf}} \boldsymbol{q}_{\mathrm{e}}^{\mathrm{lf}} \\ \overline{\boldsymbol{K}}_{\mathrm{e}}^{\mathrm{hf}} \boldsymbol{q}_{\mathrm{e}}^{\mathrm{hf}} \end{bmatrix}, \tag{3.32}$$

where the dynamics of the high frequency modes vanish as quasistatic behavior is assumed. In [Tschigg20] it is stated that the high frequencies are of minor interest and are not excited. Therefore, the high frequency elastic coordinates $\boldsymbol{q}_{\mathrm{e}}^{\mathrm{hf}}$ are quasistatic and the reduced mass matrix $\overline{\boldsymbol{M}}_{\mathrm{e}}^{\mathrm{hf}}$ can be neglected, see [SherifEtAl11]. However, the formulation in Eq. (3.32) may be inaccurate if high rotational velocities occur. The algebraic equations in the last row of Eq. (3.32) represent the balance of the contact forces $\boldsymbol{h}_{\mathrm{d}}^{\mathrm{hf}}$ and the inner forces $\overline{\boldsymbol{K}}_{\mathrm{e}}^{\mathrm{hf}} \boldsymbol{q}_{\mathrm{e}}^{\mathrm{hf}}$. Without these remaining equations, the elastic deformations in the contact area could not be accurately represented. The differential-algebraic system (3.32) can be solved directly, or the algebraic quasistatic contact equations are solved separately. However, the direct solution seems to be numerically challenging. Therefore, the algebraic equations

$$\boldsymbol{f}_{\mathrm{qs}}(\boldsymbol{q}_{\mathrm{e}}^{\mathrm{hf}}) = \boldsymbol{q}_{\mathrm{e}}^{\mathrm{hf}} - \left(\overline{\boldsymbol{K}}_{\mathrm{e}}^{\mathrm{hf}}\right)^{-1} \boldsymbol{h}_{\mathrm{d}}^{\mathrm{hf}}(\boldsymbol{r}_{\mathrm{IR}}, \boldsymbol{\beta}_{\mathrm{IR}}, \boldsymbol{q}_{\mathrm{e}}^{\mathrm{lf}}, \boldsymbol{q}_{\mathrm{e}}^{\mathrm{hf}}) = \boldsymbol{0} \tag{3.33}$$

are solved separately from the differential equations of motion

$$\begin{bmatrix} m\boldsymbol{E} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{C}_{\mathrm{r}}^{\mathrm{lf}\,\mathsf{T}} \\ \boldsymbol{0} & \boldsymbol{C}_{\mathrm{r}}^{\mathrm{lf}} & \overline{\boldsymbol{M}}_{\mathrm{e}}^{\mathrm{lf}} \end{bmatrix} \begin{bmatrix} {}^{\mathrm{R}}\dot{\boldsymbol{v}}_{\mathrm{IR}} \\ {}^{\mathrm{R}}\dot{\boldsymbol{\omega}}_{\mathrm{IR}} \\ \ddot{\boldsymbol{q}}_{\mathrm{e}}^{\mathrm{lf}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{h}_{\mathrm{dt}} \\ \boldsymbol{h}_{\mathrm{dr}} \\ \boldsymbol{h}_{\mathrm{d}}^{\mathrm{lf}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{h}_{\mathrm{bt}} \\ \boldsymbol{h}_{\mathrm{br}} \\ \boldsymbol{h}_{\mathrm{be}}^{\mathrm{lf}} \end{bmatrix} - \begin{bmatrix} \boldsymbol{h}_{\omega\mathrm{t}} \\ \boldsymbol{h}_{\omega\mathrm{r}} \\ \boldsymbol{h}_{\omega\mathrm{e}}^{\mathrm{lf}} \end{bmatrix} - \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \\ \overline{\boldsymbol{K}}_{\mathrm{e}}^{\mathrm{lf}} \boldsymbol{q}_{\mathrm{e}}^{\mathrm{lf}} \end{bmatrix}. \tag{3.34}$$

The quasistatic contact equations $\boldsymbol{f}_{\mathrm{qs}}$ are solved in every time step for the high frequency elastic coordinates $\boldsymbol{q}_{\mathrm{e}}^{\mathrm{hf}}$. The straightforward approach for the solution is Newton's method. The required Jacobian $\boldsymbol{J}_{\mathrm{qs}}(\boldsymbol{q}_{\mathrm{e}}^{\mathrm{hf}})$ of Eq. (3.33) is computed numerically by first order finite differences. To counter rounding and approximation errors of the Jacobian, [Salane86] presents a method to adjust the step sizes to control these errors. This adjustment is used in the MATLAB function *odenumjac* to numerically approximate a Jacobian, see [Mathworks23]. Nevertheless, the solution of the quasistatic contact equation (3.33) can be challenging. Especially when the penalty factor is too high. This can cause rank issues in Newton's method [Tschigg20, RückwaldEtAl23]. Therefore, the penalty factor needs to be chosen with care. To save computing time, the Jacobian $\boldsymbol{J}_{\mathrm{qs}}(\boldsymbol{q}_{\mathrm{e}}^{\mathrm{hf}})$ is not computed numerically in each Newton iteration. Given a large number of high frequency elastic coordinates $\boldsymbol{q}_{\mathrm{e}}^{\mathrm{hf}}$, this would mean many function evaluations. Updating the Jacobian with Broyden's method [Broyden65] reduces the computational effort, which was also used in [Tschigg20].

The following Alg. 1 solves the quasistatic contact equation (3.33) and particularly monitors numerical challenges. The algorithm was previously presented in [RückwaldEtAl23, RückwaldEtAl24], and the description given here closely follows that presentation. Monitoring the numerics is relevant to adaptively determine the penalty factor in Sect. 6.1. Algorithm 1 is explained in the following:

In line 1 and 2 the Newton counter k_{newton} and a flag to reset Newton's method are initialized. Newton's method starts with a while loop in line 3. First of all, the counter k_{jac} of the Jacobian $\boldsymbol{J}_{\mathrm{qs}}(\boldsymbol{q}_{\mathrm{e}}^{\mathrm{hf}})$ is checked. If the counter reaches its maximum value $k_{\mathrm{jac}}^{\mathrm{max}} = 15$, determined from experience, a new Jacobian is computed by finite differences. Otherwise, the previous Jacobian updated with Broyden's method [Broyden65] is used. Then, the Jacobian is checked for singularity in line 10. A computationally efficient indicator for singularity is the reciprocal condition number (RCN) measuring the condition of a matrix [GolubLoan96]. If the RCN is near zero, the matrix is poorly conditioned. A value near one represents a well conditioned matrix. The RCN is determined with the MATLAB function *rcond*, see [Mathworks23]. If the RCN is less then $\epsilon_{\mathrm{rcn}}^{\mathrm{min}} = 10^{-10}$, the error "1" occurs. If the Jacobian is not finite, e.g. not a number (NaN) or infinite (inf), the error "2" occurs. The next but one paragraph describes how to respond to these two critical errors.

With Newton's method, the Jacobian and the previous solution of the high frequency elastic coordinates $\boldsymbol{q}_{\mathrm{e},k}^{\mathrm{hf}}$, a new solution $\boldsymbol{q}_{\mathrm{e},k+1}^{\mathrm{hf}}$ is computed in line 15. Then, Broyden's method is applied to the Jacobian $\boldsymbol{J}_{\mathrm{qs}}(\boldsymbol{q}_{\mathrm{e}}^{\mathrm{hf}})$ in line 16. Newton's

Algorithm 1 Solution of the quasistatic contact equation (3.33).

Input: $\boldsymbol{q}_{\mathrm{e}}^{\mathrm{hf}},\ \boldsymbol{J}_{\mathrm{qs}}(\boldsymbol{q}_{\mathrm{e}}^{\mathrm{hf}}),\ k_{\mathrm{jac}}$
Output: $\boldsymbol{q}_{\mathrm{e}}^{\mathrm{hf}},\ \boldsymbol{J}_{\mathrm{qs}}(\boldsymbol{q}_{\mathrm{e}}^{\mathrm{hf}}),\ k_{\mathrm{jac}}$
1: $k_{\mathrm{newton}} \Leftarrow 1$ ▷ init.: Newton counter
2: $flag \Leftarrow$ true ▷ init.: reset flag
3: **while** true **do** ▷ start Newton's method
4: **if** $k_{\mathrm{jac}} = k_{\mathrm{jac}}^{\max}$ **then**
5: $\boldsymbol{J}_{\mathrm{qs}}(\boldsymbol{q}_{\mathrm{e}}^{\mathrm{hf}})$ ▷ compute: new Jacobian
6: $k_{\mathrm{jac}} \Leftarrow 1$ ▷ reset: Jacobian counter
7: **else** ▷ reuse Jacobian
8: $k_{\mathrm{jac}} \Leftarrow k_{\mathrm{jac}} + 1$ ▷ reuse Jacobian and increase: Jacobian counter
9: **end if**
10: **if** $rcond(\boldsymbol{J}_{\mathrm{qs}}(\boldsymbol{q}_{\mathrm{e}}^{\mathrm{hf}})) < \epsilon_{\mathrm{rcn}}^{\min}$ **then** ▷ reciprocal condition number (rcn)
11: $error \Leftarrow 1$ ▷ error: singular
12: **else if** $notfinite(\boldsymbol{J}_{\mathrm{qs}}(\boldsymbol{q}_{\mathrm{e}}^{\mathrm{hf}}))$ **then**
13: $error \Leftarrow 2$ ▷ error: not finite
14: **end if**
15: $\boldsymbol{q}_{\mathrm{e},k+1}^{\mathrm{hf}} \Leftarrow \boldsymbol{q}_{\mathrm{e},k}^{\mathrm{hf}} - \boldsymbol{J}_{\mathrm{qs}}(\boldsymbol{q}_{\mathrm{e}}^{\mathrm{hf}})^{-1} \boldsymbol{f}_{\mathrm{qs}}(\boldsymbol{q}_{\mathrm{e}}^{\mathrm{hf}})$ ▷ Newton step
16: apply Broyden's method to Jacobian $\boldsymbol{J}_{\mathrm{qs}}(\boldsymbol{q}_{\mathrm{e}}^{\mathrm{hf}})$
17: **if** $max(|\boldsymbol{q}_{\mathrm{e},k}^{\mathrm{hf}} - \boldsymbol{q}_{\mathrm{e},k+1}^{\mathrm{hf}}|) < \epsilon$ **then**
18: break ▷ successful
19: **else if** $k_{\mathrm{newton}} = k_{\mathrm{newton}}^{\max}$ **then**
20: $error \Leftarrow -1$ ▷ error: maximum Newton steps
21: **end if**
22: **if** $error \neq 0$ and $flag$ **then** ▷ reset algorithm and try again
23: $flag \Leftarrow$ false ▷ reset is allowed once
24: $k_{\mathrm{newton}} \Leftarrow 1$ ▷ reset: Newton counter
25: $k_{\mathrm{jac}} \Leftarrow k_{\mathrm{jac}}^{\max}$ ▷ reset: Jacobian counter
26: $\boldsymbol{q}_{\mathrm{e},k}^{\mathrm{hf}} \Leftarrow \boldsymbol{0}$ ▷ reset: elastic coordinates
27: **else if** $error > 0$ **then** ▷ critical error
28: return $error$
29: **else if** $error < 0$ **then** ▷ non-critical error: successful
30: break
31: **else**
32: $k_{\mathrm{newton}} \Leftarrow k_{\mathrm{newton}} + 1$ ▷ increase: Newton counter
33: **end if**
34: **end while**
35: $k_{\mathrm{jac}} \Leftarrow 0$ ▷ reset: Jacobian counter

method terminates successfully in line 17 if the solution does not change with the tolerance ϵ, here, $\epsilon = 10^{-15}$ is chosen. If the maximum number of Newton steps $k_{\mathrm{newton}}^{\max} = 3k_{\mathrm{jac}}^{\max} = 45$ is reached, the non-critical error "-1" occurs. Again,

this parameter is determined by experience.

From line 22, the occurring errors "1", "2", and "−1" are evaluated. If any error occurs the first time during the current Newton iteration, Newton's method is reset. The reset includes the Newton counter k, the Jacobian counter k_{jac}, and the high frequency elastic coordinates $\boldsymbol{q}_{\mathrm{e},k}^{\mathrm{hf}}$. The algorithm may only be reset once per Newton iteration. If an error occurs despite the reset, a distinction is made between the critical errors "1" and "2" as well as the non-critical error "−1". If the error in line 27 is critical, the quasistatic algorithm stops. If the error in line 29 is non-critical, Newton's method terminates successfully. If no error occurs, Newton's method continues in line 32.

After the while loop, the discrete forces $\boldsymbol{h}_{\mathrm{d}}^{\mathrm{lf}}$ can be computed, and the Jacobian counter k_{jac} is set back to zero in line 35. Therefore, the Jacobian can be used more than the $k_{\mathrm{jac}}^{\mathrm{max}} = 15$ times. Testing showed, that this greatly improves the computation time while the results of the contact forces are nearly identical. Without the reset of the Jacobian counter k_{jac}, 3 to 4 Newton steps are required until convergence. Resetting the Jacobian counter increases the required Newton steps to 6 to 7, but reduces the number of computed Jacobians. Since computing the Jacobian is the most computationally expensive part of the quasistatic contact, the computational performance is increased even though more Newton steps are required.

3.6 Energy of the Quasistatic Contact Model

In order to validate the methods presented in this work, the total energy of the flexible multibody system is monitored. Thereby, only conservative systems and Buckens-frames [SchwertassekWallrapp14] as floating frames are considered. The energy of the system is composed of two parts: First, the energy of rigid or flexible bodies described by the floating frame of reference formulation, see Sect. 2.1. Second, the energy temporarily stored in the virtual springs of the penalty method, see Sect. 3.3.

In the floating frame of reference formulation, the motion of a flexible body is composed by its rigid body motion and the elasticity is assumed to be linear, see [SchwertassekWallrapp14]. Therefore, the energy of a single flexible body is composed of its rigid and flexible body motion as well of its kinetic and potential

energy. The kinetic energy $E_{\text{kinetic}}^{\text{rigid}}$ of the rigid body motion is given by

$$E_{\text{kinetic}}^{\text{rigid}} = \underbrace{\frac{1}{2}\left({}^{\text{R}}\boldsymbol{v}_{\text{IR}}\right)^{\intercal} m \left({}^{\text{R}}\boldsymbol{v}_{\text{IR}}\right)}_{\text{translation}} + \underbrace{\frac{1}{2}\left({}^{\text{R}}\boldsymbol{\omega}_{\text{IR}}\right)^{\intercal} \boldsymbol{I} \left({}^{\text{R}}\boldsymbol{\omega}_{\text{IR}}\right)}_{\text{rotation}}. \tag{3.35}$$

It is composed of a translational and rotational component. If the body experiences gravity, the potential energy $E_{\text{potential}}^{\text{rigid}}$ reads

$$E_{\text{potential}}^{\text{rigid}} = mg(h - h_0) \tag{3.36}$$

consisting of the potential height h and its zero level h_0. The energy resulting from linear elasticity is composed of the low and high frequency elastic coordinates $\boldsymbol{q}_{\text{e}}^{\text{lf}}$ and $\boldsymbol{q}_{\text{e}}^{\text{hf}}$. Since the high frequency modes are neglected in the equations of motion (3.34), the kinetic energy $E_{\text{kinetic}}^{\text{flexible}}$ of the flexible body motion reads

$$E_{\text{kinetic}}^{\text{flexible}} = \underbrace{\frac{1}{2}\left(\dot{\boldsymbol{q}}_{\text{e}}^{\text{lf}}\right)^{\intercal} \overline{\boldsymbol{M}}_{\text{e}}^{\text{lf}} \dot{\boldsymbol{q}}_{\text{e}}^{\text{lf}}}_{\text{low frequency}} + \underbrace{0}_{\text{high frequency}}. \tag{3.37}$$

Equation (3.37) only includes the low frequency elastic coordinates $\boldsymbol{q}_{\text{e}}^{\text{lf}}$. However, since the high frequency elastic coordinates $\boldsymbol{q}_{\text{e}}^{\text{hf}}$ are quasistatic, they occur in the potential energy

$$E_{\text{potential}}^{\text{flexible}} = \underbrace{\frac{1}{2}\left(\boldsymbol{q}_{\text{e}}^{\text{lf}}\right)^{\intercal} \overline{\boldsymbol{K}}_{\text{e}}^{\text{lf}} \boldsymbol{q}_{\text{e}}^{\text{lf}}}_{\text{low frequency}} + \underbrace{\frac{1}{2}\left(\boldsymbol{q}_{\text{e}}^{\text{hf}}\right)^{\intercal} \overline{\boldsymbol{K}}_{\text{e}}^{\text{hf}} \boldsymbol{q}_{\text{e}}^{\text{hf}}}_{\text{high frequency}} \tag{3.38}$$

of the flexible body motion denoted by $E_{\text{potential}}^{\text{flexible}}$. If the flexible body is in contact, energy is stored in the virtual springs of the penalty method. Based on the contact force in Eq. (3.22) and Eq. (3.23), the energy $E_{\text{potential}}^{\text{contact}}$ of the penalty contact evaluated at all $n_{\text{points}}^{\text{eval}}$ contact evaluation points reads

$$E_{\text{potential}}^{\text{contact}} = \sum_{i=1}^{\text{n}_{\text{points}}^{\text{eval}}} \frac{1}{2} c_{\text{p}} g_{\text{n}}^2 \hat{w}_i \det(\boldsymbol{J}). \tag{3.39}$$

Here, $\det(\boldsymbol{J})$ is the determinant of the Jacobian from Eq. (2.31), and $\hat{w}_i$ is the collocation weight of the current collocation point, see Sect. 3.2. If the flexible body is axisymmetric, Eq. (3.39) is multiplied by 2π and the perpendicular distance between the current evaluation point and its axis of rotation. The energy $E_{\text{potential}}^{\text{contact}}$ of the penalty contact is evaluated at every contact evaluation point $\boldsymbol{x}_{\text{C},i}$. Since the roles of the contact and target body are swapped to eliminate the distinction between contact and target body, the energy $E_{\text{potential}}^{\text{contact}}$ of the penalty contact is averaged. Finally, the total energy of a flexible body in contact reads

$$E = E_{\text{kinetic}}^{\text{rigid}} + E_{\text{potential}}^{\text{rigid}} + E_{\text{kinetic}}^{\text{flexible}} + E_{\text{potential}}^{\text{flexible}} + E_{\text{potential}}^{\text{contact}}. \tag{3.40}$$

Software Implementation

The Institute of Mechanics and Ocean Engineering (MuM) has been working for many years in the field of flexible multibody systems using the floating frame of reference formulation. The application areas include, for instance, topology optimization [Moghadasi19], real-time simulation of vehicle dynamics [Schmitt19], contact simulations [Tschigg20], and control strategies for flexible robots [Morlock23]. The works mentioned above use the floating frame of reference formulation and isoparametric elements to retrieve global shape functions of flexible bodies. This chapter introduces the Matlab toolbox RIGA in Sect. 4.1 its computational performance is briefly analyzed in Sect. 4.2.

4.1 RIGA

A workflow similar to the one in Fig. 4.1 is used in the previous works [Moghadasi19, Schmitt19, Tschigg20, Morlock23] to simulate flexible multibody systems. First, the geometry is modeled in a CAD software, e.g. Inventor. Second, the geometry is meshed with isoparametric elements in a FEM software, e.g. Ansys [Ansys19]. The FEM model can be saved in a file which includes the assembled mass and stiffness matrix as in Eq. (2.34). Third, the inhouse Matlab toolbox RED developed at MuM reads the file saved by Ansys. The toolbox then stores the FEM model in a data structure which will be used in all following tasks. With RED, the model is reduced, e.g. with modal truncation or the Craig-Bampton method, and the SID file is computed and stored, see [SchwertassekWallrapp14]. Fourth, the SID file is loaded in another inhouse Matlab toolbox named DynManto, see [HeldEtAl20]. Here, the flexible multibody system is set up and simulated.

A milestone of this work is to substitute the first two steps in Fig. 4.1 and replace them with the IGA as shown in Fig. 4.2. The general workflow in Fig. 4.1 should remain. Therefore, the IGA Matlab toolbox Reduced Isogeometric Analysis (RIGA) is developed to work smoothly with the existing two toolboxes. In total, the toolbox RIGA consists of 10 000 lines of code. As usual

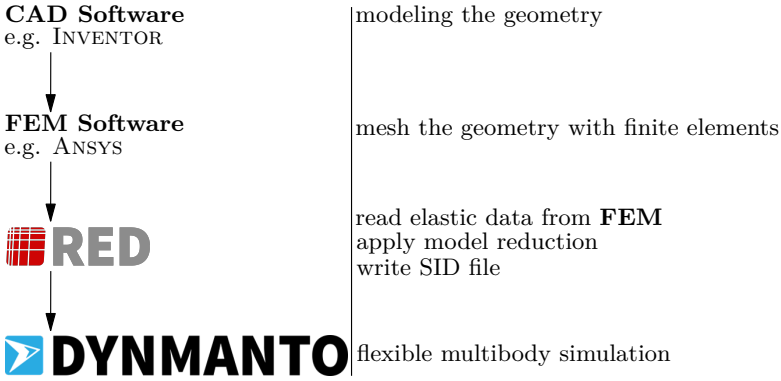


Figure 4.1: Workflow of a flexible multibody simulation with isoparametric elements.

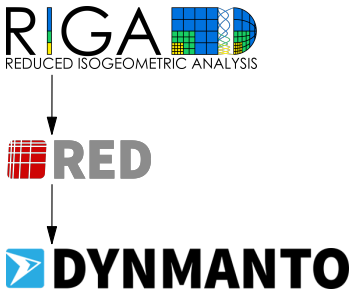


Figure 4.2: Workflow of a flexible multibody simulation with RIGA.

in IGA research [CottrellEtAl09], the toolbox is limited to simple geometries such as circles, spheres, and rods, as the focus is set on the development of new algorithms. Therefore, arbitrary CAD models cannot be imported into RIGA. As side note, although CAD software uses NURBS to define the geometry, the geometry cannot be directly used to derive the equations of motion. For complex models, CAD software uses T-splines or subdivision surfaces with multiple patches [CottrellEtAl09]. Another challenge is that the geometry in CAD is build on surfaces, not solid bodies. This may be ideal for the boundary element method, but applications in the field of the FEM such as impact simulations require solid models.

The toolbox RIGA is implemented in MATLAB. In practice, MATLAB is an easy-to-use coding language but its computational performance is low. However, the already existing workflow shown in Fig. 4.1 is implemented in MATLAB. The model generation could be implemented in a faster language, but the IGA contact routine must still be used during time integration in DYNMANTO. Therefore,

MATLAB is chosen. To integrate RIGA into the existing workflow from Fig. 4.1, RIGA uses the same data structure of the full model as RED. This means that only very few adjustments are required in RED and DYNMANTO.

The toolbox RIGA uses object oriented programming. Different tasks and data structures are separated into different classes. Each class has methods, which are similar to functions, and properties, e.g. variables. As an example, the method *finish* of the class RIGA is briefly summarized in Alg. 2. Before calling the

Algorithm 2 Process of the method *finish* of the class RIGA.

```
1: check optional inputs (e.g. constraints)
2: if hierarchically refined then
3:     prepare hierarchical B-splines
4: end if
5: set up elements
6: set up control points
7: compute element mass and stiffness matrices
8: assemble mass and stiffness matrix
9: add interface control points
10: find repeating control points
11: apply constraints
```

method *finish*, the user defines the geometry by entering the control points, the weights, the knot vectors, the order of the B-splines, the material parameters, such as the Young's modulus, and finally the refinement. With the Alg. 2, the IGA model is generated and can be imported into RED afterwards. First, in line 1, it is checked for optional inputs such as constraints. If the IGA model is hierarchically refined, the linear combinations of finer B-splines is computed as briefly explained in Sect. 2.5. In line 5, the elements, also known as knot spans, are set up. The class of the control points is initialized in line 6 assigning each control point its degrees of freedom. Then, in line 7, the element mass and stiffness matrices are computed as defined in Eq. (2.29) and Eq. (2.30). The matrices are assembled to the full equations of motion (2.34) in line 8. In line 9, interface control points are defined which act as a joint to connect to other bodies. As mentioned before in Sect. 2.4, control points may lay on top of each other and can be constraint by linear constraint equations. This procedure is executed in line 10. All constraints are imposed on the equations of motion (2.34) and the method finishes in line 11.

