

TECHNISCHE UNIVERSITÄT KAISERSLAUTERN

**Isogeometric Shell Discretizations for
Flexible Multibody Dynamics**

Anmol Goyal

Vom Fachbereich Mathematik der Technischen Universität Kaiserslautern zur
Verleihung des akademischen Grades

Doktor der Naturwissenschaften
(Doctor rerum naturalium, Dr. rer. nat.)

genehmigte Dissertation.

Gutachter:

1. Prof. Dr. Bernd Simeon, Technische Universität Kaiserslautern, Kaiserslautern
2. Univ.-Prof. Dr. Bert Jüttler, Johannes Kepler Universität, Linz

Datum der Disputation: 24. April 2015

D386

To my family.

Acknowledgements

It gives me immense pleasure to express my deepest sense of gratitude to my mentor Prof. Dr. Bernd Simeon for providing me with an opportunity to work under his able supervision. I would always be indebted to him for his invaluable guidance and belief in my abilities throughout the duration of my research.

I sincerely thank all my dear colleagues Julia Vuong, Anh-Vu Vuong, Daniela Fußeder, Mané Harutyunyan, Dennis Merkert for providing me with a very relaxed and conducive work environment. I also take this opportunity to thank some of them for their exceptional support - Julia and Dennis, for being extremely helpful and most patient office mates; Daniela, for meticulously proof-reading this work and giving me helpful suggestions to make it better. Not to forget our occasional breaks along with Christian Stinner, Alexander Hunt and Stefanie Sonner, which comprised not only of delicious cakes but also some very enriching conversations. I also thank Oliver Weeger for our discussions and collaboration about viscoelastic models during his various visits to Kaiserslautern. A special thanks to Michael Dörfel and Anh-Vu Vuong, whose initial works and codes helped me get started with my research and develop an understanding of the implementation.

I also gratefully acknowledge the contribution of the following people in enhancing my practical knowledge of this work field. Dr. Josef Kiendl for our healthy discussions and his tips in dealing with shells at a summer school in Italy and via emails. Dr. Martin Schulze and Dr. Stefan Dietz from the Simpack AG, who as our industry partners in projects provided us interesting practical models to work with and also supported me to connect our shell code to the Simpack software.

I thank my parents for letting me do whatever I wished for in life and being supportive of it, although that meant we not seeing each other for long time periods. I also thank my dearest friends Chhitiz and Rina for always being there as our family. They made me feel at home in Germany with their company. Most of all, I am grateful to my wife Priyanka, who was always there for me by my side. Without her love, support and patience, I could not have completed this thesis.

The work presented in this thesis was funded by the German Ministry for Education and Research through the research projects "SOFA" and "MusiKa", which provided me with an opportunity to collaborate with university and industry partners for joint research goals. I thank them from the bottom of my heart.

Anmol Goyal
April, 2015

Contents

1	Introduction	3
2	Multibody Dynamics	8
2.1	Differential Algebraic Equation (DAE)	8
2.1.1	Basic Forms of DAEs	8
2.1.2	Index Concepts	10
2.1.3	Time Integration	11
2.2	Multibody System (MBS)	15
2.2.1	Equations of Motion	15
2.2.2	Index and Minimax Characterization	18
2.2.3	Stiff System	20
2.2.4	Scaling of Equations of Motion	21
2.2.5	Flexible Multibody System	24
3	Shell Theory	26
3.1	Choice of Kinematic Model	26
3.2	Kirchhoff-Love Model	28
3.2.1	Kinematics, Material Law and Energy Expressions	29
3.2.2	External Energy	32
3.2.3	From the Functional to the Weak Form	33
3.3	Dirichlet Boundary Conditions	35
3.4	Stress Recovery	36
3.5	Viscoelasticity	38
4	Isogeometric Analysis on Shells	41
4.1	Geometry Fundamentals	41
4.1.1	Bézier Curves	43
4.1.2	B-Splines	43
4.1.3	NURBS	46
4.1.4	Geometry Refinement	48
4.1.5	Point Projection	48
4.2	Isogeometric Analysis (IGA) using NURBS	50
4.2.1	Elements	50
4.2.2	Mesh Refinement	50
4.2.3	Dirichlet Boundary Conditions	51
4.2.4	Advantages	51

4.3	Application of IGA to Shells	52
4.4	Multipatch Shells	54
4.4.1	Motivation	55
4.4.2	Penalty Approach - Bending Strip Method	55
4.4.3	Scaling on the Bending Strip Method	61
5	Penalty-free Formulations for Multipatch Shells	63
5.1	Alternative Formulation with Bending Strips	63
5.1.1	Connection with Stiff Systems	64
5.1.2	Redundancy Issues	65
5.1.3	Treating Redundancy	65
5.2	Alternative Formulation with Continuity Constraints	78
5.3	Summary	79
6	Applications and Examples	81
6.1	Extracting NURBS Surfaces from CAD Files	81
6.1.1	CAD File with NURBS Data	81
6.1.2	Make Analysis Suitable	81
6.1.3	Extract NURBS Surface Data	82
6.1.4	Initialize IGA Shell Solver	82
6.2	IGA Shell Solver	84
6.2.1	Underlying Equation System	84
6.2.2	Information Flow	85
6.3	MBS-Shell Coupling	87
6.3.1	The Coupling	88
6.3.2	The Simulation	90
6.4	Numerical Results	90
6.4.1	Single Patch	91
6.4.2	Multipatch	94
6.4.3	Viscoelasticity	102
6.4.4	Coupled MBS	104
7	Conclusion	107
	Bibliography	109

Chapter 1

Introduction

In today's competitive world, many tasks are performed using complex machinery for efficiency reasons. To reduce costs, it is important to analyze these structures via computer simulations before actually producing them. A structural analysis helps predict the expected behavior of individual components and suggests amongst others the deformation, developed strains and stresses or the critical areas where a part might fail under a particular loading. One can not only study individual rigid or elastic parts, but also a system created by interconnecting different bodies. A collection of rigid bodies connected through joints or force elements is called a multibody system, whereas if elastic bodies are also included then it is called a flexible multibody system. Examples of real life objects analyzed as flexible multibody systems are automobiles, aeroplanes, wind turbines, or for that matter all kinds of mechanical devices.

Flexible multibody systems aim at applications such as the analysis of lightweight and high-precision mechanical systems where the elasticity of certain bodies needs to be taken into account. In practice, it is often sufficient to represent the elastic deformation in terms of a few modes that are derived by model reduction techniques [CK06]. To some extent, the corresponding computational methods are provided by commercial finite element packages, and specific interfaces have been developed to import the resulting discrete elastic model into a multibody code [Wal91]. This approach, however, has its limitations from a multibody system perspective. For one, it is based on the assumptions of linear elasticity, and large deformation or complex nonlinearities are mostly not covered. A second issue is the necessity to connect to a finite element code for the discretization and preprocessing.

Due to the increasing demand for refined elastic models that include specific nonlinearities, tailored spatial discretizations that are generated within a multibody formalism become more attractive. For a long time, specific beam models have been standard [BP92], but also nonlinear shells have been studied in combination with so-called absolute nodal coordinates [BS09] that do not rely on the method of floating reference frames [Sha98]. In this contribution, we also skip the concept of a floating reference frame, take up the latest development in finite element methods, the approach of isogeometric analysis [HCB05, CHB09], and apply it to shells within a multibody system.

Isogeometric analysis (IGA) extends isoparametric finite elements to more general functions such as B-Splines and Non-Uniform Rational B-Splines (NURBS) [PT97, Rog01]. In this way, exact geometries at the coarsest level of discretization are obtained, and geometry errors are eliminated from the very beginning. The resulting discretization still fits into the

variational framework of the finite element method, and the usage of NURBS as basis functions in the numerical approximation offers substantial savings with respect to the number of degrees of freedom (dofs) due to an improved convergence behavior. A particular advantage is the promise of simplifying the mesh generation step, and mesh refinement is easily performed by eliminating the need for communication with the geometry representation in a Computer-Aided Design (CAD) [RA90, Far96, FHK02] tool.

In multibody dynamics, this new type of finite element discretization seems particularly promising in combination with geometrically exact shell models, as shell analysis is extremely sensitive to geometric imperfections. Shells are thin walled structures where one dimension is significantly smaller than the other two. The use of shells minimizes the material used by reducing dead loads. When designed optimally, they carry loads parallel to their centerline. Moreover, they are also computationally efficient, since for the analysis dimension reduction is possible, leading to large savings in dofs and computing times. Following [KBLW09], we concentrate in our work on nonlinear, thin to moderately thick Kirchhoff-Love shells. To solve an elasticity problem on such a shell, the parametrization of the shell's mid-surface needs a high regularity, which can be easily achieved with NURBS but is hard to realize with standard finite elements. The higher smoothness leads also to impressive convergence rates, and even for a coarse mesh, higher order splines such as cubics, quartics or quintics yield already quite small errors. Target applications are twist beam rear suspensions and wind turbine blades.

The geometries of elastic bodies involved are often quite complicated and require a multipatch description for their modeling. Also, multipatches are required when the underlying regularity requirements of the model are not fulfilled at all the areas inside a patch. For the analysis of a multipatch structure, each patch is first analyzed separately and then joined together taking care of the interface conditions. These may also be classified under the domain decomposition methods (DDM), where the original domain is split into many smaller domains due to various reasons such as when the domains have different physics, for contact problems, for parallel computing, etc. The study of DDM in the framework of the IGA is carried out in [ASWB14], where various techniques such as the penalty approach, Lagrange multiplier method, perturbed Lagrange method and the Nitsche methods for joining the domains are discussed. The IETI method proposed in [KPJT12] is another technique used to join domains with C^0 continuity in the framework of the NURBS based IGA.

To join multipatch shell structures, in addition to the C^0 continuity, the interface conditions must additionally preserve the angle of kink in case of the C^0 continuous patch interfaces and the G^1 continuity for smooth connections. They are needed in order to transfer the moments across the intersection and prevent the hinge like effect at the joint. A penalty approach to join isogeometric shell patches is studied in [KBH⁺10], where the violation of the joint properties is penalized using bending strips. Another approach to handle isogeometric shell patch intersections is proposed in [BHB⁺13], where the thin and thick shell theories are blended together and each triple of control points at the joint is considered as a rigid body and its dynamics are introduced in the complete system.

It is one of the main goals of this work to point out the connection between the penalty approach and the class of so-called stiff mechanical systems [Lub93, Sim13], with particular emphasis on transient problems and Kirchhoff-Love shells. A stiff mechanical system consists of the equations of motion

$$\mathbf{M}(\mathbf{d})\ddot{\mathbf{d}} = \mathbf{f}_n(\mathbf{d}, \dot{\mathbf{d}}) - \frac{1}{\epsilon^2} \nabla U(\mathbf{d}) \quad (1.1)$$

where $\mathbf{d}(t)$ stands for the degrees of freedom after spatial discretization, \mathbf{f}_n denotes the non-

stiff forces and the stiff part U a potential that leads to the stiff force term $\epsilon^{-2}\nabla U(\mathbf{d})$ with penalty parameter ϵ . Moreover, \mathbf{M} is the mass matrix of the system.

We will show that the recently introduced bending strip method [KBH⁺10] for the treatment of multipatch shell structures leads to equations of motion in the form (1.1). As a consequence, the performance of numerical methods both in the stationary and transient case might suffer from problems such as ill-conditioning and poor convergence.

The reason for these problems stems from the fact that (1.1) is a *singularly perturbed system* that in the limit $\epsilon \rightarrow 0$ tends to a differential-algebraic equation (DAE) of index-3 that can be written as

$$\begin{aligned} \mathbf{M}(\mathbf{d}^0)\ddot{\mathbf{d}}^0 &= \mathbf{f}_n(\mathbf{d}^0, \dot{\mathbf{d}}^0) - \mathbf{\Gamma}^T(\mathbf{d}^0)\boldsymbol{\mu}^0, \\ \mathbf{0} &= \boldsymbol{\gamma}(\mathbf{d}^0) \end{aligned} \quad (1.2)$$

in terms of the smooth limit solution \mathbf{d}^0 and additional Lagrange multipliers $\boldsymbol{\mu}^0$. The constraints $\boldsymbol{\gamma}(\mathbf{d}^0)$ vanish whenever $\nabla U(\boldsymbol{\gamma}(\mathbf{d}^0)) = \mathbf{0}$, and $\mathbf{\Gamma} = \partial\boldsymbol{\gamma}/\partial\mathbf{d}$ stands for the constraint Jacobian.

It is well-known that the direct numerical solution of the DAE (1.2) is challenging and requires special care [HW96], but the alternative treatment of the stiff system (1.1) actually exhibits additional phenomena such as high frequency oscillations and a dependence of the maximum time step size on the penalty parameter whenever this parameter is small enough. For this reason, we study here alternative formulations that are penalty-free. Our overall goal is the implementation of isogeometric Kirchhoff-Love shells in flexible multibody dynamics due to various benefits, see [GDSV13]. There it is quite important that the resulting system we get is non-stiff, as otherwise the time integration is badly affected.

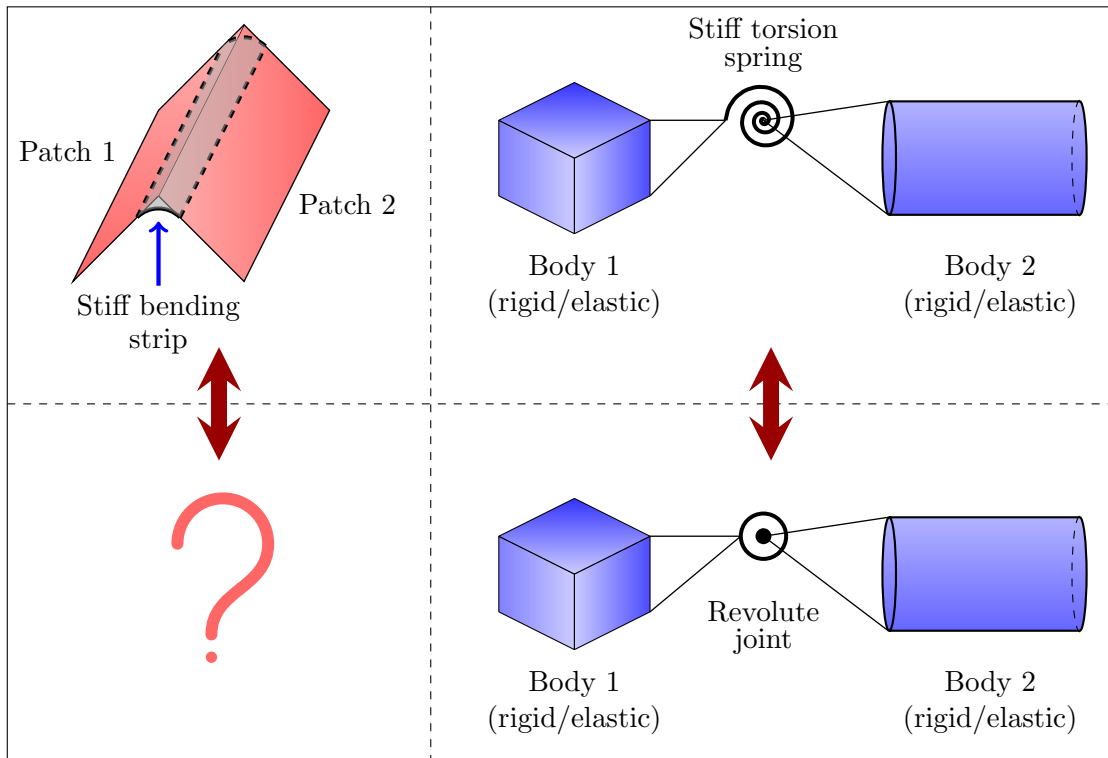


Figure 1.1: The bending strip approach and its analogy to stiff mechanical systems

Fig. 1.1 illustrates our approach. We make use of the analogy between mechanical multi-body systems where we may choose between joints or stiff springs to express the coupling between pairs of bodies. The bending strip can be viewed as the analogue of a stiff spring, and correspondingly we investigate the limit case of infinite stiffness, leading to a constraint $\mathbf{0} = \gamma(\mathbf{d}^0)$ as coupling condition. It turns out, however, that the resulting alternative formulation has redundant constraints, i.e., the Jacobian $\mathbf{\Gamma}$ has a rank deficiency. We address this issue and introduce an algorithm that uses an LU decomposition with full pivoting in order to eliminate the redundancy [dJL13]. Finally, we study also another alternative formulation with continuity constraints applied to triples of control points at the interface. The approach presented here to tackle stiff systems is quite general and can be applied to all penalty problems fulfilling some regularity requirements from [Lub93].

This work develops shell models that can be studied in the multipatch setting for large deformations. We include effects such as viscoelasticity to introduce dissipation, which is necessary to remove high frequency oscillations in the transient case. Furthermore, we establish a coupling between the multibody systems and the shell structures, either solving the complete system monolithically in a single solver or via co-simulation in specialized solvers for the multibody system and elastic shell structures respectively.

Outline of the thesis

In Chapter 2, we discuss the fundamentals of the multibody dynamics. We study the theory of the differential algebraic equations, which serve as the backbone of the equations of the multibody systems. We introduce the basic forms of the DAEs and the index concept, which points to the difficulty in solving them. We also present some important aspects about time integration and look into time integration schemes that can be used to solve general DAEs and in process its sub-class, the multibody systems. Then, we derive the equations of the multibody system and study their properties. Stiff systems which can lead to numerical troubles like ill-conditioning are introduced. We discuss the idea of scaling, which can up to some extent alleviate the conditioning issues. Lastly, we define the flexible multibody systems and their equations, where later we replace the elastic body equations with the shell system for our applications.

In Chapter 3, we introduce the shell theory employed throughout this work. We state the assumptions of two famous shell models, namely the Reissner-Mindlin and the Kirchhoff-Love models. Following our application goals, we choose the Kirchhoff-Love model, which is used for thin to moderately thick shells. We consider various aspects of the model such as the kinematics, material law, energy expressions, boundary conditions, etc., that are important to define the elasticity problem on the shells. In order to be able to discretize this problem for numerical computations, we derive a weak formulation of the elasticity problem, which is solved by applying finite element schemes. The goal is to obtain the system unknowns like the displacements. This is discussed in Chapter 4. For engineering applications, the displacement might not be the final quantity of interest, but physical quantities like strains and stresses. We present the theory of stress recovery to obtain them in the post processing step. Lastly, the shell theory is extended to include the effects of dissipation in the transient analysis. This is based on the Kelvin-Voigt model for viscoelasticity, following [LLT13].

In Chapter 4, we explain the approach of isogeometric analysis and apply it to study the Kirchhoff-Love shells. We start with the geometry fundamentals. The Non-Uniform Rational B-Splines are introduced as generalizations of the well known Bézier curves and B-Splines. A NURBS surface is defined, that is used to describe the centerline of a shell. We explain

the concepts of refinement and point projection for NURBS surfaces, in order to use them later for analysis. Furthermore, we introduce the concept of isogeometric analysis, which uses NURBS as basis functions to solve the weak problem from the previous chapter in a discretized subspace. The procedures to construct computational meshes and apply refinement are explained. With NURBS as basis functions, the Dirichlet boundary condition cannot be applied pointwise in the general setting and is introduced through the weak formulation. The various benefits of using the NURBS based IGA are stated. Lastly, the handling of the multipatch shell structures is discussed and in particular the bending strip method, which is a penalty approach, is analyzed. Furthermore, the ill-conditioning sources of the method are identified. A scaling technique, following [Bau11], is applied to alleviate some of the conditioning issues. One such ill-conditioning source is the penalty term stemming from the artificial bending strip stiffness. This penalty parameter dependence, that additionally introduces a time step size restriction for transient analysis, is eliminated in the next chapter.

In Chapter 5, we develop penalty free alternative formulations. One alternative formulation is based on the bending strip method, where we point to an analogy of the system to the stiff mechanical systems. Following this, the stiff bending strip forces are identified and written as separate constraints, following [Lub93]. The methodology employed is however quite general and can be used to treat all penalty approaches under some mild convexity and smoothness assumptions. Our goal is to observe the limit case of the bending strip method, fill in the gap mentioned in Fig. 1.1 and derive joint constraints between the patches. The bending strip constraints obtained in the process are however redundant. To tackle this issue, we study different approaches such as the analysis of the bending strip force constraints, using the perturbed Lagrange method and using linear algebra to work with the independent equations in each Newton iteration while solving the nonlinear problem. We also use an alternative formulation that employs directly the continuity constraints applied to each triple of control points at the joints, instead of the bending strip constraints. Finally, we summarize the benefits of using the alternative formulations over the penalty approach.

In Chapter 6, we focus on applications and present the procedure to solve shell problems. We begin by extracting the geometry data like the NURBS information, control points, etc. from the standard CAD data formats used in the industry. The geometry data along with material data and boundary conditions help initialize the IGA shell solver, whose schematic is also presented. An important application is the connection of our shell solver to the multibody systems. We present a pointwise interaction approach to transfer data between both the systems. Lastly, we present our numerical results from various solvers and highlight their benefits over the standard techniques.

Chapter 7 briefly summarizes the main topics and achievements of this work.

Chapter 2

Multibody Dynamics

In this chapter, we present the fundamentals of the multibody systems and give an overview on the mathematical models to study the multibody dynamics. To begin with, the underlying theory of differential algebraic equations, which is the backbone of multibody system equations is discussed. The difficulty in solving such equations is studied in form of the index concepts. Moreover, we discuss the important aspects about the time integration of such systems. We derive the equations of motion for the multibody systems. We study the index of the mathematical model and show the minimax characterization of the constraints, which points to an analogy with the famous inf-sup condition for constraints on elastic bodies. Then, a classification of the system in terms of stiff system is identified. To improve the properties of the multibody system, we apply scaling. Finally, the flexible multibody systems are defined for a general setting, where the equations of the elastic body will be replaced later with the equations of shells that we derive in the Chapter 3.

2.1 Differential Algebraic Equation (DAE)

The dynamical behavior of physical systems is modeled via differential equations. If the states of such systems are constrained, for example due to presence of a translation joint such that the body can move only along a straight line, then the mathematical model also includes algebraic equations to describe these constraints. Such systems consisting of both differential and algebraic equations are called the *differential-algebraic systems* or the *DAEs*. This section presents only a brief discussion on the theory of DAEs. We refer to the monographs of Brennan, Campbell and Petzold [BCP96], Hairer and Wanner [HW96], Kunkel and Mehrmann [KM06] for an in-depth analysis of the topic.

2.1.1 Basic Forms of DAEs

The most general form of a time dependent differential equation is the fully implicit system

$$\mathbf{f}(\dot{\mathbf{x}}, \mathbf{x}, t) = \mathbf{0} \tag{2.1}$$

with state variables $\mathbf{x}(t) \in \mathbb{R}^{n_x}$ and a nonlinear, vector-valued function \mathbf{f} of the corresponding dimension. If the $n_x \times n_x$ Jacobian $\partial \mathbf{f} / \partial \dot{\mathbf{x}}$ is invertible, then by implicit function theorem, it is in principle possible to transform (2.1) to a system of *ordinary differential equations* or

ODEs at least locally. These are the differential equations without constraints. Otherwise, if $\partial\mathbf{f}/\partial\dot{\mathbf{x}}$ is singular, (2.1) constitutes a *fully implicit system of differential-algebraic equations*.

In most of the applications, more information about the problem at hand is available. One can usually build more specific DAEs with a better structure. An important class are the *linear-implicit systems* of the form

$$\mathbf{E}\dot{\mathbf{x}} = \phi(\mathbf{x}, t) \quad (2.2)$$

with a singular matrix $\mathbf{E} \in \mathbb{R}^{n_x \times n_x}$.

Assuming a constant matrix \mathbf{E} , we apply either the Gaussian elimination with full pivoting or the singular value decomposition [TB97] to transform \mathbf{E} as

$$\mathbf{U}\mathbf{E}\mathbf{V} = \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \quad (2.3)$$

with invertible matrices \mathbf{U}, \mathbf{V} . The block matrix on the right consists of an identity matrix \mathbf{I} of the same size as the rank of \mathbf{E} . Taking

$$\mathbf{V}^{-1}\mathbf{x} =: \begin{pmatrix} \mathbf{y} \\ \mathbf{z} \end{pmatrix} \quad (2.4)$$

of appropriate dimensions and multiplying (2.2) from the left by \mathbf{U} , we get

$$\begin{pmatrix} \dot{\mathbf{y}} \\ \mathbf{0} \end{pmatrix} = \mathbf{U}\phi\left(\mathbf{V}\begin{pmatrix} \mathbf{y} \\ \mathbf{z} \end{pmatrix}, t\right) =: \begin{pmatrix} \mathbf{a}(\mathbf{y}, \mathbf{z}, t) \\ \mathbf{b}(\mathbf{y}, \mathbf{z}, t) \end{pmatrix}. \quad (2.5)$$

Writing this in the autonomous form by adding the equation $\dot{t} = 1$ and appending the time variable to vector \mathbf{y} , we obtain the *semi-explicit system*

$$\dot{\mathbf{y}} = \mathbf{a}(\mathbf{y}, \mathbf{z}), \quad (2.6a)$$

$$\mathbf{0} = \mathbf{b}(\mathbf{y}, \mathbf{z}). \quad (2.6b)$$

The differential-algebraic system (2.6) shows a clear separation into n_y differential equations (2.6a) for the *differential variable* $\mathbf{y}(t) \in \mathbb{R}^{n_y}$ and n_z constraints (2.6b), which define the *algebraic variables* $\mathbf{z} \in \mathbb{R}^{n_z}$.

For the convergence analysis of numerical time integration methods, the system (2.6) is usually the easiest starting point. If the method is invariant under a transformation from the linear-implicit system (2.2) to (2.6), the convergence results then also hold for (2.2).

Example 2.1. *In the limit $\epsilon \rightarrow 0$, van der Pol's equation [VdP26]*

$$\dot{y} = z, \quad (2.7a)$$

$$\epsilon\dot{z} = (1 - y^2)z - y, \quad (2.7b)$$

is an example of a semi-explicit DAE system with scalar variables y and z . It is also a singularly perturbed differential equation system, since the differential equation (2.7b) leads to an algebraic constraint in the limit $\epsilon \rightarrow 0$.

Such a close relation between a singularly perturbed ODE system and a DAE system is quite common and can be found in various application fields. See Section 2.2.3, where we present the stiff mechanical systems as singularly perturbed systems in the limit case and construct the corresponding DAE form following [Lub93].

2.1.2 Index Concepts

The index of a DAE is a measure of its well-posedness. Also as a rule of thumb, the greater the index of a DAE is, the more difficult it gets to solve the problem. Many definitions of the index exist and have evolved over several decades. We discuss here the two most frequently used variants, namely the differentiation and the perturbation index.

2.1.2.1 Differentiation Index

This index concept is applicable to the fully implicit system (2.1). We define the index k , such that

$k = 0$: If $\partial \mathbf{f} / \partial \dot{\mathbf{x}}$ is non-singular, the index is 0.

$k > 0$: Otherwise, consider the system of equations

$$\begin{aligned} \mathbf{f}(\dot{\mathbf{x}}, \mathbf{x}, t) &= \mathbf{0}, \\ \frac{d}{dt} \mathbf{f}(\dot{\mathbf{x}}, \mathbf{x}, t) &= \frac{\partial}{\partial \dot{\mathbf{x}}} \mathbf{f}(\dot{\mathbf{x}}, \mathbf{x}, t) \mathbf{x}^{(2)} + \dots = \mathbf{0}, \\ &\vdots \\ \frac{d^s}{dt^s} \mathbf{f}(\dot{\mathbf{x}}, \mathbf{x}, t) &= \frac{\partial}{\partial \dot{\mathbf{x}}} \mathbf{f}(\dot{\mathbf{x}}, \mathbf{x}, t) \mathbf{x}^{(s+1)} + \dots = \mathbf{0} \end{aligned} \quad (2.8)$$

as a system in the separate dependent variables $\dot{\mathbf{x}}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(s+1)}$, with \mathbf{x} and t as independent variables. Then the *differentiation index* k is the smallest s for which it is possible, using algebraic manipulations only, to extract an ODE $\dot{\mathbf{x}} = \boldsymbol{\psi}(\mathbf{x}, t)$ from (2.8).

We consider next the semi-explicit system (2.6). We assume the Jacobian matrix

$$\frac{\partial \mathbf{b}}{\partial \mathbf{z}}(\mathbf{y}, \mathbf{z}) \in \mathbb{R}^{n_z \times n_z} \quad \text{is invertible} \quad (2.9)$$

in a neighborhood of the solution. Differentiating (2.6b) then leads to

$$\mathbf{0} = \frac{d}{dt} \mathbf{b}(\mathbf{y}, \mathbf{z}) = \frac{\partial \mathbf{b}}{\partial \mathbf{y}}(\mathbf{y}, \mathbf{z}) \dot{\mathbf{y}} + \frac{\partial \mathbf{b}}{\partial \mathbf{z}}(\mathbf{y}, \mathbf{z}) \dot{\mathbf{z}}.$$

This implies

$$\dot{\mathbf{z}} = - \left(\frac{\partial \mathbf{b}}{\partial \mathbf{z}}(\mathbf{y}, \mathbf{z}) \right)^{-1} \frac{\partial \mathbf{b}}{\partial \mathbf{y}}(\mathbf{y}, \mathbf{z}) \cdot \mathbf{a}(\mathbf{y}, \mathbf{z}), \quad (2.10)$$

which is the desired ODE for the variable \mathbf{z} . This implies that (2.6) has an index $k = 1$, if assumption (2.9) holds. The underlying ODE system comprises of (2.6a), (2.10). An initial value $(\mathbf{y}_0, \mathbf{z}_0)$ is said to be *consistent* if the constraint $\mathbf{0} = \mathbf{b}(\mathbf{y}_0, \mathbf{z}_0)$ is satisfied.

2.1.2.2 Perturbation Index

The *perturbation index* introduced by Hairer, Lubich and Roche [HLR89] measures the sensitivity of the solutions to the perturbations in the equations.

The system (2.1) has index $k \geq 1$ along a solution $\mathbf{x}(t)$ on $[t_0, t_1]$ if k is the smallest integer, such that for all functions $\hat{\mathbf{x}}$ having a defect

$$\mathbf{f}(\dot{\hat{\mathbf{x}}}, \hat{\mathbf{x}}, t) = \boldsymbol{\delta}(t),$$

there exists on $[t_0, t_1]$ an estimate

$$\|\hat{\mathbf{x}}(t) - \mathbf{x}(t)\| \leq c \left(\|\hat{\mathbf{x}}(t_0) - \mathbf{x}(t_0)\| + \max_{t_0 \leq \xi \leq t} \|\delta(\xi)\| + \cdots + \max_{t_0 \leq \xi \leq t} \|\delta^{(k-1)}(\xi)\| \right)$$

whenever the expression on the right hand side is sufficiently small. Note that the constant c depends only on \mathbf{f} and on the length of the interval, but not on the perturbation δ . The perturbation index $k = 1$, if

$$\|\hat{\mathbf{x}}(t) - \mathbf{x}(t)\| \leq c \left(\|\hat{\mathbf{x}}(t_0) - \mathbf{x}(t_0)\| + \max_{t_0 \leq \xi \leq t} \left\| \int_{t_0}^{\xi} \delta(\tau) d\tau \right\| \right),$$

which is satisfied for ODEs. If the perturbation index exceeds 1, derivatives of the perturbation show up in the estimate and indicate a certain degree of ill-posedness.

2.1.3 Time Integration

In this section, we discuss important aspects about time integration of DAEs. A brief insight is provided into the basic theory for the stability and convergence of time integration methods. We refer the reader to [DB02, BCP96, AB07] for a detailed analysis on the topic. Furthermore, we study the backward differentiation formulas [BCP96] and the generalized- α method extended to DAEs [AB07] as time integration schemes one can employ to solve multibody systems.

We begin by a formal definition of the initial value problem.