4.2 Computational Performance

Computational performance in MATLAB is an issue, especially for contact handling during time simulation. Compared to the linear isoparametric elements, the nonlinear local shape functions of the IGA, the NURBS, are computationally more expensive. When using hierarchically refined IGA models, the computational requirements are even higher. MATLAB offers toolboxes to improve the performance. The straightforward approach is to parallelize the code, e.g. parallel *for* loops. As an example, the contact check of several collocation points or the determination of the element mass and stiffness matrices, as in Eq. (2.29) and Eq. (2.30), are parallelized. However, parallelization needs to be used with care. On the one hand, parallelization is not possible if the iterations of the *for* loop are dependent on each other. On the other hand, in practice, a parallelized *for* loop may be slower than a sequential *for* loop.

A greater increase in performance can be achieved with the MATLAB CODER Toolbox. Individual MATLAB functions are automatically translated to C-code and compiled into a MATLAB EXECUTABLE (MEX) file. A MEX-file is similar to a compiled Dynamic Link Library (DLL) file but is easier to include into MATLAB. This work uses the open source MINGW-W64 compiler [MathWorks24] to create MEX-files. Compared to the standard MATLAB compiler, the MINGW-W64 compiler can also generate parallelized code. This provides the greatest increase in performance compared to standard MATLAB without parallelization.

As an example, the impact of two axisymmetric spheres is simulated in 2D. The models are hierarchically refined with $n_{\mathrm{lvl}} = 6$ levels and a quasistatic contact model is chosen. For more details on this setup, see the application example discussed in Sect. 5.1.5. The processor used is the Intel W-2295 model with 18 cores. It should be noted that the following qualitative observations depend on the particular application example. The setup is simulated in four different ways, including sequential and parallel computing, as well as standard MATLAB and MEX. Each of the four simulation types is run three times and the average computation time is calculated. The final comparison is shown in Fig 4.3. It can be seen that parallelizing the contact routine and compiling it as a MEX-file improves the computational performance by a factor of up to 15 compared to the sequentially computed MATLAB code. As a side note, the sequential simulations still take advantage of multiple cores. This is because MATLAB automatically performs matrix operations on all available cores, e.g. for solving the equations of motion (3.34).

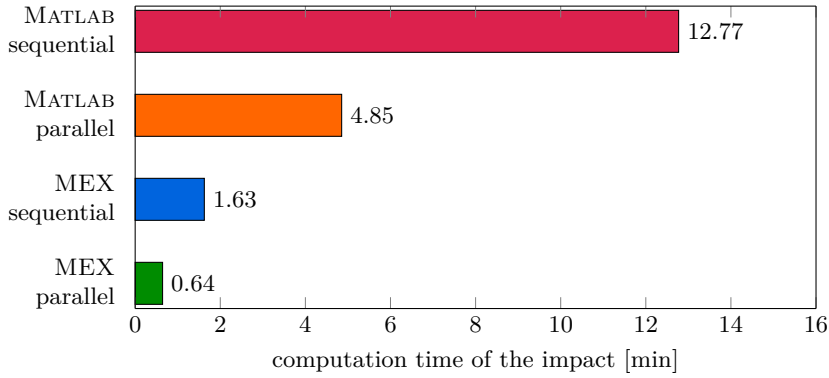


Figure 4.3: Comparing the computation time of sequential and parallel computing in MATLAB for an IGA impact simulation.

Investigation of IGA Impact Modeling

In this chapter, IGA based flexible multibody systems are validated in impact simulations to show its overall functionality. Factors that affect accuracy and computation time are evaluated. In the first testing example in Sect. 5.1, the impact of two axisymmetric spheres is simulated. This setup is 2D. In the second testing example in Sect. 5.2 the setup is extended to 3D. Some of the results presented in this chapter have already been published in [RückwaldEtAl21c, RückwaldEtAl21d, RückwaldEtAl22, RückwaldEtAl23].

5.1 Testing Example I: Axisymmetric Spheres

The setup of the first testing example is illustrated in Fig. 5.1. The axisymmetric setup in 2D represents the impact of two flexible spheres modeled by two semi-

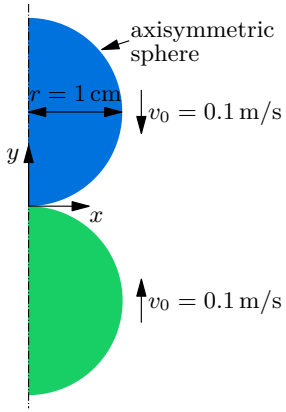


Figure 5.1: Impact of two axisymmetric spheres.

circles. The radius of the spheres is $r = 1\,\text{cm}$ and their initial velocity is defined with $v_0 = 0.1\,\text{m/s}$. The spheres are made of steel. Therefore, the Young's modulus is chosen as $E = 210\,\text{GPa}$, the density as $\rho = 7850\,\text{kg/m}^3$, and the Poisson's ratio as $\nu = 0.3$. The setup is well suited since the analytic solution by Hertz can be used as a reference, see Sect. 3.1. The goal of the first testing example is to validate the IGA in a flexible multibody impact simulation. This includes the different contact discretization methods from Sect. 3.2, the quasistatic contact model from Sect. 3.5, and hierarchical refinement from Sect. 2.5. Besides the analytic solution by Hertz, an isoparametric model is used as a second reference.

This testing example is divided into five tests. In each test, the setup is simulated for 0.1 ms, which is a little more than the duration of the impact. An overview of the various tests is provided in Tab. 5.1. First, the IGA contact discretization methods are tested including Gauss points and collocation points, e.g. Greville, Botella, and Chebyshev points. Second, the effect of order elevation (p-refinement) is evaluated, see Sect. 2.4. Third, the IGA model is compared with an isoparametric model. This model will be simulated by a reduced isoparametric (RISO) model as well as a full FEM model in Ansys. Up to and including the third test, only modal damping is used to reduce the numerical stiffness of the Craig-Bampton model. Fourth, modal damping and the quasistatic model are compared. Fifth, hierarchical refinement is tested and compared to the standard approach for refinement in IGA, which is global refinement.

The geometry of the IGA model is based on the semicircle in Fig. 2.5. The order of

Table 5.1: Goals of the first testing example.

	goal	focus of analysis
Test 1 Sect. 5.1.1	investigate the IGA discretization methods (see Sect. 3.2)	Gauss points Greville points Botella points Chebyshev points
Test 2 Sect. 5.1.2	evaluate order elevation (p-refinement) in IGA (see Sect. 2.4)	variation of order elevation
Test 3 Sect. 5.1.3	compare the IGA with isoparametric elements in an impact simulation	reduced IGA (RIGA) reduced isoparametric model full Ansys model
Test 4 Sect. 5.1.4	compare modally damped IGA models with quasistatic contact models (see Sect. 3.5)	modal damping quasistatic contact model
Test 5 Sect. 5.1.5	compare globally and hierarchically refined IGA models (see Sect. 2.4 and Sect. 2.5)	global refinement hierarchical refinement

the B-splines is elevated by $n_\mathrm{p} = 2$ in ξ- and η-direction. To represent the overall flexibility, 15 knots are inserted in ξ-direction and 24 knots in η-direction between each existing knot. To accurately simulate the occurring contact forces, the contact area needs to be refined. Special care is taken to ensure an element size of 10 µm in the contact area. The same refinement is applied to the isoparametric model which will be used as a reference in Sect. 5.1.3. In Fig. 5.2, the IGA model and the isoparametric model are visualized. The overall refinement of the bodies is similar and both contact areas are equally refined. The contact area of both models consists of 25 contact elements. As a reference, the isoparametric model will be used in a full FEM simulation in Ansys and as a reduced model to be simulated in the same way as the IGA model. The reduced models are obtained by applying the Craig-Bampton method and use $n_\mathrm{q}^\mathrm{lf} = 10$ low frequency normal modes to represent the overall flexibility. The number of high frequency constrained modes is slightly higher for the IGA model. Selecting 25 contact elements results in 26 interface nodes for the isoparametric model. Since the IGA model uses local shape functions of order $p = 4$ and the isoparametric model uses linear local shape functions, the IGA model requires three more interface control points yielding 29 interface control points. However, the number of contact evaluation points is identical given that the IGA model uses collocation points, e.g. Greville points. A detailed overview of both models is given in Tab. 5.2. Here, the green colored properties are identical for both models, and the red colored properties are different. As a side note, it is mentioned in Sect. 3.2 that for collocation points, the number of contact evaluation points is equal to the number of control points on the exterior surface. This also applies to the IGA model in Fig. 5.2. However, the number of interface control points is larger than the number of contact evaluation points, see Tab. 5.2. This is because the contact interval is set to the size of 25 contact elements, and only the relevant collocation and interface control points are considered.

Table 5.2: Globally refined IGA and isoparametric model of an axisymmetric sphere.

	reduced IGA (RIGA) model	reduced isoparametric (RISO) model
model	3136 control points	3764 nodes
order of local shape functions	$p = 4$ and $q = 3$	linear
contact elements	25	25
dimension of contact elements	10 µm	10 µm
low frequency modes n_q^lf	10	10
interface	29 interface control points	26 interface nodes
high frequency modes n_q^hf	$29 \times 2 = 58$	$26 \times 2 = 52$
model reduction	Craig-Bampton	Craig-Bampton
modal damping	✓	✓
contact evaluation points	26 Greville points	26 nodes

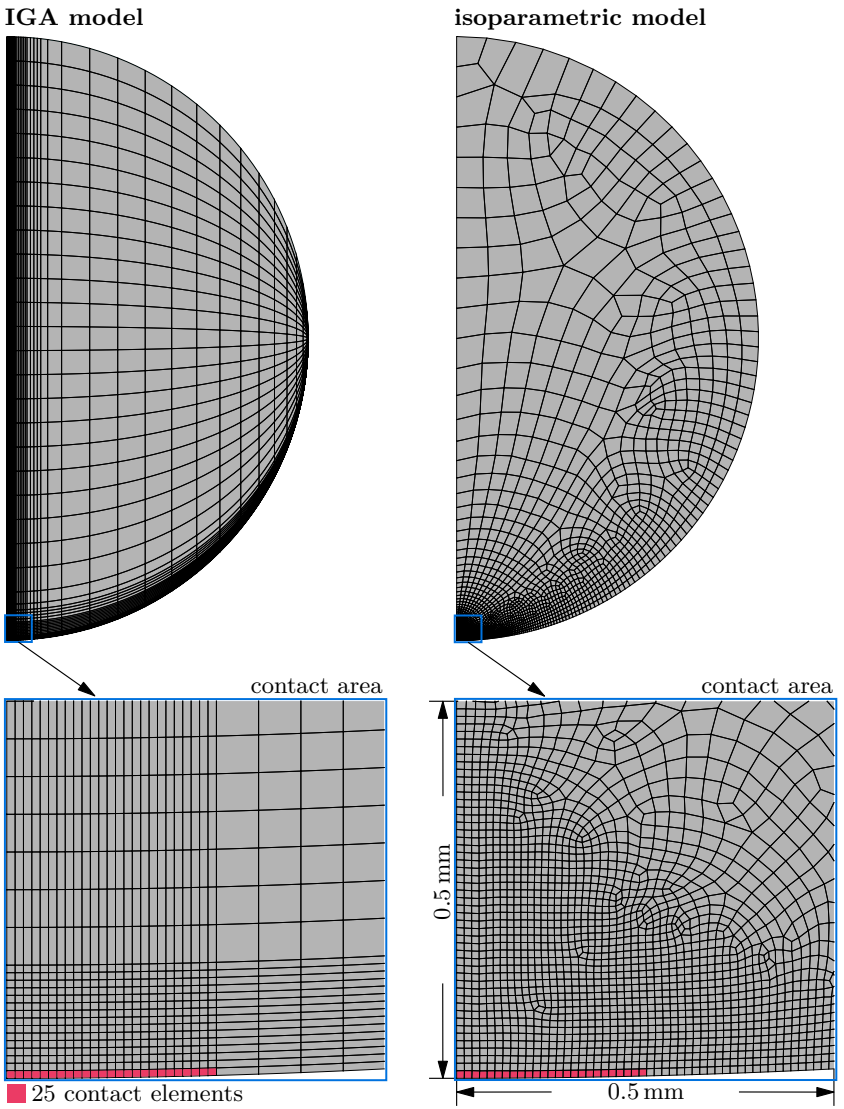


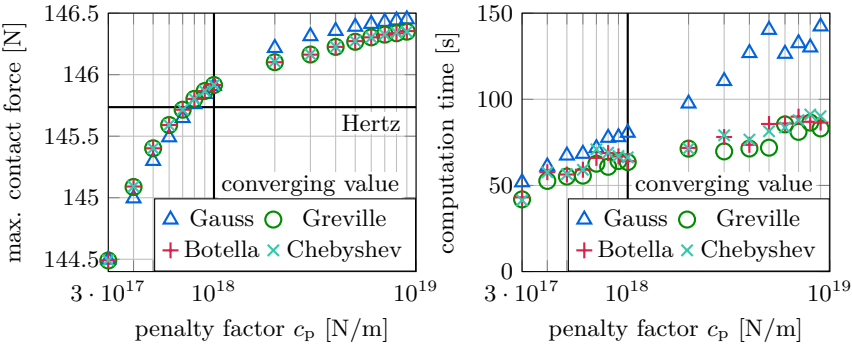
Figure 5.2: Globally refined IGA and isoparametric model of an axisymmetric sphere.

5.1.1 Test 1: IGA Contact Discretization

In the first test, Gauss points and collocation points are compared for contact discretization, see Sect. 3.2. The straightforward approach for selecting points to evaluate the contact are Gauss points. However, using Gauss points results in a overdetermined contact formulation since it involves more evaluation points than control points on the exterior surface of the body. This is not the case for collocation points. Therefore, it is expected that the collocation points show lower computation times than Gauss points. However, it is to be determined how both methods differ in their accuracy.

This work uses the penalty method which requires the penalty factor c_p to be determined. As mentioned before, its value is increased until the results become mostly independent of the chosen parameter [SeifriedEtAl03]. Increasing the penalty factor beyond its converging value increases the computation time since the equations of motion (2.5) become numerically stiffer. All simulations in this work are run three times and then the average computation time is calculated. The results of such a study on the penalty factor are shown in Fig. 5.3. Here, the convergence of the penalty factor ranging from $c_\mathrm{p} = 3 \cdot 10^{17}\,\mathrm{N/m}$ up to $c_\mathrm{p} = 9 \cdot 10^{18}\,\mathrm{N/m}$ is analyzed. The penalty factor is chosen so that the contact force does not change significantly, see Fig. 5.3a. Increasing the penalty factor beyond its converging value increases the computation time, see Fig. 5.3b. In this way, the penalty factor $c_\mathrm{p} = 1 \cdot 10^{18}\,\mathrm{N/m}$, is chosen for all four contact discretization methods for further analysis.

There are no significant differences between the considered collocation points in Fig. 5.3. The overall course, the calculation time and the maximum contact force are almost identical. The same applies to the maximum contact force when



a) Maximum contact force. **b)** Computation time (impact phase).

Figure 5.3: Convergence of the penalty factor.

using Gauss points. However, the computation time is higher since the contact formulation is overdetermined. This observation can also be made in Fig. 5.4, where the computation time for $c_\mathrm{p} = 1 \cdot 10^{18}\,\mathrm{N/m}$ is shown in detail. Using Gauss points take the most computation time. Using collocation points results in a faster computation of the impact. The time differences between the three collocation methods are minor.

Regarding the accuracy of the results, the contact force is plotted in Fig. 5.5 with respect to the time t. The analytic solution by Hertz is used as a reference. All four discretization methods show accurate results and do not differ significantly.

Finally, the translational rigid body velocity in y-direction of the upper sphere in Fig. 5.1 is shown in Fig. 5.6. The initial velocity of the spheres is defined with $v_0 = 0.1\,\mathrm{m/s}$. The impact of two spheres according to Hertz is an elastic impact without damping. Therefore, the coefficient of restitution, see [Wriggers06], is equal to one and the velocity after the impact is still v_0 with a change of sign. Although the high frequency modes are modally damped to reduce numerical stiffness, all four discretization methods represent the elastic impact accurately with the respect to the analytic solution by Hertz.

Similar results are achieved in [Matzen15], where no major differences between the collocation methods is observed. For consistency, Greville points are used for all subsequent IGA simulations.

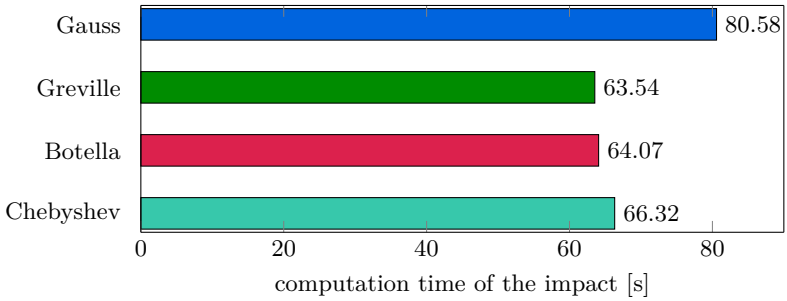


Figure 5.4: Comparing the computation time of the different discretization methods using $c_\mathrm{p} = 1 \cdot 10^{18}\,\mathrm{N/m}$.

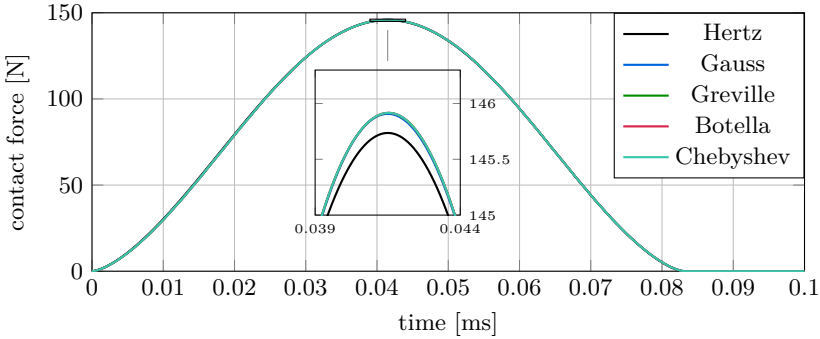


Figure 5.5: Contact force over time for $c_\mathrm{p} = 1 \cdot 10^{18}\,\mathrm{N/m}$.

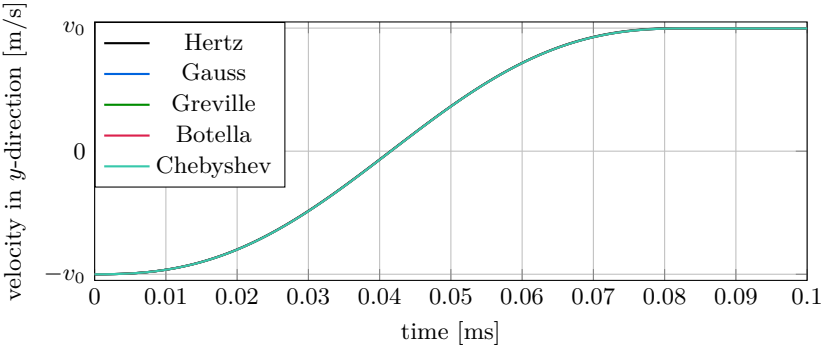


Figure 5.6: Translational rigid body velocity in y-direction of the upper sphere in Fig. 5.1 for $c_\mathrm{p} = 1 \cdot 10^{18}\,\mathrm{N/m}$.

5.1.2 Test 2: IGA Order Elevation

The second test evaluates the effect of order elevation on accuracy and computation time. Therefore, order elevation is varied between $n_\mathrm{p} = 0$ (no elevation) and $n_\mathrm{p} = 4$. Greville collocation points and the penalty factor $c_\mathrm{p} = 1 \cdot 10^{18}\,\mathrm{N/m}$ are used. In Fig. 5.7, the contact force is visualized. It can be seen that the force converges to the solution by Hertz as the order increases. The only exception is the order elevation of $n_\mathrm{p} = 3$. Comparing the computation times in Fig. 5.8 shows that increasing the order also increases the computation time. This is because higher order local shape functions are more computationally expensive. An order elevation of $n_\mathrm{p} = 2$ appears as a good compromise between computation time and accuracy in this test. Therefore, this value is used for the remaining tests in Sect. 5.1.

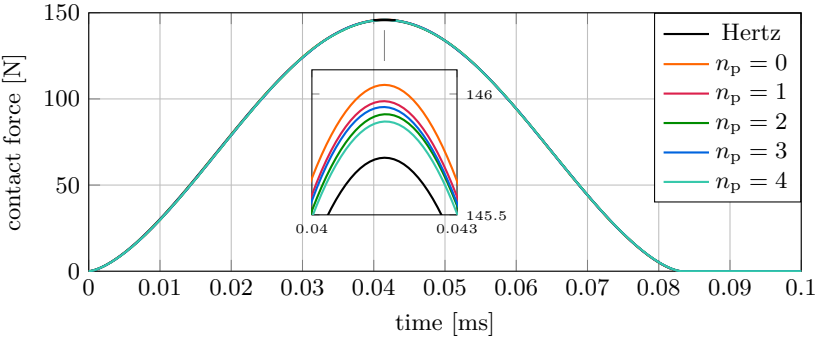


Figure 5.7: Contact force over the time while varying the order.

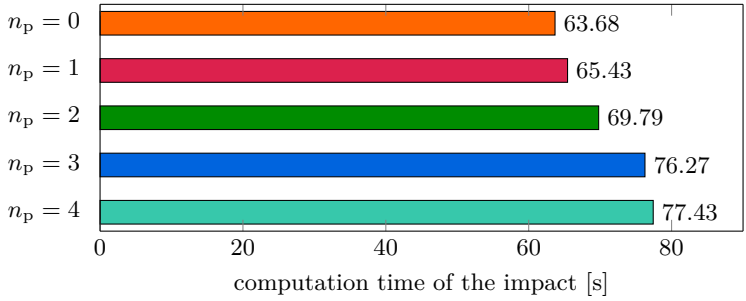


Figure 5.8: Computation time of different levels of order elevation.

5.1.3 Test 3: IGA and Isoparametric Models

The third test investigates the accuracy and computational performance of the IGA model by comparing it with the isoparametric reference model from Fig. 5.2. The isoparametric model is simulated in two ways: First, it is simulated as a RISO model using the Craig-Bampton method for reduction, modal damping to reduce the stiffness of the high frequency modes, and the impact is modeled with a penalty method. Therefore, the RIGA and the RISO model are simulated in the same way. Second, the isoparametric model is also simulated in a full Ansys simulation using the Lagrange multiplier method as a reference.

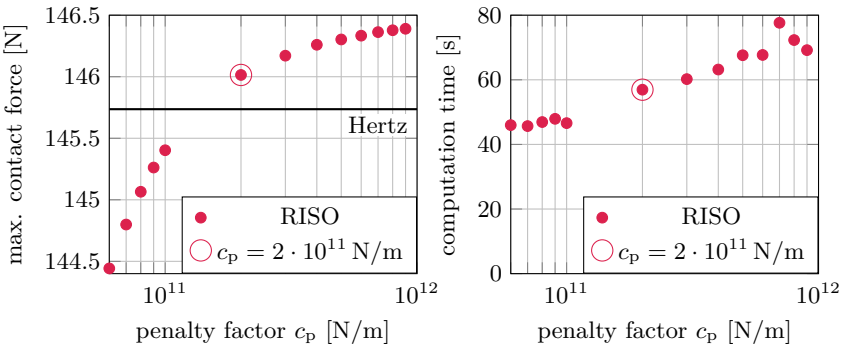
On the one hand, the IGA seems to be more accurate than classical isoparametric models. This is because the geometry is exactly preserved in IGA and high order local shape functions are used. On the other hand, the high order local shape function based on B-splines are computationally more expensive than linear shape functions of isoparametric elements. This test checks whether these assumptions

can be reproduced by means of a simulation.