Definition 2.1. *The **initial value problem**, referred further as the **IVP**, is to find $\mathbf{x}(t) \in \mathbb{R}^{n_x}$ such that for $\mathbf{f}: \mathbb{R}^{2n_x+1} \rightarrow \mathbb{R}^{n_x}$ the fully implicit DAE (2.1)*

$$\mathbf{f}(\dot{\mathbf{x}}, \mathbf{x}, t) = \mathbf{0} \text{ with } \mathbf{x}(0) = \mathbf{x}_0 \quad (2.11)$$

in the compact interval $t \in [0, T] \subset \mathbb{R}$, $T < \infty$.

To solve the IVP, we discretize the time domain: $0 = t_0 < t_1 < \cdots < t_n = T$. We take a fixed step size h such that $t_i = t_0 + ih$, $i = 0, \dots, n$. However, the following theory can be extended to variable step sizes. A discretization scheme is employed to compute the *discrete solution* $\mathbf{x}_i \in \mathbb{R}^{n_x}$ at time t_i based on the solutions from the previous time steps. We denote the *exact solution* at time t_i by $\mathbf{x}(t_i) \in \mathbb{R}^{n_x}$. The goal is to find the discrete solution \mathbf{x}_n that converges to the exact solution $\mathbf{x}(t_n)$ for $n \rightarrow \infty$.

We represent a one step *discretization scheme* for the IVP (2.11) by the iteration

$$\mathbf{x}_i = \mathbf{g}(\mathbf{f}, \mathbf{x}_{i-1}, h, t_i), \quad (2.12)$$

where $\mathbf{x}_i \in \mathbb{R}^{n_x}$ is the resulting discrete solution at time t_i based on the value from only one previous time step t_{i-1} . One might also employ a k -step discretization scheme that uses values from k previous time steps, but for the ease of notation we use here a one step scheme.

Definition 2.2. *The discretization scheme (2.12) is called **consistent with order p** , if the error generated after one iteration satisfies*

$$\|\mathbf{x}(t_i) - \mathbf{g}(\mathbf{f}, \mathbf{x}(t_{i-1}), h, t_i)\| \leq c_1 h^{p+1}, \quad (2.13)$$

with a constant c_1 independent of h .

Definition 2.3. The discretization scheme (2.12) is called **stable**, if

$$\|\mathbf{g}(\mathbf{f}, \mathbf{x}(t_{i-1}), h, t_i) - \mathbf{g}(\mathbf{f}, \mathbf{x}_{i-1}, h, t_i)\| \leq (1 + c_2 h) \|\mathbf{x}(t_{i-1}) - \mathbf{x}_{i-1}\| \quad (2.14)$$

with a constant c_2 independent of h .

Definition 2.4. The discretization scheme (2.12) is called **convergent with order p** , if the global error satisfies

$$\|\mathbf{x}(t_n) - \mathbf{x}_n\| \leq c_3 h^p, \quad n \rightarrow \infty \quad (2.15)$$

provided that the initial values satisfy $\|\mathbf{x}(t_0) - \mathbf{x}_0\| \leq c_4 h^p$ with the constants c_3, c_4 independent of h .

Theorem 2.1. If the discretization scheme (2.12) is stable and consistent with order p , then it is **convergent with order p** .

Proof. Refer to the book of Kunkel and Mehrmann [KM06]. □

With a goal to solve stiff systems in this work later, another important stability concept is the *A-stability*. The stiff systems appear either due to stiff components in a mechanical system or due to the finite elements structures with multipatches joined using stiff bending strips in Section 4.4. Stiff systems have large spectral radii that have either strongly negative real parts; or lie on or close to the imaginary axis with low real parts. Such systems therefore require the more restrictive A-stable discretization in order to converge. For a detailed discussion on the numerical analysis and issues related to stiff systems, we refer to the monograph from Simeon [Sim13].

Let us consider the one dimensional Dahlquist test equation

$$\dot{x} = \lambda x, \quad \lambda \in \mathbb{C}^-, \quad (2.16)$$

where $\mathbb{C}^- = \{z \in \mathbb{C} | \operatorname{Re}(z) < 0\}$.

Applying the discretization scheme (2.12) to (2.16) and simplifying gives us

$$x_i = R(\mu)x_{i-1}, \quad (2.17)$$

where $\mu := \mu(h, \lambda)$. In the literature, $R(\mu)$ is called a *stability function*.

Definition 2.5. A discretization scheme is said to be **A-stable**, if the stability function satisfies

$$|R(\mu)| \leq 1, \quad \forall \mu \in \mathbb{C}^-. \quad (2.18)$$

It can be easily shown that an *A-stable*, consistent discretization scheme also satisfies the stability condition (2.14) and hence leads to a convergent scheme by Theorem 2.1.

2.1.3.1 Backward Differentiation Formulas

k-step *Backward Differentiation Formulas* or the *BDF-k* approximate the derivative of a function $\dot{\mathbf{x}}(t)$ at time t_i in terms of the values $\mathbf{x}_{i-k}, \dots, \mathbf{x}_{i-1}$ from the last k steps as well as the current unknown approximation \mathbf{x}_i . To solve a fully-implicit DAE (2.1), take

$$\mathbf{f}(D_h \mathbf{x}_i, \mathbf{x}_i, t_i) = \mathbf{0}, \quad \dot{\mathbf{x}}(t_i) \approx D_h \mathbf{x}_i = \frac{1}{h} \sum_{l=0}^k \alpha_{k-l} \mathbf{x}_{i-l}, \quad (2.19)$$

where D_h denotes a *backward difference operator*.

Implicit Euler. BDF-1 scheme or more popularly the *implicit-Euler* is a first order, A-stable scheme with the backward difference operator

$$D_h \mathbf{x}_i = \frac{1}{h} (\mathbf{x}_i - \mathbf{x}_{i-1}) = \dot{\mathbf{x}}(t_i) + \mathcal{O}(h). \quad (2.20)$$

BDF-2. BDF-2 scheme is a second order, A-stable scheme with the backward difference operator

$$D_h \mathbf{x}_i = \frac{1}{h} \left(\frac{3}{2} \mathbf{x}_i - 2 \mathbf{x}_{i-1} + \frac{1}{2} \mathbf{x}_{i-2} \right) = \dot{\mathbf{x}}(t_i) + \mathcal{O}(h^2). \quad (2.21)$$

We only consider in this section the BDF-1 and BDF-2 due to the restriction from the *Dahlquist second barrier*. It states that for ODEs a linear multistep method which is A-stable, cannot have an order more than 2. Moreover for the DAE case, although the methods are convergent, they are observed to undergo order reduction depending on the index of the DAEs [HW96, BCP96]. For index-1 DAEs, no order reduction is observed, whereas for index-2 cases there is a reduction in the local order of the algebraic variable, although since the BDFs possess no memory, this does not affect the global convergence order. For the index-3 case, eg. constrained mechanical systems, the global order is reduced by 1.

After the application of the appropriate BDF-k method, we obtain in each time step t_i in the general case a nonlinear problem $\mathbf{f}(D_h \mathbf{x}_i, \mathbf{x}_i, t_i) = \mathbf{0}$ with unknowns \mathbf{x}_i . This can be solved by using a variant of the Newton method.

2.1.3.2 Generalized- α Method

We now discuss a special integration method, that is a second order discretization scheme with *unconditional stability* which is equivalent to A-stability for a particular choice of parameters. This scheme is the so-called *generalized- α method* by Chung and Hulbert [CH93], that is designed especially for the applications in *structural dynamics*. It is especially suitable for stiff mechanical systems due to the possibility of controlling the numerical dissipation with a *dissipation parameter* $\rho_\infty \in [0, 1]$. Numerical dissipation is a desirable property whenever the high frequency motion of stiff systems need not be resolved. For $\rho_\infty = 0$, the high frequency system oscillations are damped out and for $\rho_\infty = 1$, there is no dissipation.

The extensions of the scheme for constrained mechanical systems has been actively studied for the last few decades. We discuss next the approach presented by Arnold and Brüls [AB07]. A particular benefit of the approach is that no order reduction is observed for both differential as well as algebraic variables. This is in contrast to the order reduction observed especially for the index-3 DAEs with the BDF methods.

The starting point are the equations of motion of a constrained mechanical system (2.37)

$$\mathbf{M}(\mathbf{q}) \ddot{\mathbf{q}} = \mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}, t) - \mathbf{G}(\mathbf{q})^T \boldsymbol{\lambda}, \quad (2.22a)$$

$$\mathbf{0} = \mathbf{g}(\mathbf{q}), \quad (2.22b)$$

where \mathbf{q} are generalized coordinates, \mathbf{M} is a symmetric mass matrix, \mathbf{f} is a force vector. The system constraints are given by \mathbf{g} , with constraint Jacobian \mathbf{G} and $\boldsymbol{\lambda}$ as the corresponding Lagrange multipliers. (2.22) is an index-3 DAE, whose derivation, assumptions and index calculation are discussed in Section 2.2.

The algorithm uses discrete values $\mathbf{q}_i, \dot{\mathbf{q}}_i, \ddot{\mathbf{q}}_i, \boldsymbol{\lambda}_i$ at time t_i that satisfy the dynamic equations (2.22). The auxiliary variables for the accelerations \mathbf{a} are computed using

$$(1 - \alpha_m)\mathbf{a}_i + \alpha_m\mathbf{a}_{i-1} = (1 - \alpha_f)\ddot{\mathbf{q}}_i + \alpha_f\ddot{\mathbf{q}}_{i-1}, \quad \mathbf{a}_0 = \ddot{\mathbf{q}}_0. \quad (2.23)$$

The auxiliary acceleration variables further give us

$$\mathbf{q}_i = \mathbf{q}_{i-1} + h\dot{\mathbf{q}}_{i-1} + h^2 \left(\frac{1}{2} - \beta \right) \mathbf{a}_{i-1} + h^2\beta\mathbf{a}_i, \quad (2.24)$$

$$\dot{\mathbf{q}}_i = \dot{\mathbf{q}}_{i-1} + h(1 - \gamma)\mathbf{a}_{i-1} + h\gamma\mathbf{a}_i. \quad (2.25)$$

The numerical parameters $\alpha_m, \alpha_f, \beta, \gamma$ define the properties of the scheme and are chosen based on [CH93].

Algorithm 2.1 is used to determine the discrete values $\mathbf{q}_i, \dot{\mathbf{q}}_i, \ddot{\mathbf{q}}_i, \boldsymbol{\lambda}_i$ at time t_i , which involves in the correction step the parameters

$$\beta' = \frac{1 - \alpha_m}{h^2\beta(1 - \alpha_f)}, \quad \gamma' = \frac{\gamma}{h\beta}, \quad (2.26)$$

and the iteration matrix

$$\mathbf{S}_t = \begin{pmatrix} (\mathbf{M}\beta' + \mathbf{C}_t\gamma' + \mathbf{K}_t) & \mathbf{G}^T \\ \mathbf{G} & \mathbf{0} \end{pmatrix}. \quad (2.27)$$

The tangent stiffness matrix $\mathbf{K}_t = \partial(\mathbf{M}\ddot{\mathbf{q}} - \mathbf{f} + \mathbf{G}^T\boldsymbol{\lambda})/\partial\mathbf{q}$ and the tangent damping matrix $\mathbf{C}_t = -\partial\mathbf{f}/\partial\dot{\mathbf{q}}$.

Algorithm 2.1 [$\mathbf{q}_i, \dot{\mathbf{q}}_i, \ddot{\mathbf{q}}_i, \boldsymbol{\lambda}_i, \mathbf{a}$] = AlphaStep($\mathbf{q}_{i-1}, \dot{\mathbf{q}}_{i-1}, \ddot{\mathbf{q}}_{i-1}, \mathbf{a}$)

```

1:  $\mathbf{q}_i := \mathbf{q}_{i-1} + h\dot{\mathbf{q}}_{i-1} + h^2(0.5 - \beta)\mathbf{a}$ 
2:  $\dot{\mathbf{q}}_i := \dot{\mathbf{q}}_{i-1} + h(1 - \gamma)\mathbf{a}$ 
3:  $\boldsymbol{\lambda}_i := \mathbf{0}$ 
4:  $\mathbf{a} := 1/(1 - \alpha_m)(\alpha_f\ddot{\mathbf{q}}_{i-1} - \alpha_m\mathbf{a})$ 
5:  $\mathbf{q}_i := \mathbf{q}_i + h^2\beta\mathbf{a}$ 
6:  $\dot{\mathbf{q}}_i := \dot{\mathbf{q}}_i + h\gamma\mathbf{a}$ 
7:  $\ddot{\mathbf{q}}_i := \mathbf{0}$ 
8: for  $j = 1$  to  $j_{\max}$  do
9:   Compute the residuals  $\mathbf{r}^q, \mathbf{r}^\lambda$  from (2.22)
10:  if  $\sqrt{\|\mathbf{r}^q\|^2 + \|\mathbf{r}^\lambda\|^2} < tol$  then
11:    break
12:  end if
13:   $\begin{pmatrix} \Delta\mathbf{q} \\ \Delta\boldsymbol{\lambda} \end{pmatrix} := -\mathbf{S}_t^{-1} \begin{pmatrix} \mathbf{r}^q \\ \mathbf{r}^\lambda \end{pmatrix}$ 
14:   $\mathbf{q}_i := \mathbf{q}_i + \Delta\mathbf{q}$ 
15:   $\dot{\mathbf{q}}_i := \dot{\mathbf{q}}_i + \gamma'\Delta\mathbf{q}$ 
16:   $\ddot{\mathbf{q}}_i := \ddot{\mathbf{q}}_i + \beta'\Delta\mathbf{q}$ 
17:   $\boldsymbol{\lambda}_i := \boldsymbol{\lambda}_i + \Delta\boldsymbol{\lambda}$ 
18: end for
19:  $\mathbf{a} := \mathbf{a} + (1 - \alpha_f)/(1 - \alpha_m)\ddot{\mathbf{q}}_i$ 

```

Refer to [AB07] for a detailed analysis of this scheme.

2.2 Multibody System (MBS)

In this section, we discuss the basic theory of multibody systems (MBS), their formulations and solutions as DAEs. For extensive details on this topic, refer to [EF98, VS99, Sha98, Sim13].

Definition 2.6. A multibody system, see Fig. 2.1, is a collection of interconnected rigid bodies, coupled using joints and/or force elements. The joints like the revolute, sliding joints, etc., limit the relative motion between pairs of bodies, whereas the force elements like springs, dampers, etc., give rise to reaction forces and moments. The bodies possess some defined mass and geometry, whereas the interconnections are assumed to be massless.

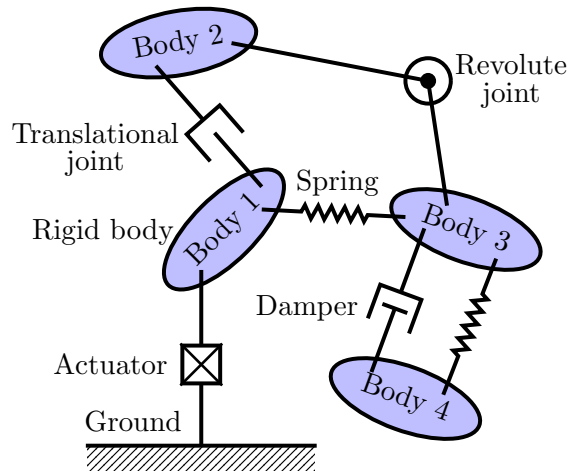


Figure 2.1: Multibody system with typical interconnections

2.2.1 Equations of Motion

We now define the well-known equations of motion of a multibody system.

2.2.1.1 Kinematics

Let $\mathbf{q}(t) \in \mathbb{R}^{n_q}$ be a vector that denotes n_q position and orientation coordinates of all the bodies in the system at time t . We discuss here the geometry of the motion \mathbf{q} without considering the forces that produce it. We express the conditions on \mathbf{q} resulting from the presence of the joints in terms of n_λ holonomic *constraint equations*

$$\mathbf{g}(\mathbf{q}) = \mathbf{0}, \quad \mathbf{g} : \mathbb{R}^{n_q} \rightarrow \mathbb{R}^{n_\lambda} \quad (2.28)$$

with the rectangular *constraint Jacobian matrix*

$$\mathbf{G}(\mathbf{q}) := \frac{\partial \mathbf{g}(\mathbf{q})}{\partial \mathbf{q}} \in \mathbb{R}^{n_\lambda \times n_q}. \quad (2.29)$$

A meaningful model requires that $n_\lambda < n_q$. Holonomic constraints are independent of the velocities $\dot{\mathbf{q}} := \frac{d\mathbf{q}}{dt}$. Moreover, there may also exist time t dependent constraints $\mathbf{0} = \mathbf{g}(\mathbf{q}, t)$, $\mathbf{g} : \mathbb{R}^{n_q+1} \rightarrow \mathbb{R}^{n_\lambda}$. But we omit this dependence for notational simplicity.

Remark 2.1. A standard assumption on the constraint Jacobian is the full rank condition

$$\text{rank } \mathbf{G}(\mathbf{q}) = n_\lambda, \quad (2.30)$$

which means that the constraint equations are linearly independent. In such a case, the degrees of freedom n_p is defined by $n_p = n_q - n_\lambda$.

At this point, two ways to formulate the equations of motion branch off. In the first approach, the governing equations are expressed in terms of the *redundant coordinates* \mathbf{q} and it uses additional *Lagrange multipliers* $\boldsymbol{\lambda}(t) \in \mathbb{R}^{n_\lambda}$ to take the constraint equations into account. Alternatively, in the second approach we introduce *minimal coordinates* $\mathbf{p}(t) \in \mathbb{R}^{n_p}$ for which the constraint equation (2.28) is automatically satisfied for all choices of \mathbf{p} . The redundant variables \mathbf{q} can be written as a function $\mathbf{q}(\mathbf{p})$ such that

$$\mathbf{g}(\mathbf{q}(\mathbf{p})) = \mathbf{0}, \quad \forall \mathbf{p}. \quad (2.31)$$

Until quite lately, the traditional approach for deriving the equation of motion has been in terms of the generalized (minimal) coordinates which automatically satisfy the constraints. However, the preferred choice of coordinates has shifted towards redundant coordinates. This is because, finding a set of minimal coordinates is not very easy and in case of complicated constraints e.g. for non-holonomic constraints (explicitly dependent on velocities), it becomes impossible to find such coordinates. Also, it might not be desirable from a numerical point of view to choose the minimal coordinates, e.g. in case of stiff systems in Section 2.2.3. Moreover, it is easier to interpret the physical meaning of the redundant coordinates.

2.2.1.2 Dynamics

Now we study the forces and their influence on the motion of the system. We derive the equations of motion with respect to redundant and minimal coordinates, leading to Lagrange equations of the first and the second kind respectively.

Lagrange Equations of the First Kind. Using the redundant coordinates \mathbf{q} and the Lagrange multipliers $\boldsymbol{\lambda}$, we derive following [VS99] the *Lagrange Equations of the first kind* using *Hamilton's principle of least action*.

Let the functionals $T := T(\mathbf{q}, \dot{\mathbf{q}})$, $T : \mathbb{R}^{2n_q} \rightarrow \mathbb{R}$ and $U := U(\mathbf{q})$, $U : \mathbb{R}^{n_q} \rightarrow \mathbb{R}$ denote the *kinetic* and the *potential energy* of an MBS respectively. Associate a Lagrange multiplier λ_i , $i = 1, \dots, n_\lambda$ with each one of the n_λ constraint equations (2.28). Then the system's *Lagrangian* is defined by

$$\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}}, \boldsymbol{\lambda}) := T(\mathbf{q}, \dot{\mathbf{q}}) - U(\mathbf{q}) - \mathbf{g}(\mathbf{q})^T \boldsymbol{\lambda}, \quad (2.32)$$

Hamilton's principle of least action states that given the positions $\mathbf{q}^{(0)}$ and $\mathbf{q}^{(1)}$ of the system at time instances t_0 and t_1 respectively, the line integral

$$\int_{t_0}^{t_1} (\mathcal{L} + W_{nc}) dt \rightarrow \text{stationary !} \quad (2.33)$$

called the *action* of the system is an extremum for the actual path of motion. W_{nc} denotes the work done by non-conservative forces \mathbf{f}_{nc} . Solving (2.33), [VS99, Sha98] gives

$$\frac{d}{dt} \frac{\partial T(\mathbf{q}, \dot{\mathbf{q}})}{\partial \dot{\mathbf{q}}} - \frac{\partial T(\mathbf{q}, \dot{\mathbf{q}})}{\partial \mathbf{q}} + \frac{\partial U(\mathbf{q})}{\partial \mathbf{q}} + \mathbf{G}(\mathbf{q})^T \boldsymbol{\lambda} = \mathbf{f}_{nc}(\mathbf{q}, \dot{\mathbf{q}}, t) \quad (2.34)$$

where $\mathbf{G}(\mathbf{q})^T \boldsymbol{\lambda}$ are the *constraint forces*, and $\mathbf{f}_{nc}(\mathbf{q}, \dot{\mathbf{q}}, t)$ is the vector of non-conservative forces due to W_{nc} . Additionally, the constraints (2.28) must be satisfied.

The kinetic energy T is defined as

$$T(\mathbf{q}, \dot{\mathbf{q}}) = \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{M}(\mathbf{q}) \dot{\mathbf{q}} \quad (2.35)$$

where $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{n_q \times n_q}$ denotes the *mass matrix* which includes the masses as well as moments and products of inertia. It gives us

$$\frac{d}{dt} \frac{\partial T(\mathbf{q}, \dot{\mathbf{q}})}{\partial \dot{\mathbf{q}}} = \frac{\partial^2 T(\mathbf{q}, \dot{\mathbf{q}})}{\partial \dot{\mathbf{q}}^2} \ddot{\mathbf{q}} + \frac{\partial^2 T(\mathbf{q}, \dot{\mathbf{q}})}{\partial \mathbf{q} \partial \dot{\mathbf{q}}} \dot{\mathbf{q}},$$

wherein

$$\frac{\partial^2 T(\mathbf{q}, \dot{\mathbf{q}})}{\partial \dot{\mathbf{q}}^2} = \mathbf{M}(\mathbf{q}), \quad \frac{\partial^2 T(\mathbf{q}, \dot{\mathbf{q}})}{\partial \mathbf{q} \partial \dot{\mathbf{q}}} = \frac{\partial \mathbf{M}(\mathbf{q}) \dot{\mathbf{q}}}{\partial \mathbf{q}} = \frac{d}{dt} \mathbf{M}(\mathbf{q}) =: \dot{\mathbf{M}}(\mathbf{q}, \dot{\mathbf{q}}).$$

We consolidate the non-conservative forces, the generalized Coriolis and centrifugal forces and the potential (or conservative) forces in a vector of generalized forces \mathbf{f} ,

$$\mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}, t) := \mathbf{f}_{nc}(\mathbf{q}, \dot{\mathbf{q}}, t) - \dot{\mathbf{M}}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} + \frac{\partial}{\partial \mathbf{q}} \left(\frac{1}{2} \dot{\mathbf{q}}^T \mathbf{M}(\mathbf{q}) \dot{\mathbf{q}} \right) - \frac{\partial U(\mathbf{q})}{\partial \mathbf{q}}. \quad (2.36)$$

Substituting the terms above in (2.34) and taking the constraints (2.28), we get the equations of motion of the MBS

$$\mathbf{M}(\mathbf{q}) \ddot{\mathbf{q}} = \mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}, t) - \mathbf{G}(\mathbf{q})^T \boldsymbol{\lambda}, \quad (2.37a)$$

$$\mathbf{g}(\mathbf{q}) = \mathbf{0}. \quad (2.37b)$$

This form is called the *Lagrange Equations of the first kind*.

The above equation is a system of *second order differential equations* with additional constraints. More precisely, it is a DAE system of *index-3* as shown later in Section 2.2.2.1. Due to problems with the numerical treatment of an index-3 DAE, (2.37) is reduced to a lower index DAE system, mostly index-1. For the numerical treatment of the index-1 DAEs, the most common method is the BDF method, see Section 2.1.3.1. The various solvers used commercially for solving the DAEs are DASSL [Pet82], RADAU5 [HW88]. DASSL uses the BDF method to solve index-1 DAEs. RADAU5 uses the 3-stage Radau collocation method to solve index - 1,2,3 DAEs.

Several other methodologies for the derivation of the governing equations are successfully applied in multibody dynamics like the principles of *d'Alembert* and *Jourdain*. These approaches as discussed in [BP92] are basically equivalent and lead to the same mathematical model. The choice of the coordinates is however crucial in deciding the obtained model.

Lagrange Equations of the Second Kind. Using the minimal coordinates \mathbf{p} as defined by (2.31), the equations of motion for the MBS can be written as:

$$\mathbf{M}(\mathbf{q}(\mathbf{p})) \ddot{\mathbf{q}}(\mathbf{p}) = \mathbf{f}(\mathbf{q}(\mathbf{p}), \dot{\mathbf{q}}(\mathbf{p}), t). \quad (2.38)$$

This form is called the *minimal form*, *Lagrange Equations of the second kind* or most popularly as the *state-space form*. It is a linearly implicit second order system of ordinary differential equations (ODEs).

2.2.2 Index and Minimax Characterization

In this section, we first compute the index of the equations of motion of a mechanical system from (2.37) and in turn comment on the uniqueness of the solution. Secondly, we reformulate the full rank condition (2.30) in terms of a singular value minimax condition from [Sim13].

2.2.2.1 Index Determination

We rewrite the system (2.37) in the first order form

$$\dot{\mathbf{q}} = \mathbf{v}, \quad (2.39a)$$

$$\mathbf{M}(\mathbf{q})\dot{\mathbf{v}} = \mathbf{f}(\mathbf{q}, \mathbf{v}, t) - \mathbf{G}(\mathbf{q})^T \boldsymbol{\lambda}, \quad (2.39b)$$

$$\mathbf{0} = \mathbf{g}(\mathbf{q}) \quad (2.39c)$$

with $\mathbf{v} \in \mathbb{R}^{n_q}$ introduced as velocity variables.

Differentiating the constraints (2.39c) with respect to time, we obtain the *velocity level* constraints

$$\mathbf{0} = \frac{d}{dt} \mathbf{g}(\mathbf{q}) = \mathbf{G}(\mathbf{q})\dot{\mathbf{q}} = \mathbf{G}(\mathbf{q})\mathbf{v}. \quad (2.40)$$

Differentiating a second time gives us the *acceleration level* constraints

$$\mathbf{0} = \frac{d^2}{dt^2} \mathbf{g}(\mathbf{q}) = \mathbf{G}(\mathbf{q})\dot{\mathbf{v}} + \boldsymbol{\kappa}(\mathbf{q}, \mathbf{v}), \quad \boldsymbol{\kappa}(\mathbf{q}, \mathbf{v}) := \frac{\partial \mathbf{G}(\mathbf{q})}{\partial \mathbf{q}}(\mathbf{v}, \mathbf{v}). \quad (2.41)$$

The combination of the dynamic equation (2.39b) and the acceleration constraints (2.41) leads to the linear system

$$\begin{pmatrix} \mathbf{M}(\mathbf{q}) & \mathbf{G}(\mathbf{q})^T \\ \mathbf{G}(\mathbf{q}) & \mathbf{0} \end{pmatrix} \begin{pmatrix} \dot{\mathbf{v}} \\ \boldsymbol{\lambda} \end{pmatrix} = \begin{pmatrix} \mathbf{f}(\mathbf{q}, \mathbf{v}, t) \\ -\boldsymbol{\kappa}(\mathbf{q}, \mathbf{v}) \end{pmatrix}, \quad (2.42)$$

with the matrix on the left hand side having a *saddle point structure* [BGL05], i.e. the matrix is indefinite with both positive and negative eigenvalues. It must be noted that the solution of the above system (2.42) in combination with (2.39a) does not necessarily satisfy the original system (2.37), since the differentiation of the constraints leads to a loss of integration constraints. In order to be consistent, the initial values $(\mathbf{q}_0, \mathbf{v}_0)$ must satisfy the position and velocity invariants

$$\mathbf{0} = \mathbf{g}(\mathbf{q}_0), \quad \mathbf{0} = \mathbf{G}(\mathbf{q}_0)\mathbf{v}_0. \quad (2.43)$$

These invariants are however not preserved under discretization and the numerical solution may turn unstable, leading to the so-called *drift-off phenomenon*. This has been studied in the last decades and approaches such as the Baumgarte stabilization [Bau72], etc have become popular to overcome the challenges.

In order to solve the system (2.42), it is required that

$$\begin{pmatrix} \mathbf{M}(\mathbf{q}) & \mathbf{G}(\mathbf{q})^T \\ \mathbf{G}(\mathbf{q}) & \mathbf{0} \end{pmatrix} \text{ is invertible} \quad (2.44)$$

at least in a neighborhood of the solution. A necessary but not sufficient condition for (2.44) is the full rank of the constraint Jacobian \mathbf{G} as mentioned in (2.30). If additionally the mass matrix \mathbf{M} is symmetric positive definite, then (2.44) automatically holds true.

Assuming the condition (2.44) is fulfilled and a symmetric positive definite mass matrix \mathbf{M} , we can solve the linear system (2.42) for the acceleration $\dot{\mathbf{v}}$ and the Lagrange multiplier $\boldsymbol{\lambda}$, leading to the explicit expressions

$$\dot{\mathbf{v}} = \mathbf{M}(\mathbf{q})^{-1} (\mathbf{f}(\mathbf{q}, \mathbf{v}, t) - \mathbf{G}(\mathbf{q})^T \boldsymbol{\lambda}), \quad (2.45a)$$

$$\boldsymbol{\lambda} = (\mathbf{G}(\mathbf{q})\mathbf{M}(\mathbf{q})^{-1}\mathbf{G}(\mathbf{q})^T)^{-1} (\mathbf{G}(\mathbf{q})\mathbf{M}(\mathbf{q})^{-1}\mathbf{f}(\mathbf{q}, \mathbf{v}, t) + \boldsymbol{\kappa}(\mathbf{q}, \mathbf{v})). \quad (2.45b)$$

Inserting $\boldsymbol{\lambda}$ from (2.45b) into (2.45a) leads to an ODE for unknown velocities \mathbf{v} . The unique solution (\mathbf{q}, \mathbf{v}) is guaranteed for (2.39a) and (2.45a) under the assumption of Lipschitz continuity of the corresponding right hand side. In turn, the Lagrange multiplier $\boldsymbol{\lambda}$ is also uniquely determined.

Following the two differentiation steps to obtain the explicit expressions (2.45), a third differentiation step on (2.45b) yields an ODE, leading to a differentiation index-3.

2.2.2.2 Minimax Characterization of Constraints

The singular value decomposition of the constraint Jacobian \mathbf{G} gives us

$$\mathbf{U}^T \mathbf{G}(\mathbf{q}) \mathbf{V} = \text{diag}(\sigma_1, \dots, \sigma_{n_\lambda}) \in \mathbb{R}^{n_\lambda \times n_q}$$

with orthogonal matrices $\mathbf{U} \in \mathbb{R}^{n_\lambda \times n_\lambda}$ and $\mathbf{V} \in \mathbb{R}^{n_q \times n_q}$. The singular values are ordered as

$$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_{n_\lambda} \geq 0$$

and for the full rank of \mathbf{G} , we require $\sigma_{\min} := \sigma_{n_\lambda} > 0$.