As in the previous test, the penalty factor is manually determined in Fig. 5.9. For the RISO model, the convergence of the penalty factor ranging from $c_\mathrm{p} = 6 \cdot 10^{10}\,\mathrm{N/m}$ up to $c_\mathrm{p} = 9 \cdot 10^{11}\,\mathrm{N/m}$ is analyzed in Fig. 5.9. The analysis of the penalty factor for the RISO model in results in the penalty factor $c_\mathrm{p} = 2 \cdot 10^{11}\,\mathrm{N/m}$ which is highlighted in Fig. 5.9. Comparing the convergence behavior with the Greville points in Fig. 5.3, both reduced models converge to the same maximum contact force. However, the RISO model is computed slightly faster.

However, the models converge for different penalty factors. The penalty factor of the RISO model is about seven magnitudes smaller than that of the RIGA model. This is because of the calculation of the contact force in the IGA, see Eq. (3.22). It includes the determinant $\det(\boldsymbol{J})$ of the Jacobian $\boldsymbol{J}$, see Eq. (2.31), and the collocation weight $\hat{w}_i$. Both terms are not included in the contact force calculation of isoparametric models, because the contact is evaluated at the nodes, which are also the degrees of freedom. As an example, the collocation weight of the first Greville point in the lower left of Fig. 5.2 is $\hat{w}_1 \approx 3.13 \cdot 10^{-4}$. The determinant of the Jacobian of the corresponding element is $\det(\boldsymbol{J}) \approx 2.83 \cdot 10^{-4}$. Multiplying both factors gives $8.86 \cdot 10^{-8} \approx 10^{-7}$, which explains the difference of seven decades in the penalty factors.

Regarding the computation times listed in Fig. 5.10, the RISO model is slightly faster than the RIGA model. The difference is only minor since both models are simulated using the floating frame of reference formulation with a similar number of elastic coordinates, see Tab. 5.2. This includes the transformation from the body reference frame to the inertial frame and the assembly of the discrete forces $\boldsymbol{h}_\mathrm{d}$, see Sect. 3.3. The actual contact check only takes approximately 10 %



a) Maximum contact force.

b) Computation time (impact phase).

Figure 5.9: Convergence of the penalty factor of a reduced isoparametric model.

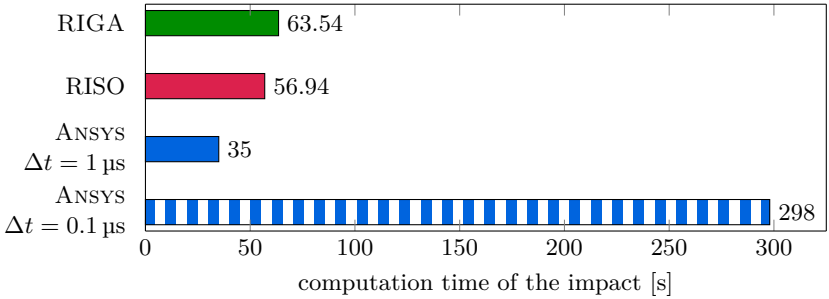


Figure 5.10: Comparing the computation time of the different discretization methods.

of the computation time for the RISO model and 30 % for the RIGA model. Both reduced models are simulated with a variable step size, which is on average $\Delta t \approx 0.7\,\mu s$. In addition to the simulation of the RISO model, the isoparametric model is simulated with Ansys. The simulation is performed using the step sizes $\Delta t = 1\,\mu s$ and $\Delta t = 0.1\,\mu s$. The simulation with the larger step size is the fastest of all four simulations. However, it has a low resolution of only 100 time steps and the results show slight oscillations in the energy over time. The simulation with the smaller step size takes approximately ten times longer and generates 1000 time steps. Since the results appear of higher quality, the Ansys simulation with the time step size $\Delta t = 0.1\,\mu s$ is used in this test.

As in the previous test, the contact force is shown in Fig. 5.11. All simulations can replicate the analytic solution by Hertz well. However, in this example, the RIGA and the RISO model are closest to the analytic solution.

When simulating the Hertzian contact, the relationship in Eq. (3.3) between the

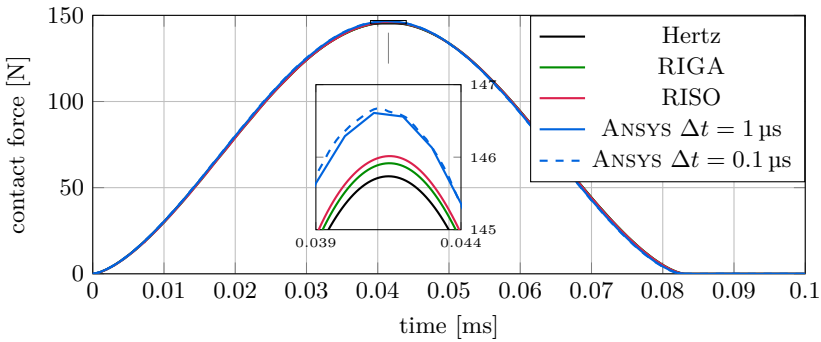


Figure 5.11: Contact force over the time.

contact force and the relative displacement δ between the centers of mass can be visualized, see Fig. 5.12. The nonlinear relationship is well represented by all models and the differences are minor.

The normal gap g_n with respect to the time t is plotted in Fig. 5.13. Here, the maximum penetration of all contact evaluation points is visualized. Although a penalty method is used, the unphysical penetration of the bodies is very small. The maximum normal gap of the RISO model is $g_\mathrm{n} = 0.46\,\mu\mathrm{m}$. For the RIGA model, it is only $g_\mathrm{n} \approx 0.13\,\mu\mathrm{m}$. Since Ansys uses the Lagrange multiplier method, no penetration occurs. As the normal gap is the highest with the RISO model, it can be assumed that the penalty factor $c_\mathrm{p} = 2 \cdot 10^{11}\,\mathrm{N/m}$ may have been set too low. However, even with the significantly higher penalty factor $c_\mathrm{p} = 9 \cdot 10^{11}\,\mathrm{N/m}$, such a small normal gap as with the RIGA model is not achieved. Therefore, the RIGA model is slightly more accurate than the reduced isoparametric model in this testing example.

The von Mises stress along the symmetry axis given in Eq. (3.8) is visualized in Fig 5.14. For comparison, the time step was chosen at which the occurring stresses are maximum in the respective simulation. The maximum stress occurring in the flexible body is slightly better represented by the isoparametric model than by the isogeometric model. A reason for this may be the slightly higher maximum force of the Ansys solution, see Fig. 5.11. However, the overall stress is well represented by both models. Overall, the differences are very small and are of minor practical importance. The distribution of the von Mises stresses in the contact area is shown in Fig 5.15. It shows that the RIGA model reproduces the stresses predicted by the full Ansys model accurately.

To validate the conservation of energy, the energy of the system is computed according to Sect. 3.6 and plotted over the time t in Fig. 5.16. Since the impacts are elastic, the energy should stay constant. The initial energy E of the system

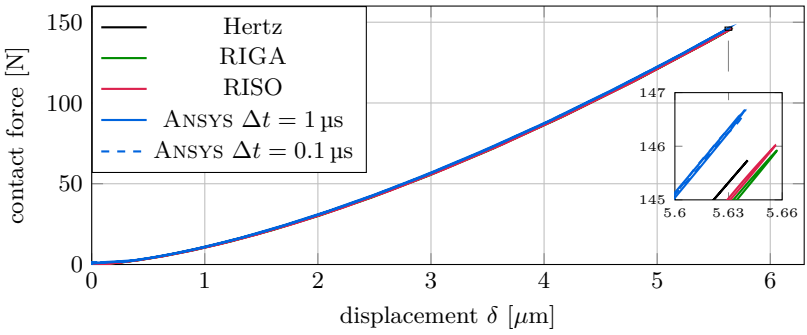


Figure 5.12: Displacement δ with respect to the contact force force, see Fig. 3.2.

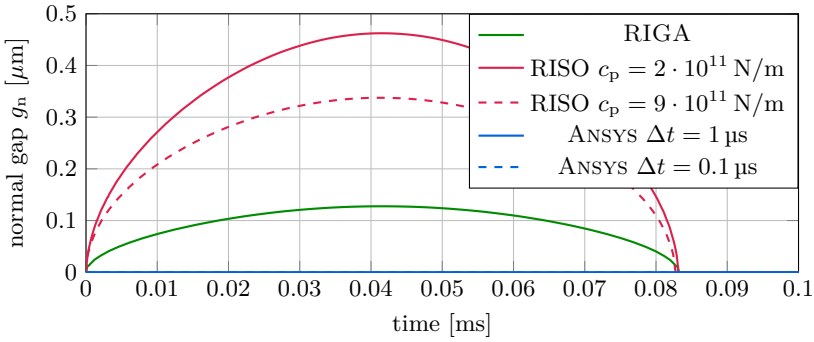


Figure 5.13: Normal gap g_n over the time.

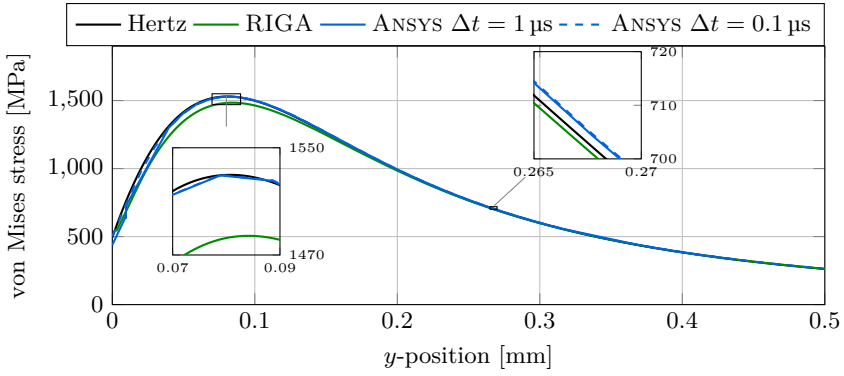


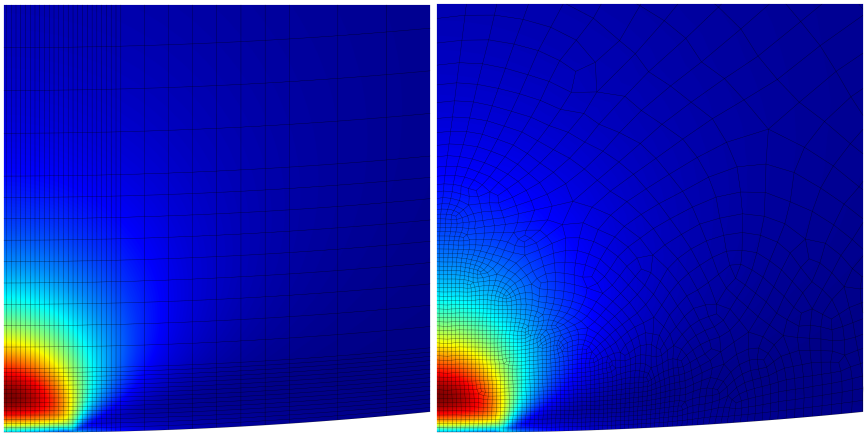
Figure 5.14: Maximum von Mises stresses along the symmetry axis.

is given by

$$E = 2\frac{1}{2}mv_0^2 \tag{5.1}$$

$$m = \frac{4}{3}\pi r^3 \rho, \tag{5.2}$$

where m is the mass of the sphere and v_0 its initial velocity. It can be seen that all models show a slight decrease of the total energy after impact, however the energy change remains well below one percent. Since the RIGA and the RISO model include damping of the high-frequency modes, the energy decreases during the impact. The energy of the Ansys solution oscillates during the impact. Using the larger time step size $\Delta t = 1\,\mu s$ results in an even larger oscillation of the energy. After the impact, the total energy of the Ansys simulations is close to the initial energy. However, this property may not be specific to isoparametric



a) RIGA. **b)** Ansys.

Figure 5.15: Von Mises Stress in the contact area.

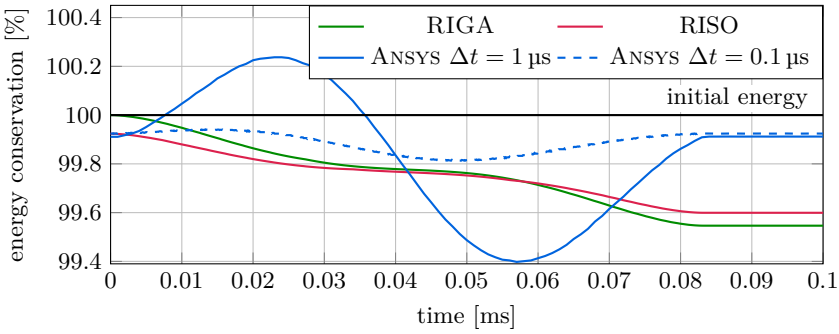


Figure 5.16: Energy of the system for test 3.

elements, but rather depends on the time integration method and the Lagrange multiplier method used.

Additionally, it appears in Fig. 5.16 that the initial energy of the RISO and the Ansys solutions are lower than the energy calculated in Eq. (5.1). Both simulations use the isoparametric model. However, the initial energy of the RIGA solution exactly matches the predetermined energy in Eq. (5.1). The reason for the difference is the mass of the models. Each model and its mass are compared in Tab. 5.3. As a reference, the mass defined in Eq. (5.2) is computed. The RIGA model exactly matches the reference mass. However, the mass of the isoparametric model is slightly lower. To be precise, the mass of the isoparametric

Table 5.3: Mass of the 2D axisymmetric spheres.

model	mass of the sphere
reference: mass according to Eq. (5.2)	$32.882\,003\,\mathrm{g} = \frac{4}{3}\pi(1\,\mathrm{cm})^3 7850\,\mathrm{kg/m^3}$
IGA model	$32.882\,003\,\mathrm{g}$
isoparametric model	$32.856\,878\,\mathrm{g}$

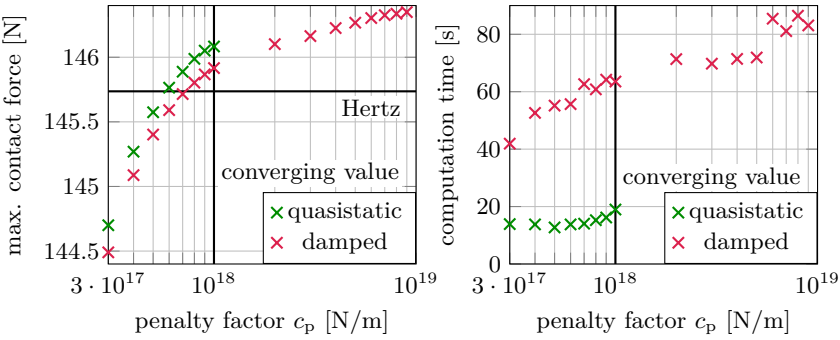
model corresponds to 99.92 % of the reference mass. This corresponds exactly to the initial energy of the RISO and Ansys solution in Fig. 5.16. The slightly lower mass may be due to discretization errors of the isoparametric elements. In the IGA, there are no discretization errors in the representation of the geometry.

To summarize the comparison between isogeometric and isoparametric models, the differences are small. The reduced IGA model is slightly more accurate and minimally more computationally expensive than the reduced isoparametric model. A general statement as to whether a simulation with RIGA is faster or more accurate than a full Ansys simulation is not easy to answer in this test. Depending on the time step size in Ansys, the computation time can be lower or higher than that of the reduced models, see Fig. 5.10. However, the energy of the Ansys solution shows large oscillations and the mass cannot be discretized exactly, see Fig. 5.16. The comparison of a RIGA model and an Ansys model is further discussed in Sect. 5.2.2 with a 3D setup.

5.1.4 Test 4: Quasistatic Contact Model

In the previous tests, the high frequency modes are critically damped to improve the computational performance of the reduced contact models. The fourth test uses quasistatic contact models for the RIGA models instead of modal damping. In the previous work [Tschigg20], the quasistatic contact model is applied to isoparametric models. The computational performance is drastically increased while the results remain accurate compared to modally damped models. In this test, it is checked whether these observations can also be made for IGA models.

In the first test in Fig. 5.3, the penalty factor has been determined with $c_\mathrm{p} = 1 \cdot 10^{18}\,\mathrm{N/m}$ for the modally damped RIGA model. In Fig. 5.17, the analysis of the convergence of the penalty factor is repeated by using the quasistatic contact model. The maximum contact force is displayed in Fig. 5.17a and the required computation time in Fig. 5.17b. For the quasistatic contact model, the maximum possible penalty factor is $c_\mathrm{p} = 1 \cdot 10^{18}\,\mathrm{N/m}$. A further increase results in a singular Jacobian $\boldsymbol{J}_\mathrm{qs}(\boldsymbol{q}_\mathrm{e}^\mathrm{hf})$ of Eq. (3.33). Overall, the results of the maximum contact force in Fig. 5.17a show similar behavior between the damped and the quasistatic contact model. Significant differences are visible in Fig. 5.17b with respect to the computation time. The quasistatic contact model takes only a



a) Maximum contact force.

b) Computation time (impact phase).

Figure 5.17: Convergence of the penalty factor for test 4.

third of the time of the damped model.

As in the previous test, the contact force is shown in Fig. 5.18. Both contact models can replicate the analytic solution by Hertz very well. The contact force of the quasistatic model is only slightly higher.

To validate the conservation of energy, the energy of the system is displayed over time in Fig. 5.19. As in Fig. 5.16 of the third test, it is found that the energy of the damped model decreases. The energy of the quasistatic contact model remains almost constant. Only at the end of the contact, at time $t \approx 0.08$ ms, the energy decreases slightly. The reason for this remains unknown. For comparison, the energy of the full Ansys simulation is displayed in Fig. 5.19. The energy changes in Ansys are larger compared to the energy of the quasistatic model.

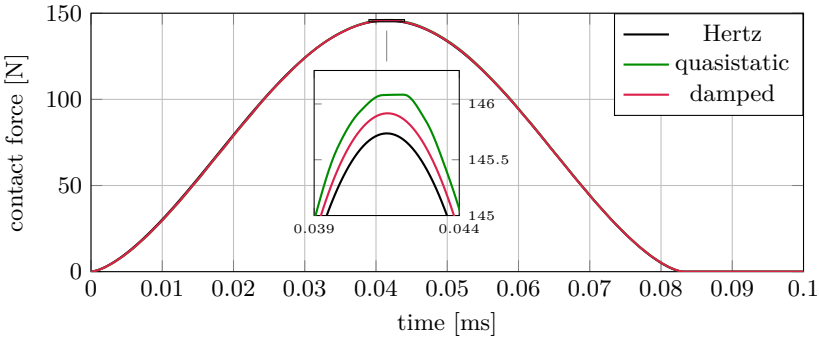


Figure 5.18: Contact force over time for test 4.

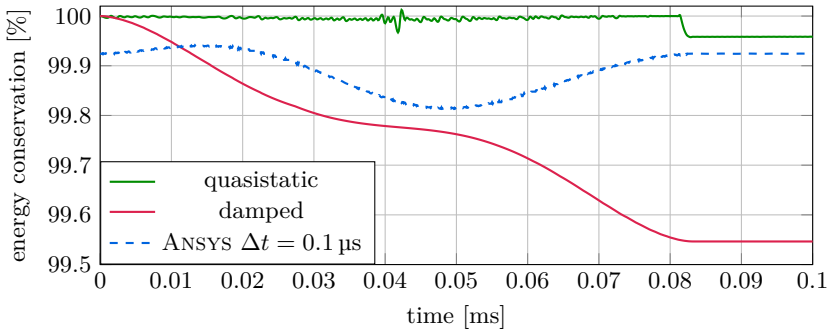


Figure 5.19: Energy of the system for test 4.

However, the Ansys solution seems to be more energy conserving since the energy at the beginning and end of the contact is almost identical.

5.1.5 Test 5: Hierarchical Refinement

The aim of this test is to compare globally and hierarchically refined IGA models in the context of an impact simulation in a flexible multibody system. It is investigated whether a hierarchically refined model has the same accuracy as an equivalent globally refined model and how hierarchically and globally refined models differ in computation time. The benchmark has already been carried out in [RückwaldEtAl21c, RückwaldEtAl22] using modally damped models. In this work, quasistatic contact models are used due to the increased computational performance, see Sect. 5.1.4.

Four hierarchically refined models are compared in the following studies, in which the number of hierarchy levels n_{lvl} is varied between three and six. Each of these four hierarchically refined models has an equivalent globally refined reference model. The equivalence is that the knot vectors of the globally refined models are identical to the knot vectors in the lowest level of the hierarchically refined models. The investigated hierarchically refined models are visualized in Fig. 5.20. Additionally, Tab. 5.4 contains more details about the models. Properties that are colored green are identical for all models, and properties colored red are different between models, see Tab. 5.4. As described in Sect. 2.5, hierarchically refined models require less elements and less control points to achieve the same level of refinement in the contact area compared to globally refined models. It can also be seen in Tab. 5.4 that a higher number of levels increases this effect. The number of low and high frequency modes as well as the number of Greville points in the contact interval is identical for all models regardless of whether

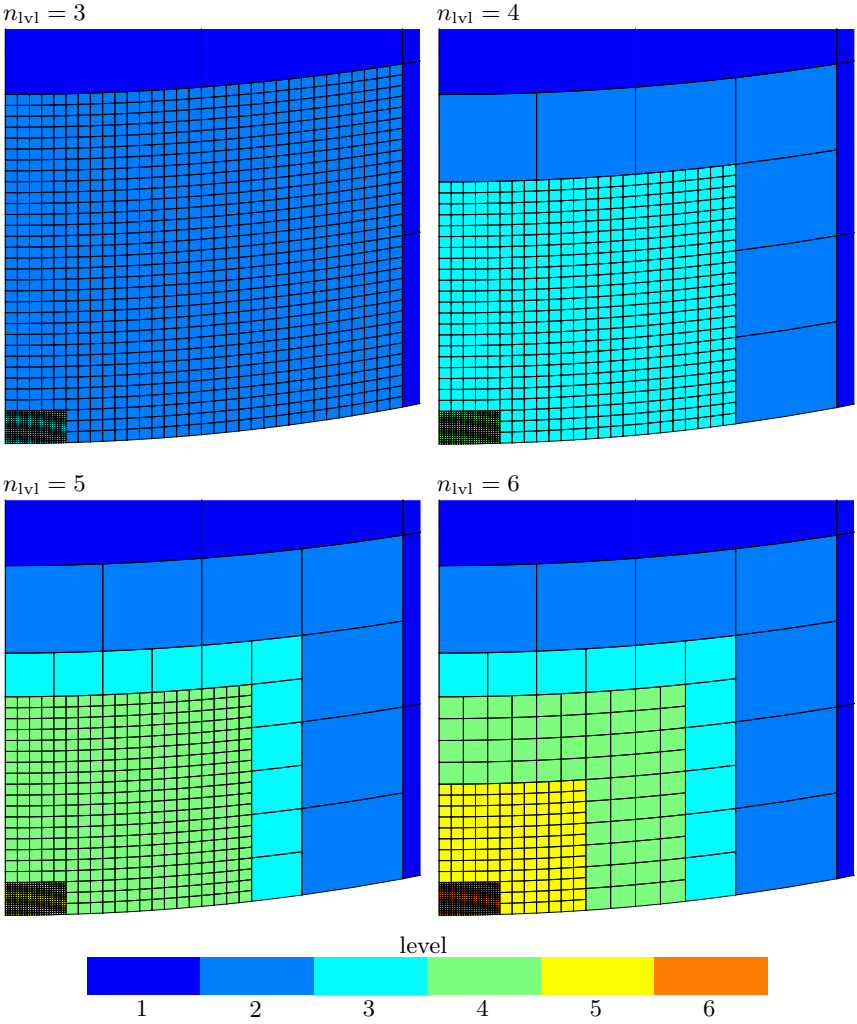


Figure 5.20: Detail plot of the contact area of the hierarchically refined axisymmetric models.

the model is globally or hierarchically refined. In terms of preprocessing, the hierarchically refined models are slightly faster because they have fewer degrees of freedom than the globally refined models. Therefore, the computation and the assembly of the mass and stiffness matrix as well as the model reduction is faster for the hierarchically refined models. As a side note, the globally refined reference model of the hierarchically refined model with $n_{\mathrm{lvl}} = 6$ levels is used in

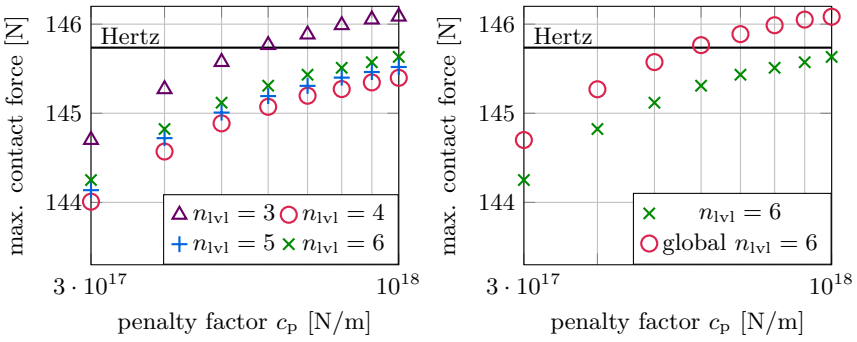
Table 5.4: Globally and locally refined models of an axisymmetric sphere.

hierarchically refined	$n_{\mathrm{lvl}} = 3$	$n_{\mathrm{lvl}} = 4$	$n_{\mathrm{lvl}} = 5$	$n_{\mathrm{lvl}} = 6$
control points n_{cp}	1940	1504	1342	1147
low frequency modes $n_{\mathrm{q}}^{\mathrm{lf}}$	10	10	10	10
high frequency modes $n_{\mathrm{q}}^{\mathrm{hf}}$	58	58	58	58
Greville points	26	26	26	26
preprocessing time	2.0 s	1.6 s	1.5 s	1.3 s
globally refined reference				
control points n_{cp}	4900	3969	3600	3136
low frequency modes $n_{\mathrm{q}}^{\mathrm{lf}}$	10	10	10	10
high frequency modes $n_{\mathrm{q}}^{\mathrm{hf}}$	58	58	58	58
Greville points	26	26	26	26
preprocessing time	5.2 s	3.3 s	3.0 s	2.5 s

all previous tests.

The influence of the penalty factor on the maximum contact force is visualized in Fig. 5.21. In Fig. 5.21a, all hierarchically refined models are tested. For each model, the maximum possible penalty factor is $c_{\mathrm{p}} = 1 \cdot 10^{18}$ N/m as in Sect. 5.1.4. All models show approximately the same convergence behavior. In Fig. 5.21b, the hierarchically refined model with $n_{\mathrm{lvl}} = 6$ levels is compared against its globally refined reference model. The maximum contact force of the globally refined model is slightly higher than that of the hierarchical model. The same observation is made in Fig. 5.22. However, both models reproduce the analytical solution by Hertz well and the differences are minor.

Finally, the influence of the number of hierarchy levels n_{lvl} on the computation



a) All locally refined models. **b)** Local model and its global reference.

Figure 5.21: Maximum contact force compared to analytic solution by Hertz.

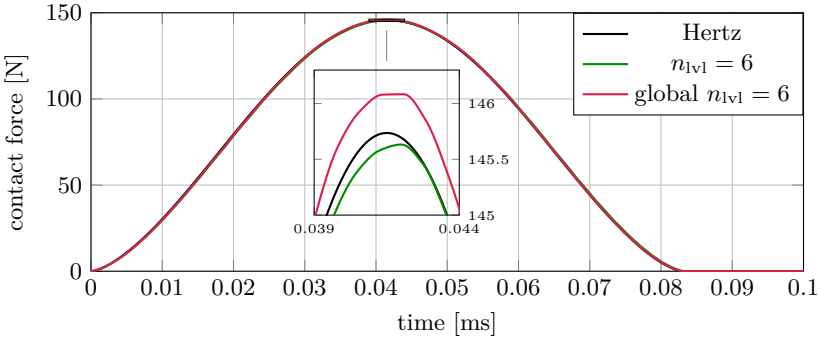
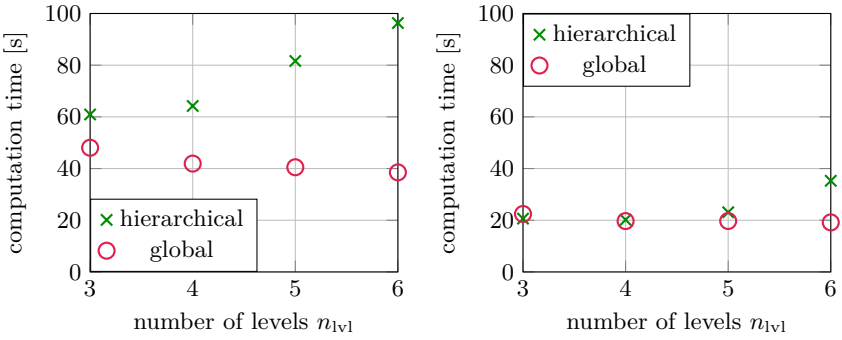


Figure 5.22: Contact force over the time.

time of the impact simulation shown in Fig. 5.23 is discussed. The computation times do not include the preprocessing, only the time integration. Although the contact search can be parallelized with respect to the evaluation points, sequentially executed simulations are more independent of the architecture of the computer, the MEX implementation of parallel for-loops, and the used operating system WINDOWS. Therefore, the sequential computation allows a better relative comparison of the individual models.

Observing the computation time of the sequential simulation in Fig. 5.23a, it can be seen that the computation time increases with the number of hierarchy levels. In contrast, the computation time of the globally refined reference models decreases. The latter observation can be attributed to the fact that the number of control points decreases, see Table 5.4. Since the globally refined models have



a) Sequential computing.

b) Parallelized computing.

Figure 5.23: Comparison of computation times.

only one level in the parameter space, it is likely that the number of control points directly influences the computation time. The number of control points n_{cp} correlates to the number of degrees of freedom n_{dof}. If the number of degrees of freedom n_{dof} is smaller, the matrix of global shape functions $\boldsymbol{\Phi} \in \mathbb{R}^{n_{\mathrm{dof}} \times \left(n_{\mathrm{q}}^{\mathrm{lf}} + n_{\mathrm{q}}^{\mathrm{hf}}\right)}$ is also smaller. Thereby, the matrix multiplications in the course of the contact algorithm, e.g. Eq. (3.16) and Eq. (3.25), are computed faster for models with fewer control points. However, this effect is rather small, as can be seen in the computation time of the globally refined models in Fig. 5.23a. This can be explained by the fact that matrix multiplications are relatively inexpensive to calculate in contrast to, for example, matrix inversions.

Although, the number of control points of the hierarchically refined models decreases as the number of hierarchy levels increases, the aforementioned effect does not seem to be dominant here. In the literature [SchillingerEtAl12, D'AngellaEtAl18, NoëlEtAl20, ZimmermannSauer17], the reduced number of control points saved by hierarchical refinement are listed as an advantage. This advantage does not apply in this example since the models are reduced with the Craig-Bampton method. The reduced mass and stiffness matrix are then used in the evaluation of the equations of motion in the context of the floating frame of reference formulation. However, the number of control points of hierarchically refined models is lower compared to globally refined models but the number of elastic coordinates $n_{\mathrm{q}}^{\mathrm{lf}}$ and $n_{\mathrm{q}}^{\mathrm{hf}}$ is identical regardless of the type of refinement, see Tab. 5.4. After model reduction, the number of control points only has an influence on the global shape functions $\boldsymbol{\Phi}$. As described before, the influence of the size of the global shape functions is rather small. Since the number and position of Greville points are also identical for the respective globally and locally refined models, the evaluation of the NURBS in the course of the contact evaluation remains the last possible reason for the difference in computation time. In fact, additional computational effort in computing hierarchical NURBS described in Sect. 2.5 is responsible for the higher computational time in Fig. 5.23a.

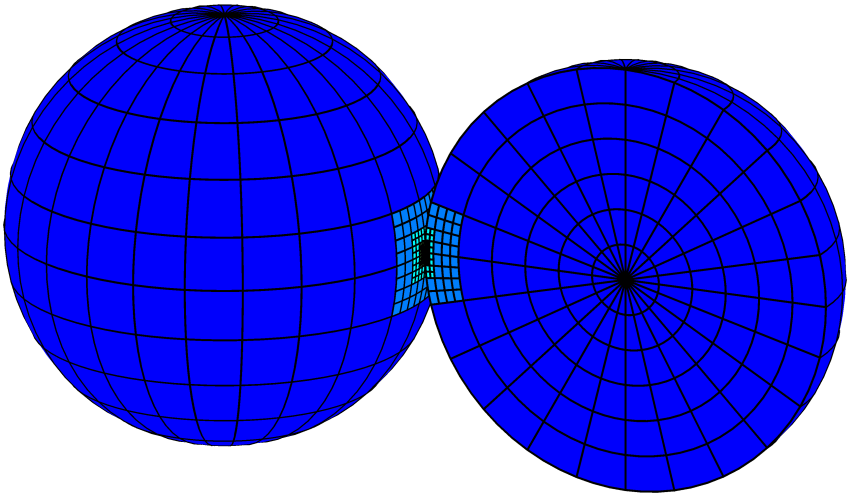
In the parallel computation shown in Fig. 5.23b, the globally refined models are likewise faster. However, the difference between the globally and locally refined models is much smaller compared to the sequential computation in Fig. 5.23a. In this case, the locally refined models seem to benefit more from the parallel computation than the globally refined models, whereby the MEX implementation could be responsible for this.

5.2 Testing Example II: 3D Spheres

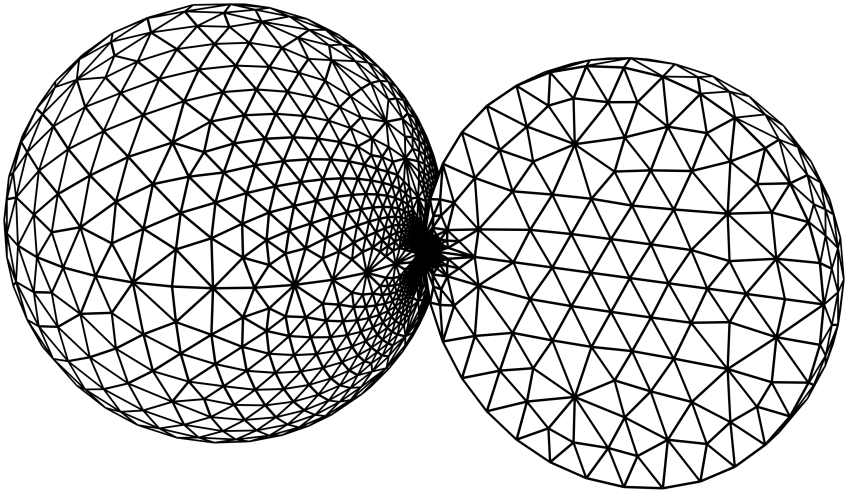
In the second testing example, the first testing example from Sect. 5.1 is extended and simulated with 3D sphere models instead of 2D axisymmetric spheres. Again, the two spheres are made of steel and have the initial velocity of $v_0 = 0.1\,\mathrm{m/s}$. The setup is shown in Fig. 5.24 in a cutaway view. This testing example is divided into two tests listed in Tab. 5.5. The focus is on the same aspects as in test 3 and test 5 from the first testing example in Sect. 5.1. However, it is being investigated whether the findings from 2D models can also be applied to 3D models. First, the study of hierarchical refinement is extended to 3D. Second, a final benchmark between Ansys and RIGA is performed. This includes an evaluation of order elevation in IGA.

Table 5.5: Goals of the second testing example.

	goal	focus of analysis
Test 1 Sect. 5.2.1	compare globally and hierarchically refined IGA models (see Sect. 2.4 and Sect. 2.5)	global refinement hierarchical refinement
Test 2 Sect. 5.2.2	final benchmark between RIGA and Ansys	RIGA (vary order elevation) Ansys (vary step size)



a) Hierarchically refined IGA model with $n_{\text{lvl}} = 5$ levels.



b) Isoparametric model.

Figure 5.24: Cutaway view of the spheres.

5.2.1 Test 1: Hierarchical Refinement

In the first test, the study on the hierarchical refinement in Sect. 5.1.5 is extended and simulated with 3D sphere models instead of 2D axisymmetric spheres.

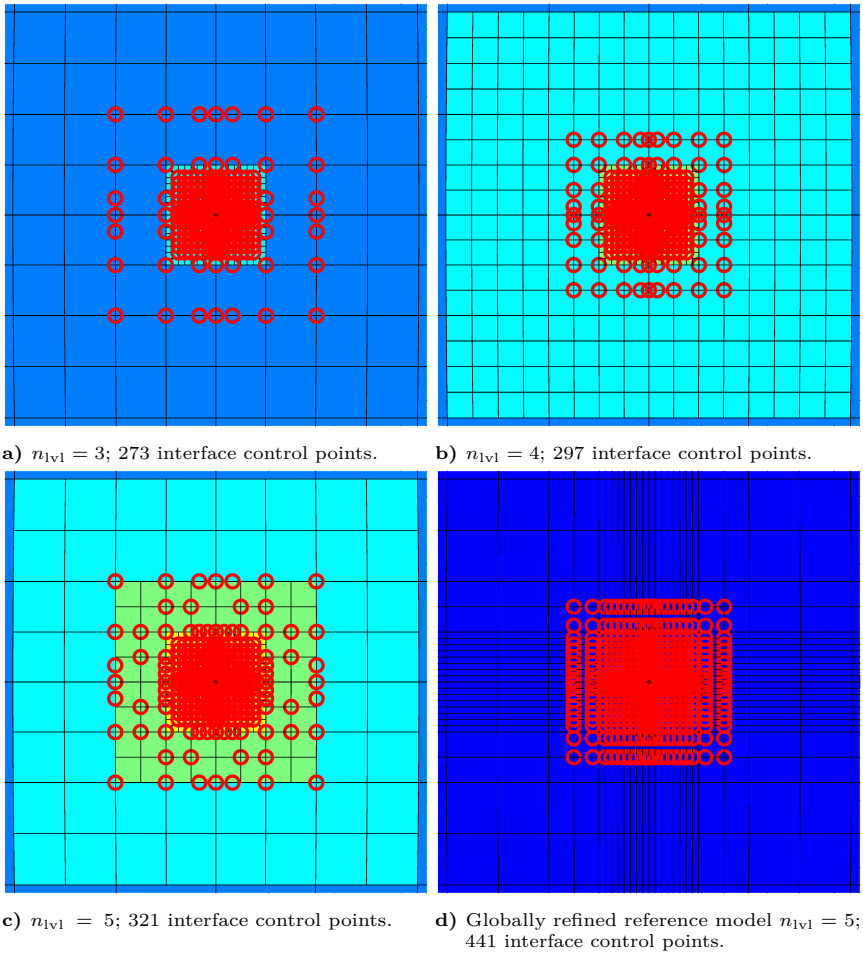
The motivation of this extension is to monitor the behavior of hierarchically refinement for 3D models. The benchmark has already been carried out in the [RückwaldEtAl22] using modally damped models. In this work, the same IGA models are used, but in combination with quasistatic contact models due to increased computational performance, see Sect. 5.1.4.

As described in Sect. 2.5, the computation of hierarchical B-splines is more expensive due to the intersection and linear combination of different hierarchy levels. This is also observed in the 2D setup in Sect. 5.1.5. The question arises whether global refinement is also computationally more efficient than hierarchical refinement due to more linear combinations of B-splines in three dimensions instead of two, or if other effects dominate. In the course of this analysis, three hierarchically refined models with three to five levels and their corresponding globally refined reference model are compared. The penalty factor is again determined to be $c_\mathrm{p} = 1 \cdot 10^{18}\,\mathrm{N/m}$ for the IGA models following the procedure as in the first testing example. The number of degrees of freedom before and after the model reduction are summarized in Tab. 5.6. Here, the lowest level of the locally refined models is defined as the contact region. The number and size of contact elements is identical for all hierarchically and globally refined models. For each model, the order of B-splines is elevated by $n_\mathrm{p} = 1$.

The differences in the number of control points n_cp, the number of high frequency modes n_q^lf, the Greville points, and the preprocessing time are now significantly larger compared to the 2D example in Sect. 5.1.5. As in the 2D example in Sect. 5.1.5, the number of control points n_cp decreases when the number of levels increases. However, the number of high frequency modes n_q^hf varies for the hierarchically refined models and is only constant for the globally refined models. However, it is noticeable that all locally refined models have less high frequency modes than the globally refined ones. The reason for this difference can be monitored in Fig. 5.25. The number of elastic coordinates corresponds to

Table 5.6: Globally and locally refined models of a 3D sphere.

hierarchically refined	$n_\mathrm{lvl} = 3$	$n_\mathrm{lvl} = 4$	$n_\mathrm{lvl} = 5$
control points n_cp	7411	6095	5350
low frequency modes n_q^lf	25	25	25
high frequency modes n_q^hf	819	891	963
Greville points	237	241	249
preprocessing time	46 s	40 s	39 s
globally refined reference			
control points n_cp	52 155	44 415	35 518
low frequency modes n_q^lf	25	25	25
high frequency modes n_q^hf	1323	1323	1323
Greville points	289	289	289
preprocessing time	14 min	12 min	9 min



a) $n_{\text{lvl}} = 3$; 273 interface control points.

b) $n_{\text{lvl}} = 4$; 297 interface control points.

c) $n_{\text{lvl}} = 5$; 321 interface control points.

d) Globally refined reference model $n_{\text{lvl}} = 5$; 441 interface control points.

Figure 5.25: Contact areas of spheres for various refinement levels

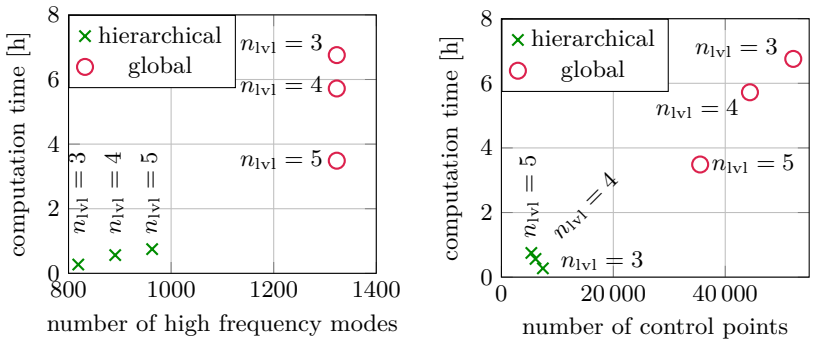
the number of interface control points, of which the model in Fig. 5.25a has the fewest. Considering Fig. 5.25, it can be seen that the respective levels adjacent to the contact areas have different element resolutions. The difference in element resolution between the lowest level and the second lowest level is largest for the model with $n_{\text{lvl}} = 3$ levels in Fig. 5.25a. Therefore, the number of interface

control points at the edges of the hierarchical model with $n_{\mathrm{lvl}} = 3$ levels is the lowest compared to the other models. As a side note, the difference in element resolution does not depend on the number of hierarchy levels. It is a design choice how the knots are inserted with respect to the levels.

The effect of the differences is visualized in Fig. 5.26 representing the computation times. In contrast to the 2D setup in Sect. 5.1.5, the simulations are only evaluated with parallel computing. Figure 5.26a shows that the hierarchical model with $n_{\mathrm{lvl}} = 3$ levels requires the least amount of computation time. The same observation was made in the 2D setup in Fig. 5.23 and was reasoned with the number of intersecting levels. In this 3D setup, the effect is further enhanced by the small number of elastic coordinates. The hierarchical model with $n_{\mathrm{lvl}} = 5$ levels takes the most computation time of all hierarchically refined models due to the large number of high frequency modes, see Fig. 5.26a. However, all hierarchical models require less computation time than the globally refined reference models.

Although the number of high frequency modes is identical for all globally refined models in Fig. 5.26a, there are significant differences in computation time. In contrast to 2D, the differences in the number of control points become more relevant in 3D, see Fig. 5.26b. As mentioned in Sect. 5.1.5, the number of degrees of freedom n_{dof} influences the size of the matrix of global shape functions $\boldsymbol{\Phi} \in \mathbb{R}^{n_{\mathrm{dof}} \times \left(n_{\mathrm{q}}^{\mathrm{lf}} + n_{\mathrm{q}}^{\mathrm{hf}}\right)}$. Thereby, the matrix multiplications in the course of the contact algorithm, e.g. Eq. (3.16) and Eq. (3.25), are computed more slowly for models with more control points. Therefore, the globally refined model corresponding to the model with $n_{\mathrm{lvl}} = 3$ levels takes the longest computation time.

The small number of elastic coordinates of the hierarchically refined models



a) Computation time compared to $n_{\mathrm{q}}^{\mathrm{hf}}$.

b) Computation time compared to n_{cp}.

Figure 5.26: Computation times of hierarchical and global refinement in 3D.

compared to the globally refined models raises the questions whether the reduced computation time is bought by a reduced accuracy. However, there are no significant differences in accuracy between globally and hierarchically refined models, see Fig. 5.27. This is also observed in [RückwaldEtAl22]. Since the outer contact area is less important, the number of interface control points of the globally refined model could be further reduced. By hand-picking the interface control points in the outer contact area, a simulation time which is equivalent to a hierarchically refined model may be achieved. However, the hierarchical refinement achieves a reduced number of elastic coordinates fully automatically. This is beneficial for the adaptive refinement presented in the next chapter.

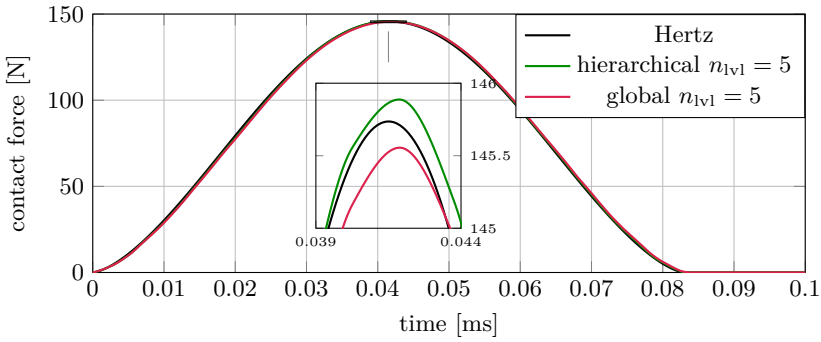


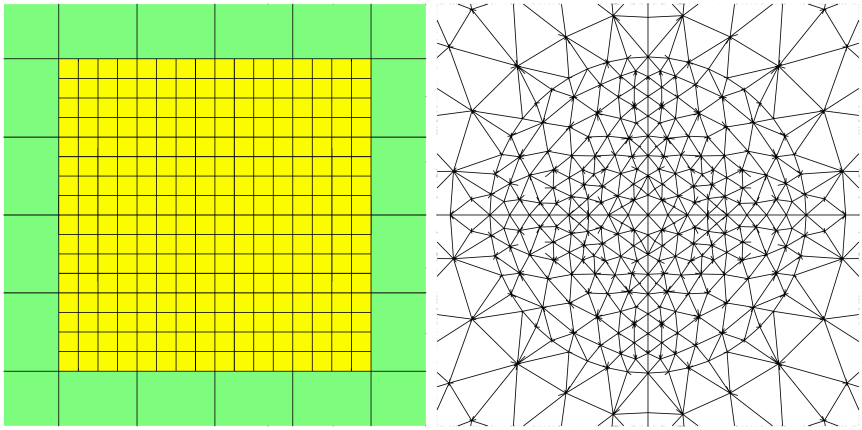
Figure 5.27: Contact force over time for test 1.

5.2.2 Test 2: RIGA and Ansys

The second test is a final benchmark between RIGA and Ansys. The goal is to show that it is not easy to compare a commercial finite element program such as Ansys with a research toolbox such as RIGA. It is observed in Sect. 5.1.2 that order elevation introduces a trade-off between accuracy and computation time. The same trade-off is found in Sect. 5.1.3 where the effect of the step size in Ansys is observed. This test focuses on these two tuning parameters. However, there are many more parameters that can be tuned, e.g. optimizing the model and solver settings.

The IGA models used is the hierarchically refined model with $n_{\mathrm{lvl}} = 5$ of the previous test in Sect. 5.2.1. Based on its element resolution in the contact area, an equivalent isoparametric model is generated. The contact area of both models can be compared in Fig. 5.28. The isoparametric model uses tetrahedral elements because Ansys does not support automatic meshing with hexahedral elements. In total, the IGA model consists of 2700 elements and the isoparametric model consists of 21724 elements.