We can reformulate this criterion for any vector $\mathbf{0} \neq \boldsymbol{\lambda} \in \mathbb{R}^{n_\lambda}$ and $\boldsymbol{\mu} := \mathbf{U}^T \boldsymbol{\lambda}$ such that

$$\frac{\boldsymbol{\lambda}^T \mathbf{G} \mathbf{G}^T \boldsymbol{\lambda}}{\boldsymbol{\lambda}^T \boldsymbol{\lambda}} = \frac{\boldsymbol{\mu}^T \text{diag}(\sigma_1^2, \dots, \sigma_{\min}^2) \boldsymbol{\mu}}{\boldsymbol{\mu}^T \boldsymbol{\mu}} \geq \sigma_{\min}^2.$$

For $\boldsymbol{\mu} = (0, \dots, 0, 1)^T \in \mathbb{R}^{n_\lambda}$, this inequality is sharp and we conclude

$$\sigma_{\min}^2 = \min_{\boldsymbol{\lambda}} \frac{\boldsymbol{\lambda}^T \mathbf{G} \mathbf{G}^T \boldsymbol{\lambda}}{\boldsymbol{\lambda}^T \boldsymbol{\lambda}} \iff \sigma_{\min} = \min_{\boldsymbol{\lambda}} \frac{\|\mathbf{G}^T \boldsymbol{\lambda}\|_2}{\|\boldsymbol{\lambda}\|_2}.$$

Using additionally the definition of the operator norm gives us

$$\|\mathbf{G}^T \boldsymbol{\lambda}\|_2 = \max_{\mathbf{v}} \frac{\|\mathbf{v}^T \mathbf{G}^T \boldsymbol{\lambda}\|_2}{\|\mathbf{v}\|_2} = \max_{\mathbf{v}} \frac{\boldsymbol{\lambda}^T \mathbf{G} \mathbf{v}}{\|\mathbf{v}\|_2}$$

since $\|\mathbf{v}^T \mathbf{G}^T \boldsymbol{\lambda}\|_2 = |\boldsymbol{\lambda}^T \mathbf{G} \mathbf{v}|$. Overall, we obtain the *minimax characterization* [Sim13]

$$\sigma_{\min}(\mathbf{G}) = \min_{\boldsymbol{\lambda}} \max_{\mathbf{v}} \frac{\boldsymbol{\lambda}^T \mathbf{G} \mathbf{v}}{\|\mathbf{v}\|_2 \|\boldsymbol{\lambda}\|_2} > 0. \quad (2.46)$$

Important is to notice that this minimax condition for the regularity of the rigid body constraints resembles the well known *inf-sup condition* or *LBB condition* [Bre90] for constraints on elastic bodies. It shows that both these conditions require the constraints to be independent or full rank and hence well defined, once in the finite dimensional rigid body case and once in the infinite dimensional setting for the elastic bodies.

2.2.3 Stiff System

In this section, we present a class of mechanical systems that are considered to be stiff. Stiff systems are ill-conditioned and often undergo numerical issues.

A *stiff multibody system* is characterized as in [Lub93, Sim01] by a force vector that splits into a non-stiff part \mathbf{f}_n and a stiff part \mathbf{f}_s such that

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} = \mathbf{f}_n(\mathbf{q}, \dot{\mathbf{q}}) - \underbrace{\frac{1}{\epsilon^2} \frac{\partial U(\mathbf{q})}{\partial \mathbf{q}}}_{=: \mathbf{f}_s}. \quad (2.47)$$

The non-stiff part \mathbf{f}_n is supposed to contain only small stiffness terms while the stiff part \mathbf{f}_s is written as the gradient of the potential U with prefactor $1/\epsilon^2$, $\epsilon \ll 1$. This is a *singularly perturbed system* with perturbation parameter ϵ .

Under mild convexity and smoothness assumptions on the stiff potential U , it is proved in [Lub93] that for the system (2.47) smooth motion \mathbf{q}^ϵ can be derived by the outer expansion

$$\mathbf{q}^\epsilon(t) = \mathbf{q}^0(t) + \epsilon^2 \mathbf{q}^1(t) + \epsilon^4 \mathbf{q}^2(t) + \dots + \epsilon^{2N} \mathbf{q}^N(t) + \mathcal{O}(\epsilon^{2N+2}), \quad (2.48)$$

such that the coefficient function \mathbf{q}^0 satisfies the index-3 DAE

$$\mathbf{M}(\mathbf{q}^0)\ddot{\mathbf{q}}^0 = \mathbf{f}_n(\mathbf{q}^0, \dot{\mathbf{q}}^0) - \mathbf{\Gamma}^T(\mathbf{q}^0)\boldsymbol{\lambda}^0, \quad (2.49a)$$

$$\mathbf{0} = \boldsymbol{\gamma}(\mathbf{q}^0), \quad (2.49b)$$

where (2.49b) is satisfied whenever $\mathbf{0} = \partial U(\boldsymbol{\gamma}(\mathbf{q}^0))/\partial \mathbf{q}$ and $\mathbf{\Gamma} = \partial \boldsymbol{\gamma}/\partial \mathbf{q}$. The next coefficient functions follow in a similar fashion and lead to a chain of further DAEs.

Note that we have used here the MBS equations in the state-space form. The theory is also valid for the Lagrange equations of the first kind.

Example 2.2. Stiff Pendulum. *Let us consider a mathematical stiff spring pendulum, consisting of a massless spring with Hooke's constant $1/\epsilon^2$, with a planar motion $\mathbf{q} = (q_1, q_2)^T \in \mathbb{R}^2$ having the following ODE form as in (2.47)*

$$\underbrace{\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}}_{\mathbf{M}(\mathbf{q})} \underbrace{\begin{pmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{pmatrix}}_{\ddot{\mathbf{q}}} = \underbrace{\begin{pmatrix} 0 \\ -1 \end{pmatrix}}_{\mathbf{f}_n} - \underbrace{\frac{1}{\epsilon^2} \frac{\sqrt{q_1^2 + q_2^2} - 1}{\sqrt{q_1^2 + q_2^2}}}_{\mathbf{f}_s} \begin{pmatrix} q_1 \\ q_2 \end{pmatrix} \quad (2.50)$$

with a small parameter $\epsilon \ll 1$. It is assumed that we have unit mass, unit gravity and in the rest position, the spring has unit length. This is in particular a singularly perturbed problem.

Since U is convex and sufficiently smooth, we derive an index-3 formulation as above. Assuming that ϵ is small, we take that $\mathbf{q} := \mathbf{q}^\epsilon = \mathbf{q}^0$ and the constraints obtained are

$$\mathbf{0} = \frac{\sqrt{q_1^2 + q_2^2} - 1}{\sqrt{q_1^2 + q_2^2}} \begin{pmatrix} q_1 \\ q_2 \end{pmatrix},$$

similar to (2.49b). It must be noted that for a valid motion, $q_1, q_2 \neq 0$. Hence, we get two equivalent or redundant constraints as

$$0 = \frac{\sqrt{q_1^2 + q_2^2} - 1}{\sqrt{q_1^2 + q_2^2}} \quad \text{or} \quad 0 = q_1^2 + q_2^2 - 1.$$

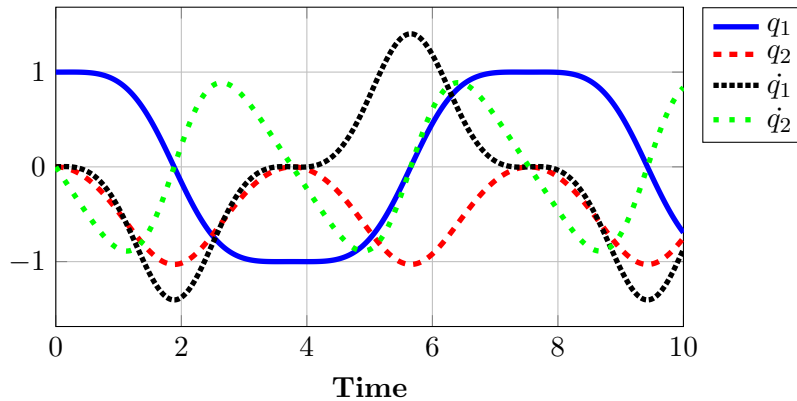


Figure 2.2: Smooth solution of the stiff pendulum

Taking one of them to fulfill the full rank requirements of Remark 2.1, we then get the index-3 DAE

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{pmatrix} = \begin{pmatrix} 0 \\ -1 \end{pmatrix} - \lambda \begin{pmatrix} 2q_1 \\ 2q_2 \end{pmatrix}, \quad (2.51)$$

$$0 = q_1^2 + q_2^2 - 1.$$

Starting with the initial values $q_1 = 1$, $q_2 = 0$, $\dot{q}_1 = 0$, $\dot{q}_2 = 0$, we solve both the ODE (2.50) and the index-3 DAE (2.51) formulations from time 0s to 10s using the BDF-2 solver from Section 2.1.3.1. The result obtained for both cases is a smooth solution as in Fig. 2.2. It is observed that for stiff systems, i.e. $\epsilon \ll 1$, the higher the stiffness or in other words the lower the value of ϵ , the more Newton iterations and hence computational time are required to solve the ODE system. On the other hand, the changing stiffness had no effect on the computations for the DAE case. At best the number of Newton iterations for the ODE case were the same as the DAE case. It supports the claim from [Lub93] that stiff ODE systems are at least as difficult to solve as the corresponding DAE form, if not more.

Remark 2.2. It is shown in [Lub93] that when solving stiff mechanical systems, the modified Newton iterations employed for the solution of the nonlinear problem suffers a severe time step size restriction, which is a function of the perturbation parameter ϵ , except when the potential U is quadratic.

We encounter the problem of the time step size restriction as mentioned in Remark 2.2, when dealing with stiff systems that arise in Section 4.4.2 as a result of the application of the bending strip method, which is a penalty approach used to join multipatch shell structures.

2.2.4 Scaling of Equations of Motion

In this section, we discuss a scaling technique as proposed in [Bau11], which is used to improve the overall conditioning of a constrained mechanical system. This increases the numerical stability of the system. Also refer to [BEB07, BBC07] for scaling techniques for index-3 mechanical systems.

2.2.4.1 Scaling Technique

In order to represent a constrained mechanical system, we take the following index-3 representation

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{D}\dot{\mathbf{q}} + \mathbf{K}\mathbf{q} + \mathbf{G}^T \boldsymbol{\lambda} = \mathbf{h}, \quad (2.52a)$$

$$\mathbf{g}(\mathbf{q}, t) = \mathbf{0}. \quad (2.52b)$$

It has the structure of (2.37), where the force $\mathbf{f} = \mathbf{h} - \mathbf{K}\mathbf{q} - \mathbf{D}\dot{\mathbf{q}}$ is expressed in terms of \mathbf{D} , \mathbf{K} , \mathbf{h} , which are the damping and stiffness matrices and remaining forces respectively. The system constraints $\mathbf{g}(\mathbf{q}, t) = \mathbf{0}$ are normalized to unity.

In the stationary case, the time dependent terms vanish and we obtain

$$\mathbf{K}\mathbf{q} + \mathbf{G}^T \boldsymbol{\lambda} = \mathbf{h}, \quad (2.53a)$$

$$\mathbf{g}(\mathbf{q}) = \mathbf{0}. \quad (2.53b)$$

For dynamical systems, in anticipation of the problem of the time step size affecting the conditioning of the system Jacobian, we replace the time t in (2.52) with $t = h\tau$, where h is the time step size and τ the normalized time [Bau11]. Furthermore, multiplying (2.52a) with h^2 , we get

$$\mathbf{M}\mathbf{q}'' + h\mathbf{D}\mathbf{q}' + h^2\mathbf{K}\mathbf{q} + h^2\mathbf{G}^T \boldsymbol{\lambda} = h^2\mathbf{h}, \quad (2.54a)$$

$$\mathbf{g}(\mathbf{q}, \tau) = \mathbf{0}, \quad (2.54b)$$

where $(\cdot)'$ is the derivative with respect to the normalized time τ .

Two obvious numerical issues might arise from the equation system for both stationary and transient settings. Firstly, large values of \mathbf{K} in stationary case and \mathbf{M} , \mathbf{D} , \mathbf{K} in transient case might make the constraints invisible for the numerics due to their relatively low values. Secondly, the system unknowns \mathbf{q} and $\boldsymbol{\lambda}$ might have different orders of magnitude since \mathbf{q} represents the system displacements and $\boldsymbol{\lambda}$ the constraint forces, which differ usually by many orders of magnitude.

To correct these issues, [Bau11] suggests the following. In the stationary case, multiply the constraint equations $\mathbf{g}(\mathbf{q}) = \mathbf{0}$ with a scaling factor $s = \|\mathbf{K}\|_\infty$ and take $\boldsymbol{\lambda} := s\hat{\boldsymbol{\lambda}}$, leading to

$$\mathbf{K}\mathbf{q} + s\mathbf{G}^T \hat{\boldsymbol{\lambda}} = \mathbf{h}, \quad (2.55a)$$

$$s\mathbf{g}(\mathbf{q}) = \mathbf{0}. \quad (2.55b)$$

In the transient case, multiply the constraint equations $\mathbf{g}(\mathbf{q}, \tau) = \mathbf{0}$ with a scaling factor $s = m_r + d_r h + k_r h^2$, such that

$$m_r = \|\mathbf{M}\|_\infty, \quad d_r = \|\mathbf{D}\|_\infty, \quad k_r = \|\mathbf{K}\|_\infty.$$

Then, take

$$h^2 \boldsymbol{\lambda} = s\hat{\boldsymbol{\lambda}}, \quad (2.56)$$

leading to the equations

$$\mathbf{M}\mathbf{q}'' + h\mathbf{D}\mathbf{q}' + h^2\mathbf{K}\mathbf{q} + s\mathbf{G}^T \hat{\boldsymbol{\lambda}} = h^2\mathbf{h}, \quad (2.57a)$$

$$s\mathbf{g}(\mathbf{q}, \tau) = \mathbf{0}, \quad (2.57b)$$

Hence, the values in the constraint equations (2.55b), (2.57b), due to their multiplication with an appropriate scaling factor s , have the same order as of the values in the equations of motion (2.55a), (2.57a). Also, the new unknown $\hat{\boldsymbol{\lambda}}$ has a magnitude of similar order as that of the displacement variable \mathbf{q} .

We rewrite the transient equation system (2.57) in a compact form for analysis

$$\mathbf{M}\mathbf{q}'' + s\mathbf{G}^T\hat{\boldsymbol{\lambda}} = h^2\mathbf{f}, \quad (2.58a)$$

$$s\mathbf{g}(\mathbf{q}, \tau) = \mathbf{0}, \quad (2.58b)$$

such that $\mathbf{f} = \mathbf{h} - \mathbf{K}\mathbf{q} - \mathbf{D}\mathbf{q}'/h$.

2.2.4.2 Scaling Analysis

In this section, we analyze the scaled equations of motion (2.55), (2.58) from the previous section, first for the stationary and then for the transient case. Without loss of generality, we take the normalized time $\tau = 1$ for our analysis.

Stationary case Applying the Newton method to the equations of the stationary case (2.55), we get at each iteration step j the following linearized equation system

$$\mathbf{J}_j\Delta\mathbf{x}_j = -\mathbf{c}_j, \quad (2.59)$$

where $\Delta\mathbf{x}_j = (\Delta\mathbf{q}^T, \Delta\hat{\boldsymbol{\lambda}}^T)_j^T$,

$$\mathbf{J}_j = \begin{pmatrix} \mathbf{K} & s\mathbf{G}^T \\ s\mathbf{G} & \mathbf{0} \end{pmatrix}_j, \quad \mathbf{c}_j = \begin{pmatrix} \mathbf{K}\mathbf{q} + s\mathbf{G}^T\hat{\boldsymbol{\lambda}} - \mathbf{h} \\ s\mathbf{g} \end{pmatrix}_j. \quad (2.60)$$

Due to the multiplication of the constraint Jacobian \mathbf{G}_j with the scaling factor s , the values of the components of the Jacobian matrix \mathbf{J}_j lie in a similar range and hence, the overall condition of the Jacobian matrix \mathbf{J}_j is improved.

Transient case We analyze the scaled equations of motion (2.58) in the transient case. We discretize the equations with respect to the normalized time τ using the implicit Euler scheme and obtain in each time window (t_i, t_f)

$$\mathbf{M}_f(\mathbf{q}_f - \mathbf{q}_i - \mathbf{v}_i) + s\mathbf{G}_f^T\hat{\boldsymbol{\lambda}}_f - h^2\mathbf{f}_f = \mathbf{0}, \quad (2.61a)$$

$$s\mathbf{g}_f = \mathbf{0}, \quad (2.61b)$$

where the subscripts i, f represent the quantities evaluated at the start and end times of the time window respectively. We solve this nonlinear system for $\mathbf{q}_f, \hat{\boldsymbol{\lambda}}_f$ and apply the Newton method to get in each iteration step j the linearized problem

$$\mathbf{J}_j\Delta\mathbf{x}_j = -\mathbf{c}_j, \quad (2.62)$$

where $\Delta\mathbf{x}_j = (\Delta\mathbf{q}_f^T, \Delta\hat{\boldsymbol{\lambda}}_f^T)_j^T$ and

$$\mathbf{J}_j = \begin{pmatrix} \mathbf{M}_f - h^2\nabla\mathbf{f}_f & s\mathbf{G}_f^T \\ s\mathbf{G}_f & \mathbf{0} \end{pmatrix}_j = \begin{pmatrix} \mathcal{O}(h^0) & \mathcal{O}(h^0) \\ \mathcal{O}(h^0) & \mathbf{0} \end{pmatrix}, \quad (2.63)$$

$$\mathbf{c}_j = \begin{pmatrix} \mathbf{M}_f(\mathbf{q}_f - \mathbf{q}_i - \mathbf{v}_i) + s\mathbf{G}_f^T \hat{\boldsymbol{\lambda}}_f - h^2 \mathbf{f}_f \\ s\mathbf{g}_f \end{pmatrix}_j = \begin{pmatrix} \mathcal{O}(h^0) \\ \mathcal{O}(h^0) \end{pmatrix}. \quad (2.64)$$

Following

$$\lim_{h \rightarrow 0} (\mathbf{J}_j \Delta \mathbf{x}_j) = \lim_{h \rightarrow 0} (\mathbf{J}_j) \lim_{h \rightarrow 0} (\Delta \mathbf{x}_j) = - \lim_{h \rightarrow 0} (\mathbf{c}_j), \quad (2.65)$$

we get $\lim_{h \rightarrow 0} (\Delta \mathbf{x}_j) = \mathcal{O}(h^0)$. This implies that the update of the system unknowns is independent of the time step size. Also the condition number of the Jacobian, $\kappa(\mathbf{J}_j) = \|\mathbf{J}_j\|_\infty \|\mathbf{J}_j^{-1}\|_\infty = \mathcal{O}(h^0)$.

Remark 2.3. Due to the introduction of the new Lagrange multiplier $\hat{\boldsymbol{\lambda}}_{f,j}$ from (2.56), we get in the Jacobian \mathbf{J}_j the term $s\mathbf{G}_{f,j}^T$, instead of $h^2\mathbf{G}_{f,j}^T$, which removes the time step size dependence of the condition number of the Jacobian and hence of the updates of the system unknowns. This independence is also observed in case of other time integration schemes like the BDF-2, generalized- α , etc.

In the case of the unscaled equations, application of the implicit Euler scheme followed by the Newton method leads to $\mathbf{J}_j \Delta \mathbf{x}_j = -\mathbf{c}_j$ such that

$$\mathbf{J}_j = \begin{pmatrix} \frac{1}{h^2} \mathbf{M}_f - \nabla \mathbf{f}_f & \mathbf{G}_f^T \\ \mathbf{G}_f & \mathbf{0} \end{pmatrix}_j = \begin{pmatrix} \mathcal{O}(h^{-2}) & \mathcal{O}(h^0) \\ \mathcal{O}(h^0) & \mathbf{0} \end{pmatrix},$$

$$\mathbf{c}_j = \begin{pmatrix} \frac{1}{h^2} \mathbf{M}_f(\mathbf{q}_f - \mathbf{q}_i - h\mathbf{v}_i) + \mathbf{G}_f^T \boldsymbol{\lambda}_f - \mathbf{f}_f \\ \mathbf{g}_f \end{pmatrix}_j = \begin{pmatrix} \mathcal{O}(h^{-2}) \\ \mathcal{O}(h^0) \end{pmatrix}.$$

The inverse Jacobian is

$$\mathbf{J}_j^{-1} = \begin{pmatrix} \mathcal{O}(h^2) & \mathcal{O}(h^0) \\ \mathcal{O}(h^0) & \mathcal{O}(h^{-2}) \end{pmatrix}.$$

It then follows that the condition number of the Jacobian matrix, $\kappa(\mathbf{J}_j)$, exhibits a strong dependency on the time step size, $\kappa(\mathbf{J}_j) = \mathcal{O}(h^{-4})$, and

$$\Delta \mathbf{q}_f = \mathcal{O}(h^0), \quad \Delta \boldsymbol{\lambda}_f = \mathcal{O}(h^{-2}).$$

Such step size dependence is particularly harmful for small step sizes h , where the system Jacobian is ill-conditioned and the Lagrange multipliers also badly affected.

2.2.5 Flexible Multibody System

A *flexible multibody system* is an extension of the standard multibody system with an inclusion of elastic bodies additional to the rigid bodies. This is done in order to simulate systems with greater details utilizing the tremendous increase in the computational capabilities of modern computers.

Elastic members in a multibody system are typically modeled by means of a preprocessing step where the deformation is split into rigid body motion, i.e., translation and rotation, and certain elastic mode shapes. In general, the result is a coupled system of differential-algebraic equations that can be written as

$$\mathbf{M}(\mathbf{q}, \mathbf{d}) \begin{pmatrix} \ddot{\mathbf{q}} \\ \ddot{\mathbf{d}} \end{pmatrix} = \mathbf{f}(\mathbf{q}, \mathbf{d}, \dot{\mathbf{q}}, \dot{\mathbf{d}}, t) - \mathbf{G}(\mathbf{q}, \mathbf{d})^T \boldsymbol{\lambda}, \quad (2.66a)$$

$$\mathbf{0} = \mathbf{g}(\mathbf{q}, \mathbf{d}). \quad (2.66b)$$

Here, $\mathbf{q}(t)$ are the variables that describe the rigid body motions of both the rigid and the elastic bodies, while $\mathbf{d}(t)$ stands for the elastic displacements of the latter. The mass matrix \mathbf{M} might be state-dependent and the vector \mathbf{f} contains all force relations between the bodies, including generalized Coriolis and centrifugal terms. Holonomic constraints between the bodies lead to additional algebraic equations (2.66b), with constraint Jacobian $\mathbf{G} = \partial\mathbf{g}/\partial(\mathbf{q}, \mathbf{d})$ and corresponding Lagrange multipliers $\boldsymbol{\lambda}(t)$, see e.g. [Sim00].

Assuming a single elastic body for sake of simplicity, the corresponding displacement field $\mathbf{u}(\mathbf{x}, t)$ is approximated by

$$\mathbf{u}_h(\mathbf{x}, t) = \sum_{i=1}^{n_{\text{DOF}}} \boldsymbol{\Psi}_i(\mathbf{x}) d_i(t) \quad (2.67)$$

where $\boldsymbol{\Psi}_i$ are the chosen spatial mode shapes and d_i the corresponding displacement coefficients that form the vector $\mathbf{d} = (d_i)_{i=1}^{n_{\text{DOF}}}$. We refer to the literature [CK06] for the adept choice of the mode shapes $\boldsymbol{\Psi}_i$. An important point in this context is the role of the floating reference frame that splits the kinematics of the flexible body into rigid motion plus elastic offset. This results in a partitioned mass matrix \mathbf{M} that contains extra off-diagonal coupling blocks.

The methodology outlined so far is state-of-the-art in many codes, and in this way it is even possible to include large-scale finite element structures in a multibody simulation by means of a few carefully chosen elastic modes. The approach, however, has also its limitations, as discussed in Chapter 1. An exception are beam models where nonlinear or the so-called geometric stiffening effects have become quite standard and corresponding features are directly included in the multibody codes [SW99, Wal91].

Due to the increasing demand for refined elastic models that include specific nonlinearities and in view of the above issues, tailored spatial discretizations that are generated within a multibody formalism become more and more attractive. Recently, nonlinear shells have been studied in combination with so-called absolute nodal coordinates [BS09, Sha98] that do not rely on the method of floating reference frames, and in this contribution, we also skip the concept of a floating reference frame and take up the latest development in finite element methods, the approach of isogeometric analysis [CHB09]. In this way, we aim at a direct incorporation of the discrete nonlinear elastic structure without an a-priori model reduction. If the elastic degrees of freedom \mathbf{d} include the rigid body motion, the coupling in the partitioned equations of motion (2.66a) is weak, and the mass matrix \mathbf{M} possesses no coupling blocks, i.e., it becomes block-diagonal. The equations of motion simplify to

$$\mathbf{M}_r(\mathbf{q})\ddot{\mathbf{q}} = \mathbf{f}_r(\mathbf{q}, \mathbf{d}, \dot{\mathbf{q}}, \dot{\mathbf{d}}, t) - \frac{\partial\mathbf{g}}{\partial\mathbf{q}}(\mathbf{q}, \mathbf{d})^T \boldsymbol{\lambda}, \quad (2.68a)$$

$$\mathbf{M}_h\ddot{\mathbf{d}} = \mathbf{f}_h(\mathbf{q}, \mathbf{d}, \dot{\mathbf{q}}, \dot{\mathbf{d}}, t) - \frac{\partial\mathbf{g}}{\partial\mathbf{d}}(\mathbf{q}, \mathbf{d})^T \boldsymbol{\lambda}, \quad (2.68b)$$

$$\mathbf{0} = \mathbf{g}(\mathbf{q}, \mathbf{d}). \quad (2.68c)$$

Moreover, the force vector of the elastic body on the right hand side of the dynamic equations splits into

$$\mathbf{f}_h(\mathbf{q}, \mathbf{d}, \dot{\mathbf{q}}, \dot{\mathbf{d}}, t) = -\mathbf{K}(\mathbf{d})\mathbf{d} - \mathbf{D}\dot{\mathbf{d}} + \mathbf{b}(\mathbf{d}, \dot{\mathbf{d}}, \mathbf{q}, \dot{\mathbf{q}}, t) \quad (2.69)$$

where $\mathbf{K}(\mathbf{d})$ is the nonlinear stiffness matrix, \mathbf{D} the damping matrix, and \mathbf{b} the load vector. These equations of motion can either be directly solved in a dynamic simulation or treated by means of a co-simulation approach where different time integration schemes are used for both subsystems. See Section 6.3 for details.

Chapter 3

Shell Theory

Shells are structures where one dimension is significantly smaller than the other two. They are thin walled structures arising from a natural optimization that reduces the dead loads and minimizes the material used. Thus, they are characterized by their curvature and small thickness. A *plate* is a special case of shells with no curvature. Examples of shells in nature are eggs, seashells, nutshells or parts of living organisms like human skulls, turtle shell, etc. Their first use by mankind as building structures was, as early as 126 AD, in form of the concrete dome of the Pantheon. Many of the world's most famous structures like the Taj Mahal in India, the Hagia Sophia in Greece, the St. Paul's Church in England, etc have shell domes since centuries and are still standing strong. Their versatility is evident by their use in different fields of sciences like civil, automotive, naval and aerospace engineering. Even in everyday life, we come across shells in the form of cans, bottles, balls, body of electrical appliances and many more. The benefits of using shells are their cost efficiency as they are hollow and save dead load of material. They are also thin and light weight structures, that can be handled or transported quite easily. They are visually pleasing and good for acoustics, as we see from various auditoriums, theaters, churches, etc which have shell roofs. Lastly, they are also computationally efficient, since dimension reduction can be done and huge savings in the degrees of freedom and computing times can therefore be observed.

When designed properly, shells carry the load exclusively in terms of membrane forces, which are parallel to the mid-surface or the centerline. The membrane state is desirable because it optimally makes use of the material, keeping at least approximately the stress distribution constant across the thickness. During the bending, the shells are much more flexible due to the material fibers in the thickness direction coming into play. The membrane forces strain the centerline of the shell, whereas the bending forces preserve the extension (or compression) of the centerline. In practice, usually both the membrane and the bending actions are involved. But a good design ensures that the membrane action dominates.

3.1 Choice of Kinematic Model

The invention of the modern plate theory dates back to 1850 and is accredited to Gustav Kirchhoff (1824–1887). He made the following assumptions for his model:

- thickness of the plate remains constant
- straight cross-sections remain straight after the deformation

- cross-sections perpendicular to the mid-surface remain perpendicular after deformation

This was extended to the theory of thin shells in 1888 by August E. H. Love (1863–1940). It is called the *classical shell theory*. Since it is based on the assumptions of Kirchhoff, the model became popular as the *Kirchhoff-Love model*. Sometimes it is also referred to as the *Koiter shell* in the literature, named after Warner T. Koiter, who proposed the nonlinear theory of thin shells [Koi66] again based on the same assumptions.

This theory is however too strict for thicker plates or shells, where it is also important to consider the transverse shear deformations, which can be understood as the sliding of the surfaces parallel to the centerline over each other. Such effects were first studied almost a century after the advent of the plate theory, by Eric Reissner in 1945 and Raymond Mindlin in 1951 for their plate models. The idea was to have the rotations independent of the centerline deformation. In other words, the assumption from Kirchhoff of the cross-sections remaining perpendicular was dropped. It was later extended to shells, which are famously known as the *first order shear deformable shells* or the *Reissner-Mindlin shells*.

We present the basic idea of the two approaches mathematically by means of a direct approach. This means that the models are directly implemented as two dimensional surfaces with the proper kinematic assumptions and are not derived from a three dimensional theory from the continuum mechanics. For other approaches to derive the various shell theories, we refer to the work of Bischoff et. al [BWBR04].

The kinematic description of a shell is given by a differentiable manifold

$$\mathcal{B} = \{(\mathbf{c}, \mathbf{d}) : \omega \times \mathbb{R} \rightarrow \mathbb{R}^3 \times \mathcal{S}^2, \omega \subset \mathbb{R}^2\}, \quad (3.1)$$

where \mathcal{S}^2 is defined as a unit sphere

$$\mathcal{S}^2 = \{\mathbf{d} \in \mathbb{R}^3 : \|\mathbf{d}\| = 1\}. \quad (3.2)$$

The parameter space $\omega \subset \mathbb{R}^2$ is an open set, representing the mid-surface of the shell. The mapping $\mathbf{c} : \omega \times \mathbb{R} \rightarrow \mathbb{R}^3$ characterizes the *shell centerline* from a 2D parametric space at a particular time $t \in \mathbb{R}$ to 3D physical space. $\mathbf{d} : \omega \times \mathbb{R} \rightarrow \mathcal{S}^2$ represents a unit director at each point of the centerline at time t pointing in the thickness direction. The parametric space is defined in terms of the *curvilinear* coordinates ϑ_i , $i = 1, 2, 3$. The deformed state $S \subset \mathbb{R}^3$ of such a shell is described by a mapping Θ for the *deformation*, see Fig. 3.1, such that

$$S := \{\Theta \in \mathbb{R}^3 : \Theta(\vartheta_1, \vartheta_2, \vartheta_3, t) = \mathbf{c}(\vartheta_1, \vartheta_2, t) + \vartheta_3 \mathbf{d}(\vartheta_1, \vartheta_2, t) \\ \text{with } (\mathbf{c}, \mathbf{d}) \in \mathcal{B} \text{ and } \vartheta_3 \in [-\epsilon, \epsilon]\}, \quad (3.3)$$

where the shell has a constant thickness of 2ϵ . The centerline can be further written in terms of the displacement field $\mathbf{u} \in \mathbb{R}^3$

$$\mathbf{c}(\vartheta_1, \vartheta_2, t) = \mathbf{c}_0(\vartheta_1, \vartheta_2) + \mathbf{u}(\vartheta_1, \vartheta_2, t), \quad (3.4)$$

where \mathbf{c}_0 represents the undeformed shell centerline.

Reissner-Mindlin Shells The kinematics introduced above assume that the thickness remains constant and the cross-sections remain straight. It correlates to the first order shear deformable theory, where the director has no particular restrictions except the length and

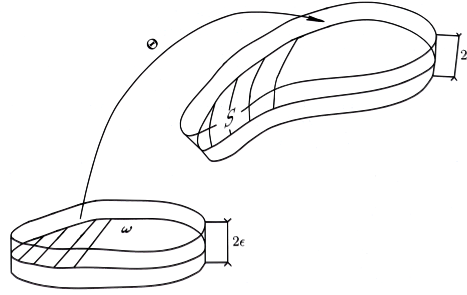


Figure 3.1: Mapping $\Theta : \omega \times [-\epsilon, \epsilon] \rightarrow S$ from a parametric to physical space [Cia05]

is independent of the centerline deformations. We obtain here the 5-parameter model with three displacement and two rotation unknowns as free variables per node.

The introduction of a free director field leads to the curvature of the centerline being determined by the first derivatives of these directors. This model can therefore work with C^0 -continuous, piecewise linear approximations without inducing any regularity issues, contrary to the Kirchhoff-Love models. This reduced regularity requirement is actually the main cause of the early shift from the Kirchhoff-Love shell to the Reissner-Mindlin models after the arrival of the finite element methods.

On one hand, taking into account the transverse shear deformations reduces the error with respect to the three dimensional theory in case of thick shells, but on the other hand, these models undergo a *locking* in case of thin shells leading to errors in the computation of transverse shear forces. More details about this locking phenomenon can be found in Bischoff et al. [BWBR04].

Kirchhoff-Love Shells To restrict the kinematic description (3.1) to Kirchhoff assumptions, one assumes that the shell director \mathbf{d} is always normal to the centerline, i.e. $\mathbf{d} = \mathbf{n}$, where \mathbf{n} is the normal at the centerline. This gives a 3-parameter model with three displacement unknowns as free variables and a fully dependent rotation.

The Kirchhoff-Love shells are known to bring along the difficulty of having to ensure global C^1 -continuity of the discrete surface. This is because the bending strains depend on the curvature of the centerline, which are derived from the second derivatives of the centerline, see Remark 3.8. The global C^1 -continuity is on the other hand easy to achieve with the advent of the new technique of isogeometric analysis, where arbitrary smoothness of the basis functions can be achieved by the use of splines as basis functions. Also, the model is free from all kinds of locking phenomenon.

Considering our requirement to study thin shells in the context of the multibody framework, we choose the Kirchhoff-Love shell theory for our modeling process. It also provides the benefits of being locking free and requiring lesser number of degrees of freedom. Any further mention of shells in this thesis will henceforth refer to the Kirchhoff-Love shells.

3.2 Kirchhoff-Love Model

We next consider the so-called Kirchhoff-Love shells in more detail. The theory presented here is applicable for large deformation or nonlinear setting.

3.2.1 Kinematics, Material Law and Energy Expressions

Starting with the mapping from (3.3), we make a restriction on the choice of the director field \mathbf{d} and choose it to be normal to the centerline at each point \mathbf{n} , giving us the mapping

$$\Theta(\vartheta_1, \vartheta_2, \vartheta_3, t) = \mathbf{c}(\vartheta_1, \vartheta_2, t) + \vartheta_3 \mathbf{n}(\vartheta_1, \vartheta_2, t), \quad \text{where } \mathbf{n} = \frac{\partial_1 \mathbf{c} \times \partial_2 \mathbf{c}}{|\partial_1 \mathbf{c} \times \partial_2 \mathbf{c}|}. \quad (3.5)$$

Besides constant thickness and straight cross-sections, the Kirchhoff-Love assumptions require that sections perpendicular to the centerline remain perpendicular.

Remark 3.1. *We refer to the book of Ciarlet [Cia05] for the details of the mathematical model and remark that for a well-defined 3D elasticity problem, the map Θ has to be an injective immersion that is also a C^2 -diffeomorphism. If the shell centerline \mathbf{c} in (3.5) is such that $\mathbf{c} \in C^3(\bar{\omega})$ is an injective immersion, then there exists an $\epsilon > 0$, such that the above stated requirement on the map Θ is fulfilled.*

With respect to the displacement field \mathbf{u} , a prescribed motion on a boundary Γ_0 represents a *Dirichlet boundary condition*

$$\mathbf{u}(\vartheta_1, \vartheta_2, t) = \mathbf{u}_0(\vartheta_1, \vartheta_2, t) \quad \text{on } \Gamma_0. \quad (3.6)$$

The elasticity problem is to find the deformation of the centerline \mathbf{c} in terms of the physical components of the displacement field \mathbf{u} or in terms of the covariant components $\tilde{\mathbf{u}} = (u_1, u_2, u_3)$, which are connected by

$$\mathbf{c} = \mathbf{c}_0 + \mathbf{u} = \mathbf{c}_0 + u_1 \mathbf{a}_0^1 + u_2 \mathbf{a}_0^2 + u_3 \mathbf{a}_0^3 = \mathbf{c}_0 + \mathbf{F}_0^{-T} \tilde{\mathbf{u}}. \quad (3.7)$$

Here, \mathbf{a}_0^i are the contravariant basis vectors, i.e., the columns of the matrix \mathbf{F}_0^{-T} where \mathbf{F} is the first part of the *deformation gradient*

$$\nabla \Theta = \underbrace{(\partial_1 \mathbf{c}, \partial_2 \mathbf{c}, \mathbf{n})}_{=: \mathbf{F}} + \vartheta_3 \underbrace{(\partial_1 \mathbf{n}, \partial_2 \mathbf{n}, \mathbf{0})}_{=: \tilde{\mathbf{F}}}. \quad (3.8)$$

The subscript 0 in (3.7) denotes the value of the respective variables in undeformed state.

Remark 3.2. *In practice, it is advantageous to choose the physical components of the displacement field \mathbf{u} over the covariant components $\tilde{\mathbf{u}}$ due to several reasons. Firstly, choosing $\tilde{\mathbf{u}}$ instead of \mathbf{u} requires higher smoothness of the undeformed shell centerline. Secondly, the block diagonal structure of the mass matrix is only retained on choosing \mathbf{u} , which makes the time integration of the shell structure less expensive. Lastly, it is easier to interpret and calculate the displacements in terms of \mathbf{u} , since they are simply the displacements of the control points.*

To measure local changes of length elements, we compute the strain. The deformation gradient $\nabla \Theta$ as a strain measure suffers from the drawback that it is not invariant under rigid motion, which is particularly disadvantageous for multibody applications, where the elastic bodies might undergo rigid motion. We consider instead a strain measure that fulfills the invariance property and take the covariant components of the Green-Lagrange strain tensor \mathcal{E} defined with respect to the undeformed contravariant basis $\mathbf{a}_0^i \otimes \mathbf{a}_0^j$, $i, j = 1, 2, 3$

$$\mathcal{E} = \frac{1}{2} \left(\nabla \Theta^T \nabla \Theta - \nabla \Theta_0^T \nabla \Theta_0 \right) = \mathbf{E}^m + \vartheta_3 \mathbf{E}^f + \mathcal{O}(\vartheta_3^2) \quad (3.9)$$

with membrane part \mathbf{E}^m and flexural or bending part \mathbf{E}^f , i.e.,

$$\mathbf{E}^m := \frac{1}{2} \left(\mathbf{F}^T \mathbf{F} - \mathbf{F}_0^T \mathbf{F}_0 \right) \quad \text{and} \quad \mathbf{E}^f := \mathbf{F}^T \tilde{\mathbf{F}} - \mathbf{F}_0^T \tilde{\mathbf{F}}_0. \quad (3.10)$$

The membrane part concerns itself with the extension or compression along the centerline, whereas the flexural part with the bending of the centerline, such that its length is preserved.

Remark 3.3. Taking the deformation gradient from (3.8) and putting into (3.9), we get

$$\mathbf{E}^f = \frac{1}{2} \left(\mathbf{F}^T \tilde{\mathbf{F}} + \tilde{\mathbf{F}}^T \mathbf{F} - \mathbf{F}_0^T \tilde{\mathbf{F}}_0 - \tilde{\mathbf{F}}_0^T \mathbf{F}_0 \right).$$

But, due to the second fundamental form of the curvilinear coordinates, see [BWBR04, Cia05], we get $\mathbf{F}^T \tilde{\mathbf{F}} = \tilde{\mathbf{F}}^T \mathbf{F}$ and hence the form as in (3.10).

Substituting the definition of \mathbf{F} and $\tilde{\mathbf{F}}$ from (3.8) into (3.10), we obtain the strain matrices

$$\mathbf{E}^m = \frac{1}{2} \begin{pmatrix} \partial_1 \mathbf{c} \cdot \partial_1 \mathbf{c} - \partial_1 \mathbf{c}_0 \cdot \partial_1 \mathbf{c}_0 & \partial_1 \mathbf{c} \cdot \partial_2 \mathbf{c} - \partial_1 \mathbf{c}_0 \cdot \partial_2 \mathbf{c}_0 & 0 \\ \partial_1 \mathbf{c} \cdot \partial_2 \mathbf{c} - \partial_1 \mathbf{c}_0 \cdot \partial_2 \mathbf{c}_0 & \partial_2 \mathbf{c} \cdot \partial_2 \mathbf{c} - \partial_2 \mathbf{c}_0 \cdot \partial_2 \mathbf{c}_0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad (3.11)$$

$$\text{and } \mathbf{E}^f = \begin{pmatrix} \partial_{11} \mathbf{c}_0 \cdot \mathbf{n}_0 - \partial_{11} \mathbf{c} \cdot \mathbf{n} & \partial_{12} \mathbf{c}_0 \cdot \mathbf{n}_0 - \partial_{12} \mathbf{c} \cdot \mathbf{n} & 0 \\ \partial_{12} \mathbf{c}_0 \cdot \mathbf{n}_0 - \partial_{12} \mathbf{c} \cdot \mathbf{n} & \partial_{22} \mathbf{c}_0 \cdot \mathbf{n}_0 - \partial_{22} \mathbf{c} \cdot \mathbf{n} & 0 \\ 0 & 0 & 0 \end{pmatrix}. \quad (3.12)$$

Remark 3.4. Although the plate like geometry of a shell structure prompts to a plane stress state, we obtain the strains in (3.11), (3.12) as in plane strain due to the Kirchhoff assumption of directors being perpendicular to centerline. The membrane transverse shear strain

$$\mathbf{E}_{3\alpha}^m = \mathbf{E}_{3\alpha}^m = \partial_\alpha \mathbf{c} \cdot \mathbf{n} - \partial_\alpha \mathbf{c}_0 \cdot \mathbf{n}_0 = 0 - 0 = 0, \quad \alpha = 1, 2,$$

since the director \mathbf{n} is a normal to $\partial_\alpha \mathbf{c}$ by construction as in (3.5). Moreover, since $\|\mathbf{n}\|_2 = 1$, the membrane normal strain

$$\mathbf{E}_{33}^m = \mathbf{n} \cdot \mathbf{n} - \mathbf{n}_0 \cdot \mathbf{n}_0 = 1 - 1 = 0.$$

The flexural transverse shear and normal strain $\mathbf{E}_{i3}^f = 0$, $i = 1, 2, 3$, due to the dot product with zero vector.

$$\mathbf{E}_{3\alpha}^f = \mathbf{n} \cdot \partial_\alpha \mathbf{n} - \mathbf{n}_0 \cdot \partial_\alpha \mathbf{n}_0 = 0 - 0 = 0, \quad \alpha = 1, 2,$$

since $\partial_\alpha \mathbf{n}$ lies in the tangent plane of \mathbf{n} , see Bischoff et al. [BWBR04].

Assuming a St. Venant-Kirchhoff material, we get the following linear dependency between stress and strain

$$\mathbf{S} = \mathbf{C} \mathbf{E} = \mathbf{S}^m + \vartheta_3 \mathbf{S}^f \quad (3.13)$$

giving us the *second Piola-Kirchhoff stress tensor*. It is defined with respect to the undeformed covariant basis $((\mathbf{a}_i)_0 \otimes (\mathbf{a}_j)_0)$, where $(\mathbf{a}_i)_0$ are the columns of the matrix \mathbf{F}_0 , i.e. $(\mathbf{a}_\alpha)_0 = \partial_\alpha \mathbf{c}_0$, $\alpha = 1, 2$ and $(\mathbf{a}_3)_0 = \mathbf{n}_0$. The stress tensor is split into two parts

$$\mathbf{S}^m := \mathbf{C} \mathbf{E}^m \quad \text{and} \quad \mathbf{S}^f := \mathbf{C} \mathbf{E}^f, \quad (3.14)$$

where \mathcal{C} is the 4th order tensor which states the constitutive relation. It depends on the material parameters (E-modulus E and Poisson's ratio ν or the Lamé constants λ and μ , respectively) as well as on the initial configuration \mathbf{c}_0 . In this case

$$\mathcal{C}^{\alpha\beta\sigma\tau} = 4 \frac{\lambda\mu}{\lambda + 2\mu} a_0^{\alpha\beta} a_0^{\sigma\tau} + 2\mu(a_0^{\alpha\sigma} a_0^{\beta\tau} + a_0^{\alpha\tau} a_0^{\beta\sigma}), \quad \alpha, \beta, \sigma, \tau = 1, 2, \quad (3.15)$$

where $a_0^{\alpha\beta} = \mathbf{a}_0^\alpha \cdot \mathbf{a}_0^\beta$. This relation incorporates a plane stress state.

It follows that the stress is also two dimensional and to reduce some complexity in the notation, the Voigt's notation is used from now on and the curvilinear components of the strain and stress tensors are rewritten as the covariant and contravariant vectors respectively,

$$\vec{\mathbf{E}}^m = \begin{pmatrix} \mathbf{E}_{11}^m \\ \mathbf{E}_{22}^m \\ 2 \mathbf{E}_{12}^m \end{pmatrix}, \quad \vec{\mathbf{E}}^f = \begin{pmatrix} \mathbf{E}_{11}^f \\ \mathbf{E}_{22}^f \\ 2 \mathbf{E}_{12}^f \end{pmatrix} \quad \text{and} \quad \vec{\mathbf{S}}^m = \begin{pmatrix} \mathbf{S}^{m,11} \\ \mathbf{S}^{m,22} \\ \mathbf{S}^{m,12} \end{pmatrix}, \quad \vec{\mathbf{S}}^f = \begin{pmatrix} \mathbf{S}^{f,11} \\ \mathbf{S}^{f,22} \\ \mathbf{S}^{f,12} \end{pmatrix} \quad (3.16)$$

and the fourth order tensor \mathcal{C} as the matrix

$$\mathbf{C} = \begin{pmatrix} \mathcal{C}^{1111} & \mathcal{C}^{1122} & \mathcal{C}^{1112} \\ \mathcal{C}^{2211} & \mathcal{C}^{2222} & \mathcal{C}^{2212} \\ \mathcal{C}^{1211} & \mathcal{C}^{1222} & \mathcal{C}^{1212} \end{pmatrix}. \quad (3.17)$$

Note that it can be easily verified that the material matrix \mathbf{C} in (3.17) is *symmetric* from its definition using (3.15).

Remark 3.5. *The material matrix \mathbf{C} is dependent on the undeformed configuration of the centerline due to the choice of curvilinear coordinates. It is also possible to use the regular plane strain material matrix*

$$\mathbf{D} = \frac{E}{1 - \nu^2} \begin{pmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{pmatrix}, \quad (3.18)$$

which is independent of the undeformed configuration. To work with this matrix, we transform the strains with respect to the local Cartesian basis $\mathbf{e}_i \otimes \mathbf{e}_j$ using the relation

$$\hat{\mathbf{E}}_{kl}^{m/f} = \vec{\mathbf{E}}_{ij}^{m/f} (\mathbf{e}_k \cdot \mathbf{a}_0^i) (\mathbf{a}_0^j \cdot \mathbf{e}_l), \quad k, l, i, j = 1, 2, 3, \quad (3.19)$$

with the local Cartesian axis, see [BWBR04], defined by

$$\mathbf{e}_1 = \frac{(\mathbf{a}_1)_0}{\|(\mathbf{a}_1)_0\|}, \quad \mathbf{e}_2 = \frac{\mathbf{a}_0^2}{\|\mathbf{a}_0^2\|}, \quad \mathbf{e}_3 = (\mathbf{a}_3)_0 = \mathbf{n}_0 = \frac{(\mathbf{a}_1)_0 \times (\mathbf{a}_2)_0}{\|(\mathbf{a}_1)_0 \times (\mathbf{a}_2)_0\|}. \quad (3.20)$$

The stresses in the Cartesian coordinates are then determined by

$$\hat{\mathbf{S}}^m = \mathbf{D} \hat{\mathbf{E}}^m \quad \text{and} \quad \hat{\mathbf{S}}^f = \mathbf{D} \hat{\mathbf{E}}^f. \quad (3.21)$$

The notation $(\hat{\cdot})$ denotes values with respect to the Cartesian coordinates. For future calculations, we however stick to the curvilinear coordinates unless otherwise stated.

The internal energy of the shell reads

$$\begin{aligned}
U_{\text{int}}(\mathbf{u}) &= \int_{\omega \times [-\epsilon, \epsilon]} \frac{1}{4} \mathbf{S} : \boldsymbol{\varepsilon} \nu_0 \, d\boldsymbol{\vartheta} \\
&= \frac{1}{4} \int_{\omega \times [-\epsilon, \epsilon]} \left(\vec{\mathbf{S}}^m \cdot \vec{\mathbf{E}}^m + 2\vartheta_3 \vec{\mathbf{S}}^m \cdot \vec{\mathbf{E}}^f + \vartheta_3^2 \vec{\mathbf{S}}^f \cdot \vec{\mathbf{E}}^f \right) \nu_0 \, d\boldsymbol{\vartheta} \\
&= \int_{\omega} \left(\frac{\epsilon}{2} \vec{\mathbf{S}}^m \cdot \vec{\mathbf{E}}^m + \frac{\epsilon^3}{6} \vec{\mathbf{S}}^f \cdot \vec{\mathbf{E}}^f \right) \nu_0 \, d(\vartheta_1, \vartheta_2)
\end{aligned} \tag{3.22}$$

where $\nu_0 = \det \mathbf{F}_0$ and appears due to the integral transformation from the physical to parametric space. The kinetic energy is given by

$$T(\dot{\mathbf{u}}) = \frac{1}{2} \int_{\omega \times [-\epsilon, \epsilon]} \rho \dot{\mathbf{c}} \cdot \dot{\mathbf{c}} \nu_0 \, d\boldsymbol{\vartheta} = \epsilon \int_{\omega} \rho \dot{\mathbf{c}} \cdot \dot{\mathbf{c}} \nu_0 \, d(\vartheta_1, \vartheta_2). \tag{3.23}$$

Remark 3.6. *It must be noted that actually*

$$T = \frac{1}{2} \int_{\omega \times [-\epsilon, \epsilon]} \rho \dot{\boldsymbol{\Theta}} \cdot \dot{\boldsymbol{\Theta}} \nu_0 \, d\boldsymbol{\vartheta} = \int_{\omega} \left(\epsilon \rho \dot{\mathbf{c}} \cdot \dot{\mathbf{c}} + \frac{\epsilon^3}{3} \rho \dot{\mathbf{n}} \cdot \dot{\mathbf{n}} \right) \nu_0 \, d(\vartheta_1, \vartheta_2). \tag{3.24}$$

Since the normal \mathbf{n} is fully determined by the shell centerline \mathbf{c} and is attached to it, the velocities of \mathbf{c}, \mathbf{n} should have the same magnitude. Therefore, the contribution from $\frac{\epsilon^3}{3} \rho \dot{\mathbf{n}} \cdot \dot{\mathbf{n}}$ is neglected for thin shells, since ϵ is small.

3.2.2 External Energy

The external energy contributions come from the external Neumann forces to the shell such as body, surface, boundary and point forces or their respective moments. We show below the energy contributions coming from various kind of forces.

Body force For a body or volume force \mathbf{p}_v , we have

$$U_{\text{ext}}^v = \int_{\omega \times [-\epsilon, \epsilon]} \mathbf{p}_v \cdot \mathbf{u} \nu_0 \, d\boldsymbol{\vartheta} = 2\epsilon \int_{\omega} \mathbf{p}_v \cdot \mathbf{u} \nu_0 \, d(\vartheta_1, \vartheta_2). \tag{3.25}$$

Surface force For a surface force \mathbf{p}_s , we have

$$U_{\text{ext}}^s = \int_{\omega} \mathbf{p}_s \cdot \mathbf{u} \nu_0 \, d(\vartheta_1, \vartheta_2). \tag{3.26}$$

Boundary force If the surface force is a line force \mathbf{p}_b along the part of a boundary Γ , where ϑ_α varies

$$U_{\text{ext}}^b = \int_{\Gamma} \mathbf{p}_b \cdot \mathbf{u} \nu_\alpha \, d\vartheta_\alpha, \quad \alpha = 1 \text{ or } 2, \tag{3.27}$$

where $\vartheta_\alpha = \|\partial_\alpha \mathbf{c}_0\|$.

Point force If the surface force is a point force \mathbf{p}_p at $\mathbf{c}^* = \mathbf{c}(\vartheta_1^*, \vartheta_2^*)$, it leads to

$$U_{\text{ext}}^p = (\mathbf{p}_p \cdot \mathbf{u})(\vartheta_1^*, \vartheta_2^*). \quad (3.28)$$

Remark 3.7. *The loads are usually specified on the physical domain. For body, surface or boundary loads, the parametric domain is quite easy to identify, since the limits are well known. On the other hand, for the application of point load at a random point $\mathbf{c}^* = \mathbf{c}(\vartheta_1^*, \vartheta_2^*)$, it is required that we apply a point projection algorithm in order to find out the corresponding parameters $(\vartheta_1^*, \vartheta_2^*)$. Point projection for the NURBS surface is shown in Section 4.1.5.*

Note that we do not distinguish between a force applied at the top or at the bottom of the shell. The net external energy is the sum of all the energies from (3.25)–(3.28). For our calculations ahead, we represent all the applied forces \mathbf{p} in terms of the most general body force form.

$$U_{\text{ext}} = 2\epsilon \int_{\omega} \mathbf{p} \cdot \mathbf{u} \nu_0 \, d(\vartheta_1, \vartheta_2) = U_{\text{ext}}^v + U_{\text{ext}}^s + U_{\text{ext}}^b + U_{\text{ext}}^p. \quad (3.29)$$

Thus, we have now all energy expressions available in order to derive the equations of motion by invoking Hamilton's principle of least action. In case of additional dissipative terms or more general loads, appropriate extensions can be used, see Section 3.5 on viscoelasticity. It is to be noted that in the limit case of small strain, the nonlinear terms with respect to \mathbf{u} can be neglected from \mathbf{E}^m , \mathbf{E}^f and all the resulting expressions.

3.2.3 From the Functional to the Weak Form

For the conservative case, to find the displacement $\mathbf{u} \in (H^2(\omega))^3 \otimes C^2(t)$, $t \in [t_0, t_1]$, we apply the *principle of least action*, also known as *Hamilton's principle* to the *Lagrangian* $\mathcal{L} = T - U$ as in the MBS case, see (2.32),

$$\int_{t_0}^{t_1} \mathcal{L} \, dt := \int_{t_0}^{t_1} (T - U) \, dt = \int_{t_0}^{t_1} (T - U_{\text{int}} + U_{\text{ext}}) \, dt \rightarrow \text{stationary!} \quad (3.30)$$

To find the stationary point, we perturb the displacement \mathbf{u} using a variation $\theta \mathbf{v}$, where $\theta \in \mathbb{R}$ and \mathbf{v} is the *admissible* test function. Admissible \mathbf{v} means that the perturbed displacement $\mathbf{u} + \theta \mathbf{v}$ must satisfy the Dirichlet boundary condition (3.6) and hence $\mathbf{v}(\vartheta_1, \vartheta_2, t) = \mathbf{0}$ for $(\vartheta_1, \vartheta_2) \in \Gamma_0$ at any time $t_0 \leq t \leq t_1$. Also, $\mathbf{v}(\vartheta_1, \vartheta_2, t_0) = \mathbf{v}(\vartheta_1, \vartheta_2, t_1) = \mathbf{0}$ for all $(\vartheta_1, \vartheta_2) \in \omega$. Note that in the engineering literature, the perturbation is applied using the virtual displacements $\delta \mathbf{u}$ instead of $\theta \mathbf{v}$.

Next, we define the functional

$$J(\theta) := \int_{t_0}^{t_1} (T(\mathbf{u} + \theta \mathbf{v}) - U(\mathbf{u} + \theta \mathbf{v})) \, dt \quad (3.31)$$

and postulate due to the stationarity condition (3.30)

$$0 = \frac{d}{d\theta} J(\theta) \Big|_{\theta=0}. \quad (3.32)$$

This gives us

$$0 = \int_{t_0}^{t_1} \left(\int_{\omega} \left(2\epsilon \rho \dot{\mathbf{c}} \cdot \dot{\mathbf{v}} - \epsilon \bar{\mathbf{S}}^m \cdot \bar{\mathbf{E}}_*^m(\mathbf{v}) - \frac{\epsilon^3}{3} \bar{\mathbf{S}}^f \cdot \bar{\mathbf{E}}_*^f(\mathbf{v}) + 2\epsilon \mathbf{p} \cdot \mathbf{v} \right) \nu_0 \, d(\vartheta_1, \vartheta_2) \right) dt. \quad (3.33)$$

We assume that the applied forces \mathbf{p} are independent of the displacement \mathbf{u} . Due to the application of the product rule for the derivatives and therein using the symmetricity of the material matrix \mathbf{C} , i.e. $\mathbf{C} = \mathbf{C}^T$, we get the variations in terms of the star-ed first variation vectors of strain $\vec{\mathbf{E}}_\star^m$ and $\vec{\mathbf{E}}_\star^f$.

$$\vec{\mathbf{E}}_\star^m = \begin{pmatrix} \partial_1 \mathbf{v} \cdot \partial_1 \mathbf{c} \\ \partial_2 \mathbf{v} \cdot \partial_2 \mathbf{c} \\ \partial_1 \mathbf{v} \cdot \partial_2 \mathbf{c} + \partial_2 \mathbf{v} \cdot \partial_1 \mathbf{c} \end{pmatrix} \quad \text{and} \quad \vec{\mathbf{E}}_\star^f = \begin{pmatrix} \partial_{11} \mathbf{v} \cdot \mathbf{n} + \partial_{11} \mathbf{c} \cdot \mathbf{n}^\star(\mathbf{v}) \\ \partial_{22} \mathbf{v} \cdot \mathbf{n} + \partial_{22} \mathbf{c} \cdot \mathbf{n}^\star(\mathbf{v}) \\ 2 \partial_{12} \mathbf{v} \cdot \mathbf{n} + 2 \partial_{12} \mathbf{c} \cdot \mathbf{n}^\star(\mathbf{v}) \end{pmatrix}$$

where $\mathbf{n}^\star(\mathbf{v})$ is again the first variation, this time of the normal, being defined as

$$\mathbf{n}^\star(\mathbf{v}) = \mathbf{m}_\mathbf{v} - (\mathbf{n} \cdot \mathbf{m}_\mathbf{v})\mathbf{n}, \quad \text{where} \quad \mathbf{m}_\mathbf{v} = \frac{1}{|\partial_1 \mathbf{c} \times \partial_2 \mathbf{c}|} (\partial_1 \mathbf{v} \times \partial_2 \mathbf{c} + \partial_1 \mathbf{c} \times \partial_2 \mathbf{v}).$$

Integrating by parts the first term in (3.33) with respect to time t leads to

$$\int_{t_0}^{t_1} \int_\omega 2\epsilon\rho \dot{\mathbf{c}} \cdot \dot{\mathbf{v}} \nu_0 \, d(\vartheta_1, \vartheta_2) \, dt = \int_\omega 2\epsilon\rho \dot{\mathbf{c}} \cdot \mathbf{v} \nu_0 \, d(\vartheta_1, \vartheta_2) \Big|_{t_0}^{t_1} - \int_{t_0}^{t_1} \int_\omega 2\epsilon\rho \ddot{\mathbf{c}} \cdot \mathbf{v} \nu_0 \, d(\vartheta_1, \vartheta_2) \, dt.$$

Due to the time independence of the undeformed shell centerline \mathbf{c}_0 from (3.4), we get $\dot{\mathbf{c}} = \ddot{\mathbf{u}}$. Also, the first term on the right hand side vanishes due to $\mathbf{v}(\vartheta_1, \vartheta_2, t_0) = \mathbf{v}(\vartheta_1, \vartheta_2, t_1) = \mathbf{0}$ and results in the *variational formulation*

$$0 = \int_{t_0}^{t_1} \left(\int_\omega \left(2\epsilon\rho \ddot{\mathbf{u}} \cdot \mathbf{v} + \epsilon \vec{\mathbf{S}}^m \cdot \vec{\mathbf{E}}_\star^m(\mathbf{v}) + \frac{\epsilon^3}{3} \vec{\mathbf{S}}^f \cdot \vec{\mathbf{E}}_\star^f(\mathbf{v}) - 2\epsilon\mathbf{p} \cdot \mathbf{v} \right) \nu_0 \, d(\vartheta_1, \vartheta_2) \right) dt, \quad (3.34)$$

which must hold true for all admissible functions \mathbf{v} . This leads to the form

$$\begin{aligned} 0 &= \int_\omega (2\epsilon\rho \ddot{\mathbf{u}} \cdot \mathbf{v}) \nu_0 \, d(\vartheta_1, \vartheta_2) + \underbrace{\int_\omega \left(\epsilon \vec{\mathbf{S}}^m \cdot \vec{\mathbf{E}}_\star^m(\mathbf{v}) + \frac{\epsilon^3}{3} \vec{\mathbf{S}}^f \cdot \vec{\mathbf{E}}_\star^f(\mathbf{v}) \right) \nu_0 \, d(\vartheta_1, \vartheta_2)}_{=: a(\mathbf{u}, \mathbf{v})} \\ &\quad - \underbrace{\int_\omega (2\epsilon\mathbf{p} \cdot \mathbf{v}) \nu_0 \, d(\vartheta_1, \vartheta_2)}_{=: b(\mathbf{v})}, \end{aligned} \quad (3.35)$$

which is known as the *weak form* or the so-called *principle of virtual work*. It is linear in \mathbf{v} but nonlinear in \mathbf{u} .

In order to solve such a problem on computers, the finite element method is applied to obtain a discretization of the weak form. We delay a discussion of this topic to the next chapter. Nonetheless, anticipating the appearance of the Jacobian matrix of the force vector or more popularly the *stiffness matrix* in the solution of this nonlinear problem later on, we derive an analytical expression for the Jacobian $j(\mathbf{u}, \mathbf{v}, \mathbf{w})$. To obtain it, we again apply the calculus of variations on the right hand side of (3.35) in the time independent setting, leading to

$$\begin{aligned} j(\mathbf{u}, \mathbf{v}, \mathbf{w}) &= \frac{d}{d\theta} a(\mathbf{u} + \theta\mathbf{w}, \mathbf{v}) \Big|_{\theta=0} \\ &= \int_\omega \left(\epsilon \left(\vec{\mathbf{S}}^m \cdot \vec{\mathbf{E}}_{\star\star}^m(\mathbf{v}, \mathbf{w}) + \vec{\mathbf{S}}_\star^m(\mathbf{w}) \cdot \vec{\mathbf{E}}_\star^m(\mathbf{v}) \right) \right. \\ &\quad \left. + \frac{\epsilon^3}{3} \left(\vec{\mathbf{S}}^f \cdot \vec{\mathbf{E}}_{\star\star}^f(\mathbf{v}, \mathbf{w}) + \vec{\mathbf{S}}_\star^f(\mathbf{w}) \cdot \vec{\mathbf{E}}_\star^f(\mathbf{v}) \right) \right) \nu_0 \, d(\vartheta_1, \vartheta_2), \end{aligned} \quad (3.36)$$

where the first variations of the stress vectors follow simply from the constitutive relation, i.e. $\vec{\mathbf{S}}^m = \mathbf{C}\vec{\mathbf{E}}^m$ and $\vec{\mathbf{S}}^f = \mathbf{C}\vec{\mathbf{E}}^f$ and the double star-ed vectors are the second variation of the strain vectors resulting in

$$\vec{\mathbf{E}}_{**}^m = \begin{pmatrix} \partial_1 \mathbf{v} \cdot \partial_1 \mathbf{w} \\ \partial_2 \mathbf{v} \cdot \partial_2 \mathbf{w} \\ \partial_1 \mathbf{v} \cdot \partial_2 \mathbf{w} + \partial_2 \mathbf{v} \cdot \partial_1 \mathbf{w} \end{pmatrix}$$

and

$$\vec{\mathbf{E}}_{**}^f = \begin{pmatrix} \partial_{11} \mathbf{v} \cdot \mathbf{n}^*(\mathbf{w}) + \partial_{11} \mathbf{w} \cdot \mathbf{n}^*(\mathbf{v}) + \partial_{11} \mathbf{c} \cdot \mathbf{n}^{**}(\mathbf{v}, \mathbf{w}) \\ \partial_{22} \mathbf{v} \cdot \mathbf{n}^*(\mathbf{w}) + \partial_{22} \mathbf{w} \cdot \mathbf{n}^*(\mathbf{v}) + \partial_{22} \mathbf{c} \cdot \mathbf{n}^{**}(\mathbf{v}, \mathbf{w}) \\ 2 \partial_{12} \mathbf{v} \cdot \mathbf{n}^*(\mathbf{w}) + 2 \partial_{12} \mathbf{w} \cdot \mathbf{n}^*(\mathbf{v}) + 2 \partial_{12} \mathbf{c} \cdot \mathbf{n}^{**}(\mathbf{v}, \mathbf{w}) \end{pmatrix},$$

where

$$\mathbf{n}^{**}(\mathbf{v}, \mathbf{w}) = \mathbf{m}_{\mathbf{v}}^*(\mathbf{w}) - (\mathbf{m}_{\mathbf{v}} \cdot \mathbf{n}^*(\mathbf{w}) + \mathbf{n} \cdot \mathbf{m}_{\mathbf{v}}^*(\mathbf{w}))\mathbf{n} - (\mathbf{n} \cdot \mathbf{m}_{\mathbf{v}})\mathbf{n}^*(\mathbf{w})$$

with

$$\mathbf{m}_{\mathbf{v}}^*(\mathbf{w}) = \mathbf{m}_{\mathbf{vw}} - (\mathbf{n} \cdot \mathbf{m}_{\mathbf{w}})\mathbf{m}_{\mathbf{v}} \quad \text{and} \quad \mathbf{m}_{\mathbf{vw}} = \frac{1}{|\partial_1 \mathbf{c} \times \partial_2 \mathbf{c}|} (\partial_1 \mathbf{v} \times \partial_2 \mathbf{w} + \partial_1 \mathbf{w} \times \partial_2 \mathbf{v}).$$

Remark 3.8. *The calculation of the flexural components $\vec{\mathbf{E}}^f, \vec{\mathbf{E}}_{**}^f$ in the weak form (3.35) requires the computation of double derivatives of the centerline $\partial_{\alpha\beta} \mathbf{c}$ and the test functions $\partial_{\alpha\beta} \mathbf{v}$, $\alpha, \beta = 1, 2$. This demands a high regularity of at least H^2 or C^1 for the Kirchhoff-Love kinematics.*

3.3 Dirichlet Boundary Conditions

The solution of the elasticity problem on shells requires that the solution satisfies the system invariants (3.6). Instead of choosing the function space and admissible functions such that the Dirichlet boundary conditions (3.6), i.e. $\mathbf{u} = \mathbf{u}_0$ on Γ_0 are automatically satisfied, there is another possibility of enforcing the boundary conditions in the weak formulation using Lagrange multipliers on the boundary Γ_0 . The advantage of such an approach is that the boundary conditions in this case must not be invariants; and can change throughout and act as system constraints. This is particularly useful for flexible multibody system applications, where the elastic bodies are also connected to other bodies via joints, which can impart rigid motion to the elastic bodies. In such a case, the prescribed motion \mathbf{u}_0 is not known beforehand.