Four simulations are compared in this test. As in Sect. 5.1.3, the Lagrange multiplier method is selected in Ansys, and the time step sizes $\Delta t = 1\,\mu\mathrm{s}$ and $\Delta t = 0.1\,\mu\mathrm{s}$ are used. The IGA model uses variable step size control in Matlab. For the IGA model, the model is simulated without order elevation $n_{\mathrm{p}} = 0$ and with order elevation $n_{\mathrm{p}} = 1$. As side note, in IGA, order elevation does not influence the shape of the elements. Depending on the requirements on accuracy and computation time, the same IGA model can be simulated with different



a) Hierarchically refined IGA model. **b)** Isoparametric model simulated in Ansys.

Figure 5.28: Contact area of the refined IGA and isoparametric model of a 3D sphere.

order elevations. As shown in Tab. 5.7, order elevation influences the number of control points and the number of degrees of freedom in the contact area.

The computation times are presented in Fig. 5.29. As with the 2D test in Sect. 5.1.2, increasing the order and decreasing the time step size increase the computation time. With the parameters selected in this test, the IGA is minimally faster.

Regarding the contact force, the time step size has only minor influence on the accuracy, see Fig. 5.30. In contrast, the order of the B-splines has a larger influence on the accuracy. The simulation with order elevation represent the analytical solution more accurately. Comparing RIGA with Ansys, the accuracy is about the same. The difference is very small.

Finally, the energy conservation is compared in Fig. 5.31. In contrast to the 2D test in Fig. 5.19, the isoparametric model can represent the initial energy more accurately. This is because of a more detailed refinement to better approximate the mass of the sphere. In the 2D test in Tab. 5.3, the mass of the isoparametric model was already inaccurate at the second digit after the decimal point. In this test, the mass only deviates at the third digit, see Tab. 5.8. The mass of the IGA models is again accurate because the IGA can represent the geometry exactly without discretization errors. If more digits after the decimal point are considered, differences also occur with the IGA models. However, these can be attributed to the machine accuracy.

Considering the course of the energy in Fig. 5.31, the energy of the Ansys simulation deviates more from the reference during the impact than the solution of RIGA. The simulation with the larger step $\Delta t = 1\,\mu s$ size is significantly less accurate than the simulation with the smaller step size $\Delta t = 0.1\,\mu s$. The observation is different for the energy after the impact at $t \approx 0.085\,ms$. Here, the Ansys simulations preserve the energy more accurately than the solutions of RIGA.

Finally, it is not easy to determine whether Ansys or RIGA is faster. It depends on the accuracy which is desired. The IGA can represent the mass more accurately with a significantly lower number of elements compared to isoparametric elements. The same IGA model can be simulated with different accuracy by easily changing the order elevation. In this test, RIGA is faster than Ansys. However, the

Table 5.7: Hierarchically refined 3D sphere with and without order elevation.

hierarchically refined: $n_{\text{lvl}} = 5$	$n_{\text{p}} = 0$	$n_{\text{p}} = 1$
control points n_{cp}	3859	5350
low frequency modes n_{q}^{lf}	25	25
high frequency modes n_{q}^{hf}	795	963
Greville points	241	249

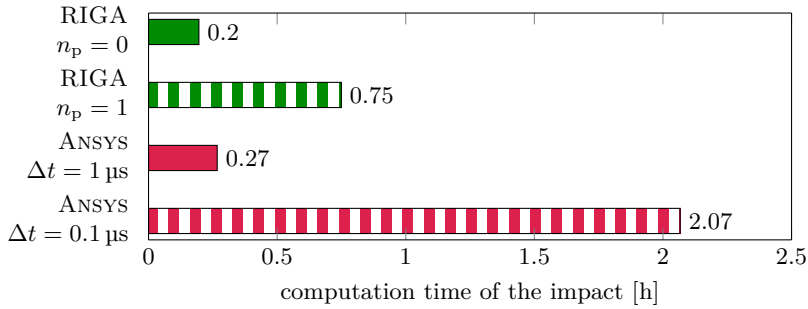


Figure 5.29: Comparing the computation time of the different discretization methods.

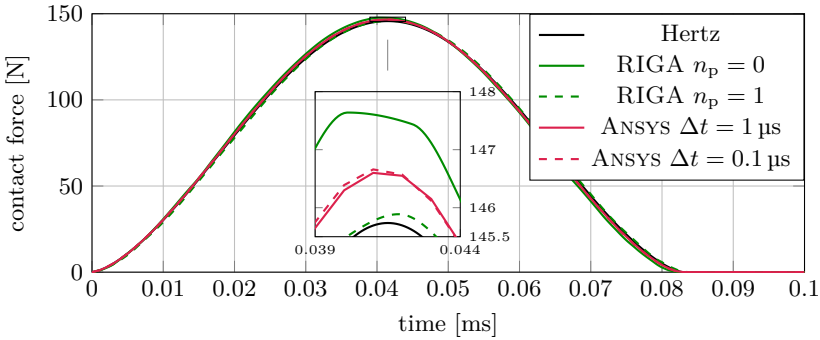


Figure 5.30: Contact force over time for test 2.

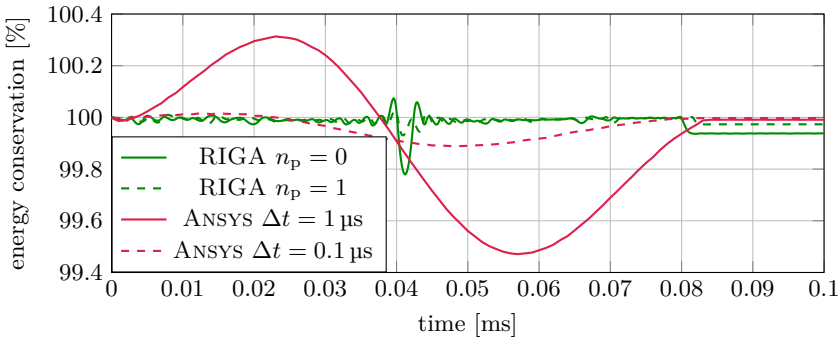


Figure 5.31: Energy of the system for test 2.

Table 5.8: Mass of the 3D spheres.

model	mass of the sphere
reference: mass according to Eq. (5.2)	$32.882\,003\,\mathrm{g} = \frac{4}{3}\pi(1\,\mathrm{cm})^3 7850\,\mathrm{kg/m^3}$
IGA model ($n_\mathrm{p} = 0$ and $n_\mathrm{p} = 1$)	$32.882\,003\,\mathrm{g}$
isoparametric model	$32.881\,338\,\mathrm{g}$

accuracy of Ansys can be reduced be selecting a coarser mesh and using a large time step size, resulting in higher computational performance.

Besides the tuning parameters, such as order elevation or step size, comparing Ansys with RIGA in terms of computation time is not easy due to the implementation. Ansys is a commercial finite element program developed for many years and uses fast programming languages such as C, C++, and Fortran. RIGA is a new research toolbox implemented in Matlab.

As motivated in the introduction in Chap. 1, using the floating frame of reference formulation allows fast simulation of systems undergoing large rigid body motion. In contrast to Ansys, the "free flying" phases between impacts can be simulated relatively fast using flexible multibody systems. An adequate adaptive application example is covered in Sect. 7.2. First, however, an adaptive procedure to automatically simulate impacts in flexible multibody systems is presented.

Adaptive Procedure for Impact Simulations

Usually, an impact in a flexible multibody system simulation requires burdensome manual setup. The user setting up the flexible multibody system needs to predetermine the location of the impact point, refine the model in the respective contact area, and determine the penalty factor for which the results converge to the physical correct contact force. Based on the insights obtained in Chap. 5, an adaptive procedure is presented here to adaptively simulate impacts without manual setup. This chapter closely follows [RückwaldEtAl24], where this approach was published. The procedure is shown in Fig. 6.1 and includes the automation of the flexible multibody setup and the determination of the three aforementioned variables. This is accomplished by dividing the process into three parts: non-impact phase, adaptive processing, and impact phase. In the beginning, the

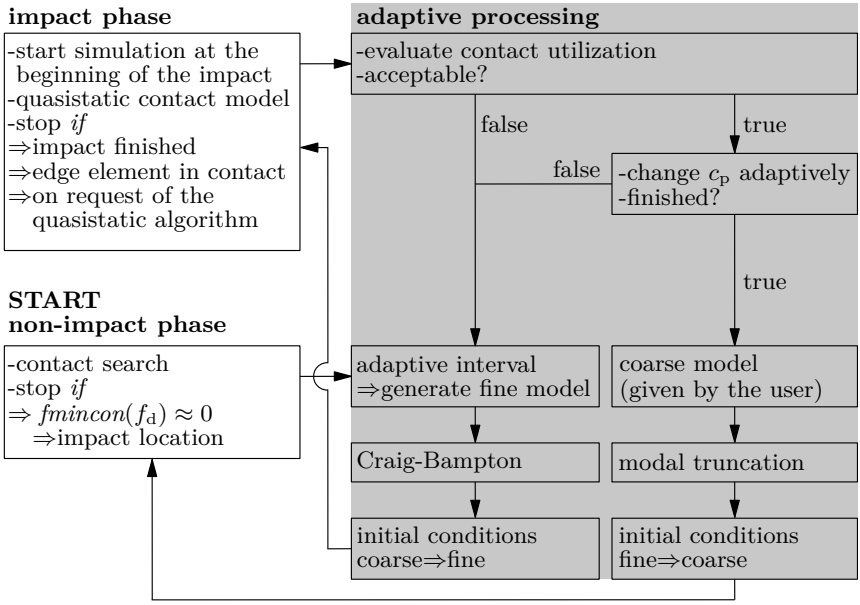


Figure 6.1: Overall procedure of the adaptive impact simulation.

user sets up the rigid and flexible bodies. The flexible bodies, which might get in contact, are represented by IGA models and have already been coarsely refined. The coarse models are globally refined via k-refinement to represent the overall elastic deformation and are used in phases between impacts. As described in Sect. 2.4, k-refinement first increases the order of the B-splines and then inserts additional knots. Regarding model reduction, coarse models are modally truncated by the first n_q normal modes.

The model refined later in the process is called the fine model. Fine models are hierarchically refined models that can represent the overall elastic deformation and the local deformations in the contact area, see Sect. 2.5. The Craig-Bampton method [CraigBampton68] will be applied to the fine model used in the impact phase. For the Craig-Bampton method, the number of low frequency normal modes n_q^lf is identical to the number of normalmodes n_q of the coarse model reduced by modal truncation. Thus, the number of the equations of motion from Eq. (2.5) remains constant throughout the adaptive simulation. Without the quasistatic contact model, the number of equations of motion would change and the presented algorithm would not work. This is because otherwise the n_q^hf constraint modes would be part of the equations of motion.

For the solution of the equations of motion (2.5), different integrators and settings are chosen. The non-impact phase uses the MATLAB *ode23tb* [Mathworks23] integrator with the maximum time step size $\Delta t_\mathrm{non-impact}^\mathrm{max} = 0.1\,\mathrm{ms}$ for the examples presented. The maximum step size $\Delta t_\mathrm{impact}^\mathrm{max} = \frac{1}{2f^\mathrm{max}}$ in the impact phase is determined by the Nyquist-Shannon sampling theorem, see e.g. [Pintelon12], where f^max is the maximum eigenfrequency within the system. An explicit integrator is selected for the impact phase. An implicit integrator would chain the solution of two sets of nonlinear equations, the quasistatic contact equation (3.33) and the equations of motion (3.34), into each other. Instead, the explicit MATLAB *ode45* [Mathworks23] integrator is modified. Two modifications are made. First, contact forces and high frequency elastic coordinates are saved during the time integration. Otherwise, post-processing is as time-consuming as the simulation itself. Second, the integrator can reset the quasistatic contact model. Usually, MATLAB integrators use variable step sizes. If an integration step is discarded, the initial conditions of Alg. 1, see Sect. 3.5, i.e. the Jacobian $\boldsymbol{J}_\mathrm{qs}(\boldsymbol{q}_\mathrm{e}^\mathrm{hf})$, from the last successful step are used. In the following, the procedure presented in Fig. 6.1 is detailed:

The simulation starts with the non-impact phase shown in the lower left of Fig. 6.1. The goal of the non-impact phase is to simulate large rigid body motions with a large time step size and to detect contacts. To detect a contact during the non-impact phase, the distance and the impact location can be determined with the distance function f_d. If the distance function value

$$f_\mathrm{d}(\xi_\mathrm{C}, \eta_\mathrm{C}, \zeta_\mathrm{C}, \xi_\mathrm{T}, \eta_\mathrm{T}, \zeta_\mathrm{T}) = ||\boldsymbol{r}_\mathrm{C}(\xi_\mathrm{C}, \eta_\mathrm{C}, \zeta_\mathrm{C}) - \boldsymbol{r}_\mathrm{T}(\xi_\mathrm{T}, \eta_\mathrm{T}, \zeta_\mathrm{T})||_2 \tag{6.1}$$

is minimized with the MATLAB optimizer *fmincon* [Mathworks23] and $f_\mathrm{d} \approx 0$ is obtained, a contact occurs. The minimization of the distance function then yields the physical points $\boldsymbol{r}_\mathrm{C}$ and $\boldsymbol{r}_\mathrm{T}$ on the bodies where they are closest, and their corresponding local coordinates ξ_C, η_C, ζ_C, ξ_T, η_T and ζ_T in the parameter space. This will be the location of the impact and this information will be necessary for the generation of the adaptive model. To reduce the computational effort of minimizing Eq. (6.1), the minimization is not performed in every time step but with a larger time step size Δt_search. As a consequence, the contact and target body may already penetrate each other in the non-impact phase. Therefore, the over-calculated time steps are discarded until no more contact occurs. Then, the impact phase is simulated. Another well known and computationally more efficient approach is the use of bounding spheres or boxes [Wriggers06]. They have been already applied in the IGA [GaoEtAl19, OishiYagawa20]. If the geometries of involved IGA bodies are more complex and non-convex, bounding spheres or boxes may be required in addition to minimizing Eq. (6.1). This is because it is challenging to find the global minimum of non-convex functions. However, since this work focuses on simple geometries typically used in research, bounding spheres or boxes are not additionally used in this work.

The next phase in Fig. 6.1 is the adaptive processing. The goal is to determine the penalty factor and to generate fine models for the impact phase. The fine model is based on its contact interval. The contact interval is defined in the IGA parameter space and indicates where the contact is checked on the contact and target body. Within the contact interval, there are several elements that can come into contact. The center of the interval is determined by minimizing the distance function f_d in Eq. (6.1). The width of the contact interval is determined adaptively. The procedure is explained in Sect. 6.2. An initial fine model is used in the first iteration of adaptive processing. For this model, knots are automatically inserted in the center of the contact interval based on the coarse model provided by the user. The procedure is described in Sect. 6.3. The fine model is reduced with the Craig-Bampton method. Although the overall shape of the eigenmodes of the fine and the coarse model is identical, the global shape functions $\boldsymbol{\Phi}$ and thus the elastic coordinates $\boldsymbol{q}_\mathrm{e}$ vary. Therefore, the initial conditions of the elastic coordinates $\boldsymbol{q}_\mathrm{e}$ are transformed from the coarse to the fine model. The procedure of this transformation is described in the following Sect. 6.4.

After the adaptive processing, the impact phase starts. The quasistatic contact model in Alg. 1 is evaluated. Initially, a user given penalty factor is used. The impact simulation stops if the impact is finished, e.g. no further contact forces occur. The simulation will also stop if an element at the edge of the contact interval is in contact, or the quasistatic contact algorithm requests a stop, e.g. due to numerical issues. If an edge element is in contact, the width of the contact interval is too small and it is pointless to continue the simulation. Therefore, the model is updated and the impact simulation starts again.

Back to the adaptive processing, the utilization of the contact interval is evaluated. If the utilization is not acceptable, the width of the interval is adaptively modified and a new fine model is generated. The evaluation of utilization and the adaptive modification of the interval is detailed in the following Sect. 6.2. If the utilization is acceptable, the penalty factor can be adjusted according to the following Sect. 6.1. If the penalty factor is not yet finally determined, the penalty factor is modified, the fine model is updated and a new iteration begins. The impact phase terminates successfully, if the impact is fully simulated and the utilization and penalty factor are acceptable. In the adaptive processing, the user given coarse model is reused, it is modally truncated, and the initial conditions are transformed. The procedure now returns back to the non-impact phase. Therefore, the initial conditions of the elastic coordinates $\boldsymbol{q}_\mathrm{e}$ are transformed from the fine to the coarse model. The procedure is described in the following Sect. 6.4.

6.1 Adaptive Penalty Factor

In the course of the adaptive impact simulation in Fig. 6.1, the penalty factor is determined adaptively. When simulating multibody systems, the corresponding penalty factor c_p is often chosen heuristically or by hand. As mentioned before, the penalty factor should be chosen large enough such that the results become somewhat independent of the chosen parameter [SeifriedEtAl03]. This trial-and-error approach was used to determine the penalty factor in the works [RückwaldEtAl21b, RückwaldEtAl21c, RückwaldEtAl21d, RückwaldEtAl22, RückwaldEtAl23], where the high frequency modes were critically damped. If the penalty factor is increased beyond its converging value, the equations of motion (2.5) become numerically stiffer. This increases the computation time, or the numerical integration might even terminate unsuccessfully.

In this work, the penalty method is combined with the quasistatic contact model. The trial-and-error approach for the determination of the penalty factor also works here. It turns out, that the value for which the penalty factor converges is identical for the modally damped contact model and quasistatic contact model. In practice, however, the penalty factor cannot be increased beyond its converging value as for a modally damped model. If the penalty factor is too high, numerical errors occur or the algorithm terminates as described in Alg. 1. In practice, it is desired that the penalty factor is determined automatically in order to obtain an accurate result. Due to the numerical efficiency of the quasistatic contact model, it is coupled with the following method to automatically determine the penalty factor.

For a simpler parametrization, the penalty factor is composed of a multiplier $c_\mathrm{p}^\mathrm{mul}$ and an exponential part $c_\mathrm{p}^\mathrm{exp}$ via

$$c_\mathrm{p} = c_\mathrm{p}^\mathrm{mul} \cdot 10^{c_\mathrm{p}^\mathrm{exp}}. \tag{6.2}$$

This work proposes the procedure in Fig. 6.2 to determine the penalty factor. The procedure is divided into two phases: quick increase and slow decrease of the penalty factor. In the first phase, the impact is not completely simulated, but only for $n_\mathrm{con} = 30$ time steps. Testing shows that if there are no errors in the first $n_\mathrm{con} = 30$ time steps of the contact, the rest of the contact will also be mostly error free. If no error occurs, the exponent $c_\mathrm{p}^\mathrm{exp}$ is increased by 1 and the impact simulation restarts. If an error occurs, the second phase begins and the penalty factor is decreased slowly by reducing the multiplicator $c_\mathrm{p}^\mathrm{mul}$ by 1. Then, the simulation continues at the time step, where the error occurred. During the second phase, it may occur that the penalty factor has to be reduced multiple times online.

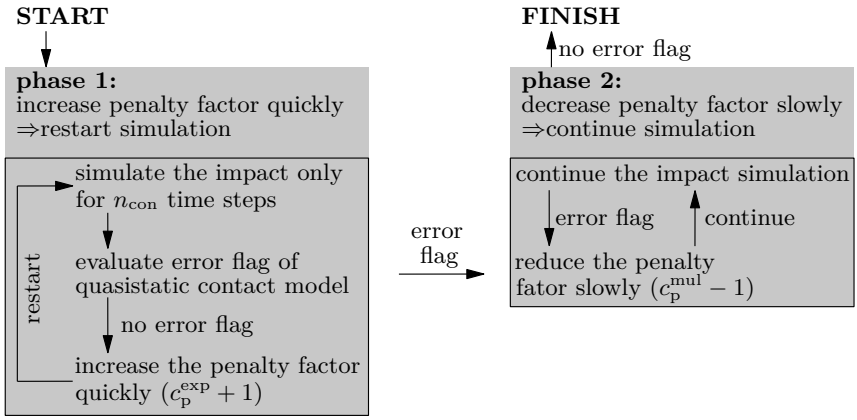


Figure 6.2: Adaptively determine the penalty factor c_p.

6.2 Adaptive Width of the Contact Interval

Besides the penalty factor, the model is refined adaptively. The model is generated in three steps: Firstly, the utilization of the contact interval in the parameter space is determined after an impact simulation. As mentioned in Fig. 6.1, the impact simulation stops when the impact is fully simulated, an edge element is in contact, or the quasistatic algorithm requests a stop, see Fig. 6.2. Secondly, the width of the contact interval is modified based on the so-called utilization $\mathcal{U}_{\mathrm{intvl}}$. The utilization depends on how many elements within the contact interval are actually in use. Thirdly, the model is generated considering the updated contact interval. After that, the fine model is reduced with the Craig-Bampton method, and the impact simulation is performed again.

The determination of the utilization $\mathcal{U}_{\mathrm{intvl}}$ of the contact interval depends on the type of the model. It is distinguished between a 2D model (plane strain or plane stress), an axisymmetric 2D model, and a 3D model. Figure 6.3 shows an example for each case. In the first two cases, the contact interval is 1D representing an exterior curve of a surface. The contact interval of a 3D solid is represented by a 2D exterior surface. Figure 6.3 shows three different types of elements: interval elements, edge elements, and contact elements. All elements inside the interval are interval elements. Elements at the edge of the interval are named edge elements. Contact elements are the elements, which are actually in contact. To determine the utilization $\mathcal{U}_{\mathrm{intvl}}$ of the contact interval, the number of elements $n^{\mathrm{ele}}_{\mathrm{contact}}$ that are in contact is divided by the number of elements in the contact interval $n^{\mathrm{ele}}_{\mathrm{intvl}}$. For a 3D model, the utilization is considered individually for both directions.

In the next step, the width of the contact interval is updated based on the utilization. The procedure is detailed in Alg. 3. In line 1, if any contact element is an edge element, the contact interval is too small and the impact seems to spread over a larger area. The model is classified as unacceptable, which prevents the penalty factor from being adaptively changed as shown in Fig. 6.2. The width of the interval is increased by 50 % across the board. The model is also classified as unacceptable in line 4 if the utilization is below the minimum utilization $\mathcal{U}^{\mathrm{min}}_{\mathrm{intvl}} = 50\,\%$. As a consequence, the width of the interval is decreased. The updated width w_{intvl} is determined with

$$w_{\mathrm{intvl}} \Leftarrow w_{\mathrm{intvl}} \frac{\mathcal{U}_{\mathrm{intvl}}}{\mathcal{U}^{\mathrm{target}}_{\mathrm{intvl}}}. \tag{6.3}$$

The maximum rate of change is limited to 50 %. For the next two cases in line 7 and 10, the model is acceptable, and the penalty factor can be adjusted according to Fig. 6.2. The target value of the utilization is $\mathcal{U}^{\mathrm{target}}_{\mathrm{intvl}} = 70\,\%$. If the utilization is below the target value, the width w_{intvl} of the interval is decreased. Otherwise the width w_{intvl} is increased. The updated width is determined with Eq. (6.3).

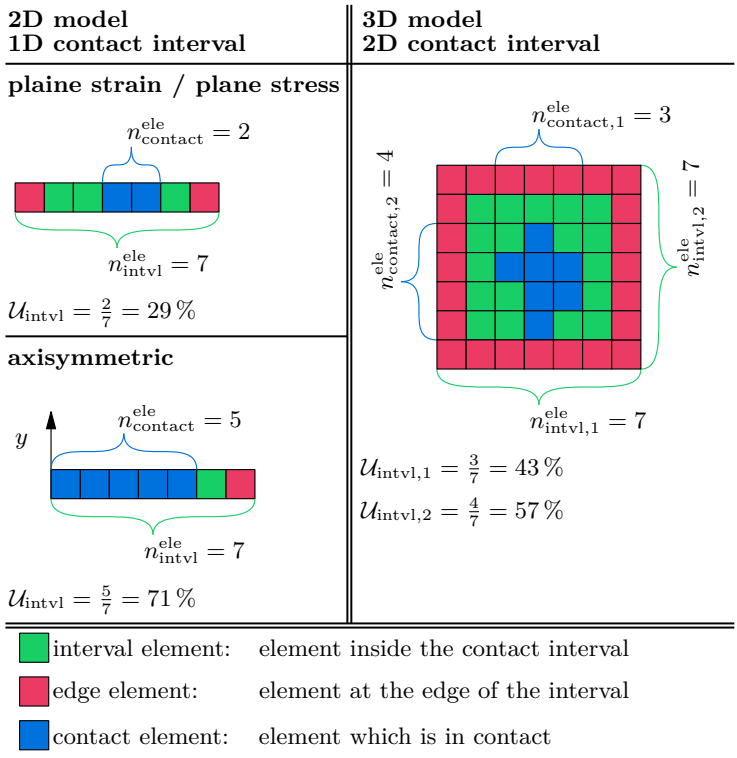


Figure 6.3: Examples of how the utilization of the contact interval is determined.

Algorithm 3 Adaptive adjustment of the contact interval.

if any '*contact element*' is an '*edge element*' **then** ▷ interval is too small
  model is not acceptable ▷ penalty factor cannot be modified
  increase width of interval by 50 %
else if $\mathcal{U}_{\text{intvl}} < \mathcal{U}^{\text{min}}_{\text{intvl}}$ **then** ▷ $\mathcal{U}^{\text{min}}_{\text{intvl}} = 50\,\%$
  model is not acceptable ▷ penalty factor cannot be modified
  decrease width of interval, see Eq. (6.3)
else if $\mathcal{U}_{\text{intvl}} < \mathcal{U}^{\text{target}}_{\text{intvl}}$ **then** ▷ $\mathcal{U}^{\text{target}}_{\text{intvl}} = 70\,\%$
  model is acceptable ▷ penalty factor can be modified
  decrease width of interval, see Eq. (6.3)
else if $\mathcal{U}_{\text{intvl}} > \mathcal{U}^{\text{target}}_{\text{intvl}}$ **then** ▷ $\mathcal{U}^{\text{target}}_{\text{intvl}} = 70\,\%$
  model is acceptable ▷ penalty factor can be modified
  increase width of interval, see Eq. (6.3)
end if

6.3 Adaptive Contact Model

In the previous Sect. 6.2, the width w_{intvl} of the contact interval is adaptively adjusted and the center of the contact interval is determined during the non-impact phase by minimizing the distance function f_{d} in Eq. (6.1). With this information, the adaptive model can be built in this section. As a reminder, the width and the center of the interval are given in the parameter space. Likewise, the model is generated in the parameter space. In the IGA, element refinement is achieved by inserting knots. The procedure is divided into two steps. Firstly, the interval is delimited by inserting knots. Secondly, knots representing the contact elements are inserted within the interval.

At the beginning of the first step, the coarse model defined by the user is given in the upper part of Fig. 6.4. It shows the existing knots spanning up the elements and the contact interval in the parameter space. The goal is to insert knots until both boundaries of the interval are represented by knots. This procedure is illustrated by an example in Fig. 6.4. In this example, three knots are required to fit the lower and upper boundary of the interval. In practice, the interval does not have to be exactly represented by knots. A one percent deviation from the

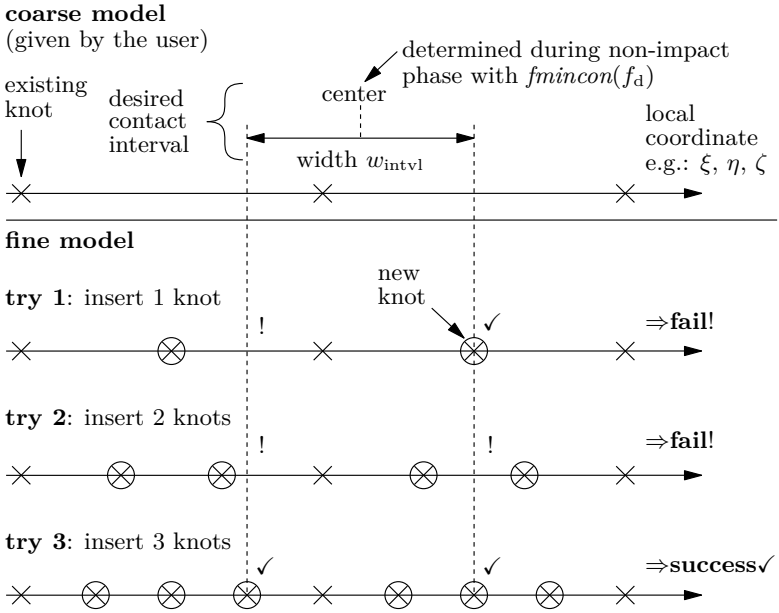


Figure 6.4: Iteratively build the fine model by inserting knots.

width of the contact interval is sufficient.

However, the goal is for the refined model to be hierarchically refined. As mentioned before in Sect. 2.5, hierarchical refinement reduces the number of elements and degrees of freedom, see Fig. 2.8. Especially for 3D IGA models, hierarchical refinement drastically reduces the computation time, see Fig. 5.26 in Sect. 5.2.1. For the hierarchical model generation, the number of levels is limited to $n_{\text{lvl}}^{\text{max}} = 5$ and the maximum of inserted knots per level is limited to $n_{\text{knots}}^{\text{max}} = 10$. The number of levels is limited because increasing the number of levels increases the computational effort to compute the NURBS basis which is required in the evaluation of the contact forces, see Fig. 5.23 in Sect. 5.1.5 and Fig. 5.26 in Sect. 5.2.1. The maximum number of inserted nodes per layer is limited to avoid inserting all knots in one layer. Inserting all knots in one layer gives the same effect as the global refinement in Fig. 2.8. Many elements and control points would be created.

But not any number of inserted knots can be divided over a fixed number of levels. An example is given in Fig. 6.5, which attempts to insert three and two knots over three levels. As a side note, the first level represents the coarse model and the deeper levels the refinement. On the left in Fig. 6.5, three knots are inserted, which is equivalent to four inserted elements. Four is not a prime, it can be divided by two. As a result, two elements are inserted in the second level and the other two elements are added in the third level. Therefore, one knot is inserted in each level. However, it is not possible to distribute two knots over three levels, as shown on the right in Fig. 6.5. Inserting two knots is equivalent to three inserted elements. The number three is a prime number and a distribution to multiple levels is not possible.

In order to comply with the maximum number of levels $n_{\text{lvl}}^{\text{max}} = 5$ and the maximum number of inserted knots $n_{\text{knots}}^{\text{max}} = 10$ per level, the number of inserted knots cannot be gradually increased as done in Fig. 6.4. An allowed number of

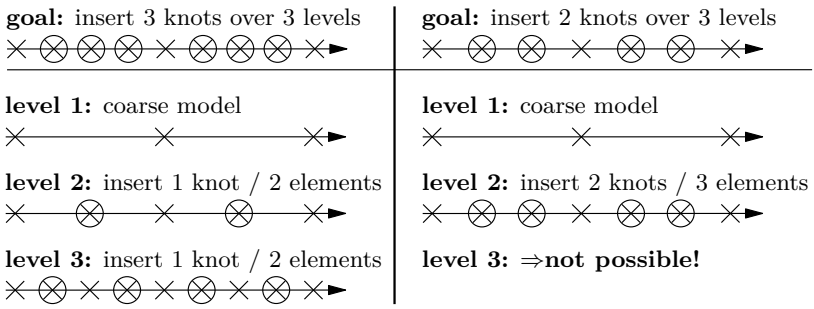


Figure 6.5: An example that not any number of inserted knots can be divided over a fixed number of levels during hierarchical refinement, see Fig. 2.8.

knots to approximate the contact interval is determined with Alg. 4. In line 2 of

Algorithm 4 Determine an allowed number of knots to map the contact interval.

Input: n_{knots}
Output: $n_{\text{knots}},\ \boldsymbol{n}_{\text{knots}}^{\text{lvl}}$

1: **while** true **do**
2: $n_{\text{knots}} \Leftarrow n_{\text{knots}} + 1$ ▷ new guess
3: $n^{\text{ele}} \Leftarrow n_{\text{knots}} + 1$ ▷ number of elements ⇐ number of knots
4: $\boldsymbol{n}_{\text{lvl}}^{\text{ele}} \Leftarrow \mathit{factor}(n^{\text{ele}})$ ▷ integer factorization (sorted ascending)
5: **while** $\mathit{length}(\boldsymbol{n}_{\text{lvl}}^{\text{ele}}) + 2 > n_{\text{lvl}}^{\max}$ **do** ▷ restrict number of levels
6: $\boldsymbol{n}_{\text{lvl}}^{\text{ele}} \Leftarrow [\boldsymbol{n}_{\text{lvl}}^{\text{ele}}(1) \cdot \boldsymbol{n}_{\text{lvl}}^{\text{ele}}(2),\ \boldsymbol{n}_{\text{lvl}}^{\text{ele}}(3 : \text{end})]$ ▷ merge two levels
7: **end while**
8: $\boldsymbol{n}_{\text{knots}}^{\text{lvl}} \Leftarrow \boldsymbol{n}_{\text{lvl}}^{\text{ele}} - \mathbf{1}$ ▷ number of knots ⇐ number of elements
9: **if** $\mathit{all}(\boldsymbol{n}_{\text{knots}}^{\text{lvl}} \leq n_{\text{knots}}^{\max})$ **then** ▷ restrict number of inserted knots per level
10: break ▷ new guess found
11: **end if**
12: **end while**

Alg. 4, the number of previously inserted knots is increased by one. The new guess is now validated. The number of knots is translated to the number of inserted elements in line 3. The integer factorization in line 4 splits up the elements over the levels. Line 5 and 6 restrict the number of levels. The addition by two in line 5 represents the coarse model in the first level and the final refinement with contact elements within the interval. The number of elements is converted back to the number of knots in line 8, and it is checked in line 9 whether the maximum number of inserted knots per level is respected. With this value n_{knots}, a tolerance can be used to check whether the interval can be mapped as in Fig. 6.4. If the search was successful, the refinement of the different levels is performed using integer factorization.

In the second step, the last hierarchy level is inserted. The user defines a number of elements $n_{\text{intvl}}^{\text{ele}}$ per coordinate direction that must be present in the interval. In this thesis, the parameter is found by experience as $n_{\text{intvl}}^{\text{ele}} = 16$. Since this number cannot be maintained for every configuration, it is possible to take this into account when determining the number of inserted knots n_{knots} in Alg. 4. If it is a 3D model where the interval is 2D, this procedure is performed for both local coordinate directions consecutively.

6.4 Transformation of Initial Conditions

Regarding the procedure of the adaptive simulation in Fig. 6.1, the elastic coordinates $\boldsymbol{q}_\mathrm{e}$ need to be transformed from the coarse to the fine model and vice versa. The concept is shown in Fig. 6.6. For the coarse model, the elastic coordinates $\boldsymbol{q}_\mathrm{e}^\mathrm{coarse}$ are normal modes of the modal truncation. The fine model is reduced with the Craig-Bampton method. Since a quasistatic contact model is used, only the low frequency modes $\boldsymbol{q}_\mathrm{e}^\mathrm{lf} = \boldsymbol{q}_\mathrm{e}^\mathrm{fine}$ are transformed. The solution of the last time step in the previous phase is required to compute the initial conditions of the next phase. The first half of this section describes the transformation from a coarse to a fine model, and the second half describes the transformation from a fine to a coarse model.

The transition from a non-impact phase to an impact phase requires a transformation from a coarse model to a fine model. Since the fine model is based on the coarse model, this fact can be exploited. The refinement is achieved by inserting knots. The control points $\boldsymbol{p}_{i,j,k}^\mathrm{fine}$ of the fine model are computed with the recursive knot insertion algorithm in [CottrellEtAl09]. The same algorithm can be used to transform the deformations $\boldsymbol{u}_\mathrm{e}$ of the control points. After the non-impact phase, the deformations $\boldsymbol{u}_\mathrm{e}^\mathrm{coarse}$ of the coarse model are computed with

$$\boldsymbol{u}_\mathrm{e}^\mathrm{coarse} = \boldsymbol{\Phi}^\mathrm{coarse} \boldsymbol{q}_\mathrm{e}^\mathrm{coarse}. \tag{6.4}$$

Afterwards, the recursive knot insertion algorithm [CottrellEtAl09] is applied to the control points $\boldsymbol{p}_{i,j,k}^\mathrm{coarse}$ and their deformations $\boldsymbol{u}_\mathrm{e}^\mathrm{coarse}$. This results in the refined control points $\boldsymbol{p}_{i,j,k}^\mathrm{fine}$ and deformations $\boldsymbol{u}_\mathrm{e}^\mathrm{fine}$. Finally, the elastic coordinates $\boldsymbol{q}_\mathrm{e}^\mathrm{fine}$ must be calculated. The connection in Eq. (6.4) also applies to the fine model. However, the global shape functions $\boldsymbol{\Phi}^\mathrm{fine}$ are a non-square matrix and therefore not invertible. Therefore, the elastic coordinates $\boldsymbol{q}_\mathrm{e}^\mathrm{fine}$ can be computed with

$$\boldsymbol{q}_\mathrm{e}^\mathrm{fine} = ((\boldsymbol{\Phi}^\mathrm{fine})^\intercal \boldsymbol{\Phi}^\mathrm{fine})^{-1} (\boldsymbol{\Phi}^\mathrm{fine})^\intercal \boldsymbol{u}_\mathrm{e}^\mathrm{fine} \tag{6.5}$$

by solving a linear least squares problem.

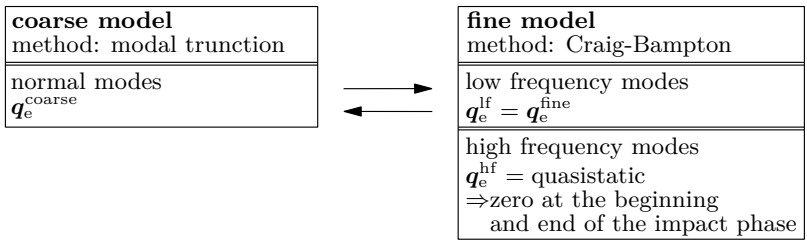


Figure 6.6: Transformation of elastic coordinates.

After an impact phase, a non-impact phase starts and the elastic coordinates need to be transformed from a fine to a coarse model. This transition is more challenging since the fine model has more degrees of freedom than the coarse model.

After the impact phase, the fine model deformations $\boldsymbol{u}_{\mathrm{e}}^{\mathrm{fine}}$ are computed with the global shape functions $\boldsymbol{\Phi}^{\mathrm{fine}}$ in the same way as in Eq. (6.4). This work proposes to transform the initial conditions by maintaining the elastic displacement of the two models whereby

$$\boldsymbol{d}^{\mathrm{coarse}}(\xi,\eta,\zeta) \approx \boldsymbol{d}^{\mathrm{fine}}(\xi,\eta,\zeta) \tag{6.6}$$

holds at an arbitrary point $\begin{bmatrix}\xi & \eta & \zeta\end{bmatrix}^{\mathsf{T}}$ in the parameter space. Remembering the computation of the elastic displacement in Eq. (2.21), it can be rewritten as

$$\underbrace{\begin{bmatrix} d_i^{\mathrm{x}}(\xi_i,\eta_i,\zeta_i) \\ d_i^{\mathrm{y}}(\xi_i,\eta_i,\zeta_i) \\ d_i^{\mathrm{z}}(\xi_i,\eta_i,\zeta_i) \end{bmatrix}}_{\boldsymbol{d}_i(\xi_i,\eta_i,\zeta_i)} = \underbrace{\begin{bmatrix} u^{\mathrm{x}}_{1,1,1} & u^{\mathrm{x}}_{1,1,2} & \cdots & u^{\mathrm{x}}_{\mathrm{p+1,q+1,r+1}} \\ u^{\mathrm{y}}_{1,1,1} & u^{\mathrm{y}}_{1,1,2} & \cdots & u^{\mathrm{y}}_{\mathrm{p+1,q+1,r+1}} \\ u^{\mathrm{z}}_{1,1,1} & u^{\mathrm{z}}_{1,1,2} & \cdots & u^{\mathrm{z}}_{\mathrm{p+1,q+1,r+1}} \end{bmatrix}}_{\boldsymbol{U}_{\mathrm{mat}}^{\mathrm{coarse}}} \underbrace{\begin{bmatrix} R^{\mathrm{p,q,r}}_{1,1,1}(\xi_i,\eta_i,\zeta_i) \\ R^{\mathrm{p,q,r}}_{1,1,2}(\xi_i,\eta_i,\zeta_i) \\ \vdots \\ R^{\mathrm{p,q,r}}_{\mathrm{p+1,q+1,r+1}}(\xi_i,\eta_i,\zeta_i) \end{bmatrix}}_{\boldsymbol{n}_i^{\mathrm{coarse}}(\xi_i,\eta_i,\zeta_i)}, \tag{6.7}$$

fine model — coarse model

where $\boldsymbol{d}_i \in \mathbb{R}^{3\times 1}$ are the fine model displacements at a point $\begin{bmatrix}\xi_i & \eta_i & \zeta_i\end{bmatrix}^{\mathsf{T}}$ in the parameter space, $\boldsymbol{U}_{\mathrm{mat}}^{\mathrm{coarse}} \in \mathbb{R}^{3\times n_{\mathrm{cp}}}$ are the deformations of the control points of an element in matrix notation, and $\boldsymbol{n}_i^{\mathrm{coarse}} \in \mathbb{R}^{n_{\mathrm{cp}}\times 1}$ are the corresponding NURBS basis of the coarse model. Equation (6.7) cannot be solved directly for the deformations $\boldsymbol{U}_{\mathrm{mat}}^{\mathrm{coarse}}$, since the NURBS basis $\boldsymbol{n}_i^{\mathrm{coarse}}$ is a vector and the system in Eq. (6.7) is underdetermined.

This work proposes the evaluation of Eq. (6.7) at Gauss points of an element of the coarse model. The coarse model is globally refined. Therefore, the number of control points $n_{\mathrm{cp}} = (p+1)(q+1)(r+1)$ is identical to the number of Gauss points in the element of a coarse model. As an example, Fig. 6.7 shows an element of a coarse model with Gauss points. By comparing the elastic displacement at all n_{cp} Gauss points at once, a square and invertible matrix $\boldsymbol{N}_{\mathrm{mat}}^{\mathrm{coarse}}$ of the NURBS basis can be assembled to

$$\boldsymbol{N}_{\mathrm{mat}}^{\mathrm{coarse}} = \begin{bmatrix} \boldsymbol{n}_1^{\mathrm{coarse}} & \boldsymbol{n}_2^{\mathrm{coarse}} & \cdots & \boldsymbol{n}_{\mathrm{n_{cp}}}^{\mathrm{coarse}} \end{bmatrix} \in \mathbb{R}^{n_{\mathrm{cp}}\times n_{\mathrm{cp}}}. \tag{6.8}$$

Thus, the elastic displacements $\boldsymbol{D}_{\mathrm{mat}}^{\mathrm{fine}}$ of the fine model at all Gauss points must be available in

$$\boldsymbol{D}_{\mathrm{mat}}^{\mathrm{fine}} = \begin{bmatrix} \boldsymbol{d}_1 & \boldsymbol{d}_2 & \cdots & \boldsymbol{d}_{\mathrm{n_{cp}}} \end{bmatrix} \in \mathbb{R}^{3\times n_{\mathrm{cp}}}. \tag{6.9}$$

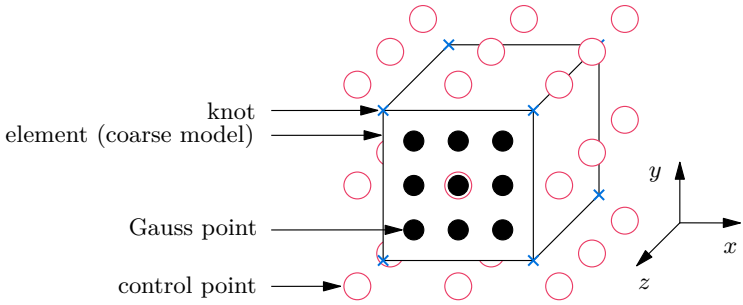


Figure 6.7: Example element of a coarse model of order $p = q = r = 2$.

Equation (6.7) is then extended to

$$\boldsymbol{D}_{\text{mat}}^{\text{fine}} = \boldsymbol{U}_{\text{mat}}^{\text{coarse}} \boldsymbol{N}_{\text{mat}}^{\text{coarse}} \tag{6.10}$$

and solved for the deformations of the coarse model with

$$\boldsymbol{U}_{\text{mat}}^{\text{coarse}} = \boldsymbol{D}_{\text{mat}}^{\text{fine}} (\boldsymbol{N}_{\text{mat}}^{\text{coarse}})^{-1}. \tag{6.11}$$

This approach results in redundant results because control points are part of multiple elements. Therefore, the results are averaged over all elements and converted back to the vector format $\boldsymbol{u}_{\text{e}}^{\text{coarse}}$ including the deformations of all control points in the coarse model. The elastic coordinates $\boldsymbol{q}_{\text{e}}^{\text{coarse}}$ are computed with the global shape functions $\boldsymbol{\Phi}^{\text{coarse}}$ and via linear least squares as in Eq. (6.5). The functionality of this procedure is checked in the application example in Sect. 7.1. There, it is shown that no energy is created or lost when the simulation phases change.

Adaptive Application Examples

In this chapter, two additional application examples are performed. In contrast to the testing examples in Chap. 5, the impacts are simulated adaptively according to the procedure presented in Chap. 6. This includes the detection of a contact, the finding of the contact location on the bodies, automatic refinement and model reduction of the contact model, determination of the penalty factor, and the transformation of the initial conditions between the coarse model used in non-impact phases and the fine model used in impact phases. For validating the adaptive procedure, especially the latter feature, monitoring the energy conservation of the wave propagation setup in Sect. 7.1 is suitable. Finally, the impact of flexible double pendulums in Sect. 7.2 is a suitable setup for the floating frame of reference formulation since large rigid body motions occur. Both adaptive application examples are first presented in [RückwaldEtAl24] and this chapter closely follows that publication.

7.1 Application Example I: Wave Propagation

The first application example, is 2D axisymmetric and shows significant elastodynamic effects. The setup was first presented in the previous work [RückwaldEtAl21d] for testing non-adaptive IGA contacts. Now, the developed adaptive procedure is applied to this setup. There are no large rigid body motions, which would make the use of the floating frame of reference formulation unnecessary. However, the goal of this setup is to validate the adaptive procedure by preserving the wave propagation and energy of the system. Validation is necessary since the transformation of the initial conditions described in Sect. 6.4 may be a possible source of error.

The axisymmetric setup visualized in Fig. 7.1 consists of two spheres and a long cylindrical rod. The spheres are made of steel and the rod is made of aluminum. As material parameters for steel, the Young's modulus is chosen as $E = 210\,\mathrm{GPa}$, the density as $\rho = 7850\,\mathrm{kg/m^3}$, and the Poisson's ratio as $\nu = 0.3$. The material

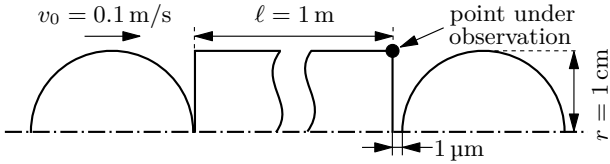


Figure 7.1: Wave propagation setup.

parameters for aluminum are given by $E = 72.8\,\mathrm{GPa}$, $\rho = 2789\,\mathrm{kg/m^3}$, and $\nu = 0.33$. The spheres and the rod have a radius of $r = 1\,\mathrm{cm}$, and the length of the rod is $\ell = 1\,\mathrm{m}$. The sphere on the left hand side in Fig. 7.1 starts with the initial velocity $v_0 = 0.1\,\mathrm{m/s}$. It impacts the resting rod, and a wave propagates to the right side of the rod. This wave induces a second impact with the resting sphere on the right hand side in Fig. 7.1. Between the rod and the right sphere, there is a small gap of length $1\,\mu\mathrm{m}$. As a result, the impacts on the left and the right side of the rod occur separately in time, and the simulation can be split into different phases. Thus, the initial conditions must be transformed between the impact and non-impact phase.