The basic idea is to append the boundary conditions $\mathbf{u} = \mathbf{u}_0$ as a constraint in the Lagrangian to which Hamilton's principle is applied [Sim13], giving us

$$\int_{t_0}^{t_1} \left(T - U - \int_{\Gamma_0} \boldsymbol{\mu}^T (\mathbf{u} - \mathbf{u}_0) \nu_0 \, d(\vartheta_1, \vartheta_2) \right) dt \rightarrow \text{stationary}, \quad (3.37)$$

where $\boldsymbol{\mu} = \boldsymbol{\mu}(\vartheta_1, \vartheta_2, t)$ is the *Lagrangian multiplier* defined on the boundary Γ_0 . Moreover, the displacement \mathbf{u} is now arbitrary and need not satisfy (3.6) a-priori.

Similar to (3.31), we define the functional

$$J(\theta) := \int_{t_0}^{t_1} \left(T(\mathbf{u} + \theta \mathbf{v}) - U(\mathbf{u} + \theta \mathbf{v}) - \int_{\Gamma_0} (\boldsymbol{\mu} + \theta \mathbf{v})^T (\mathbf{u} + \theta \mathbf{v} - \mathbf{u}_0) \nu_0 \, d(\vartheta_1, \vartheta_2) \right) dt, \quad (3.38)$$

where the perturbations $\mathbf{v}(\vartheta_1, \vartheta_2, t_0) = \mathbf{v}(\vartheta_1, \vartheta_2, t_1) = \mathbf{0}$ for all $(\vartheta_1, \vartheta_2) \in \omega$ and $\mathbf{v}(\vartheta_1, \vartheta_2, t_0) = \mathbf{v}(\vartheta_1, \vartheta_2, t_1) = \mathbf{0}$ for all $(\vartheta_1, \vartheta_2)$ on Γ_0 . Taking the shell energies T, U from Section 3.2.3 and following the stationarity condition (3.32) on J , we get

$$0 = \int_{t_0}^{t_1} \left(\int_{\omega} \left(2\epsilon\rho \ddot{\mathbf{u}} \cdot \mathbf{v} + \epsilon \vec{\mathbf{S}}^m \cdot \vec{\mathbf{E}}_*^m(\mathbf{v}) + \frac{\epsilon^3}{3} \vec{\mathbf{S}}^f \cdot \vec{\mathbf{E}}_*^f(\mathbf{v}) - 2\epsilon\mathbf{p} \cdot \mathbf{v} \right) \nu_0 \, d(\vartheta_1, \vartheta_2) + \int_{\Gamma_0} \left(\boldsymbol{\mu}^T \mathbf{v} + \mathbf{v}^T (\mathbf{u} - \mathbf{u}_0) \right) \nu_0 \, d(\vartheta_1, \vartheta_2) \right) dt, \quad (3.39)$$

which must hold for all admissible \mathbf{v} and \mathbf{v} .

The variational formulation (3.39) must hold true in case \mathbf{v} or \mathbf{v} vanish. We get the system

$$0 = \int_{\omega} (2\epsilon\rho \ddot{\mathbf{u}} \cdot \mathbf{v}) \nu_0 \, d(\vartheta_1, \vartheta_2) + \underbrace{\int_{\omega} \left(\epsilon \vec{\mathbf{S}}^m \cdot \vec{\mathbf{E}}_*^m(\mathbf{v}) + \frac{\epsilon^3}{3} \vec{\mathbf{S}}^f \cdot \vec{\mathbf{E}}_*^f(\mathbf{v}) \right) \nu_0 \, d(\vartheta_1, \vartheta_2)}_{=: a(\mathbf{u}, \mathbf{v})} - \underbrace{\int_{\omega} (2\epsilon\mathbf{p} \cdot \mathbf{v}) \nu_0 \, d(\vartheta_1, \vartheta_2)}_{=: b(\mathbf{v})} + \underbrace{\int_{\Gamma_0} \boldsymbol{\mu}^T \mathbf{v} \nu_0 \, d(\vartheta_1, \vartheta_2)}_{=: g(\mathbf{v}, \boldsymbol{\mu})}, \quad (3.40a)$$

$$0 = \underbrace{\int_{\Gamma_0} \mathbf{v}^T \mathbf{u} \nu_0 \, d(\vartheta_1, \vartheta_2)}_{=: g(\mathbf{u}, \mathbf{v})} - \underbrace{\int_{\Gamma_0} \mathbf{v}^T \mathbf{u}_0 \nu_0 \, d(\vartheta_1, \vartheta_2)}_{=: \ell(\mathbf{v})}, \quad (3.40b)$$

which must hold for all admissible \mathbf{v} and \mathbf{v} . The system (3.40) is called the *weak constraint form*. It is a *time-dependent saddle point problem*, which leads in the discretized form to a system similar to a constrained mechanical system as we see in Chapter 4.

3.4 Stress Recovery

The elasticity problem on shells as in Section 3.2 gives us the deformed geometry of the shell. Post processing this information, we can also calculate other physical quantities such as stresses which are important from an engineering point of view. Computing the stresses from the obtained displacements is called the *stress recovery*. In this section, we show the calculations for various frequently used stress parameters following the approach in [Kie11].

Following Section 3.2.1, the covariant components $\boldsymbol{\mathcal{E}}$ of the strain tensor \mathbf{E} are calculated in terms of vectors $\vec{\mathbf{E}}^m$ and $\vec{\mathbf{E}}^f$ with respect to the undeformed contravariant basis $\mathbf{a}_0^i \otimes \mathbf{a}_0^j$, $i, j = 1, 2, 3$,

$$\mathbf{E} = \boldsymbol{\mathcal{E}}_{ij} \mathbf{a}_0^i \otimes \mathbf{a}_0^j. \quad (3.41)$$

Using the material law, we compute the contravariant components $\boldsymbol{\mathcal{S}}$ of the second Piola-Kirchhoff stress tensor \mathbf{S} in terms of vectors $\vec{\mathbf{S}}^m$ and $\vec{\mathbf{S}}^f$ with respect to the undeformed covariant basis $((\mathbf{a}_i)_0 \otimes (\mathbf{a}_j)_0)$,

$$\mathbf{S} = \boldsymbol{\mathcal{S}}^{ij} ((\mathbf{a}_i)_0 \otimes (\mathbf{a}_j)_0). \quad (3.42)$$

It is to be noted that these stress values $\vec{\mathbf{S}}^m, \vec{\mathbf{S}}^f$ in (3.22) are computed only on the shell centerline and the thickness direction is included in the energy calculations by pre-integration.

For an accurate stress recovery, we therefore need to include the effect of this pre-integration in the membrane and bending stress values and compute

$$\vec{\mathbf{S}}_{3D}^m = 2\epsilon\vec{\mathbf{S}}^m \quad \text{and} \quad \vec{\mathbf{S}}_{3D}^f = \frac{2}{3}\epsilon^3\vec{\mathbf{S}}^f. \quad (3.43)$$

Next we compute the contravariant components of *Cauchy stress tensor* $\boldsymbol{\sigma}$ in terms of the membrane and bending stress vectors $\vec{\boldsymbol{\sigma}}^m$ and $\vec{\boldsymbol{\sigma}}^f$ respectively. It is calculated with respect to the actual covariant basis $\mathbf{a}_i \otimes \mathbf{a}_j$, where $\mathbf{a}_\alpha = \partial_\alpha \mathbf{c}$, $\mathbf{a}_3 = \mathbf{n}$. The relation between the contravariant Cauchy stresses $\vec{\boldsymbol{\sigma}}^m, \vec{\boldsymbol{\sigma}}^f$ and the second Piola-Kirchhoff stresses $\vec{\mathbf{S}}_{3D}^m, \vec{\mathbf{S}}_{3D}^f$ is given as, see [BW00],

$$\vec{\boldsymbol{\sigma}}^{m,f} = \frac{|\partial_1 \mathbf{c}_0 \times \partial_2 \mathbf{c}_0|}{|\partial_1 \mathbf{c} \times \partial_2 \mathbf{c}|} \vec{\mathbf{S}}_{3D}^{m,f}. \quad (3.44)$$

To compute the stress values in standard units, these need to be transformed to the local Cartesian coordinates $\hat{\boldsymbol{\sigma}}_{kl}$ in the basis $\mathbf{e}_i \otimes \mathbf{e}_j$ using the relation

$$\hat{\boldsymbol{\sigma}}_{kl} = \vec{\boldsymbol{\sigma}}^{ij} (\mathbf{e}^k \cdot \mathbf{a}_i) (\mathbf{a}_j \cdot \mathbf{e}^l). \quad (3.45)$$

This implies

$$\hat{\boldsymbol{\sigma}}^m = \mathbf{T} \vec{\boldsymbol{\sigma}}^m \quad \text{and} \quad \hat{\boldsymbol{\sigma}}^f = \mathbf{T} \vec{\boldsymbol{\sigma}}^f, \quad (3.46)$$

where

$$\mathbf{T} = \begin{pmatrix} (\mathbf{e}^1 \cdot \mathbf{a}_1)(\mathbf{a}_1 \cdot \mathbf{e}^1) & (\mathbf{e}^1 \cdot \mathbf{a}_2)(\mathbf{a}_2 \cdot \mathbf{e}^1) & 2(\mathbf{e}^1 \cdot \mathbf{a}_1)(\mathbf{a}_2 \cdot \mathbf{e}^1) \\ (\mathbf{e}^2 \cdot \mathbf{a}_1)(\mathbf{a}_1 \cdot \mathbf{e}^2) & (\mathbf{e}^2 \cdot \mathbf{a}_2)(\mathbf{a}_2 \cdot \mathbf{e}^2) & 2(\mathbf{e}^2 \cdot \mathbf{a}_1)(\mathbf{a}_2 \cdot \mathbf{e}^2) \\ (\mathbf{e}^1 \cdot \mathbf{a}_1)(\mathbf{a}_1 \cdot \mathbf{e}^2) & (\mathbf{e}^1 \cdot \mathbf{a}_2)(\mathbf{a}_2 \cdot \mathbf{e}^2) & (\mathbf{e}^1 \cdot \mathbf{a}_1)(\mathbf{a}_2 \cdot \mathbf{e}^2) + (\mathbf{e}^1 \cdot \mathbf{a}_2)(\mathbf{a}_1 \cdot \mathbf{e}^2) \end{pmatrix}$$

and

$$\mathbf{e}^1 = \frac{\mathbf{a}_1}{\|\mathbf{a}_1\|}, \quad \mathbf{e}^2 = \frac{\mathbf{a}^2}{\|\mathbf{a}^2\|} \quad \text{and} \quad \mathbf{e}^3 = \mathbf{a}_3 = \mathbf{n} = \frac{\mathbf{a}_1 \times \mathbf{a}_2}{\|\mathbf{a}_1 \times \mathbf{a}_2\|}.$$

Note that this choice of \mathbf{e}^1 and \mathbf{e}^2 results in $\mathbf{e}^1 \cdot \mathbf{e}^2 = 0$ i.e. $\mathbf{e}^1 \perp \mathbf{e}^2$. Moreover, as \mathbf{a}_1 and \mathbf{a}^2 lie in the same plane and \mathbf{a}_3 is the normal to that plane, it leads to an orthogonal Cartesian axis $(\mathbf{e}^1, \mathbf{e}^2, \mathbf{e}^3)$. Of course, there are infinitely many possibilities to choose an orthogonal Cartesian axis, but for our stress calculations we use the aforementioned axis.

Also, in order to compute the membrane and bending stress resultants in the local Cartesian coordinates, we can compute similarly a matrix $\tilde{\mathbf{T}}$ for the undeformed configuration $\mathbf{e}_0^1, \mathbf{e}_0^2, (\mathbf{a}_1)_0, \mathbf{a}_0^2$ such that

$$\hat{\boldsymbol{\sigma}}^m = \tilde{\mathbf{T}} \vec{\mathbf{S}}_{3D}^m \quad \text{and} \quad \hat{\boldsymbol{\sigma}}^f = \tilde{\mathbf{T}} \vec{\mathbf{S}}_{3D}^f. \quad (3.47)$$

To obtain the *von Mises stress*, we calculate the principal membrane and bending stresses using

$$\begin{aligned} \hat{\sigma}_{X,Y}^m &= \frac{1}{2} \left((\hat{\boldsymbol{\sigma}}_{11}^m + \hat{\boldsymbol{\sigma}}_{22}^m) \pm \sqrt{(\hat{\boldsymbol{\sigma}}_{11}^m - \hat{\boldsymbol{\sigma}}_{22}^m)^2 + 4(\hat{\boldsymbol{\sigma}}_{12}^m)^2} \right), \\ \hat{\sigma}_{X,Y}^f &= \frac{1}{2} \left((\hat{\boldsymbol{\sigma}}_{11}^f + \hat{\boldsymbol{\sigma}}_{22}^f) \pm \sqrt{(\hat{\boldsymbol{\sigma}}_{11}^f - \hat{\boldsymbol{\sigma}}_{22}^f)^2 + 4(\hat{\boldsymbol{\sigma}}_{12}^f)^2} \right). \end{aligned} \quad (3.48)$$

The total stress on the upper shell surface is then obtained by

$$\sigma_{X,Y}^{up} = \frac{\hat{\sigma}_{X,Y}^m}{2\epsilon} + \frac{3}{2} \frac{\hat{\sigma}_{X,Y}^f}{\epsilon^2}. \quad (3.49)$$

Lastly, the von Mises stress $\sigma_{\text{von Mises}}$ is given by

$$\sigma_{\text{von Mises}} = \sqrt{(\sigma_X^{up})^2 + (\sigma_Y^{up})^2 - \sigma_X^{up} \sigma_Y^{up}}. \quad (3.50)$$

3.5 Viscoelasticity

For accurately predicting the transient response of structural members, it is important to consider the dissipative effects, because in reality no material has a purely elastic behavior. The materials deviate from it e.g. by exhibiting viscous properties additional to elastic characteristics. It is also desirable from a numerical point of view to consider the damping of spurious high frequency oscillations that appear when dealing with relatively stiff elastic structures in the flexible multibody dynamics.

Viscoelastic materials are those for which the relationship between stress and strain depends on time. A basic viscoelastic model is the one dimensional *Kelvin-Voigt model*, see [LC90],

$$\sigma = E\epsilon(t) + \eta \frac{d\epsilon(t)}{dt},$$

where the stress σ depends on the strain ϵ as well as the strain rate $\frac{d\epsilon}{dt}$. It is created by using spring and damper in parallel, see Fig. 3.2. This leads to equal strains in both the parts

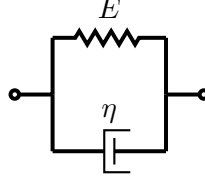


Figure 3.2: Kelvin-Voigt model

$$\epsilon = \epsilon_e = \epsilon_v,$$

where ϵ_e is the strain in the elastic part and ϵ_v the strain in the viscous part. The total stress is given by

$$\sigma = \sigma_e + \sigma_v,$$

where σ_e, σ_v are the stresses in the elastic and viscous parts respectively. Due to its simple application, this model is often extended to introduce viscous damping in nonlinear models, see Linn et. al [LLT13]. We also use it to introduce damping in our nonlinear isogeometric shell model. Note that the standard damping methods like the Rayleigh damping cannot be used, since they are well understood only for small deformations, i.e. linear setting.

Analogous to the one dimensional Kelvin-Voigt model as above, we take the viscoelastic shell model, where the stress tensor \mathcal{S} is given in terms of the elastic \mathcal{S}_e and viscous \mathcal{S}_v stress tensors as

$$\mathcal{S} = \mathcal{S}_e + \mathcal{S}_v, \quad (3.51)$$

where $\mathcal{S}_e = \mathcal{C}\mathcal{E}$ and $\mathcal{S}_v = \eta \mathcal{C}\dot{\mathcal{E}}$ with *damping factor* η and

$$\dot{\mathcal{E}} = \dot{\mathbf{E}}^m + \vartheta_3 \dot{\mathbf{E}}^f. \quad (3.52)$$

$$\dot{\mathbf{E}}^m = \frac{d\mathbf{E}^m}{dt} = \begin{pmatrix} \partial_1 \mathbf{c} \cdot \partial_1 \dot{\mathbf{c}} & \frac{1}{2}(\partial_1 \mathbf{c} \cdot \partial_2 \dot{\mathbf{c}} + \partial_1 \dot{\mathbf{c}} \cdot \partial_2 \mathbf{c}) & 0 \\ \frac{1}{2}(\partial_1 \mathbf{c} \cdot \partial_2 \dot{\mathbf{c}} + \partial_1 \dot{\mathbf{c}} \cdot \partial_2 \mathbf{c}) & \partial_2 \mathbf{c} \cdot \partial_2 \dot{\mathbf{c}} & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad (3.53)$$

$$\dot{\mathbf{E}}^f = \frac{d\mathbf{E}^f}{dt} = - \begin{pmatrix} \partial_{11} \mathbf{c} \cdot \dot{\mathbf{n}} + \partial_{11} \dot{\mathbf{c}} \cdot \mathbf{n} & \partial_{12} \mathbf{c} \cdot \dot{\mathbf{n}} + \partial_{12} \dot{\mathbf{c}} \cdot \mathbf{n} & 0 \\ \partial_{12} \mathbf{c} \cdot \dot{\mathbf{n}} + \partial_{12} \dot{\mathbf{c}} \cdot \mathbf{n} & \partial_{22} \mathbf{c} \cdot \dot{\mathbf{n}} + \partial_{22} \dot{\mathbf{c}} \cdot \mathbf{n} & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad (3.54)$$

where $\dot{\mathbf{n}}$ is the time derivative of the normal, defined as

$$\dot{\mathbf{n}} = \mathbf{m}_t - (\mathbf{n} \cdot \mathbf{m}_t) \mathbf{n}, \quad \text{where} \quad \mathbf{m}_t = \frac{1}{|\partial_1 \mathbf{c} \times \partial_2 \mathbf{c}|} (\partial_1 \mathbf{c} \times \partial_2 \dot{\mathbf{c}} + \partial_1 \dot{\mathbf{c}} \times \partial_2 \mathbf{c}).$$

Applying the total stress following the Kelvin-Voigt ansatz (3.51) and vectorizing the strain and stress tensors as before, we get the new weak form

$$\begin{aligned} 0 &= \int_{\omega} (2\epsilon \rho \ddot{\mathbf{u}} \cdot \mathbf{v}) \nu_0 d(\vartheta_1, \vartheta_2) - \underbrace{\int_{\omega} (2\epsilon \mathbf{p} \cdot \mathbf{v}) \nu_0 d(\vartheta_1, \vartheta_2)}_{=: b(\mathbf{v})} \\ &+ \underbrace{\int_{\omega} \left(\epsilon \vec{\mathbf{S}}^m(\mathbf{u}, \dot{\mathbf{u}}) \cdot \vec{\mathbf{E}}_*^m(\mathbf{v}) + (\epsilon^3/3) \vec{\mathbf{S}}^f(\mathbf{u}, \dot{\mathbf{u}}) \cdot \vec{\mathbf{E}}_*^f(\mathbf{v}) \right) \nu_0 d(\vartheta_1, \vartheta_2)}_{=: \tilde{a}(\mathbf{u}, \dot{\mathbf{u}}, \mathbf{v})} \end{aligned} \quad (3.55)$$

where $\vec{\mathbf{S}}^m, \vec{\mathbf{S}}^f$ in case of (hyper-)elasticity comprise of only the elastic stresses and in case of Kelvin-Voigt type model of the elastic as well as the viscous stresses, i.e.

$$\vec{\mathbf{S}}^m = \vec{\mathbf{S}}_e^m + \vec{\mathbf{S}}_v^m \quad \text{and} \quad \vec{\mathbf{S}}^f = \vec{\mathbf{S}}_e^f + \vec{\mathbf{S}}_v^f.$$

The strains $\vec{\mathbf{E}}^m, \vec{\mathbf{E}}^f$ as in the one dimensional model remain the same in both the cases. Again, to solve this weak form (3.55), we find $\mathbf{u} \in (H^2(\omega))^3 \times C^2(t)$, $t \in [t_0, t_1]$ for all admissible \mathbf{v} . The nonlinear problem is solved by the Newton method, which involves the Jacobian or the stiffness matrix

$$\begin{aligned} j(\mathbf{u}, \dot{\mathbf{u}}, \mathbf{v}, \mathbf{w}) &= \frac{d}{d\theta} \tilde{a}(\mathbf{u} + \theta \mathbf{w}, \dot{\mathbf{u}}, \mathbf{v}) \Big|_{\theta=0} \\ &= \int_{\omega} \left(\epsilon \left((\vec{\mathbf{S}}_e^m + \vec{\mathbf{S}}_v^m) \cdot \vec{\mathbf{E}}_{**}^m(\mathbf{v}, \mathbf{w}) + (\vec{\mathbf{S}}_{*,e}^m(\mathbf{w}) + \vec{\mathbf{S}}_{*,v}^m(\mathbf{w})) \cdot \vec{\mathbf{E}}_*^m(\mathbf{v}) \right) \right. \\ &\quad \left. + \frac{\epsilon^3}{3} \left((\vec{\mathbf{S}}_e^f + \vec{\mathbf{S}}_v^f) \cdot \vec{\mathbf{E}}_{**}^f(\mathbf{v}, \mathbf{w}) + (\vec{\mathbf{S}}_{*,e}^f(\mathbf{w}) + \vec{\mathbf{S}}_{*,v}^f(\mathbf{w})) \cdot \vec{\mathbf{E}}_*^f(\mathbf{v}) \right) \right) \nu_0 d(\vartheta_1, \vartheta_2) \end{aligned}$$

and additionally the damping matrix due to viscous terms

$$\begin{aligned} l(\mathbf{u}, \dot{\mathbf{u}}, \mathbf{v}, \mathbf{w}) &= \frac{d}{d\theta} \tilde{a}(\mathbf{u}, \dot{\mathbf{u}} + \theta \mathbf{w}, \mathbf{v}) \Big|_{\theta=0} \\ &= \int_{\omega} \left(\epsilon \vec{\mathbf{S}}_{*,e}^m(\mathbf{w}) \cdot \vec{\mathbf{E}}_*^m(\mathbf{v}) + \frac{\epsilon^3}{3} \vec{\mathbf{S}}_{*,e}^f(\mathbf{w}) \cdot \vec{\mathbf{E}}_*^f(\mathbf{v}) \right) \nu_0 d(\vartheta_1, \vartheta_2). \end{aligned}$$

Taking derivative with respect to $\dot{\mathbf{u}}$ to compute the damping matrix leaves it independent of $\dot{\mathbf{u}}$, since viscous stresses $\vec{\mathbf{S}}_v^m, \vec{\mathbf{S}}_v^f$ are linear with respect to the velocity $\dot{\mathbf{u}}$. $\vec{\mathbf{S}}_{*,e}^m, \vec{\mathbf{S}}_{*,e}^f$ are the same as $\vec{\mathbf{S}}_*^m, \vec{\mathbf{S}}_*^f$ from the elasticity theory, whereas for the viscous model we have

$$\vec{\mathbf{S}}_{*,v}^m(\mathbf{w}) = \eta \mathbf{C} \vec{\mathbf{E}}_{*,v}^m(\mathbf{w}) \quad \text{and} \quad \vec{\mathbf{S}}_{*,v}^f(\mathbf{w}) = \eta \mathbf{C} \vec{\mathbf{E}}_{*,v}^f(\mathbf{w}),$$

with

$$\vec{\mathbf{E}}_{\star,v}^m(\mathbf{w}) = \begin{pmatrix} \partial_1 \mathbf{w} \cdot \partial_1 \dot{\mathbf{c}} \\ \partial_2 \mathbf{w} \cdot \partial_2 \dot{\mathbf{c}} \\ \partial_1 \mathbf{w} \cdot \partial_2 \dot{\mathbf{c}} + \partial_2 \mathbf{w} \cdot \partial_1 \dot{\mathbf{c}} \end{pmatrix},$$

$$\vec{\mathbf{E}}_{\star,v}^f(\mathbf{w}) = - \begin{pmatrix} \partial_{11} \mathbf{c} \cdot \dot{\mathbf{n}}_{\star}(\mathbf{w}) + \partial_{11} \mathbf{v} \cdot \dot{\mathbf{n}} + \partial_{11} \dot{\mathbf{c}} \cdot \mathbf{n}_{\star}(\mathbf{w}) \\ \partial_{22} \mathbf{c} \cdot \dot{\mathbf{n}}_{\star}(\mathbf{w}) + \partial_{22} \mathbf{v} \cdot \dot{\mathbf{n}} + \partial_{22} \dot{\mathbf{c}} \cdot \mathbf{n}_{\star}(\mathbf{w}) \\ \partial_{12} \mathbf{c} \cdot \dot{\mathbf{n}}_{\star}(\mathbf{w}) + \partial_{12} \mathbf{v} \cdot \dot{\mathbf{n}} + \partial_{12} \dot{\mathbf{c}} \cdot \mathbf{n}_{\star}(\mathbf{w}) \end{pmatrix}.$$

We take for the computations above $\dot{\mathbf{v}} = \mathbf{0}$ as the basis functions are only space dependent and the time derivative of the first variation of the normal as

$$\dot{\mathbf{n}}_{\star}(\mathbf{w}) = -(\mathbf{n} \cdot \mathbf{m}_w) \dot{\mathbf{n}} + \dot{\mathbf{m}}_w - (\mathbf{n} \cdot \dot{\mathbf{m}}_w) \mathbf{n} - (\mathbf{n}_{\star}(\mathbf{w}) \cdot \mathbf{m}_t) \mathbf{n} - (\mathbf{n} \cdot \mathbf{m}_t) \mathbf{n}_{\star}(\mathbf{w}).$$

Chapter 4

Isogeometric Analysis on Shells

In order to solve the structural models on computers, finite elements are used since the 1950s. The finite element method [Cia78, Hug00, Bra01, ZTZ05] generally uses polynomials as basis functions to approximate an infinite dimensional function space. For this, the geometry is first discretized into a finite number of parts called the *finite elements* on which the basis functions reside. A quadrature formula is applied to evaluate the terms in the weak formulation. This technique usually involves basis functions that are mostly only C^0 continuous at the element boundaries. A higher continuity is difficult to achieve.

Due to this continuity restriction, the *Reissner-Mindlin* shell model became more popular than the *Kirchhoff-Love* model, despite the latter having been invented almost a century earlier. Also, in many of the shell applications, the structure under investigation is a thin shell. In such a case, Reissner-Mindlin shells are known to produce shear-locking effects.

After the advent of the isogeometric analysis, which is one of the current hot topics in the field of finite elements, the use of higher order shell theory such as the Kirchhoff-Love shells is made possible. We study in this chapter the fundamentals of the NURBS basis functions, that are often used in the CAD to define the geometry. We then study the concept of isogeometric analysis, that uses the same NURBS to discretize the unknown vector field. We mention the numerous benefits of such an approach and apply it to the Kirchhoff-Love shell theory. Moreover, we analyze the approach of the *bending strip method*, developed by Kiendl et al. [KBH⁺10], to join the multipatch shells and observe its pros and cons.

4.1 Geometry Fundamentals

In this section, we study the origin of NURBS. They are generalizations of B-Splines, which themselves arise from Bézier curves. Firstly, we briefly look at the Bézier curves, followed by a study of B-Spline geometries. Finally, we construct the NURBS from B-Splines. Moreover, we study the properties of the NURBS surfaces that are important for their use in isogeometric shell analysis, namely the concepts of continuity, refinement and point projection. For more details on the topics, refer to the following books [PT97, Rog01] and the references therein.

It is common to use polynomials such as Bernstein polynomials, B-Splines, etc as basis functions to define various geometries. Also, the polynomials as basis functions find their application in the finite element methods. But, there are a number of important curves and surfaces that cannot be represented exactly using polynomials, e.g. conic sections such as circles, ellipses, hyperbolas, cylinders, etc. It is illustrated below by taking a circle as an

example.

Lemma 4.1. *A circle cannot be represented by using polynomials.*

Proof. The proof comes from [PT97]. Without loss of generality, a circle of radius r with center at origin is taken, such that $x^2 + y^2 - r^2 = 0$ in xy plane and the statement is proved by contradiction.

A constant i.e. a polynomial representation of degree 0 obviously does not represent a circle. Therefore, we assume a polynomial representation with $n \geq 1$ of the form

$$\begin{aligned} x(u) &= \sum_{i=0}^n a_i u^i = a_0 + a_1 u + \cdots + a_n u^n, \\ y(u) &= \sum_{i=0}^n b_i u^i = b_0 + b_1 u + \cdots + b_n u^n, \end{aligned}$$

Substituting in the circle equation gives

$$\begin{aligned} 0 &= (a_0 + a_1 u + \cdots + a_n u^n)^2 + (b_0 + b_1 u + \cdots + b_n u^n)^2 - r^2 \\ &= (a_0^2 + b_0^2 - r^2) + 2(a_0 a_1 + b_0 b_1)u + (a_1^2 + 2a_0 a_2 + b_1^2 + 2b_0 b_2)u^2 \\ &\quad + \cdots + (a_{n-1}^2 + 2a_{n-2} a_n + b_{n-1}^2 + 2b_{n-2} b_n)u^{2n-2} \\ &\quad + 2(a_n a_{n-1} + b_n b_{n-1})u^{2n-1} + (a_n^2 + b_n^2)u^{2n}. \end{aligned}$$

This equation must hold true for all u , which implies that all the coefficients are zero. Starting with the highest degree and working down, it is shown in n steps that all $a_i = 0$ and $b_i = 0$ for $1 \leq i \leq n$.