The wave speed in the rod can be computed, following e.g. [Graff91], with

$$c_{\mathrm{rod}} = \sqrt{E/\rho} = 5109\,\mathrm{m/s}. \tag{7.1}$$

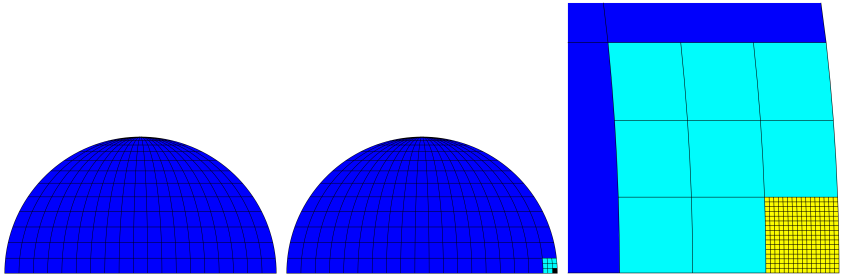
As suggested in [Seifried05] for a similar impact problem, the highest frequency of interest for this rod is $f_{\max} = 50\,\mathrm{kHz}$. The maximum step size $\Delta t^{\max}$ for a simulation of such wave propagation problem is subsequently chosen with

$$\Delta t^{\max} = \frac{1}{20 f^{\max}} = 1\,\mu\mathrm{s} \tag{7.2}$$

as described in [MoserEtAl99]. For wave propagation, the maximum length $\ell^{\max}$ of an element in the IGA rod model is then given by

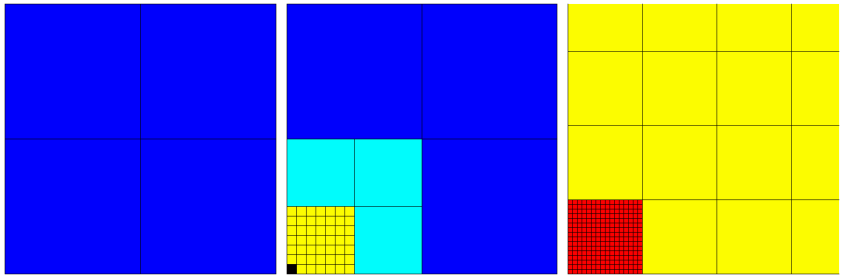
$$\ell^{\max} = \frac{c_{\mathrm{rod}}}{20 f^{\max}} = 5\,\mathrm{mm}. \tag{7.3}$$

Considering the requirement in Eq. (7.3), the coarse model of the rod and the two spheres are generated. To represent the maximum frequency $f^{\max} = 50\,\mathrm{kHz}$, the rod is modally truncated using $n_{\mathrm{q}} = 20$ modes. The two spheres are reduced using $n_{\mathrm{q}} = 10$ modes. The setup is then simulated for $0.8\,\mathrm{ms}$, which is approximately the time $4/c_{\mathrm{rod}} = 0.78\,\mathrm{ms}$. This is the time it takes for the wave to travel four times from one end to the other. The resulting model of the left sphere is displayed Fig. 7.2 and the left side of the rod is visualized in Fig. 7.3. The final adaptively refined models are used as a reference in the simulation of the quasistatic and the damped contact models without adaptivity. The adaptive procedure gives the penalty factors $c_{\mathrm{p}}^1 = 3 \cdot 10^{16}\,\mathrm{N/m}$ and $c_{\mathrm{p}}^2 = 1 \cdot 10^{17}\,\mathrm{N/m}$ for



a) Coarse model. **b)** Fine model. **c)** Close-up of fine model.

Figure 7.2: Adaptive model of the left sphere.



a) Coarse model. **b)** Fine model. **c)** Close-up of fine model.

Figure 7.3: Left side of the adaptive rod model.

the first and the second impact. These resulting penalty factors are used in the damped and the quasistatic model.

To monitor the wave propagation, the point at the right end of the rod, see Fig. 7.1, is observed. The velocity in the direction of the impact of the point is displayed in Fig. 7.4. A very good agreement of all simulations is observed. The first impact starts at the time t_0. In the beginning, both the rod and the observed point are at rest. The first impact induces a wave, which reaches the second contact area and therefore the monitored point at the time t_1. As the wave has reached the end of the rod, the wave is reflected and moves back to the left side of the rod. There it is reflected again and the wave travels to the monitored point again at t_3.

The contact forces of the two contacts are shown in Fig. 7.5. The impact and non-impact phase of the adaptive simulation can be distinguished by the gray and white areas. The first impact at the time t_0 on the left hand side of the rod and the second impact on the right hand side of the rod can be identified.

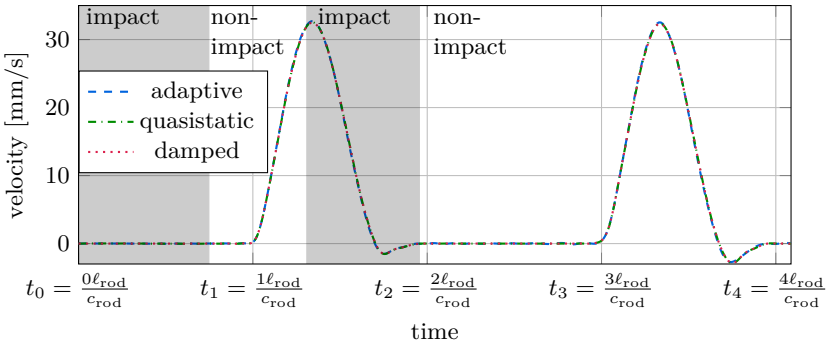


Figure 7.4: Velocity of the point under observation in the top right of the rod.

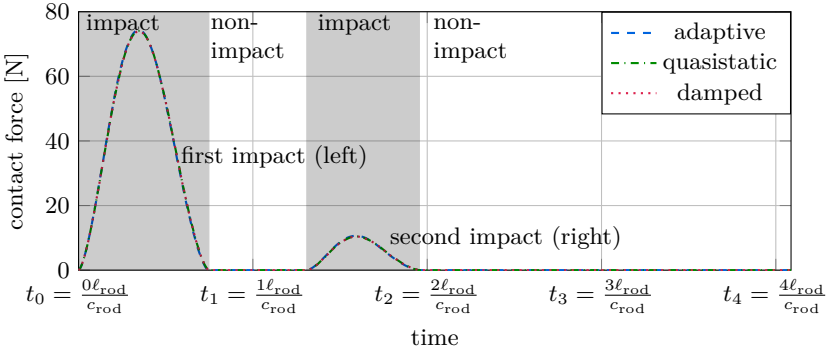


Figure 7.5: Contact force of the rod.

Considering the velocity in Fig. 7.4 and the contact force in Fig. 7.5, all three simulations show a good agreement. The results of the wave propagation show, that the transformation of the initial conditions is functioning.

To validate the conservation of energy, the energy of the system is displayed over time in Fig. 7.6. Since the impacts are elastic, the energy should stay constant. The initial energy E of the system is given by

$$E = \frac{1}{2} m_{\text{sphere}} v_0^2, \tag{7.4}$$

where m_{sphere} is the mass of the sphere and v_0 its initial velocity. It can be seen that the energy change remains well below one percent. All three simulations use the penalty method, which allows nonphysical penetration of the bodies. A minor increase of the energy is visible in the energy of the adaptive and quasistatic model. This increase is not visible in the damped model, as the damping of the

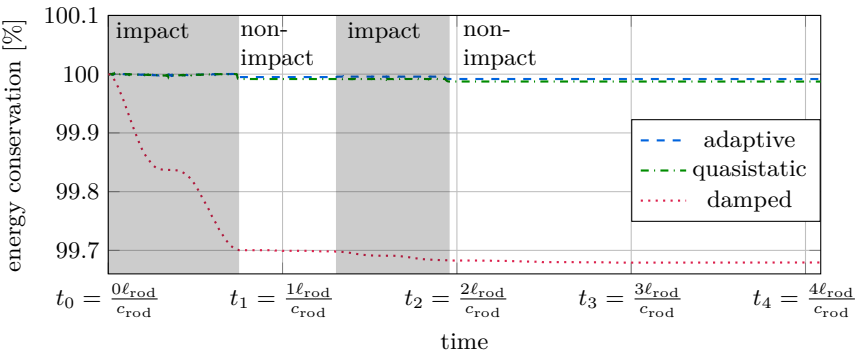


Figure 7.6: Energy of the system.

high-frequency modes predominates here. The energy of the adaptive model changes abruptly at two time points during the first impact. The reasons for the jumps is the adaptive decrease of the penalty factor. When the penalty factor is decreased, the nonphysical penetration increases. However, the overall course of the energy of the adaptive model is similar to the quasistatic and the damped model. It shows that the transformation of the elastic coordinates in the adaptive simulation does not effect the energy conservation of the system. In the quasistatic model, the penalty factor is predetermined and does not have to be reduced during the simulation. Therefore, the energy of the quasistatic model remains almost constant.

Considering the computation times, the adaptive model takes 4.4 min, the quasistatic model 2.7 min, and the damped model takes 14 min. The quasistatic model is the fastest, especially when compared to the damped model. The computational efficient quasistatic contact model outperforms the modally damped model. The adaptive model takes a bit longer than the quasistatic model without adaptivity since the contact model and the penalty factor are determined automatically. However, setting up the system and parameters manually is far more time consuming.

Finally, the utilization of the contact interval can be analyzed. In the simulation of the quasistatic and the damped model, the utilization of the left and right side of the rod vary. The utilization of the first contact on the left hand side of the rod is $\mathcal{U}_{\text{intvl}} = 10/16 = 62.5\,\%$ and the utilization of the second contact on the right side is only $\mathcal{U}_{\text{intvl}} = 6/16 = 37.5\,\%$. This is because the adaptive contact model of the first contact is used for both sides of the rod. Due to the nature of the IGA and its parameter space, the contact resolution at both ends of the rod cannot be different in the quasistatic model and the damped model. As shown in Fig. 7.5, the contact force and thus the actual contact width of the second impact

is significantly smaller than in the first impact. Therefore, the utilization of the contact interval of the second impact is smaller or even too small. This issue is not present in the adaptive model. Both impacts are simulated separately and can therefore use different contact models. The utilization of both impacts is close to the target utilization $\mathcal{U}_{\text{intvl}}^{\text{target}} = 70\,\%$. The adaptive procedure automatically selects a suitable contact interval width, which is smaller for the second impact.

7.2 Application Example II: Impact of Two Flexible Double Pendulums

This application example demonstrates the adaptive procedure in a flexible multibody system. The setup of two 3D flexible double pendulums is visualized in Fig. 7.7. The lower part of a pendulum is represented by a flexible sphere, and the connection between the suspension and the sphere is modeled by a circular flexible rod. The bodies are connected by a rotational joint allowing rigid body motion in the x-z-plane. This can be efficiently handled by the floating frame of reference formulation. This setup is already presented in the previous work [RückwaldEtAl22], where hierarchical refinement is investigated and the bodies are refined manually. Since the double pendulum behaves nonlinear, the locations of the impacts vary. Therefore, only the first impact is simulated

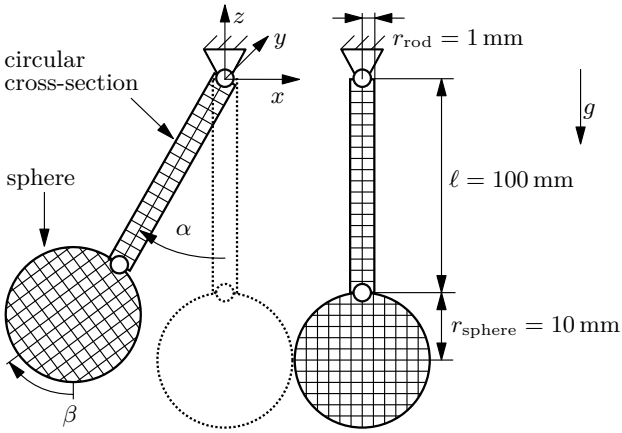


Figure 7.7: Setup of two 3D flexible double pendulums.

in [RückwaldEtAl22]. Now, the goal is to validate the adaptive procedure and simulate the system for a longer time period with multiple impacts.

The rods and the spheres are made of steel. The rods are modeled by globally refined IGA models, which are modally truncated to $n_q = 20$ elastic coordinates. The pendulum on the left-hand side in Fig. 7.7 is initially deflected by $\alpha_0 = 20°$ and $\beta_0 = 21.14°$. The two angles are chosen such that at the time of impact, it is $\alpha = \beta = 0°$. Both double pendulums have no initial velocity. Since gravity is under consideration with the gravitational constant $g = 9.81\,\mathrm{m/s^2}$, the initial values of the elastic coordinates of the four flexible bodies need to be determined. A straightforward approach is to solve the equations of motion for the elastic coordinates $\mathbf{q}_e$ so that the accelerations $\ddot{q}_e$ vanish. The time is set to $t = 0\,\mathrm{s}$ and the angles $\alpha = \alpha_0$ and $\beta = \beta_0$ are kept constant.

In this application example, five different models listed in Tab. 7.1 are compared:
1.) Hertz: In this approach, the rods are modeled by flexible IGA bodies, the spheres by rigid bodies, and the contact forces are computed according to the analytic solution by Hertz [Hertz82, Johnson04].

2.) adaptive: The adaptive contact simulation presented in Chap. 6 is performed. It includes the quasistatic contact model. This model is called "adaptive model" for simplicity.

3.) quasistatic: Only the quasistatic contact model without adaptivity is used in this simulation. For better comparability, the resulting refined sphere model from the adaptive simulation is used here. This model is called "quasistatic model" for simplicity.

4.) damped: This simulation uses damped contact models without adaptivity and without quasistatic contact models. The high frequency modes resulting

Table 7.1: Models compared in the fourth application example.

name	adaptivity	contact model	notes
Hertz	no	Hertz (see Sect. 3.1)	reference
adaptive	yes	quasistatic	test adaptive procedure (see Chap. 6)
quasistatic	no	quasistatic	uses resulting adaptive model compare computational performance
damped	no	modal damping	uses resulting adaptive model compare computational performance
coarse contact	no	quasistatic	uses the initial coarse model demonstrates the need for refinement

from the Craig-Bampton method are critically damped using modal damping. For better comparability, the resulting refined sphere model from the adaptive simulation is used here. This model is called "damped model" for simplicity.

5.) coarse contact: The coarse sphere model is used with the quasistatic contact model. This simulation shows the effect of no adaptivity, no refined contact model, and no manual adjustment of the penalty factor. This model is called "coarse contact model" for simplicity.

The simulations utilize the contact detection including Eq. (6.1) to switch integrator settings. The quasistatic model without adaptivity and the damped model can also be found in [RückwaldEtAl23]. Since the manual impact simulations with the quasistatic contact model and the damped contact model can only handle one impact, the system is only simulated for 170 ms. The other simulations are simulated for 1 s.

As mentioned before, the penalty factor in the quasistatic contact model cannot be arbitrarily increased beyond its converging value. Therefore, a penalty factor analysis is performed with the damped model in Fig. 7.8. Only minor differences are visible between the various penalty factors. In Fig. 7.9, the computation time of the impact is compared with the maximum occurring contact force. Therefore, the analysis of the penalty factor results in the converging value $c_\mathrm{p} = 10^{18}\,\mathrm{N/m}$ of the penalty factor with reasonable computation time and accuracy.

The simulation of the adaptive model also results in the penalty factor $c_\mathrm{p} =$

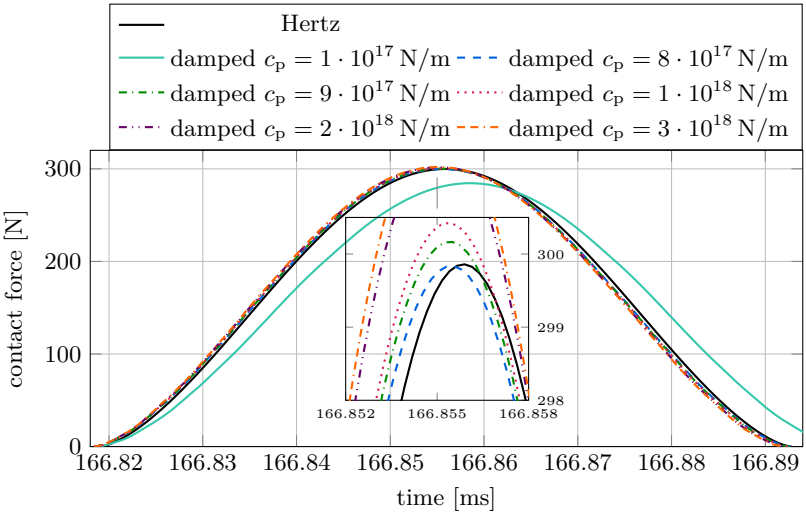


Figure 7.8: Convergence of the penalty factor for the first impact.

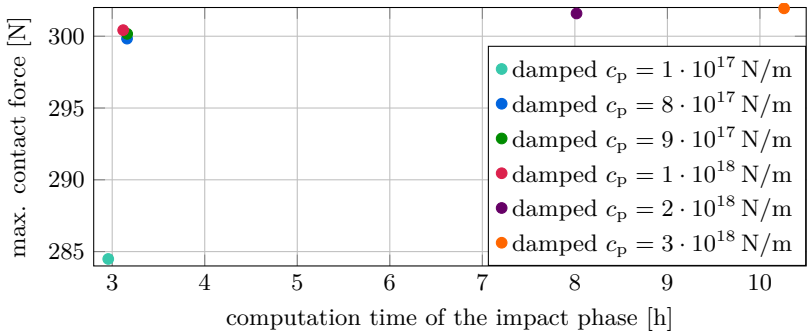


Figure 7.9: Convergence of the penalty factor for the first impact.

10^{18} N/m. The Hertz solution, the adaptive, the quasistatic, and the damped model are all compared in Fig. 7.10. Additionally, the simulation of the coarse contact model with the penalty factors $c_p = 10^{15}$ N/m and $c_p = 10^{18}$ N/m is visualized . It can be directly seen that the coarse contact model cannot represent the impact exactly. The remaining adaptive, quasistatic, and damped contact model are simulated with the penalty factor $c_p = 10^{18}$ N/m. No significant differences are visible between them and the Hertz solution. It shows, that the adaptive simulation works and, that the dynamics of the high frequency modes do not significantly affect the accuracy of the results.

Now, the adaptive simulation with three impacts is monitored in Fig. 7.11. The

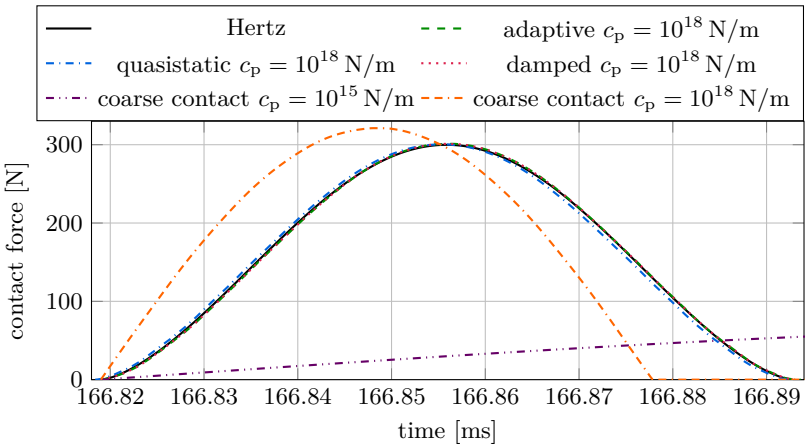


Figure 7.10: Close-up view of the first impact.

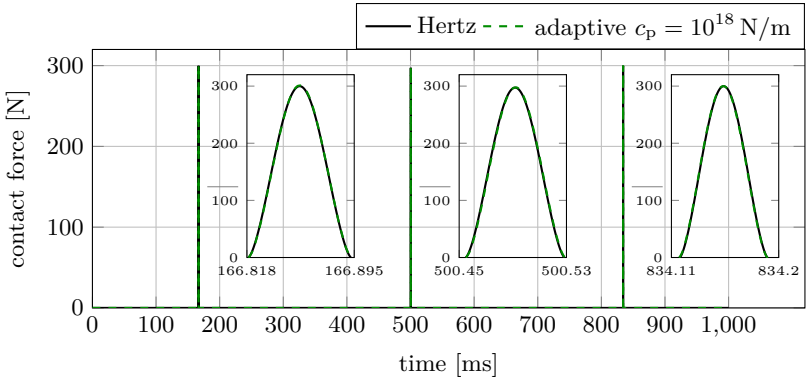


Figure 7.11: Contact force of three impacts.

three impacts nearly occur at the same time as in the Hertz reference simulation. Very small time shifts are noticeable in the second and the third impact. The Hertz solution and the adaptive model drift apart over time because the adaptive model uses an IGA model and the Hertz simulation uses rigid bodies as spheres and the Hertz contact law.

In the following, the complete computation including processing is considered. If the setup is simulated with one impact for 170 ms, the adaptive model takes 3.0 h, the quasistatic model 2.0 h, the damped model 8.2 h, and the coarse contact model 6.1 min. Although the penalty factor and the contact model are predetermined in the simulation of the damped model, this simulation takes the most time. The reason for this is the numerical stiffness and thus the computationally expensive dynamics of the high frequency modes. This can be observed especially in the second non-impact phase. The frequency band of the numerically stiff system is excited by the impact and small time steps are required.

The simulation of the quasistatic model is the fastest because, unlike in the adaptive simulation, the contact model and the penalty factor are already predetermined and taken from the adaptive simulation. The simulation of the adaptive model is more time consuming than the quasistatic without adaptivity. However, the setup of the quasistatic and the damped model is manual and would thus take even more time. The coarse contact model is the fastest because it is not refined. However, Fig. 7.10 shows that this model cannot represent the impact correctly in any way.

The detailed computation times of each part of the simulations are listed in Tab. 7.2. The listed computation times of the phases only include the time of the actual simulation without processing. The listed complete computation times include also the processing. There is a small difference in the first non-impact

phase. This phase is computed faster with the adaptive and coarse contact model than with the quasistatic model. The reason for this is that the quasistatic contact model is already refined and the contact detection with Eq. (6.1) takes longer. The second non-impact phase of the damped model takes very long. The impact induced the whole frequency band and small time steps are required. A simulation until the beginning of the second impact would take too much computation time.

It is also worth noting the attempts of the adaptive procedure. In the first three tries, the algorithm only adjusts the IGA model. The initial contact interval is too large and needs to be decreased. In the fourth try, the model is acceptable and the penalty factor is increased. The first six tries of the first impact are computed relatively fast, because the impact is not fully simulated and the impact phase terminates early. According to Fig. 6.1, the impact phase terminates early if an edge element is in contact or on request of the quasistatic algorithm. As described in Sect. 6.1 and Fig. 6.2, initially only $n_{\mathrm{con}} = 30$ contact time steps are simulated to increase the penalty factor as quickly as possible. The seventh try starts with the penalty factor $c_{\mathrm{p}} = 1 \cdot 10^{18}\,\mathrm{N/m}$ but an error occurs in the quasistatic contact model, see Alg. 1. The algorithm still converges after a reset, but the factor must not be increased any further. Therefore, the impact is fully simulated with the penalty factor $c_{\mathrm{p}} = 1 \cdot 10^{18}\,\mathrm{N/m}$. It can also be seen that only the first adaptive impact is time consuming. The following impacts benefit from the previous findings on the penalty factor and the contact interval width. Therefore, the complete computation time of three impacts for 1 s takes 8.8 h.

As can be seen in Tab. 7.2, the first six tries are required to determine values for the interval width and the penalty factor, which are close to their converging value. Providing improved initial values for these parameters would reduce the number of tries in the beginning. However, also in finite element analysis, finding appropriate contact parameters requires a trial-and-error approach. Additionally, the first tries are computed relatively fast, see Tab. 7.2, and the required pre-processing of the models only takes 1 min to 2 min for each model. As initial values, the optimal penalty factor for static simulations, see [Nour-OmidWriggers87]. Alternatively, a neural network might be used as a predictor in the future. Using the adaptive procedure in this work would help to automatically generate training data for the neural network. As in [RückwaldEtAl21a], neural networks are suitable for system identification. However, this is not in the scope of this work.

Table 7.2: Computation times of the second adaptive application example.

	adaptive		quasistatic	damped	coarse contact
			$c_\mathrm{p} = 1 \cdot 10^{18}$ N/m		
simulation	computation time	c_p [N/m]	computation time	computation time	computation time
non-impact 1 - try 1	18 s		55 s	29 min	16 s
impact 1 - try 1	1.4 min	$1 \cdot 10^{15}$	111 min	188 min	2 min
impact 1 - try 2	1.1 min	$1 \cdot 10^{15}$			
impact 1 - try 3	2.3 min	$1 \cdot 10^{15}$			
impact 1 - try 4	3.3 min	$1 \cdot 10^{16}$			
impact 1 - try 5	3.4 min	$1 \cdot 10^{17}$			
impact 1 - try 6	5.4 min	$1 \cdot 10^{17}$			
impact 1 - try 7	130 min	$1 \cdot 10^{18}$			
non-impact 2 - try 1	0.1 min		0.3 min	187 min	0.8 min
complete computation time	3.0 h		2.0 h	8.2 h	6.1 min
non-impact 2 - try 1	7.1 min				
impact 2 - try 1	160 min	$1 \cdot 10^{18}$			
non-impact 3 - try 1	6.7 min				
impact 3 - try 1	172 min	$1 \cdot 10^{18}$			
non-impact 4 - try 1	4.2 min				
complete computation time	8.8 h				

Conclusion

Overall, this thesis shows that it is feasible to obtain global shape functions for flexible multibody systems from IGA models. A model reduction with the Craig-Bampton method can be performed, and the global shape functions obtained from the IGA can be smoothly included in the floating frame of reference formulation. The toolbox RIGA developed in this thesis can be integrated into the existing workflow consisting of the toolboxes RED and DynManto with only minor adjustments. Subsequently, impact simulations can be performed using the penalty formulation. The results agree with the analytical solution of Hertz and with the results obtained by nonlinear FEM of the software Ansys.

The two testing examples in Chap. 5 serve as a validation and as a benchmark to gain insight into the behavior of the IGA in impact simulations in combination with the floating frame of reference formulation. Four insights were gained in the study of the two examples:

First, the use of collocation points instead of Gauss points increases computational performance without sacrificing accuracy. The same observation is made in [Matzen15]. This is because using Gauss points results in an overdetermined contact formulation. When using collocation points, the number of contact evaluation points is identical to the number of control points on the exterior surface.

Second, order elevation is a method to increase the accuracy of the model. Here, the order of the B-splines is increased without changing the geometry. In most cases, it is sufficient to increase the order only slightly. If desired, the accuracy of the same model can be increased. However, computation time increase, especially for 3D setups.

Third, a quasistatic contact model increases the computational performance compared to a modally damped model. This observation made with isoparametric models in [Tschigg20] is also made in the IGA. Although a quasistatic contact equation has to be solved, the system is less numerically stiff. This is not only noticeable in pure impact simulations such as in Chap. 5. But also when simulating multiple impacts, as in the simulation of two pendulums, see Sect. 7.2. If modal damping is used, the first impact induces the whole frequency band requiring small time step sizes for a subsequent ”free flying” phase without

impacts. This disadvantage disappears when using a quasistatic contact model instead.

Fourth, hierarchical refinement can be incorporated into the tool chain, including model reduction with the Craig-Bampton method and impact simulation with the floating frame of reference formulation. In contrast to other applications of hierarchical refinement found in the literature [SchillingerEtAl12, D'AngellaEtAl18, NoëlEtAl20, ZimmermannSauer17], it cannot be generally stated that the computational performance is increased compared to global refinement. This is the case, when hierarchically refined models are reduced and included in the floating frame of reference formulation. Hierarchical refinement reduces the number of control points and degrees of freedom. However, in this work, these degrees of freedom do not correspond to the degrees of freedom of the equations of motion to be solved. The models are reduced and therefore the number of interface control points in the contact area is relevant. For 2D models, the number of interface control points is identical for hierarchically and globally refined models. Therefore, the computational performance of hierarchical refinement is equal to or less than that of global refinement. However, hierarchically refined 3D models require fewer interface control points. In the transition area between the lowest and second-lowest level, the number of interface control points is lower than in global refinement. This area corresponds to the outer contact area where little or no contact forces occur. This favors the use of hierarchical refinement in impact simulations with the floating frame of reference formulation.

In the previous work [Tschigg20], the focus is on being more computationally efficient than a FEM software, such as Abaqus, while still achieving accurate results. In [Tschigg20], the quasistatic contact model is applied to isoparametric models. The same model is then simulated in Abaqus. This similarity simplifies the comparison of the two approaches. The main objective of this work is an adaptive procedure to automatically simulate impacts in flexible multibody systems. For this purpose, the IGA is proposed. Additionally, the IGA is compared with isoparametric models simulated as RISO models and simulated in Ansys. Both IGA and isoparametric models produce accurate results. In 2D, the RISO model is faster than the RIGA model due to linear shape functions of isoparametric elements. Comparing the IGA with isoparametric models in general shows that the IGA is slightly more accurate. This can be observed, for example, in the energy of the system and the mass of the bodies. Isoparametric models are discretized models of the original geometry. The true mass of the body cannot be approximated exactly, but only with a large number of elements. With the IGA, a small number of elements is sufficient to map the mass exactly. This is because the IGA can represent the geometry exactly. The accuracy and the computational performance strongly depend on the parameters of the simulation, e.g. order elevation, refinement, and time step size. Therefore, the comparison between the IGA and isoparametric elements is ambiguous.

Regarding the main objective of this work, it can be concluded that the presented adaptive procedure can be smoothly included in the floating frame of reference formulation to allow automatic impact treatment. As mentioned before, the quasistatic contact model and hierarchical refinement increase the computational performance. This is especially the case in phases where large rigid body motions occur, e.g., after an impact that excites the whole frequency band. However, the use of the two techniques also facilitate the implementation of the adaptive procedure. By using the quasistatic contact model and neglecting the high frequency modes, the number of elastic coordinates is identical for the fine and the coarse model since only the low frequency modes are included in the dynamics. Hierarchical refinement allows an easy and structured approach to automatically refine the models. The presented adaptive procedure in this work generates hierarchically refined contact models where the location and resolution of the impact are adaptively determined. Additionally, the procedure allows the adaptive determination of the penalty factor. For an efficient computation of impact and non-impact phases, a method to detect impacts is presented. To switch between impact and non-impact phases, an approach is presented in this thesis to transform the elastic coordinates back and forth between the coarse and the fine model.

The adaptive application examples show that the procedure works and that energy conservation is mainly given. The setup of two flexible double pendulums is suitable because the contact location varies. The simulation with a pre-refined model would require a generous refined contact area. However, a large number of interface control points would be required which increases the computation time. Therefore, the proposed procedure is an efficient approach to automatically handle impacts.

The research on IGA in flexible multibody system can be continued. In this work, two bodies are involved in a point contact. In future works, self and line contacts can be added. A challenging setup may be the self contact of a spring, see Fig. 8.1. The spring can be incorporated in a flexible multibody system, e.g. a car or train suspension. In this setup, automatic mesh refinement may be required.

The representation of an arbitrary CAD geometry as a solid IGA model remains a challenging task [YuEtAl22]. One possible solution may be meshing the geometry with isoparametric elements and replacing the contact area with IGA elements. This is part of active research, see [VilledaEtAl23]. However, it remains a challenging task due to the intersection of isoparametric and isogeometric elements.

Besides contact simulations, the IGA and hierarchical refinement are applied to adaptive level set topology optimization, see [NoëlEtAl20]. In topology optimization, poorly filled elements can occur and cause numerical issues. To better resolve these, the domain can be adaptively refined using hierarchical refinement. The

hierarchical refinement and the transformation of elastic coordinates are already implemented in RIGA so that the development of RIGA can be continued.

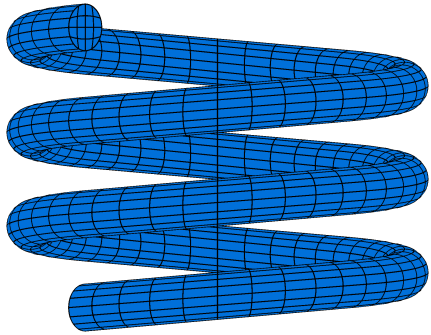


Figure 8.1: A spring as an example of a self contact.

Bibliography

[Ansys19] Ansys: *Documentation for Ansys, Release 19.5*. Canonsburg, 2019.

[Bathe14] Bathe, K.J.: *Finite Element Procedures*. Watertown, Mass.: Klaus-Jürgen Bathe, 2014.

[Boor72] de Boor, C.: On calculating with b-splines. *Journal of Approximation Theory*, Vol. 6, No. 1, pp. 50–62, 1972. DOI:10.1016/0021-9045(72)90080-9.

[Botella02] Botella, O.: On a collocation b-spline method for the solution of the navier–stokes equations. *Computers & Fluids*, Vol. 31, No. 4–7, pp. 397–420, 2002. DOI:10.1016/s0045-7930(01)00058-5.

[Broyden65] Broyden, C.G.: A class of methods for solving nonlinear simultaneous equations. *Mathematics of Computation*, Vol. 19, No. 92, pp. 577–593, 1965. DOI:10.1090/s0025-5718-1965-0198670-6.

[CottrellEtAl09] Cottrell, J.A.; Hughes, T.J.R.; Bazilevs, Y.: *Isogeometric Analysis*. Singapore: John Wiley & Sons, Ltd, 2009. DOI:10.1002/9780470749081.

[Cox71] Cox, M.: *The Numerical Evaluation of B-splines*. DNAC. National Physical Laboratory, Division of Numerical Analysis and Computing, 1971.

[CraigBampton68] Craig, R.R.; Bampton, M.C.C.: Coupling of substructures for dynamic analyses. *AIAA Journal*, Vol. 6, No. 7, pp. 1313–1319, 1968.

[D'AngellaEtAl18] D'Angella, D.; Kollmannsberger, S.; Rank, E.; Reali, A.: Multi-level bézier extraction for hierarchical local refinement of isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, Vol. 328, pp. 147–174, 2018. DOI:10.1016/j.cma.2017.08.017.

[Demko85] Demko, S.: On the existence of interpolating projections onto spline spaces. *Journal of Approximation Theory*, Vol. 43, No. 2, pp. 151–156, 1985. DOI:10.1016/0021-9045(85)90123-6.

[Eberhard00] Eberhard, P.: *Kontaktuntersuchungen durch hybride Mehrkörpersystem-Finite-Elemente-Simulationen*. Berichte aus dem Maschinenbau. Shaker, 2000.

[Farin01] Farin, G.: *Curves and Surfaces for CAGD*. Morgan Kaufmann, 2001.

[FehrEberhard10] Fehr, J.; Eberhard, P.: Simulation process of flexible multibody systems with non-modal model order reduction techniques. *Multibody System Dynamics*, Vol. 25, No. 3, pp. 313–334, 2010. DOI:10.1007/s11044-010-9238-3.

[GaoEtAl19] Gao, W.; Wang, J.; Yin, S.; Feng, Y.: A coupled 3d isogeometric and discrete element approach for modeling interactions between structures and granular matters. *Computer Methods in Applied Mechanics and Engineering*, Vol. 354, pp. 441–463, 2019. DOI:10.1016/j.cma.2019.05.043.

[GarauVázquez18] Garau, E.M.; Vázquez, R.: Algorithms for the implementation of adaptive isogeometric methods using hierarchical b-splines. *Applied Numerical Mathematics*, Vol. 123, pp. 58–87, 2018. DOI:10.1016/j.apnum.2017.08.006.

[GolubLoan96] Golub, G.H.; Loan, C.F.V.: *Matrix Computations*. London: Johns Hopkins University Press, 3rd. Edn., 1996.

[GonthierEtAl04] Gonthier, Y.; McPhee, J.; Lange, C.; Piedbœuf, J.C.: A regularized contact model with asymmetric damping and dwell-time dependent friction. *Multibody System Dynamics*, Vol. 11, No. 3, pp. 209–233, 2004. DOI:10.1023/b:mubo.0000029392.21648.bc.

[Graff91] Graff, K.F.: *Wave Motion in Elastic Solids*. Dover, New York: DOVER PUBN INC, 1991.

[HeldEtAl20] Held, A.; Moghadasi, A.; Seifried, R.: DynManto: A matlab toolbox for the simulation and analysis of multibody systems. In Volume 2: 16th International Conference on Multibody Systems, Nonlinear Dynamics, and Control (MSNDC), American Society of Mechanical Engineers, 2020.

[Hertz82] Hertz, H.: Über die Berührung fester elastischer Körper (in German). *Journal für die reine und angewandte Mathematik*, Vol. 92, pp. 156–171, 1882. DOI:10.1515/crll.1882.92.156.

[Johnson04] Johnson, K.L.: *Contact Mechanics*. Cambridge: Cambridge University Press, 2004. DOI:10.1017/CBO9781139171731.

[Koppens89] Koppens, W.W.: *The dynamics of systems of deformable bodies*. Ph.D. thesis, Technische Universiteit Eindhoven, 1989. DOI:10.6100/IR297020.

[LankaraniNikravesh90] Lankarani, H.M.; Nikravesh, P.E.: A contact force model with hysteresis damping for impact analysis of multibody sys-

tems. *Journal of Mechanical Design*, Vol. 112, No. 3, pp. 369–376, 1990. DOI:10.1115/1.2912617.

[LeePark02] Lee, B.G.; Park, Y.: Degree elevation of nurbs curves by weighted blossom. *Korean Journal of Computational & Applied Mathematics*, Vol. 9, No. 1, pp. 151–165, 2002.

[Leidinger22] Leidinger, L.: *Explicit Isogeometric B-Rep Analysis for Nonlinear Dynamic Crash Simulations*. Ph.D. thesis, Technical University of Munich, 2022.

[Mathworks23] Mathworks: *Matlab, Product Help, Matlab Release 2023b*. Mathworks Inc., 2023.

[MathWorks24] MathWorks: *MinGW-w64 C/C++/Fortran Compiler* `https://de.mathworks.com/matlabcentral/fileexchange/52848-matlab-support-for-mingw-w64-c-c-fortran-compiler`, accessed 26th March 2024.

[Matzen15] Matzen, M.E.: *Isogeometrische Modellierung und Diskretisierung von Kontaktproblemen (in German)*. Ph.D. thesis, University of Stuttgart, Institute of Structural Mechanics, 2015.

[MatzenBischoff16] Matzen, M.; Bischoff, M.: A weighted point-based formulation for isogeometric contact. *Computer Methods in Applied Mechanics and Engineering*, Vol. 308, pp. 73–95, 2016. DOI:10.1016/J.CMA.2016.04.010.

[Moghadasi19] Moghadasi, A.: *Contributions to topology optimization in flexible multibody dynamics*. Ph.D. thesis, Hamburg University of Technology, 2019. DOI:10.15480/882.1976.

[Morlock23] Morlock, M.: *Control strategies for flexible link parallel robots*. Ph.D. thesis, Hamburg University of Technology, 2023. DOI:10.15480/882.6203.

[MoserEtAl99] Moser, F.; Jacobs, L.J.; Qu, J.: Modeling elastic wave propagation in waveguides with the finite element method. *NDT & E International*, Vol. 32, No. 4, pp. 225–234, 1999.

[Nour-OmidWriggers87] Nour-Omid, B.; Wriggers, P.: A note on the optimum choice for penalty parameters. *Communications in Applied Numerical Methods*, Vol. 3, No. 6, pp. 581–585, 1987. DOI:10.1002/CNM.1630030620.

[NoëlEtAl20] Noël, L.; Schmidt, M.; Messe, C.; Evans, J.; Maute, K.: Adaptive level set topology optimization using hierarchical b-splines. *Structural and Multidisciplinary Optimization*, Vol. 62, No. 4, pp. 1669–1699, 2020. DOI:10.1007/S00158-020-02584-6.

[OishiYagawa20] Oishi, A.; Yagawa, G.: A surface-to-surface contact search method enhanced by deep learning. *Computational Mechanics*, Vol. 65, No. 4, pp. 1125–1147, 2020. DOI:10.1007/S00466-019-01811-2.

[PieglTiller97] Piegl, L.; Tiller, W.: *The NURBS Book*. Springer Berlin Heidelberg, 1997. DOI:10.1007/978-3-642-59223-2.

[Pintelon12] Pintelon, R.: *System identification : a frequency domain approach*. Hoboken, N.J: Wiley, 2nd ed. Edn., 2012. DOI:10.1002/9781118287422.

[RückwaldEtAl23] Rückwald, T.; Held, A.; Seifried, R.: A quasistatic contact model for impact analysis in flexible multibody systems based on iga. In 11th ECCOMAS Thematic Conference on Multibody Dynamics 2023, IDMEC, 2023.

[RückwaldEtAl21a] Rückwald, T.; Drücker, S.; Dücker, D.A.; Seifried, R.: Nonlinear system identification of a furuta pendulum using machine learning techniques. *91st Annual Meeting of the International Association of Applied Mathematics and Mechanics (GAMM 2021)*, 2021. DOI:10.15480/882.3903.

[RückwaldEtAl21b] Rückwald, T.; Held, A.; Seifried, R.: Flexible multibody impact simulations based on the isogeometric analysis approach. *Multibody System Dynamics*, Vol. 54, No. 1, pp. 75–95, 2021. DOI:10.1007/s11044-021-09804-x.

[RückwaldEtAl21c] Rückwald, T.; Held, A.; Seifried, R.: Flexible multibody impact simulations using hierarchically refined isogeometric models. In 10th ECCOMAS Thematic Conference on MULTIBODY DYNAMICS 2021, pp. 114–125, 2021.

[RückwaldEtAl21d] Rückwald, T.; Held, A.; Seifried, R.: Reduced isogeometric analysis models for impact simulations. In 17th International Conference on Multibody Systems, Nonlinear Dynamics, and Control, 2021. DOI:10.1115/DETC2021-67417.

[RückwaldEtAl22] Rückwald, T.; Held, A.; Seifried, R.: Hierarchical refinement in isogeometric analysis for flexible multibody impact simulations. *Multibody System Dynamics*, 2022. DOI:10.1007/s11044-022-09856-7.

[RückwaldEtAl24] Rückwald, T.; Held, A.; Seifried, R.: Adaptive impact analysis in flexible multibody systems based on hierarchically refined iga models. *Archive of Applied Mechanics*, 2024. DOI:10.1007/s00419-024-02604-7.

[Salane86] Salane, D.E.: Adaptive routines for forming jacobians numerically. *Sandia National Laboratories*, 1986.

[SchillingerEtAl12] Schillinger, D.; Dedè, L.; Scott, M.A.; Evans, J.A.; Borden, M.J.; Rank, E.; Hughes, T.J.: An isogeometric design-through-analysis methodology based on adaptive hierarchical refinement of NURBS, immersed boundary methods, and t-spline CAD surfaces. *Computer Methods in Applied Mechanics and Engineering*, Vol. 249-252, pp. 116–150, 2012. DOI:10.1016/j.cma.2012.03.017.

[Schmitt19] Schmitt, A.G.: *Real-time simulation of flexible multibody systems in vehicle dynamics*. Ph.D. thesis, Hamburg University of Technology, 2019. DOI:10.15480/882.2510.

[SchwertassekWallrapp14] Schwertassek, R.; Wallrapp, O.: *Dynamik flexibler Mehrkörpersysteme (in German)*. Wiesbaden: Vieweg & Teubner, 2014.

[Seifried05] Seifried, R.: *Numerische und experimentelle Stoßanalyse für Mehrkörpersysteme (in German)*. Aachen: Shaker, 2005.

[SeifriedEtAl03] Seifried, R.; Hu, B.; Eberhard, P.: Numerical and experimental investigation of radial impacts on a half-circular plate. *Multibody System Dynamics*, Vol. 9, No. 3, pp. 265–281, 2003.

[SeitzEtAl16] Seitz, A.; Farah, P.; Kremheller, J.; Wohlmuth, B.I.; Wall, W.A.; Popp, A.: Isogeometric dual mortar methods for computational contact mechanics. *Computer Methods in Applied Mechanics and Engineering*, Vol. 301, pp. 259–280, 2016. DOI:10.1016/J.CMA.2015.12.018.

[Shabana05] Shabana, A.: *Dynamics of multibody systems*. Cambridge New York: Cambridge University Press, 2005.

[SherifEtAl11] Sherif, K.; Witteveen, W.; Irschik, H.; Holl, H.; Mayrhofer, K.: *On the extension of global vibration modes with Ritz-vectors needed for local effects*. In Conference Proceedings of the Society for Experimental Mechanics Series, pp. 37–46. Springer New York, 2011. DOI:10.1007/978-1-4419-9305-2_4.

[SherifEtAl12] Sherif, K.; Witteveen, W.; Mayrhofer, K.: Quasi-static consideration of high-frequency modes for more efficient flexible multibody simulations. *Acta Mechanica*, Vol. 223, No. 6, pp. 1285–1305, 2012. DOI:10.1007/S00707-012-0624-1.

[SimoLaursen92] Simo, J.; Laursen, T.: An augmented lagrangian treatment of contact problems involving friction. *Computers & Structures*, Vol. 42, No. 1, pp. 97–116, 1992. DOI:10.1016/0045-7949(92)90540-G.

[TamarozziEtAl13] Tamarozzi, T.; Ziegler, P.; Eberhard, P.; Desmet, W.: Static modes switching in gear contact simulation. *Mechanism and Machine Theory*, Vol. 63, pp. 89–106, 2013.

[TemizerEtAl11] Temizer, İ.; Wriggers, P.; Hughes, T.: Contact treatment in isogeometric analysis with NURBS. *Computer Methods in Applied Mechanics and Engineering*, Vol. 200, No. 9-12, pp. 1100–1112, 2011.

[TemizerEtAl12] Temizer, İ.; Wriggers, P.; Hughes, T.: Three-dimensional mortar-based frictional contact treatment in isogeometric analysis with NURBS. *Computer Methods in Applied Mechanics and Engineering*, Vol. 209-212, pp. 115–128, 2012.

[TobiasEberhard11] Tobias, C.; Eberhard, P.: Stress recovery with krylov-subspaces in reduced elastic multibody systems. *Multibody System Dynamics*, Vol. 25, No. 4, pp. 377–393, 2011. DOI:10.1007/s11044-010-9239-2.

[Tschigg20] Tschigg, S.: *Effiziente Kontaktberechnung in Fexiblen Mehrkörpersystemen (in German)*. Ph.D. thesis, Hamburg University of Technology, 2020. DOI:10.15480/882.2709.

[TschiggSeifried18] Tschigg, S.; Seifried, R.: Efficient impact analysis using reduced flexible multibody systems and contact submodels. In 6th European Conference on Computational Mechanics: Solids, Structures and Coupled Problems, ECCM 2018 and 7th European Conference on Computational Fluid Dynamics, ECFD 2018, pp. 2711–2722, 2018.

[VilledaEtAl23] Villeda, E.G.L.; Steinbrecher, I.; Popp, A.: Towards and embedded mesh approach for isogeometric boundary layers in contact mechanics. In Book of abstracts of the ICCCM 2023 7th International Conference on Computational Contact Mechanics, pp. 137–138, 2023.

[VuongEtAl11] Vuong, A.V.; Giannelli, C.; Jüttler, B.; Simeon, B.: A hierarchical approach to adaptive local refinement in isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, Vol. 200, No. 49–52, pp. 3554–3567, 2011. DOI:10.1016/j.cma.2011.09.004.

[Wielenga84] Wielenga, T.J.: *Simplifications in the simulation of mechanisms containing flexible members*. Ph.D. thesis, The University of Michigan, 1984.

[Wriggers06] Wriggers, P.: *Computational Contact Mechanics*. Berlin: Springer-Verlag GmbH, 2006. DOI:10.1007/978-3-540-32609-0.

[YuEtAl22] Yu, Y.; Wei, X.; Li, A.; Liu, J.G.; He, J.; Zhang, Y.J.: *HexGen and Hex2Spline: Polycube-Based Hexahedral Mesh Generation and Spline Modeling for Isogeometric Analysis Applications in LS-DYNA*. In Springer INdAM Series, pp. 333–363. Springer International Publishing, 2022.

[ZimmermannSauer17] Zimmermann, C.; Sauer, R.A.: Adaptive local surface refinement based on LR NURBS and its application to contact. *Computational Mechanics*, Vol. 60, No. 6, pp. 1011–1031, 2017. DOI:10.1007/s00466-017-1455-7.