Step

1. $a_n^2 + b_n^2 = 0$ implies $a_n = b_n = 0$.
2. $a_{n-1}^2 + 2a_{n-2}a_n + b_{n-1}^2 + 2b_{n-2}b_n = 0$ and Step 1 imply that $a_{n-1}^2 + b_{n-1}^2 = 0$, which implies that $a_{n-1} = b_{n-1} = 0$.
- ⋮
- n. $a_1^2 + 2a_0 a_2 + b_1^2 + 2b_0 b_2 = 0$ and Step n-1 imply that $a_1^2 + b_1^2 = 0$, which implies that $a_1 = b_1 = 0$.

Thus $x(u) = a_0$ and $y(u) = b_0$, which is an obvious contradiction to the polynomial representation with degree $n \geq 1$. \square

Remark 4.1. *All conic curves, including the circle, can be represented using rational functions, which are defined as the ratio of two polynomials. [PT97]*

Taking cue from Lemma 4.1 and Remark 4.1, we create geometries later in this section by using rational functions to define the basis, so that a wide variety of geometries can be represented. But, we start from the polynomial setting and define next the Bézier curves.

4.1.1 Bézier Curves

An n th degree *Bézier curve* is defined by a linear combination

$$\mathbf{C}(\xi) = \sum_{i=0}^n B_{i,n}(\xi) \mathbf{P}_i, \quad 0 \leq \xi \leq 1, \quad (4.1)$$

of the basis functions $B_{i,n}(\xi) \in \mathbb{R}$, which are the classical n th degree *Bernstein polynomials* [Ber12, Lor86] given by

$$B_{i,n}(\xi) = \frac{n!}{i!(n-i)!} \xi^i (1-\xi)^{n-i}, \quad (4.2)$$

and the *control points* $\mathbf{P}_i \in \mathbb{R}^3$, $i = 0, 1, \dots, n$. For a rigorous and complete treatment of Bézier curves, refer to [Bez72, Far96]. An example of a Bézier curve of degree 5 is shown in Fig. 4.1. The Bézier curve is an approximating curve that interpolates only the two end control points $\mathbf{P}_0, \mathbf{P}_n$, giving a smooth curve generated from the control net of the various control points. The control net is the linear connection of the control points. The degree n of the basis functions is directly related to the $n + 1$ number of control points.

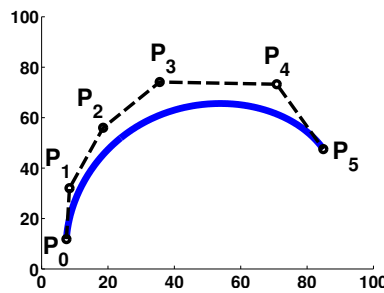


Figure 4.1: A Bézier curve of degree $n = 5$

Since the Bézier curves comprise of just one segment, representing curves with them is often disadvantageous due to their various shortcomings. Firstly, the support of the basis functions is not local. This means that changing any control point affects the whole shape instead of a local region of the curve. Moreover, with increasing number of control points, the degree of the basis functions increases, making the curves numerically highly unstable. These shortcomings can however be avoided by using piecewise polynomial functions called *splines*.

4.1.2 B-Splines

B-Splines or *basis splines* are piecewise polynomials of degree p that are connected in so-called *knots*. Let us define $(\xi_0, \dots, \xi_m) \in \mathbb{R}^{m+1}$ as the *knot vector*, consisting of non-decreasing real numbers. If all the knots are equally spaced, the knot vector is called *uniform*. If the same knot appears k times, then it is said to have a *multiplicity* k and the splines are C^{p-k} continuous at this knot. For a single knot, the spline has C^{p-1} continuity at the knot. This differs from the classical finite element basis functions where mostly C^0 continuity dominates. We assume that the first and the last knot have multiplicity $p + 1$, i.e. the splines are C^{-1} or discontinuous at these knots. Also, such knot vectors are called *open*. For this thesis, all knot vectors are taken to be open, unless otherwise stated. The interval between two adjacent unique knots is called a *knot span*.

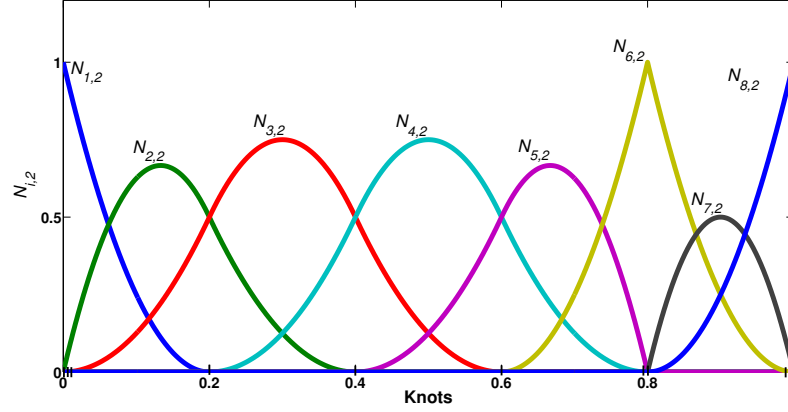


Figure 4.2: B-Splines of degree 2 with an open knot vector $\xi = (0, 0, 0, 0.2, 0.4, 0.6, 0.8, 0.8, 1, 1, 1)$

The i -th B-Spline basis function $N_{i,p}$ with degree p is defined recursively, with Cox, deBoor formulas [Cox72, dB72], as

$$N_{i,0}(\xi) = \begin{cases} 1 & \text{if } \xi_i \leq \xi < \xi_{i+1}, \\ 0 & \text{otherwise;} \end{cases} \quad (4.3a)$$

$$N_{i,p}(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi). \quad (4.3b)$$

Note that the quotient $0/0$ is defined to be zero.

We define an open knot vector as

$$\xi = (\underbrace{a, \dots, a}_{p+1}, \xi_{p+1}, \dots, \xi_{m-p-1}, \underbrace{b, \dots, b}_{p+1}). \quad (4.4)$$

A knot vector

$$(\underbrace{0, \dots, 0}_{p+1}, \underbrace{1, \dots, 1}_{p+1}) \quad (4.5)$$

yields the p degree Bernstein polynomials from (4.2).

A B-Spline basis function is C^∞ continuous inside a knot span and C^{p-k} continuous at knots with multiplicity k . As an example, in Fig. 4.2 we have plotted the degree 2 B-Spline basis functions $N_{i,2}$ for an open knot vector $\xi = (0, 0, 0, 0.2, 0.4, 0.6, 0.8, 0.8, 1, 1, 1)$. The use of a multiple knot at 0.8 reduces the degree to C^{p-2} , i.e. C^0 . This feature of the B-Splines enables the modeling of kinks or other low continuity curve features.

We list some important properties of the B-Spline basis functions.

- *Local support:* $N_{i,p}(\xi) = 0$, if ξ lies outside the interval $[\xi_i, \xi_{i+p+1})$
- *Non-negativity:* $N_{i,p}(\xi) \geq 0$, for all i, p, ξ
- *Partition of unity:* For an arbitrary knot span $[\xi_i, \xi_{i+1})$, $\sum_{j=i-p}^i N_{j,p}(\xi) = 1$, for all $\xi \in [\xi_i, \xi_{i+1})$

More properties and a thorough analysis on this topic is found in [PT97].

4.1.2.1 B-Spline Curve

A *B-Spline curve* of degree p is defined as the linear combination of the degree p B-Spline basis functions constructed using (4.3) and the control points $\mathbf{P}_i \in \mathbb{R}^3$

$$\mathbf{C}(\xi) = \sum_{i=0}^n N_{i,p}(\xi) \mathbf{P}_i, \quad \xi \in [a, b], \quad (4.6)$$

such that $a, b \in \mathbb{R}$, $n = m - p - 1$, defined on the open knot vector ξ from (4.4), see [Far96, PT97].

The curve is interpolated at the first and the last control point, such that $\mathbf{C}(a) = \mathbf{P}_0$ and $\mathbf{C}(b) = \mathbf{P}_n$. When $n = p$ and the knot vector is as in (4.5), the B-Spline curve in (4.6) represents a Bézier curve. Hence, the Bézier curve is a special case of the B-Spline curve. Another important property of the B-Spline curve relates to its first derivative at the end points, which is important when joining the curves smoothly. In [PT97], the first derivatives at the end points of a B-Spline curve are given as

$$\begin{aligned} \mathbf{C}'(a) &= \frac{p}{\xi_{p+1}} (\mathbf{P}_1 - \mathbf{P}_0), \\ \mathbf{C}'(b) &= \frac{p}{1 - \xi_{m-p-1}} (\mathbf{P}_n - \mathbf{P}_{n-1}). \end{aligned} \quad (4.7)$$

This suggests that the curve is tangential to the control polygon at its start and end. A B-Spline curve also has the property of affine invariance, which means that a rigid motion applied to the curve is equivalent to the rigid motion applied to the control points directly. This aspect is useful for our applications in the multibody dynamics, where the joint between the elastic body and another body might impart a rigid motion to the elastic body. The continuity of the curves follow the continuity of the corresponding B-Splines.

4.1.2.2 B-Spline Surface

A *B-Spline surface* is obtained by taking a bidirectional net of control points, two knot vectors ξ_1, ξ_2 , and the tensor product of the univariate B-Spline functions of degree p and q in the two parametric directions respectively. Refer to [Far96, PT97].

$$\mathbf{S}(\vartheta_1, \vartheta_2) = \sum_{i=0}^{n_1} \sum_{j=0}^{n_2} N_{i,p}(\vartheta_1) N_{j,q}(\vartheta_2) \mathbf{P}_{ij}, \quad \vartheta_1 \in [a_1, b_1], \vartheta_2 \in [a_2, b_2]. \quad (4.8)$$

The surface is interpolated at the four corner points, i.e. $\mathbf{S}(a_1, a_2) = \mathbf{P}_{00}$, $\mathbf{S}(a_1, b_2) = \mathbf{P}_{0n_2}$, $\mathbf{S}(b_1, a_2) = \mathbf{P}_{n_10}$, $\mathbf{S}(b_1, b_2) = \mathbf{P}_{n_1n_2}$. Analogous to the curves, the slopes along a boundary of the surface is defined by the first two rows of control points from the boundary. Moreover, the surface is now divided into *elements*, which are defined by the products of the knot spans in the two directions. These elements form the computational mesh within the framework of isogeometric analysis as introduced in Chapter 4.

A B-Spline solid can also be generated in a similar fashion as the surface. This time a tridirectional net of control points, three knot vectors and the tensor product of the univariate B-Spline functions in three-parametric directions are required. We do not discuss it further, since our focus in this thesis remains on shell structures, which can be constructed using surfaces.

4.1.3 NURBS

The *Non-Uniform Rational B-Splines* or the *NURBS* refer to rational functions created from polynomial B-Splines, which are defined on a non-uniform knot vector. For a detailed treatment of NURBS, refer to [Ver75, Pie91, PT97].

In one dimension, the NURBS of degree p are given by

$$R_{i,p}(\xi) = \frac{w_i N_{i,p}(\xi)}{\sum_{j \in \mathcal{J}} w_j N_{j,p}(\xi)} \quad (4.9)$$

with B-Splines $N_{i,p}$ and weights $w_i \in \mathbb{R}^+$. The index set $\mathcal{J} = \{0, 1, \dots, n = m - p - 1\}$ is used for numbering the NURBS.

By specifying single or multiple knots, one may change the smoothness of the NURBS in a similar fashion as for B-Splines. With single knots except at the ends, the global smoothness is \mathcal{C}^0 for $p = 1$, \mathcal{C}^1 for $p = 2$, and \mathcal{C}^2 for $p = 3$. Moreover, the support grows with the degree p and comprises of $p + 1$ knot spans, same as for B-Splines. Hence, the NURBS can also be used to model shapes with sharp edges or kinks. The main advantage of using such rational functions with weights is the possibility of representing conic intersections such as ellipses and hyperbolas exactly, see Remark 4.1. This makes them versatile and a well-known standard in the design community.

In the special setting where all weights w_i are set equal in (4.9), the NURBS are equivalent to B-Splines due to the *partition of unity* property of the latter. Due to the use of B-Splines as the building blocks, most of their properties such as local support, partition of unity, non-negativity, affine invariance, amongst others carry forward to NURBS.

4.1.3.1 NURBS Curve

Analogous to B-Splines, a *NURBS curve* of degree p is defined as the linear combination of the NURBS basis functions $R_{i,p}$ and the control points $\mathbf{P}_i \in \mathbb{R}^3$

$$\mathbf{C}(\xi) = \sum_{i \in \mathcal{I}} R_{i,p}(\xi) \mathbf{P}_i, \quad \xi \in [a, b], \quad (4.10)$$

with $a, b \in \mathbb{R}$, the index set $\mathcal{I} = \{0, 1, \dots, n = m - p - 1\}$, defined on the open knot vector $\boldsymbol{\xi}$ from (4.4).

A NURBS curve is also interpolated at the first and the last control points, such that $\mathbf{C}(a) = \mathbf{P}_0$ and $\mathbf{C}(b) = \mathbf{P}_n$. The first derivatives at the end points of a NURBS curve are given by

$$\begin{aligned} \mathbf{C}'(a) &= \frac{p}{\xi_{p+1}} \frac{w_1}{w_0} (\mathbf{P}_1 - \mathbf{P}_0), \\ \mathbf{C}'(b) &= \frac{p}{1 - \xi_{m-p-1}} \frac{w_{n-1}}{w_n} (\mathbf{P}_n - \mathbf{P}_{n-1}). \end{aligned} \quad (4.11)$$

This suggests that the rational curve is also tangential to the control polygon at its start and end.

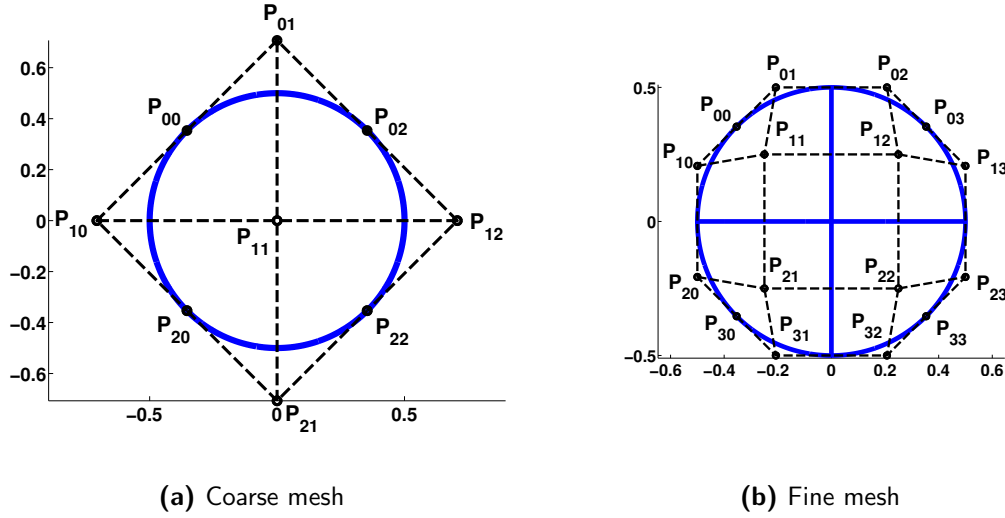


Figure 4.3: Disc surface and the control points generated with ISOGAT [VHS10]

4.1.3.2 NURBS Surface

Bivariate NURBS with degrees p and q in the two parametric directions are

$$R_{ij,pq}(\vartheta_1, \vartheta_2) = \frac{w_{ij} N_{i,p}(\vartheta_1) N_{j,q}(\vartheta_2)}{\sum_{k \in \mathcal{I}} \sum_{\ell \in \mathcal{J}} w_{k\ell} N_{k,p}(\vartheta_1) N_{\ell,q}(\vartheta_2)}, \quad (4.12)$$

with the index sets $\mathcal{I} = \{0, 1, \dots, n_1 = m_1 - p - 1\}$, $\mathcal{J} = \{0, 1, \dots, n_2 = m_2 - q - 1\}$. Note that the singularity of the denominator lies outside of the support of R_{ij} and is thus not relevant. If all weights are equal to one, the partition of unity of the underlying B-Splines gives $\sum_{k \in \mathcal{I}} \sum_{\ell \in \mathcal{J}} N_{k,p}(\vartheta_1) N_{\ell,q}(\vartheta_2) = 1$, and hence the NURBS are identical to B-Splines in this case.

A *NURBS surface* is a linear combination of the bivariate NURBS basis functions $R_{ij,pq}$ and the control points $\mathbf{P}_{ij} \in \mathbb{R}^3$

$$\mathbf{S}(\vartheta_1, \vartheta_2) = \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} R_{ij,pq}(\vartheta_1, \vartheta_2) \mathbf{P}_{ij}, \quad \vartheta_1 \in [a_1, b_1], \vartheta_2 \in [a_2, b_2]. \quad (4.13)$$

The surface interpolates the four corner points.

An example of a NURBS surface is a disc as in Fig. 4.3a with the knot vectors in the two parametric directions $\xi_j = (0, 0, 0, 1, 1, 1)$, $j = 1, 2$ and the weights $w_{00}, w_{02}, w_{11}, w_{20}, w_{22} = 1$ and $w_{01}, w_{10}, w_{12}, w_{21} = 1/\sqrt{2}$. Also in Fig. 4.3b, we observe a refined parametrization of the same geometry. This is achieved with the refinement techniques, which are discussed in Section 4.1.4. The knot vectors in this case are $\xi_j = (0, 0, 0, 0.5, 1, 1, 1)$, $j = 1, 2$ and the weights are $w_{00}, w_{03}, w_{30}, w_{33}, w_{22} = 1$ and $w_{01}, w_{02}, w_{10}, w_{11}, w_{12}, w_{13}, w_{20}, w_{21}, w_{22}, w_{23}, w_{31}, w_{32} = (1 + 1/\sqrt{2})/2$. The four corner control points are seen to lie on the surface for both the geometry descriptions due to the corner point interpolation property of the NURBS surface.

Analogously, the slopes along a boundary of the surface are defined by the first two rows of control points from the boundary. The other properties of the surface follow from the B-Spline surfaces. NURBS solids are also generated in a similar manner.

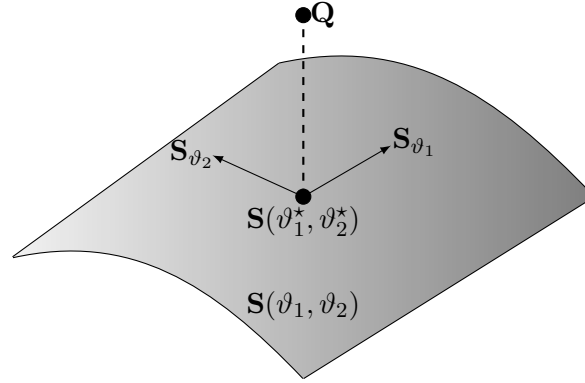


Figure 4.4: Parameter search using point projection

It is to be noted that the continuity in each parametric direction is determined by the knot multiplicities of the corresponding knot vectors and the weights have no influence on this. Therefore the tensor product like structure results in isoparametric lines, which possess the same continuity corresponding to the multiplicity of the knot along which they are defined.

4.1.4 Geometry Refinement

In this brief section, we suggest methods that are used to refine the NURBS geometry in such a way that more flexibility and control is added for their better handling. The two main algorithms to perform a refinement on the NURBS geometry are the *knot refinement* [Boe85] and *degree elevation* [Pra84]. The underlying idea is to project the existing NURBS geometry in a higher dimensional space without changing the original geometry at all. The knot refinement algorithm refers to an efficient implementation of the knot insertion method [Boe80], where multiple knots are introduced simultaneously in the knot vector. During knot refinement, the same number of control points as the additional knots are introduced. In Fig. 4.3b, we add one knot in each direction, leading to $4 \times 4 = 16$ control points instead of $3 \times 3 = 9$ in Fig. 4.3a in case of no refinement. If any knot is repeated, then the continuity at that knot is reduced by one for each repetition. In the degree elevation, the number of knot spans remains unchanged and the degree of the basis functions is increased. Moreover, the existing knots are repeated in process in order to preserve the continuity at these knots. Efficient algorithms to implement these methods are suggested in the book of Piegl and Tiller [PT97].

4.1.5 Point Projection

To implement point loads on NURBS surfaces, it is important to identify the parametric coordinates of the point \mathbf{Q} at which the loads are applied. Given a point $\mathbf{Q} = (x, y, z)^T$, assumed to lie on the NURBS surface $\mathbf{S}(\vartheta_1, \vartheta_2)$, the goal is to find the parametrization $(\vartheta_1^*, \vartheta_2^*)$, such that $\mathbf{S}(\vartheta_1^*, \vartheta_2^*) = \mathbf{Q}$. This is achieved by applying the concept of *point projection* [PT97], where the idea is to use Newton iteration to minimize the distance between the point \mathbf{Q} and the surface $\mathbf{S}(\vartheta_1^*, \vartheta_2^*)$ below a specified tolerance ϵ_1 . The sketch of the point projection is shown in Fig. 4.4. We drop (*) from the unknown parameters for the following equations for the ease of notation.

Let us define a vector function

$$\mathbf{r}(\vartheta_1, \vartheta_2) = \mathbf{S}(\vartheta_1, \vartheta_2) - \mathbf{Q}. \quad (4.14)$$

To minimize the distance, we solve two scalar dot product equations

$$\begin{aligned} f(\vartheta_1, \vartheta_2) &:= \mathbf{r}(\vartheta_1, \vartheta_2) \cdot \mathbf{S}_{\vartheta_1}(\vartheta_1, \vartheta_2) = 0, \\ g(\vartheta_1, \vartheta_2) &:= \mathbf{r}(\vartheta_1, \vartheta_2) \cdot \mathbf{S}_{\vartheta_2}(\vartheta_1, \vartheta_2) = 0, \end{aligned} \quad (4.15)$$

where \mathbf{S}_{ϑ_j} , $j = 1, 2$ are the tangents to the surface \mathbf{S} in the two parametric directions. Similarly, the subscripts ϑ_j , $j = 1, 2$ in this section refer to the derivatives of the functions with respect to the two parametric directions.

We solve (4.15) with the Newton method. Let in each iteration step i , the update be $\boldsymbol{\delta}_i$ and the residual be $\boldsymbol{\kappa}_i$ such that

$$\boldsymbol{\delta}_i = \begin{pmatrix} \Delta\vartheta_{1,i} \\ \Delta\vartheta_{2,i} \end{pmatrix} = \begin{pmatrix} \vartheta_{1,i+1} - \vartheta_{1,i} \\ \vartheta_{2,i+1} - \vartheta_{2,i} \end{pmatrix}, \quad \boldsymbol{\kappa}_i = - \begin{pmatrix} f(\vartheta_{1,i}, \vartheta_{2,i}) \\ g(\vartheta_{1,i}, \vartheta_{2,i}) \end{pmatrix}$$

and the Jacobian be

$$\mathbf{J}_i = \begin{pmatrix} f_{\vartheta_1} & f_{\vartheta_2} \\ g_{\vartheta_1} & g_{\vartheta_2} \end{pmatrix} = \begin{pmatrix} |\mathbf{S}_{\vartheta_1}|^2 + \mathbf{r} \cdot \mathbf{S}_{\vartheta_1\vartheta_1} & \mathbf{S}_{\vartheta_1} \cdot \mathbf{S}_{\vartheta_2} + \mathbf{r} \cdot \mathbf{S}_{\vartheta_1\vartheta_2} \\ \mathbf{S}_{\vartheta_1} \cdot \mathbf{S}_{\vartheta_2} + \mathbf{r} \cdot \mathbf{S}_{\vartheta_2\vartheta_1} & |\mathbf{S}_{\vartheta_2}|^2 + \mathbf{r} \cdot \mathbf{S}_{\vartheta_2\vartheta_2} \end{pmatrix}.$$

We solve in each iteration step i the following linear system

$$\mathbf{J}_i \boldsymbol{\delta}_i = \boldsymbol{\kappa}_i$$

for the unknown $\boldsymbol{\delta}_i$ and then update the unknowns using $\vartheta_{j,i+1} = \Delta\vartheta_{j,i} + \vartheta_{j,i}$, $j = 1, 2$. The iterations are performed until one of the following convergence criteria is met

- distance below the tolerance, i.e. the points coincide

$$|\mathbf{S}(\vartheta_{1,i}, \vartheta_{2,i}) - \mathbf{Q}| \leq \epsilon_1,$$

- closest point possible to the surface is found, i.e. zero cosine

$$\frac{|\mathbf{S}_{\vartheta_j}(\vartheta_{1,i}, \vartheta_{2,i}) \cdot (\mathbf{S}(\vartheta_{1,i}, \vartheta_{2,i}) - \mathbf{Q})|}{|\mathbf{S}_{\vartheta_j}(\vartheta_{1,i}, \vartheta_{2,i})| |\mathbf{S}(\vartheta_{1,i}, \vartheta_{2,i}) - \mathbf{Q}|} \leq \epsilon_2, \quad j = 1, 2,$$

- parameters do not change significantly

$$|(\vartheta_{1,i+1} - \vartheta_{1,i})\mathbf{S}_{\vartheta_1}(\vartheta_{1,i}, \vartheta_{2,i}) + (\vartheta_{2,i+1} - \vartheta_{2,i})\mathbf{S}_{\vartheta_2}(\vartheta_{1,i}, \vartheta_{2,i})| \leq \epsilon_1,$$

for small tolerances ϵ_1, ϵ_2 . It must also be made sure that the parameters stay in the corresponding range of the knot values. After the convergence, the final value of $(\vartheta_1^*, \vartheta_2^*)$ refers to the point on the surface closest to \mathbf{Q} . This concept is applied to calculate pointwise external forces on the shell structures modeled as NURBS surfaces in Section 3.2.2.

4.2 Isogeometric Analysis (IGA) using NURBS

The basic idea of *isogeometric analysis* [HCB05, CHB09], also referred in this work as *IGA*, is to employ the *isoparametric concept* [Hug00], i.e. to use the same set of functions both for representing the geometry of a structure and for approximating the unknowns such as displacements. Usually *Computer Aided Design* (CAD) [RA90, Far96, FHK02] geometries use splines, sub-division surfaces, etc. as the underlying basis functions. However, for analysis with finite elements, mostly linear piecewise polynomials are used. With isogeometric analysis the idea is to employ the CAD basis functions such as B-Splines, NURBS, T-Splines, sub-division surfaces, etc. also for analysis.

A goal is to be geometrically exact no matter how coarse the discretization. This in turn also greatly simplifies mesh refinement, since after the initial mesh is created from the CAD geometry, there is no need to communicate with it. This is because the geometry stays preserved throughout the meshing. This can lead to immense saving in time spent in meshing, which is around 80% of the total analysis time of a product [HCB05].

The basis functions for analysis are required to show properties such as linear independence and partition of unity. Many of the CAD basis functions fulfill them. Since engineering shapes are typically constructed by means of CAD software and represented in terms of surface descriptions based on Non-Uniform Rational B-Splines or NURBS, refer to Section 4.1.3, the isogeometric paradigm requires that the numerical approximation of unknown, e.g. the displacement \mathbf{u}_h of \mathbf{u} , is in the same NURBS space. The coefficients of the basis functions are the control variables in case of geometry and degrees of freedom (dofs) in case of the unknowns.

4.2.1 Elements

The knot spans defining the NURBS patch subdivide the domain into *elements*. Inside each element, quadrature rule is applied for the numerical integration of the terms of the weak formulation. A *mesh* for a NURBS patch is defined by the product of knot vectors. For example, in case of surfaces, a mesh is given by $\boldsymbol{\xi}_1 \times \boldsymbol{\xi}_2$. In Fig. 4.3b, we notice that the disc surface is divided into four quadrants, each of which represents one NURBS element.

An important difference to the standard finite element method lies in the choice of the basis functions. The support of the NURBS of degree p usually is not limited to one element but extends over $p+1$ elements. Due to this reason, the mass and stiffness matrices although still sparse, are somewhat denser than for the finite elements. As an example, in Fig. 4.2, for the one dimensional, we observe that for degree 2, the basis functions extend over 3 elements or knot spans.

4.2.2 Mesh Refinement

In order to control the quality of the approximation, it is necessary to have some kind of refinement strategies as one has for the finite element method. For IGA, the analogue of the *h-refinement* is the *knot refinement* and that of the *p-refinement* is the *degree elevation* of the NURBS. For details on knot refinement and degree elevation, see Section 4.1.4. An example of knot refinement is presented in Fig. 4.3. The original surface consists of the quadratic NURBS. A knot insertion step is performed on the coarse discretization from Fig. 4.3a with knot vectors $\boldsymbol{\xi}_j = (0, 0, 0, 1, 1, 1)$, leading to $\boldsymbol{\xi}_j = (0, 0, 0, 0.5, 1, 1, 1)$, $j = 1, 2$, the new control

points and basis functions. This process enriches the solution space by adding more basis functions of the same degree while leaving the disc surface unchanged.

One can also perform a *k-refinement* alternative by applying a combination of both knot refinement and degree elevation. Important in this case is to perform first the degree elevation and then the knot refinement. This is because with the degree elevation from p to q , $p < q$, the NURBS degree is increased to q . Inserting a knot between a knot span will lead to a continuity of C^{q-1} at this knot. If instead the process is reversed, at the new knot, the continuity will be lesser, i.e. C^{p-1} . This phenomenon and its consequences are well explained in [HCB05].

In case of NURBS curves, a local refinement is possible, since a knot can be added anywhere. This is, however, not true for NURBS surfaces, since they have tensor product like structures. Thus, whenever a new knot is added in one direction, it refines at all points in the other direction and cannot be restricted to a particular region in the other direction. This adds many more dofs than required and is inefficient. This problem can however be tackled by the use of alternative basis functions, for instance, T-Splines [SLSH12], hierarchical B-Splines [VGJS11], LR B-Splines [JKD14].

4.2.3 Dirichlet Boundary Conditions

NURBS do not provide a nodal basis. As a result, the element nodes are in general not interpolated by the control points or the dofs. This makes the application of the Dirichlet boundary conditions (3.6) not so straight forward as with the finite element method, where the nodal displacement variables on the boundary are directly put equal to the prescribed displacement.

For NURBS based IGA, in case of homogeneous boundary condition $\mathbf{u} = \mathbf{0}$ on Γ_0 , eliminating all dofs on the boundary of the NURBS surface and the corresponding entries in the stiffness matrix and the load vector works. This is because the boundary curve in this case does not move. On the other hand for the inhomogeneous boundary condition $\mathbf{u} = \mathbf{u}_0 \neq \mathbf{0}$ on Γ_0 , one can include the Dirichlet boundary as constraints, following the approach from Section 3.3.

The system generated on including the inhomogeneous boundary condition leads to a saddle point structure of the form (4.27), which must satisfy the LBB condition [Bre90] in order to be well defined.

4.2.4 Advantages

We now summarize the various benefits of using NURBS based IGA over the traditional finite elements.

- Working on the exact geometry even at the coarsest discretization. This is beneficial for structures such as shells, which are quite sensitive to geometric imperfections.
- Automatic meshing for NURBS based CAD models.
- Easier mesh refinement, without communicating with CAD geometry in each refinement step.
- Higher regularity of C^{p-1} across elements in the absence of multiple knots. This is needed especially for working with Kirchhoff-Love shells.

- Rigid motion leads to zero strains, due to the affine invariance of the NURBS. It is a suitable property for the use in flexible multibody systems.

It must be noted that this method also presents a few challenges on the way. The nice tri-diagonal structure of the system matrices is lost due to the larger support of the basis functions, leading to comparatively denser system matrices. The application of the boundary condition becomes a bit more involved. Moreover, due to the tensor product like structure, local refinement is not immediately possible with NURBS.

Throughout this thesis, NURBS are employed as basis functions. We consider a displacement based formulation, where the dofs are the displacements of the corresponding control points.

4.3 Application of IGA to Shells

We assume that the geometry of the undeformed shell mid-surface is given as a linear combination of bivariate NURBS $R_{kl,pq}$ from (4.12) and control points $\mathbf{P}_{kl} \in \mathbb{R}^3$,

$$\mathbf{c}_0(\vartheta_1, \vartheta_2) = \sum_{k \in \mathcal{I}} \sum_{\ell \in \mathcal{J}} R_{kl,pq}(\vartheta_1, \vartheta_2) \mathbf{P}_{kl}. \quad (4.16)$$

The main advantage of using rational functions with weights w_{kl} over B-Splines is the possibility of representing conic intersections such as ellipses and hyperbolas exactly. An example is a circular disc, see Fig. 4.5. To model complicated geometries such as shells with cut-out, multipatches (see Fig. 4.6) or trimming techniques [SWB12] should be employed. The latter is a topic of current research, whereas multipatches will be discussed later on. Triangular elements are possible but typically avoided.

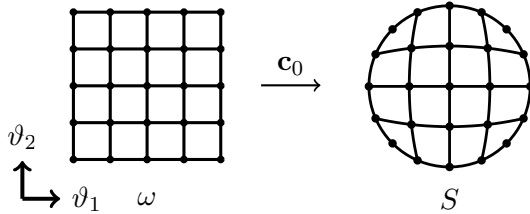


Figure 4.5: Circular disc

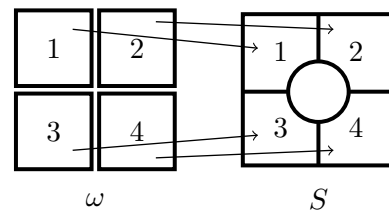


Figure 4.6: Shell cut-out using multipatch

In the stationary setting to study the Kirchhoff-Love shells, the Hamilton's principle (3.30) in Section 3.2.3 assists in computing the displacement $\mathbf{u}(\vartheta_1, \vartheta_2)$ which minimizes the energy functional, i.e.

$$\mathbf{u} = \underset{\mathbf{w}}{\operatorname{argmin}} U(\mathbf{w}), \quad U(\mathbf{w}) = U_{\text{int}}(\mathbf{w}) - U_{\text{ext}}(\mathbf{w}). \quad (4.17)$$

This minimization problem is formulated in the infinite dimensional function space $(H^2(\omega))^3$, which is a Sobolev space that includes the \mathcal{C}^1 -functions. In the instationary case, the displacement $\mathbf{u}(\vartheta_1, \vartheta_2, t)$ is characterized as a stationary point of the Lagrange function and leads to the weak form of shell elasticity problem (3.35).

For the numerical treatment of the Kirchhoff-Love shells using IGA, we apply a Galerkin projection and approximate the displacement field in a finite dimensional subspace V_h of $(H^2(\omega))^3$. For standard finite elements, one selects V_h to consist of piecewise polynomials

that are glued together such that global \mathcal{C}^1 -smoothness holds. We use instead NURBS, following the approach of isogeometric analysis as introduced in Section 4.2.

Galerkin Projection with NURBS. We choose

$$V_h \subset \text{span} \{ \Psi_i = \mathbf{e}_j R_{k\ell,pq}, j \in \{1, 2, 3\}, k \in \mathcal{I}, \ell \in \mathcal{J} \} \subset (H^2(\omega))^3. \quad (4.18)$$

Here, the scalar NURBS $R_{k\ell,pq}$ are extended to 3D space using the unit vectors $\mathbf{e}_j \in \mathbb{R}^3$. Due to the Dirichlet boundary conditions (3.6), the space V_h is actually only a subset of the NURBS space, and some of the basis functions Ψ_i in (4.18) are omitted since they must vanish. Assuming that n_{DOF} degrees of freedom remain after the Dirichlet boundary conditions have been taken into account, the displacement field \mathbf{u} from (3.7) is approximated by

$$\mathbf{u}_h(\vartheta_1, \vartheta_2, t) = \sum_{i=1}^{n_{\text{DOF}}} \Psi_i(\vartheta_1, \vartheta_2) d_i(t) \quad (4.19)$$

with elastic variables d_i . In general, these unknowns cannot be interpreted as nodal variables. They refer to the displacement of the control points and might not lie on the shell surface.

The numerical approximation of the total deformation (3.5) is given by

$$\Theta_h(\vartheta_1, \vartheta_2, \vartheta_3, t) = \mathbf{c}_h(\vartheta_1, \vartheta_2, t) + \vartheta_3 \mathbf{n}_h(\vartheta_1, \vartheta_2, t) \quad \text{where } \mathbf{n}_h = \frac{\partial_1 \mathbf{c}_h \times \partial_2 \mathbf{c}_h}{|\partial_1 \mathbf{c}_h \times \partial_2 \mathbf{c}_h|} \quad (4.20)$$

and

$$\mathbf{c}_h(\vartheta_1, \vartheta_2, t) = \mathbf{c}_0(\vartheta_1, \vartheta_2) + \mathbf{u}_h(\vartheta_1, \vartheta_2, t). \quad (4.21)$$

These approximations are inserted into the weak form (3.35), and the spatial integrals are computed by means of Gaussian quadrature rules. This assembly loop proceeds in the same way as a finite element method, with Cartesian products of NURBS knot spans playing the role of elements.

Summarizing, this discretization process results in the equations of motion

$$\mathbf{M} \ddot{\mathbf{d}} + \mathbf{f}(\mathbf{d}) = \mathbf{b}(t) \quad (4.22)$$

of the nonlinear shell, where the elastic force $\mathbf{f}(\mathbf{d}) := (a(\mathbf{u}_h, \Psi_j))_j$ and the external load vector $\mathbf{b}(t) := (b(\Psi_j))_j$, $j = 1, \dots, n_{\text{DOF}}$ from (3.35). They are solved by applying time integration schemes from Section 2.1.3. In each time integration step, we obtain a nonlinear system, which is solved using Newton method. In each Newton step, we compute the stiffness matrix

$$\mathbf{K}(\mathbf{d}) := \nabla \mathbf{f}(\mathbf{d}) = (j(\mathbf{u}_h, \Psi_i, \Psi_j))_{ij} \quad (4.23)$$

using (3.36). The mass matrix \mathbf{M} stays constant throughout the calculations and is computed as

$$M_{ij} = 2\epsilon \int_{\omega} \rho (\Psi_j \cdot \Psi_i) \nu_0 \, d(\vartheta_1, \vartheta_2), \quad (4.24)$$

where $i, j = 1, \dots, n_{\text{DOF}}$.

In case of additional Dirichlet constraints as discussed in Section 3.3, we approximate the Lagrange multiplier $\boldsymbol{\mu}$ on Γ_0 in (3.40a) by

$$\boldsymbol{\mu}_h(\vartheta_1, \vartheta_2, t) = \sum_{i=1}^{n_{\text{DOF}}^\alpha} \Phi_i^\alpha(\vartheta_1, \vartheta_2) \lambda_i(t) \quad (4.25)$$

with constraint variables λ_i and the basis functions Φ_i^α chosen from the univariate NURBS space

$$Q_h \subset \text{span} \{ \Phi_i^\alpha = \mathbf{e}_j R_{m,r}, j \in \{1, 2, 3\}, \text{ if } \alpha = 1, m \in \mathcal{I}, \text{ if } \alpha = 2, m \in \mathcal{J} \} \\ \subset (H^2(\omega))^3. \quad (4.26)$$

If Γ_0 changes along the first parametric direction ϑ_1 , then $\alpha = 1$ and if along the second parametric direction ϑ_2 , then $\alpha = 2$. Q_h is a subset of the univariate NURBS along the boundary. Some basis functions in (4.26) are removed due to the Dirichlet boundary conditions and only n_{DOF}^α are considered. The space Q_h must be chosen such that a suitable discretization for the Lagrange multipliers in (4.25) is found, which satisfies the LBB condition [Bre90].

Adding the effects of this additional discretization in system (3.40), we get

$$\mathbf{M}\ddot{\mathbf{d}} = \tilde{\mathbf{f}}(\mathbf{d}) - \mathbf{G}^T \boldsymbol{\lambda}, \quad (4.27a)$$

$$\mathbf{G}\mathbf{d} - \mathbf{g}_h = \mathbf{0}, \quad (4.27b)$$

where $\tilde{\mathbf{f}}(\mathbf{d}) = \mathbf{b}(t) - \mathbf{f}(\mathbf{d})$, $\mathbf{G} = (g(\Psi_i, \Phi_j^\alpha))_{ij}$, $\mathbf{g}_h = (\ell(\Phi_j^\alpha))_j$ and $\boldsymbol{\lambda}$ are the discretized Lagrange multipliers. The system (4.27) leads to the shell equations of motion in the constrained setting from the weak constraint form (3.40). It has a similar structure as the equations of the constrained mechanical system (2.37) from Section 2.2.

What are the advantages of such an isogeometric discretization? By construction, the motion of the discrete structure includes all rigid body modes and satisfies the usual patch test requirements at the element level [CHB09]. Moreover, the approximation properties of NURBS are much better than those of piecewise polynomials with lower global smoothness, and the higher smoothness obtained with NURBS is particularly important for the analysis of Kirchhoff-Love shells. Other pluses are that the mesh generation step becomes superfluous and that the refinement of the computational mesh simplifies to a knot insertion algorithm [PT97]. Last but not the least, design optimization also profits from an isogeometric approach [WFC08, CH09]. But there is also a price to be paid: The lack of a nodal basis necessitates an additional transformation step when exchanging forces or other physical data in a coupled simulation, and, moreover, the matrices are less sparse when compared to a finite element method. We refer to [KBLW09, DKS12, BHB⁺13, GDSV13] for a more elaborate discussion of isogeometric shell discretizations.

We finally remark that the aforementioned analysis of shells gives us displacement results. Using it for post processing, we can also calculate other physical quantities such as stresses which are important from an engineering point of view. Refer to Section 3.4 for a detailed discussion on stress recovery.

4.4 Multipatch Shells

In this section, we employ the isogeometric shell discretizations for multipatch cases. We begin by motivating the use of multipatch structures, followed by a study of the *bending strip method*, which is used to join multipatch isogeometric shells using a penalty approach [KBH⁺10]. We look at the pros and cons of the method and also suggest some ways to alleviate the underlying ill-conditioning issues at the end of this section and in the following chapter.

4.4.1 Motivation

Simple geometries are modeled using a single NURBS parametrization called a *patch* for the complete geometry. But quite often, the CAD geometries are very complex and require many patches for their modeling. Such geometries are called *multipatch structures*. Also, in order to tackle structures with special geometric regions such as the kinks, e.g. an L-shaped plate, multipatch structures might be used. In such cases, although modeling with a single patch might be possible, the representation might not fulfill the regularity requirements for the analysis.

For the analysis of a multipatch structure, each patch is first analyzed separately and then joined together taking care of the interface conditions. These may also be classified under the domain decomposition methods (DDM), where the original domain is split into many smaller domains due to various reasons such as when the domains have different physics, for contact problems, for parallel computing, etc. The study of DDM in the framework of the IGA is carried out in [ASWB14], where various techniques such as the penalty approach, Lagrange multiplier method, perturbed Lagrange method and the Nitsche methods for joining the domains are discussed. The IETI method proposed in [KPJT12] is also an elegant technique used to join domains with C^0 continuity in the framework of the NURBS based IGA.

To join multipatch shell structures, in addition to the C^0 continuity, the interface conditions must additionally preserve the angle of kink in case of the C^0 continuous patch interfaces and the G^1 continuity for smooth connections. The G^1 continuity requires the tangents to have the same direction at the intersection point. The additional conditions are needed in order to transfer the moments across the intersection and prevent the hinge like effect at the joint. A penalty approach to join isogeometric shell patches is studied in [KBH⁺10], where the violation of the joint properties is penalized. Another approach to handle isogeometric shell patch intersections is proposed in [BHB⁺13], where the thin and thick shell theories are blended together and each triple of control points at the joint is considered as a rigid body and its dynamics are introduced in the complete system. We follow a different approach and start in Section 4.4.2 with the penalty approach by Kiendl [KBH⁺10] and study its pros and cons. The problems such as the ill-conditioning and time step size restriction due to the penalty parameter dependence are identified and in Chapter 5, we develop penalty free formulations to overcome these issues.

4.4.2 Penalty Approach - Bending Strip Method

A technique to analyze multipatch isogeometric Kirchhoff-Love shell structures with the aforementioned penalty approach is the so-called *bending strip method* [KBH⁺10]. Using this method, the individual patches that are modeled as NURBS structures are joined together. The main idea is to introduce additional patches of fictitious material, namely the *bending strips*, at the patch intersections, where the patches are already joined with C^0 continuity. The bending strip is designed such that it only adds an additional bending stiffness in the direction transverse to the length of the strip, in order to penalize the change of angle at the intersection.

The design of the bending strip is explained with the help of a two-patch geometry in Fig. 4.7. The bending strip uses as control net the triples of control points at the patch interface, consisting of a shared control point and one on each side across the intersection. The parametric domain of the bending strip consists of one quadratic element in the direction

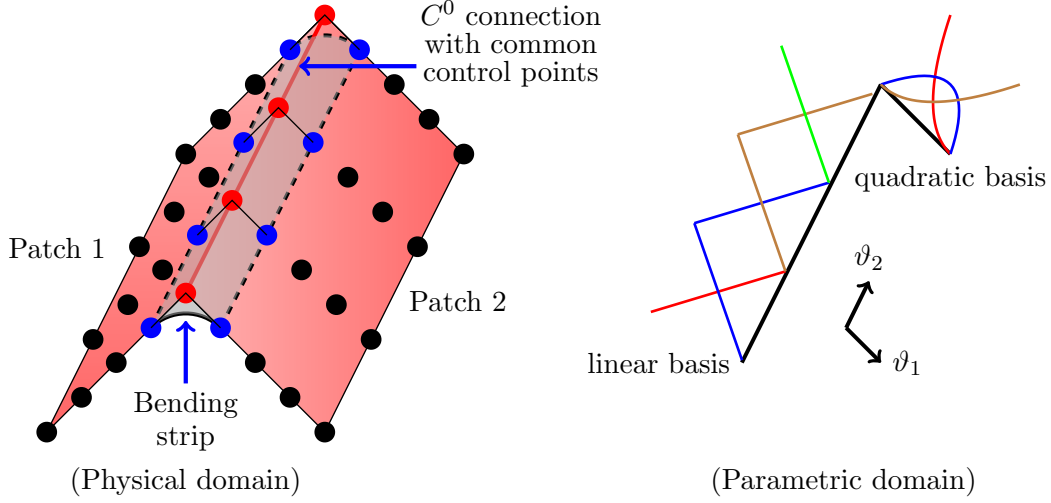


Figure 4.7: Bending strip design

transverse to the strip, and for simplicity and computational efficiency, of as many linear elements as necessary to accommodate all the control points along the length of the strip. The material is assumed to have zero mass, zero membrane stiffness; and a non-zero bending stiffness only in the direction transverse to the strip. This design permits the transfer of just the bending moments across the interface.

4.4.2.1 Material Law and Energy Expression

For our computations with the bending strip, we consider the various physical quantities with respect to the Cartesian coordinates as in Remark 3.5. The bending strip kinematics follows the theory of a regular shell patch, but the special material of the bending strips is chosen such that we have a zero membrane stiffness, which leads to a zero membrane stress

$$\hat{\mathbf{S}}_s^m = \mathbf{D}_s^m \hat{\mathbf{E}}_s^m = \mathbf{0}, \quad \text{where } \mathbf{D}_s^m = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad (4.28)$$

and a non-zero bending stiffness E_s only in the direction transverse to the length of the strip, giving us the bending stress

$$\hat{\mathbf{S}}_s^f = \mathbf{D}_s^f \hat{\mathbf{E}}_s^f, \quad \text{where } \mathbf{D}_s^f = \begin{pmatrix} E_s & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}. \quad (4.29)$$

$\hat{\mathbf{E}}_s^m, \hat{\mathbf{E}}_s^f, \hat{\mathbf{S}}_s^m, \hat{\mathbf{S}}_s^f$ represent the Voigt notation of the strain and stress tensors respectively, with its components in the Cartesian coordinates. In general for shells, the covariant components $\hat{\mathbf{E}}^m$ and $\hat{\mathbf{E}}^f$ of the strain tensor are computed with respect to the undeformed contravariant basis $\mathbf{a}_0^i \otimes \mathbf{a}_0^j$, $i, j = 1, 2, 3$, see Section 3.2.1. In order to compute the bending stress using our special material matrix \mathbf{D}_s^f , we project the bending strain in the local Cartesian coordinates with the basis $\mathbf{e}_i \otimes \mathbf{e}_j$ using the relation (3.19).

Substituting the Cartesian components of strain and stress in (3.22), we get the total energy of the bending strip U_s given by

$$U_s = \int_{\omega} \left(\frac{\epsilon}{2} \hat{\mathbf{S}}_s^m \cdot \hat{\mathbf{E}}_s^m + \frac{\epsilon^3}{6} \hat{\mathbf{S}}_s^f \cdot \hat{\mathbf{E}}_s^f \right) \nu_0 d(\vartheta_1, \vartheta_2) = \int_{\omega} \left(\frac{\epsilon^3}{6} \hat{\mathbf{S}}_s^f \cdot \hat{\mathbf{E}}_s^f \right) \nu_0 d(\vartheta_1, \vartheta_2). \quad (4.30)$$

Since the strip is massless, we have the kinetic energy $T_s = 0$. Also due to no external loading, we get the external energy $U_{s,\text{ext}} = 0$. As only one entry of the material matrix \mathbf{D}_s^f is active, the energy of the strip U_s depends only on the first component of the strain tensor $\hat{\mathbf{E}}_{s,11}^f$

$$\hat{\mathbf{E}}_{s,11}^f = \vec{\mathbf{E}}_{s,11}^f (\mathbf{e}_1 \cdot \mathbf{a}_0^1)^2 + \vec{\mathbf{E}}_{s,22}^f (\mathbf{e}_1 \cdot \mathbf{a}_0^2)^2 + 2\vec{\mathbf{E}}_{s,12}^f (\mathbf{e}_1 \cdot \mathbf{a}_0^1)(\mathbf{a}_0^2 \cdot \mathbf{e}_1) = \|(\mathbf{a}_1)_0\|^{-2} \vec{\mathbf{E}}_{s,11}^f,$$

since $\mathbf{e}_1 \cdot \mathbf{a}_0^1 = \|(\mathbf{a}_1)_0\|^{-1}$ and $\mathbf{e}_1 \cdot \mathbf{a}_0^2 = 0$ from (3.20) and

$$(\mathbf{a}_i)_0 \cdot \mathbf{a}_0^j = \delta_i^j = \begin{cases} 0 & \text{if } i \neq j \\ 1 & \text{if } i = j \end{cases}.$$

This gives us the simplified energy expression

$$U_s = \int_{\omega} \left(\frac{\epsilon^3}{6} E_s \| \partial_1 \mathbf{c}_0 \|^4 (\vec{\mathbf{E}}_{s,11}^f)^2 \right) \nu_0 d(\vartheta_1, \vartheta_2). \quad (4.31)$$

Mathematically, the bending strips are included into the system by augmenting the energy of each strip U_s to the total energy of the system. For example, for a two patch case with a single bending strip in the stationary setting, the total energy of the system is

$$U = U_1 + U_2 + U_s.$$

This design of the material constitutive matrix ensures that the bending strip adds no extra stiffness to the structure. It only penalizes the change in the angle during the deformation between the triples of control points at the patch interface. The directional bending stiffness E_s must be high enough so that the change in angle is within an acceptable tolerance. However, if E_s is chosen too high, the global stiffness matrix becomes badly conditioned, which may lead to wrong results. This issue will be addressed in detail in Section 4.4.2.2. This behavior of the parameter E_s resembles that of a penalty parameter, which leads to a stiff system. We observe it closer in Chapter 5.

Linear deformation setting $\vec{\mathbf{E}}_{s,\alpha\beta}^f$ is in general nonlinear. In the linear setting, we get from [Cia05] the linear components of the covariant strain as

$$\vec{\mathbf{E}}_{s,\alpha\beta}^f = (\partial_{\alpha\beta} \mathbf{u} - \Gamma_{\alpha\beta}^{\sigma} \partial_{\sigma} \mathbf{u}) \cdot \mathbf{n}_0, \quad (4.32)$$

where

$$\Gamma_{\alpha\beta}^{\sigma} = \mathbf{a}^{\sigma} \cdot \partial_{\alpha} \mathbf{a}_{\beta} \quad (4.33)$$

are the Christoffel symbols of the second kind defined using the covariant basis vectors $\mathbf{a}_{\beta} = \partial_{\beta} \mathbf{c}_0$ and the contravariant basis vectors $\mathbf{a}^{\sigma} = a^{\sigma\tau} \mathbf{a}_{\sigma}$, such that

$$[a^{\sigma\tau}] = [a_{\sigma\tau}]^{-1} \quad \text{and} \quad a_{\sigma\tau} = \mathbf{a}_{\sigma} \cdot \mathbf{a}_{\tau}.$$

Substituting the linear strain $\vec{\mathbf{E}}_{s,\alpha\beta}^f$ from (4.32) in (4.31), we get

$$U_s(\mathbf{u}) = \int_{\omega} \left(\frac{\epsilon^3}{6} E_s \|\partial_1 \mathbf{c}_0\|^{-4} ((\partial_{11} \mathbf{u} - \Gamma_{11}^1 \partial_1 \mathbf{u} - \Gamma_{11}^2 \partial_2 \mathbf{u}) \cdot \mathbf{n}_0)^2 \right) \nu_0 d(\vartheta_1, \vartheta_2).$$

Applying the calculus of variation once, we obtain the analytical expression for the linear force

$$a_s(\mathbf{u}, \mathbf{v}) = \int_{\omega} \left(\frac{\epsilon^3}{3} E_s \|\partial_1 \mathbf{c}_0\|^{-4} ((\partial_{11} \mathbf{v} - \Gamma_{11}^1 \partial_1 \mathbf{v} - \Gamma_{11}^2 \partial_2 \mathbf{v}) \cdot \mathbf{n}_0) \cdot ((\partial_{11} \mathbf{u} - \Gamma_{11}^1 \partial_1 \mathbf{u} - \Gamma_{11}^2 \partial_2 \mathbf{u}) \cdot \mathbf{n}_0) \right) \nu_0 d(\vartheta_1, \vartheta_2), \quad (4.34)$$

and applying it again to the force expression, we get the linear Jacobian expression

$$j_s(\mathbf{u}, \mathbf{v}, \mathbf{w}) = \int_{\omega} \left(\frac{\epsilon^3}{3} E_s \|\partial_1 \mathbf{c}_0\|^{-4} ((\partial_{11} \mathbf{v} - \Gamma_{11}^1 \partial_1 \mathbf{v} - \Gamma_{11}^2 \partial_2 \mathbf{v}) \cdot \mathbf{n}_0) \cdot ((\partial_{11} \mathbf{w} - \Gamma_{11}^1 \partial_1 \mathbf{w} - \Gamma_{11}^2 \partial_2 \mathbf{w}) \cdot \mathbf{n}_0) \right) \nu_0 d(\vartheta_1, \vartheta_2). \quad (4.35)$$

4.4.2.2 Bending Strip Equations

We now present a formal definition of a multipatch shell structure with the adjoining patches joined using bending strips.

Definition 4.1. A multipatch shell structure connected with bending strips is defined through a tuple $(\mathcal{P}, \mathcal{S}, \mathcal{T})$ with n patches $i \in \mathcal{P}$, $\mathcal{P} = \{1, \dots, n\}$ and m bending strips $j \in \mathcal{S}$, $\mathcal{S} = \{1, \dots, m\}$, connecting the j -th patch pair $\mathcal{T}_j = (k, l)$, such that $\mathcal{T}_j \in \mathcal{T}$ and

$$\mathcal{T} = \{(k, l) : \mathbf{P}_r^k = \mathbf{P}_r^l, \forall r \text{ st } \mathbf{P}_r^k \in \Gamma_k, \mathbf{P}_r^l \in \Gamma_l, k, l \in \mathcal{P}\}$$

is the set containing the C^0 connected patch pairs (k, l) , i.e. those with common control points $\mathbf{P}_r^k, \mathbf{P}_r^l$ on the patch boundaries Γ_k, Γ_l . $|\mathcal{T}| = m$.

For the analysis of a general multipatch shell structure connected with bending strips $(\mathcal{P}, \mathcal{S}, \mathcal{T})$, we solve the system

$$\int_{t_0}^{t_1} \left(\sum_{i \in \mathcal{P}} \mathcal{L}_i(\mathbf{u}_i) - \sum_{\substack{j \in \mathcal{S} \\ (k, l) \in \mathcal{T}}} U_{s,j}(\mathbf{u}_k, \mathbf{u}_l, \beta_j) \right) dt \rightarrow \text{stationary, on } \bigcup_{i \in \mathcal{P}} \omega_i \times \mathbb{R}, \quad (4.36a)$$

$$\mathbf{u}_i = \mathbf{u}_{0,i} \quad \text{on } \Gamma_{0,i}, \quad \forall i \in \mathcal{P}, \quad (4.36b)$$

$$\mathbf{u}_k = \mathbf{u}_l \quad \text{on } \Gamma_k, \Gamma_l, \quad \forall (k, l) \in \mathcal{T}, \quad (4.36c)$$

where \mathbf{u}_i is the displacement and $\mathcal{L}_i := T_i - U_i$ the Lagrangian of the i -th patch with $i \in \mathcal{P}$. $U_{s,j}$ is the energy of the j -th bending strip with $j \in \mathcal{S}$, connecting patch pair $(k, l) \in \mathcal{T}$. It is computed using the displacement information of the pair of adjoining patches $\mathbf{u}_k, \mathbf{u}_l$ due to its construction. $\beta_j = E_{s,j}/E$ is the ratio of the stiffness $E_{s,j}$ of the bending strip j and that of the shell structure E . (4.36b) represents the Dirichlet constraints on the patch boundaries and (4.36c) the C^0 continuity constraints at each patch interface $j \in \mathcal{S}$ joining the patch pairs $(k, l) \in \mathcal{T}$ respectively.

We discretize the above equations using isogeometric analysis. The Dirichlet boundary conditions on each patch (4.36b) can be fulfilled either through the choice of the appropriate subspace or using Lagrange multipliers as in Section 4.3. For brevity and ease of understanding, we take here the former approach.

First of all a weak formulation is obtained by applying the principle of least action, followed by a Galerkin projection of the system onto the subspace $\cup_{i \in \mathcal{P}} V_h^i$, where V_h^i is the finite dimensional space for individual patches $i \in \mathcal{P}$ as defined in (4.18) that satisfies the Dirichlet conditions (4.36b). The system unknowns are approximated as $\mathbf{u}_h^i = \sum_{j=1}^{n_{\text{DOF}}^i} \Psi_j^i d_j^i$.

Remark 4.2. *For the ease of notation and understanding, we consider henceforth a two patch case joined with a bending strip ($\{1, 2\}, \{1\}, \{(1, 2)\}$) for our further study. But the theory is valid for arbitrary number of patches and bending strips.*

Substituting the discretized system unknowns \mathbf{u}_h^i in the equations of motion (4.36) yield

$$\begin{pmatrix} \mathbf{M}_1 \ddot{\mathbf{d}}_1 \\ \mathbf{M}_2 \ddot{\mathbf{d}}_2 \end{pmatrix} + \begin{pmatrix} \mathbf{f}_1(\mathbf{d}_1) \\ \mathbf{f}_2(\mathbf{d}_2) \end{pmatrix} + \mathbf{f}_{s,1}(\mathbf{d}_1, \mathbf{d}_2, \beta_1) + (\mathbf{B}_{C0,1})^T \boldsymbol{\lambda}_1 = \begin{pmatrix} \mathbf{b}_1(t) \\ \mathbf{b}_2(t) \end{pmatrix}, \quad (4.37a)$$

$$\mathbf{B}_{C0,1} \begin{pmatrix} \mathbf{d}_1 \\ \mathbf{d}_2 \end{pmatrix} = \mathbf{0}, \quad (4.37b)$$

where $\mathbf{d}_i = (d_1^i, \dots, d_{n_{\text{DOF}}^i}^i)$, $\mathbf{M}_i \in \mathbb{R}^{n_{\text{DOF}}^i \times n_{\text{DOF}}^i}$ is the mass matrix, $\mathbf{f}_i(\mathbf{d}_i) \in \mathbb{R}^{n_{\text{DOF}}^i}$ the internal elastic force, $\mathbf{b}_i(t) \in \mathbb{R}^{n_{\text{DOF}}^i}$ the external force on each patch i , $i = 1, 2$ and $\mathbf{f}_{s,1}(\mathbf{d}_1, \mathbf{d}_2, \beta_1) \in \mathbb{R}^{n_{\text{DOF}}^i}$ defines the additional force on the system due to the bending strip that affects only the corresponding degrees of freedom lying on the strip and is a projection from the space $\mathbb{R}^{n_{\text{DOF},s}^1}$ to $\mathbb{R}^{n_{\text{DOF}}^i}$ by adding 0s corresponding to the dofs not lying on the bending strip. The number of bending strip dofs

$$n_{\text{DOF},s}^1 := 3 \times (3 \times \#\{j, \mathbf{P}_j^1 \in \Gamma_1\}), \quad \text{such that } (1, 2) \in \mathcal{T}. \quad (4.38)$$

The matrix $\mathbf{B}_{C0,1} \in \mathbb{R}^{(n_{\text{DOF},s}^1/3) \times n_{\text{DOF}}^i}$ contains the entries $\{0, 1, -1\}$, such that the C^0 continuity between the adjoining patches 1,2 is enforced by equating the appropriate dofs and $\boldsymbol{\lambda}_1 \in \mathbb{R}^{n_{\text{DOF},s}^1/3}$ is the vector of Lagrange multipliers. The Dirichlet conditions (4.36b) vanish here due to the choice of the subspace $\cup_{i \in \mathcal{P}} V_h^i$.

Remark 4.3. *It must be noted that (4.37a) is not variationally exact, since in practice we discretize the bending strip unknowns $\mathbf{u}_{h,s}^1$ similar to the way we do it for individual patches using separate basis functions $\Psi_{j,s}^1$, i.e.*

$$\mathbf{u}_{h,s}^1(\vartheta_1, \vartheta_2) = \sum_j \Psi_{j,s}^1(\vartheta_1, \vartheta_2) d_j^i, \quad i = \begin{cases} 1 & \text{if } (\vartheta_1, \vartheta_2) \in \Omega_1 \\ 2 & \text{else,} \end{cases} \quad (4.39)$$

and do not use the same basis functions as for $\mathbf{u}_h^1, \mathbf{u}_h^2$. This works without problem since the idea here is to penalize the change in angle at the patch intersection. The method has been created from a geometrical point of view rather than from a variational one.

The equation system (4.37) is solved by applying an appropriate time integration scheme, e.g. BDF methods from Section 2.1.3.1, generalized- α for DAEs from Section 2.1.3.2, etc., followed by the Newton method in each time step to solve the nonlinear system of equations.

Remark 4.4. A linear material law (3.13) governs that the internal force is directly proportional to the material matrix, which in turn is directly proportional to the E -modulus. Therefore for the bending strip, the net force $\mathbf{f}_{s,1}$ which equals the internal force is directly proportional to its E -modulus $E_{s,1} = \beta_1 E$ and hence to the parameter β_1 , leading to

$$\mathbf{f}_{s,1}(\mathbf{d}_1, \mathbf{d}_2, \beta_1) = \beta_1 \mathbf{f}_{s,1}(\mathbf{d}_1, \mathbf{d}_2, 1). \quad (4.40)$$

In the stationary case, the time dependent terms vanish and the system (4.37) leads to

$$\begin{pmatrix} \mathbf{f}_1(\mathbf{d}_1) \\ \mathbf{f}_2(\mathbf{d}_2) \end{pmatrix} + \mathbf{f}_{s,1}(\mathbf{d}_1, \mathbf{d}_2, \beta_1) + (\mathbf{B}_{C0,1})^T \boldsymbol{\lambda}_1 = \begin{pmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{pmatrix}, \quad (4.41a)$$

$$\mathbf{B}_{C0,1}(\mathbf{d}_1^T, \mathbf{d}_2^T)^T = \mathbf{0}, \quad (4.41b)$$

which is solved by the Newton method with the following system Jacobian

$$\begin{pmatrix} \left(\begin{pmatrix} \nabla_{\mathbf{d}_1} \mathbf{f}_1 \\ \mathbf{0} \end{pmatrix} + \nabla_{\mathbf{d}_1} \mathbf{f}_{s,1} \right) & \begin{pmatrix} \mathbf{0} \\ \nabla_{\mathbf{d}_2} \mathbf{f}_2 \end{pmatrix} + \nabla_{\mathbf{d}_2} \mathbf{f}_{s,1} \\ \mathbf{B}_{C0,1} & \mathbf{0} \end{pmatrix} \begin{pmatrix} ((\mathbf{B}_{C0,1})^T) \\ \boldsymbol{\lambda}_1 \end{pmatrix}.$$

A closer look at the system Jacobian suggests that its ill-conditioning might occur due to two reasons. Firstly, the constraint Jacobian matrix $\mathbf{B}_{C0,1}$ might have a different order of magnitude than the force Jacobians $\nabla_{\mathbf{d}_1} \mathbf{f}_1, \nabla_{\mathbf{d}_2} \mathbf{f}_2$ of patches 1 and 2 respectively. Secondly, a large value of the penalty parameter β_1 might lead to an increase in the bending strip force $\mathbf{f}_{s,1}$ and hence its corresponding Jacobians would attain large values, leading to an ill-conditioning of the system Jacobian matrix. A similar behavior is also noticed for the transient case, where in addition a time step size dependence can introduce further ill-conditioning of the Jacobian. The last effect was observed in Section 2.2.4.2.

4.4.2.3 Summary of the Bending Strip Method

In this section, we summarize the pros and cons of the bending strip method. Its main advantages are that it can tackle both cases, the patches joining at a kink as well as those joining with G^1 continuity. Not only is it possible, but the bending strip method also does not distinguish between the two cases, since in each case it penalizes the change of angle between the triples of control points at the patch interface. Moreover, since each strip adds a penalty for change of angle in a particular direction instead of additional constraint equations, overlapping strips pose no issue. For methods that involve constraints, one might run into redundancy troubles at places where more than two patches meet. Lastly, there is no increase in the degrees of freedom for the problem at hand, since the bending strip method is a penalty approach that uses the existing degrees of freedom at the interface of the patches joined by the strip.

The disadvantages are that the ill-conditioning of the system may occur due to various reasons. The constraint Jacobian might have a different order of magnitude than other Jacobian components. A high penalty parameter value badly affects the condition number of the system Jacobian and so does the time step size dependence of the Jacobian in the transient setting. Another challenge that we observe in Chapter 5 is that this penalty approach leads to stiff systems, which actually exhibit additional phenomena such as high frequency oscillations and a dependence of the time step size on the penalty parameter [Lub93]. Such time step

size restriction can lead to problems with the simulation as they might require very small step sizes to proceed. Moreover, the design of the bending strip in Fig. 4.7 suggests that the adjoining patches must have matching parametrizations in order to have a triple structure at the intersection.

4.4.3 Scaling on the Bending Strip Method

Since we see later from Lemma 5.1 that the structure of the bending strip equations (4.37) is similar to that of constrained mechanical systems, we extend the scaling technique proposed for mechanical systems from Section 2.2.4 to the bending strip method, in order to avoid some of the conditioning issues with the system Jacobian as discussed in Section 4.4.2.3.

4.4.3.1 Stationary Case

Taking the stationary bending strip system (4.41) as the starting point, we apply the scaling and obtain

$$\begin{pmatrix} \mathbf{f}_1(\mathbf{d}_1) \\ \mathbf{f}_2(\mathbf{d}_2) \end{pmatrix} + \mathbf{f}_{s,1}(\mathbf{d}_1, \mathbf{d}_2, \beta_1) + s(\mathbf{B}_{C0,1})^T \hat{\boldsymbol{\lambda}}_1 = \begin{pmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{pmatrix}, \quad (4.42a)$$

$$s\mathbf{B}_{C0,1}(\mathbf{d}_1^T, \mathbf{d}_2^T)^T = \mathbf{0}, \quad (4.42b)$$

where $s = \max_{i \in \mathcal{P}} \|\nabla \mathbf{f}_i\|_\infty$. Applying the Newton method, we get the Jacobian

$$\mathbf{J} = \begin{pmatrix} \left(\begin{pmatrix} \nabla_{\mathbf{d}_1} \mathbf{f}_1 \\ \mathbf{0} \end{pmatrix} + \nabla_{\mathbf{d}_1} \mathbf{f}_{s,1} \right) & \begin{pmatrix} \mathbf{0} \\ \nabla_{\mathbf{d}_2} \mathbf{f}_2 \end{pmatrix} + \nabla_{\mathbf{d}_2} \mathbf{f}_{s,1} \\ \left(\begin{matrix} & \\ & s\mathbf{B}_{C0,1} \end{matrix} \right) & \mathbf{0} \end{pmatrix} s(\mathbf{B}_{C0,1})^T.$$

Due to the pre-multiplication of the constraint Jacobian $\mathbf{B}_{C0,1}$ with the scaling factor s , a large difference in order of magnitude of the constraint Jacobian to patch Jacobians, which may cause ill-conditioning is avoided.

4.4.3.2 Transient Case

Applying the scaling technique to the transient bending strip system (4.37), we get

$$\begin{pmatrix} \mathbf{M}_1 \mathbf{d}_1'' \\ \mathbf{M}_2 \mathbf{d}_2'' \end{pmatrix} + \begin{pmatrix} h^2 \mathbf{f}_1(\mathbf{d}_1) \\ h^2 \mathbf{f}_2(\mathbf{d}_2) \end{pmatrix} + h^2 \mathbf{f}_{s,1}(\mathbf{d}_1, \mathbf{d}_2, \beta_1) + s(\mathbf{B}_{C0,1})^T \hat{\boldsymbol{\lambda}}_1 = \begin{pmatrix} h^2 \mathbf{b}_1(t) \\ h^2 \mathbf{b}_2(t) \end{pmatrix}, \quad (4.43a)$$

$$s\mathbf{B}_{C0,1}(\mathbf{d}_1^T, \mathbf{d}_2^T)^T = \mathbf{0}, \quad (4.43b)$$

where $s = m_r + k_r h^2$, such that

$$m_r = \max_{i \in \mathcal{P}} \|\mathbf{M}_i\|_\infty, \quad k_r = \max_{i \in \mathcal{P}} \|\nabla \mathbf{f}_i\|_\infty$$

and $(\cdot)'$ is derivative with respect to the normalized time τ , $t = h\tau$. Applying the implicit Euler method to discretize in time, followed by Newton method, we get in Newton step $i+1$ the system Jacobian \mathbf{J} as

$$\begin{pmatrix} \left(\begin{pmatrix} \mathbf{M}_1 + h^2 \nabla_{\mathbf{d}_1} \mathbf{f}_1 \\ \mathbf{0} \end{pmatrix} + h^2 \nabla_{\mathbf{d}_1} \mathbf{f}_{s,1} \right) & \begin{pmatrix} \mathbf{0} \\ \mathbf{M}_2 + h^2 \nabla_{\mathbf{d}_2} \mathbf{f}_2 \end{pmatrix} + h^2 \nabla_{\mathbf{d}_2} \mathbf{f}_{s,1} \\ \left(\begin{matrix} & \\ & s\mathbf{B}_{C0,1} \end{matrix} \right) & \mathbf{0} \end{pmatrix}_{i+1} s(\mathbf{B}_{C0,1})^T$$

and the right hand side

$$\mathbf{c} = \begin{pmatrix} \begin{pmatrix} \mathbf{M}_1(\mathbf{d}_{1,i+1} - \mathbf{d}_{1,i} - \mathbf{v}_{1,i}) + h^2\mathbf{f}_1 \\ \mathbf{M}_2(\mathbf{d}_{2,i+1} - \mathbf{d}_{2,i} - \mathbf{v}_{2,i}) + h^2\mathbf{f}_2 \end{pmatrix} + h^2\mathbf{f}_{s,1} + s(\mathbf{B}_{C0,1})^T \hat{\boldsymbol{\lambda}}_{1,i+1} - \begin{pmatrix} h^2\mathbf{b}_1(t) \\ h^2\mathbf{b}_2(t) \end{pmatrix} \\ s\mathbf{B}_{C0,1} \cdot (\mathbf{d}_{1,i+1}^T, \mathbf{d}_{2,i+1}^T) \end{pmatrix},$$

assuming that $\tau = 1$. This gives

$$\mathbf{J} = \begin{pmatrix} \mathcal{O}(h^0) & \mathcal{O}(h^0) \\ \mathcal{O}(h^0) & \mathbf{0} \end{pmatrix} \text{ and } \mathbf{c} = \begin{pmatrix} \mathcal{O}(h^0) \\ \mathcal{O}(h^0) \end{pmatrix}.$$

Therefore for the updates, we get

$$\Delta \mathbf{d}_i = \mathcal{O}(h^0) \text{ and } \Delta \boldsymbol{\lambda}_j = \mathcal{O}(h^0), \quad i = 1, 2, j = 1.$$

If we consider the problems with the conditioning of the bending strip method at the end of Section 4.4.2.3, through the scaling we can correct two of the three conditioning issues. Firstly, due to the scaling of the constraint equation, the order of magnitude of the constraint Jacobian will be same as that of the patch Jacobians. Secondly, we observe from the Jacobian and the update calculations above that they are independent of the time step size with regards to the order of magnitude, which will not be possible without a scaling, see Section 2.2.4.2.

A conditioning issue that still remains unresolved is the dependence of the Jacobian on the penalty parameter β_1 . Additionally, the bending strip system is identified in Section 5.1.1 as a stiff system and the expected time step size restriction that appears for stiff systems [Lub93] must be tackled. Approaches to handle these issues will be discussed in the following chapter.

Chapter 5

Penalty-free Formulations for Multipatch Shells

In this chapter, we develop penalty-free formulations to join multipatch shells contrary to the bending strip method from Section 4.4.2, which is a penalty approach that results in stiff systems following Lemma 5.1. This is done in order to remove the penalty parameter dependence and the ill-conditioning that comes with it. Another goal is to remove the time step size restriction that appears for stiff systems due to the Newton contractivity condition as observed in [Lub93].

We discuss two approaches which we call as *alternative formulations*, where we use constraints instead of the penalty terms to maintain the smoothness between patches. Usually the problem with such systems is that when converting the constraints from a continuous to a discretized form, the resulting saddle-point formulation must satisfy an LBB condition [Bre90], restricting the space of finite elements. In our formulations, we already start with either the discrete bending strips defined on the triples of control points at the intersection to derive the constraints or otherwise the continuity conditions are directly applied on the triples of control points. It circumvents the need for the LBB condition. Although the alternative formulations introduce additional dofs in the system via the Lagrange multipliers corresponding to the constraints, they give us well-conditioned, non-stiff systems. Such systems are also much easier to handle during a transient analysis. After the two approaches are presented, we summarize their benefits and shortcomings.

5.1 Alternative Formulation with Bending Strips

In this section, we derive an alternative formulation using the bending strips. We start from the scaled bending strip system (4.43) and establish its connection to the stiff mechanical systems in Section 5.1.1. This is followed by the separation of the stiff forces into constraints, analogous to their treatment in stiff mechanical systems, see Section 2.2.3. Such an approach is however quite general and can be applied to all penalty approaches under some mild convexity and smoothness assumptions from [Lub93]. In Section 5.1.2, we identify that the stiff force constraints are redundant due to the bending strip construction. To get well defined systems, this redundancy is removed. In Section 5.1.3, we apply various approaches to remove the redundancy such as the analysis of the stiff force constraints, using the perturbed Lagrange method and using linear algebra to work with the independent equations in each Newton

iteration while solving the nonlinear problem.

5.1.1 Connection with Stiff Systems

According to [Lub93], the numerical solution of the stiff mechanical system in ODE form (2.47) is computationally at least as expensive as that of the associated constrained mechanical system in the DAE form. Following this, we aim to show the similarity between the bending strip formulation and the stiff mechanical system of the form (2.47). Then, the stiff part is separated as constraints analogous to stiff mechanical systems, in order to improve the properties of the bending strip method via a new alternative formulation.

Lemma 5.1. *The bending strip system (4.43) is a stiff mechanical system.*

Proof. Taking the bending strip formulation (4.43) and reformulating, we get

$$\mathbf{M}\mathbf{d}'' = \mathbf{f}_n(\mathbf{d}, \dot{\mathbf{d}}) - \mathbf{f}_s(\mathbf{d}) - s(\mathbf{B}_{C0,1})^T \hat{\boldsymbol{\lambda}}_1, \quad (5.1a)$$

$$s\mathbf{B}_{C0,1} \cdot \mathbf{d} = \mathbf{0}, \quad (5.1b)$$

where

$$\mathbf{d} = \begin{pmatrix} \mathbf{d}_1 \\ \mathbf{d}_2 \end{pmatrix}, \quad \mathbf{M} = \begin{pmatrix} \mathbf{M}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_2 \end{pmatrix}, \quad (5.2)$$

and the non-stiff forces come from the net forces on individual shell patches i.e.

$$\mathbf{f}_n(\mathbf{d}, \dot{\mathbf{d}}) = \begin{pmatrix} h^2 \mathbf{b}_1(t) \\ h^2 \mathbf{b}_2(t) \end{pmatrix} - \begin{pmatrix} h^2 \mathbf{f}_1(\mathbf{d}_1) \\ h^2 \mathbf{f}_2(\mathbf{d}_2) \end{pmatrix} \quad (5.3)$$

and the stiff part is given following (2.47) by

$$\frac{1}{\epsilon^2} \frac{\partial U(\mathbf{d})}{\partial \mathbf{d}} = \mathbf{f}_s(\mathbf{d}) = h^2 \mathbf{f}_{s,1}(\mathbf{d}_1, \mathbf{d}_2, \beta_1) = \frac{1}{(1/h\sqrt{\beta_1})^2} \frac{\partial}{\partial \mathbf{d}} U_{s,1}(\mathbf{d}_1, \mathbf{d}_2, 1). \quad (5.4)$$

This system has the same structure as the stiff mechanical system from Section 2.2.3. The bending strip energy $U_{s,1}$ corresponds to the potential U and perturbation parameter ϵ is a function of the high value penalty parameter β_1 . Important point here is that the parameter β_1 comes out of the potential $U_{s,1}$ because of the linear material law for the bending strip, see Remark 4.4. \square

Analogous to the treatment of stiff mechanical systems, assuming the bending strip energy function $U_{s,1}$ to be mildly convex and smooth, we separate the stiff forces as scaled constraints, include their force contribution via Lagrange multipliers and rewrite the transient bending strip system (5.1) as

$$\begin{pmatrix} \mathbf{M}_1 \mathbf{d}_1'' \\ \mathbf{M}_2 \mathbf{d}_2'' \end{pmatrix} + \begin{pmatrix} h^2 \mathbf{f}_1(\mathbf{d}_1) \\ h^2 \mathbf{f}_2(\mathbf{d}_2) \end{pmatrix} + s(\nabla \mathbf{f}_{s,1})^T \hat{\boldsymbol{\lambda}}_1 + s(\mathbf{B}_{C0,1})^T \hat{\boldsymbol{\lambda}}_2 = \begin{pmatrix} h^2 \mathbf{b}_1(t) \\ h^2 \mathbf{b}_2(t) \end{pmatrix}, \quad (5.5a)$$

$$s\mathbf{f}_{s,1}(\mathbf{d}_1, \mathbf{d}_2, 1) = \mathbf{0}, \quad (5.5b)$$

$$s\mathbf{B}_{C0,1}(\mathbf{d}_1^T, \mathbf{d}_2^T)^T = \mathbf{0}. \quad (5.5c)$$

We call this system, which is now independent of the penalty parameter β_1 as the *alternative formulation*. It exhibits the limit case behavior, where the penalty parameter $\beta_1 \rightarrow \infty$.

Remark 5.1. (5.5b) implies that the net forces on the bending strip are zero. Since the bending strip has no external forces, this means the internal forces are zero. This suggests that the strip undergoes a rigid body motion, which in turn implies no change of angle at the intersection and is exactly as we desire.

5.1.2 Redundancy Issues

Although the system (5.5) due to its angle preservation properties and penalty parameter independence looks promising, it has an inherent issue due to its formulation which needs further treatment. The problem is that the bending strip force constraints (5.5b) are redundant. It is due to the fact that the bending strip is also a shell patch by construction. A shell patch cannot have well defined equations until it has some external influence to bring the shell in equilibrium. Since the bending strips have no external forces or boundary conditions, the constraint Jacobian matrix arising from (5.5b) cannot have full rank, which is needed for the DAE system (5.5) to be well defined.

From [KBH⁺10] and our bending strip results, we observe that the bending strip forces carry enough information to preserve the smoothness between the adjoining patches. Hence we must extract the relevant information. At this point, there is a need to identify an efficient procedure through which either the redundancy can be tackled by manipulating the equations, leading thereafter to a well defined problem; or otherwise if we can work with the independent part of the redundant bending strip force constraints from (5.5b).

5.1.3 Treating Redundancy

In this section, we present various techniques one can apply to treat the redundancy of the bending strip force constraints. Firstly we develop an analytical approach, where although very limited number of geometries in the linear deformation setting can be analyzed for the identification of the redundancies, the main goal is to obtain in process an understanding of the bending strip force constraints. Later we shortly present another technique called the *perturbed Lagrange method*, which makes certain changes to the constraints to remove the redundancy, but in process ends up introducing a new penalty parameter dependence. Lastly we present an approach using linear algebra where in each Newton step, we use only the information from the independent part of the constraints.

5.1.3.1 Analysis of Bending Strip Force Constraints

In order to remove the redundant constraints in (5.5b) and in process get a better understanding of the implementation of bending strips, we separately analyze the bending strip equations

$$\min U_{s,1} \quad \Rightarrow \quad \mathbf{f}_{s,1}(\mathbf{d}_1, \mathbf{d}_2, 1) = \mathbf{0}, \quad (5.6)$$

where the energy $U_{s,1}$ is expressed as in (4.31). We identify and tackle the redundancies from the force $\mathbf{f}_{s,1}$ in (5.6).

Substituting the nonlinear expression for strain $\mathbf{E}_{f,\alpha\beta}^s$ in the strip energy from (4.31) leads to lengthy expressions which are hard to study. In this section, we work in the linear setting and use the expressions for force (4.34) and Jacobian (4.35). In order to identify the redundancy in the force vector $\mathbf{f}_{s,1} := (a_s(\mathbf{u}_h, \mathbf{v}_j))_j$ that is linear in displacement \mathbf{u}_h , it is sufficient to identify the redundant rows of the Jacobian matrix $\mathbf{J}_{s,1} := (j_s(\mathbf{u}_h, \mathbf{v}_j, \mathbf{w}_k))_{j,k}$. The main ingredients of the Jacobian are the basis functions, initial geometry and the quadrature rule applied for numerical integration. When we substitute the basis functions in the Jacobian expression for general geometries, it is much too complicated and hard to study. Therefore, we narrow down the geometry classification and study the Jacobian. We focus on a special case, where a pair of plates is connected at an angle to each other.

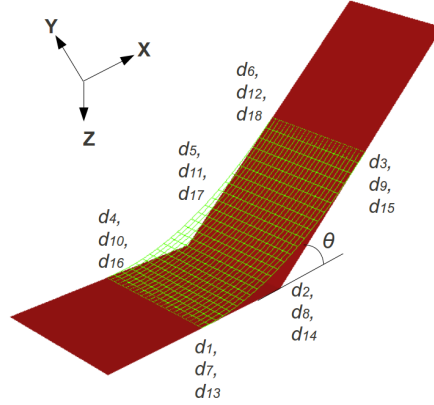


Figure 5.1: Two plates at an angle θ with each other

Plates connecting at an angle: In Fig. 5.1, we take a pair of straight plates connected at an angle θ to each other. Our goal is to identify the independent bending strip constraints generated by our alternative formulation in the linear setting. To achieve this, we study the bending strip constraint Jacobian generated for this case and identify its rank, which directly corresponds to the number of independent constraints generated.

Conjecture 5.1. *Assuming small deformation, when a pair of plates is joined at an angle θ using a bending strip, see Fig. 5.1, the number of independent constraints which equals the rank of the constraint Jacobian is given by*

$$\text{rank}(\mathbf{J}_{s,1}) = \begin{cases} n + 1 & \text{if } \theta = 0 \\ 3(n + 1) & \text{if } \theta \neq 0 \end{cases}$$

where n is the number of bending strip elements along the strip length, provided that

$$\#\text{quadrature points} \geq \begin{cases} 1 \times 2 & \text{if } \theta = 0 \\ 3 \times 2 & \text{if } \theta \neq 0 \end{cases}$$

in the two parametric directions.

We prove this statement partially using analytical expressions for the case of a few quadrature points and observe through our numerical tests that the statement holds true for higher number of quadrature points.

Without loss of generality, we assume an initial configuration such that the normal at the plates and also the bending strip lies in the x-z plane and its y-component is always zero, i.e. $\mathbf{n}_0 = (x, 0, z)^T$. Any configuration can be brought to the assumed configuration by a rigid body transformation. This is possible since a NURBS surface is invariant to rigid body transformations [PT97].

We focus our attention on the bending strip force Jacobian from (4.35)

$$\mathbf{J}_{s,1}(\mathbf{u}, \mathbf{v}_j, \mathbf{w}_k)_{j,k} = \int_{\omega} \left(\frac{\epsilon^3}{3} E_{s,1} \|\partial_1 \mathbf{c}_0\|^{-4} ((\partial_{11} \mathbf{v}_j - \Gamma_{11}^1 \partial_1 \mathbf{v}_j - \Gamma_{11}^2 \partial_2 \mathbf{v}_j) \cdot \mathbf{n}_0) \cdot ((\partial_{11} \mathbf{w}_k - \Gamma_{11}^1 \partial_1 \mathbf{w}_k - \Gamma_{11}^2 \partial_2 \mathbf{w}_k) \cdot \mathbf{n}_0) \right) \nu_0 d(\vartheta_1, \vartheta_2).$$

Due to the chosen geometry, the curvilinear axes $\mathbf{a}_1 = \partial_1 \mathbf{c}_0$ and $\mathbf{a}_2 = \partial_2 \mathbf{c}_0$ on the bending strip are perpendicular to each other and hence $a_{12} = a_{21} = 0$, leading to

$$a^{11} = \frac{1}{a_{11}} \text{ and } a^{22} = \frac{1}{a_{22}}. \quad (5.7)$$

Substituting this into the formula to compute the Christoffel symbols (4.33), we get

$$\Gamma_{11}^1 = (a^{11}\partial_1\mathbf{c}_0 + a^{12}\partial_2\mathbf{c}_0) \cdot \partial_{11}\mathbf{c}_0 = \frac{\partial_1\mathbf{c}_0 \cdot \partial_{11}\mathbf{c}_0}{\partial_1\mathbf{c}_0 \cdot \partial_1\mathbf{c}_0}, \quad (5.8a)$$

$$\Gamma_{11}^2 = (a^{21}\partial_1\mathbf{c}_0 + a^{22}\partial_2\mathbf{c}_0) \cdot \partial_{11}\mathbf{c}_0 = \frac{\partial_2\mathbf{c}_0 \cdot \partial_{11}\mathbf{c}_0}{\partial_2\mathbf{c}_0 \cdot \partial_2\mathbf{c}_0}, \quad (5.8b)$$

which depend on the slope and the curvature at the undeformed shell centerline \mathbf{c}_0 . $\Gamma_{11}^2 = 0$ for this geometry, because the covariant axis $\partial_2\mathbf{c}_0$ which points in the y-direction and the curvature vector $\partial_{11}\mathbf{c}_0$ which lies in the x-z plane are perpendicular to each other. In case the plates join with $\theta = 0$ and are of equal length, the bending strip is straight with a curvature $\partial_{11}\mathbf{c}_0 = \mathbf{0}$, leading to $\Gamma_{11}^1 = \Gamma_{11}^2 = 0$.

For the ease of analysis, we take a bending strip consisting of only one element at the patch interface, which as usual is quadratic for the bending strip in length direction and linear in the width direction. A single element due to the linear ansatz along the length of the strip has only 2 control points on the boundary Γ . Hence the number of bending strip dofs, $n_{\text{DOF},s}^1 = 3 \times 3 \times 2 = 18$ from (4.38) such that d_i , $i = 1, \dots, 18$ and the basis functions are given by

$$\Psi_{i+3j} = \Phi_i \mathbf{e}_j, \quad i = 1, \dots, 6, \quad j = 1, 2, 3,$$

where \mathbf{e}_j are the three dimensional unit vectors in j -th direction. The NURBS basis functions Φ_i and their derivatives in terms of the two parametric directions u, v are

$$\begin{aligned} \Phi_1 &= (1-u)^2(1-v), & \partial_1\Phi_1 &= -2(1-u)(1-v), & \partial_{11}\Phi_1 &= 2(1-v), \\ \Phi_2 &= 2u(1-u)(1-v), & \partial_1\Phi_2 &= 2(1-2u)(1-v), & \partial_{11}\Phi_2 &= -4(1-v), \\ \Phi_3 &= u^2(1-v), & \partial_1\Phi_3 &= 2u(1-v), & \partial_{11}\Phi_3 &= 2(1-v), \\ \Phi_4 &= (1-u)^2v, & \partial_1\Phi_4 &= -2(1-u)v, & \partial_{11}\Phi_4 &= 2v, \\ \Phi_5 &= 2u(1-u)v, & \partial_1\Phi_5 &= 2(1-2u)v, & \partial_{11}\Phi_5 &= -4v, \\ \Phi_6 &= u^2v, & \partial_1\Phi_6 &= 2uv, & \partial_{11}\Phi_6 &= 2v. \end{aligned}$$

Substituting the expressions for the basis function derivatives in the Jacobian expression (4.35) and performing numerical integration, we obtain the 18×18 Jacobian matrix with a block structure

$$(\mathbf{J}_{s,1})_{18 \times 18} = \begin{pmatrix} \mathbf{J}_{s,1,c}^{11} & \mathbf{J}_{s,1,c}^{12} & \mathbf{J}_{s,1,c}^{13} \\ \mathbf{J}_{s,1,c}^{21} & \mathbf{J}_{s,1,c}^{22} & \mathbf{J}_{s,1,c}^{23} \\ \mathbf{J}_{s,1,c}^{31} & \mathbf{J}_{s,1,c}^{32} & \mathbf{J}_{s,1,c}^{33} \end{pmatrix} = \begin{pmatrix} \mathbf{J}_{s,1,c}^{11} & \mathbf{0} & \mathbf{J}_{s,1,c}^{13} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{J}_{s,1,c}^{13} & \mathbf{0} & \mathbf{J}_{s,1,c}^{33} \end{pmatrix}, \quad (5.9)$$

getting its structure because the normal $\mathbf{n}_0 = (x, 0, z)^T$, which implies $\mathbf{J}_{s,1,c}^{k2} = 0, \mathbf{J}_{s,1,c}^{2k} = 0$, $k = 1, 2, 3$. The 6×6 Jacobian components are given by

$$(\mathbf{J}_{s,1,c}^{kl})_{6 \times 6} = \sum_i C_i \mathbf{n}_{0,i}^k \mathbf{n}_{0,i}^l w_i \begin{pmatrix} (1-v_i)^2 \mathbf{J}_{s,1,c,i} & v_i(1-v_i) \mathbf{J}_{s,1,c,i} \\ v_i(1-v_i) \mathbf{J}_{s,1,c,i} & v_i^2 \mathbf{J}_{s,1,c,i} \end{pmatrix}, \quad (5.10)$$

where $C_i = 4\epsilon^3 E_{s,1} \|\partial_1\mathbf{c}_0^i\|^{-4} \nu_0^i / 3$, $\mathbf{n}_{0,i}^k$ are k -th entry of the normal \mathbf{n}_0 evaluated at i -th

quadrature point with $k, l = 1, 2, 3$ and

$$(\mathbf{J}_{s,1,c,i})_{3 \times 3} = \begin{pmatrix} (1 + \Gamma_{11}^{1,i}(1 - u_i))^2 & (1 + \Gamma_{11}^{1,i}(1 - u_i))(-2 - \Gamma_{11}^{1,i}(1 - 2u_i)) & (1 + \Gamma_{11}^{1,i}(1 - u_i))(1 - \Gamma_{11}^{1,i}u_i) \\ (1 + \Gamma_{11}^{1,i}(1 - u_i))(-2 - \Gamma_{11}^{1,i}(1 - 2u_i)) & (-2 - \Gamma_{11}^{1,i}(1 - 2u_i))^2 & (-2 - \Gamma_{11}^{1,i}(1 - 2u_i))(1 - \Gamma_{11}^{1,i}u_i) \\ (1 + \Gamma_{11}^{1,i}(1 - u_i))(1 - \Gamma_{11}^{1,i}u_i) & (1 - \Gamma_{11}^{1,i}u_i)(-2 - \Gamma_{11}^{1,i}(1 - 2u_i)) & (1 - \Gamma_{11}^{1,i}u_i)^2 \end{pmatrix}, \quad (5.11)$$

where i corresponds to the Gaussian quadrature point for the numerical integration.

This block structure of the Jacobian matrix helps in an easy identification of the rank of the system. The rank of the symmetric matrix $\mathbf{J}_{s,1,c,i}$ in the most general setting is 2, because the sum of its three rows is zero, making them linearly dependent. Moreover from (5.10), we get

$$\text{rank}(\mathbf{J}_{s,1,c}^{kl}) = \begin{cases} \text{rank}(\mathbf{J}_{s,1,c,i}) & \text{if } v_i = 0.5, \forall i \\ 2 \text{rank}(\mathbf{J}_{s,1,c,i}) & \text{else,} \end{cases} \quad (5.12)$$

leading to the maximum rank of the full Jacobian to be

$$\max(\text{rank}(\mathbf{J}_{s,1})) = 2 \text{rank}(\mathbf{J}_{s,1,c}^{kl}).$$

This implies that for the case of two plates connecting at an angle with a single bending strip element, the number of independent rows can be at most 8 out of a total of 18 rows.

Let us now look at two specific cases.

$\theta = 0$, equal flange length: We take two plates of equal length connected at an angle $\theta = 0$. Without loss of generality we take $\mathbf{n}_0 = (0, 0, 1)^T$, leading to the following block Jacobian

$$\mathbf{J}_{s,1} = \begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{J}_{s,1,c}^{33} \end{pmatrix},$$

with the Jacobian component

$$\mathbf{J}_{s,1,c}^{33} = \sum_i C_i w_i \mathbf{J}_{s,1,c,i} \begin{pmatrix} (1 - v_i)^2 & v_i(1 - v_i) \\ v_i(1 - v_i) & v_i^2 \end{pmatrix}$$

and for i -th quadrature point

$$\mathbf{J}_{s,1,c,i} = \begin{pmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{pmatrix},$$

since $\Gamma_{11}^{1,i} = 0$ for all quadrature points due to zero curvature $\partial_{11}\mathbf{c}_0 = \mathbf{0}$. This gives a matrix $\mathbf{J}_{s,1,c,i}$ of rank one, since all the rows are linearly dependent to each other, i.e. $\text{rank}(\mathbf{J}_{s,1,c,i}) = 1$. The full Jacobian $\mathbf{J}_{s,1}$ can have at most rank two, since $\text{rank}(\mathbf{J}_{s,1}) = \text{rank}(\mathbf{J}_{s,1,c}^{33})$ and from (5.12) we have

$$\text{rank}(\mathbf{J}_{s,1,c}^{33}) = \begin{cases} \text{rank}(\mathbf{J}_{s,1,c,i}) & \text{if } v_i = 0.5, \forall i \\ 2 \text{rank}(\mathbf{J}_{s,1,c,i}) & \text{else.} \end{cases}$$

This implies that when only a single quadrature point is taken in the second parametric direction, the rank is one. Else in all other cases, the rank is two. Note that the rank is independent of the quadrature points in the first direction for this case.

When we take one Gaussian quadrature point in the first parametric direction, two in the second one and compute the constraints using the Jacobian information, we get after some rearrangements two independent constraints

$$d_{13} - 2d_{14} + d_{15} = 0, \quad (5.13a)$$

$$d_{16} - 2d_{17} + d_{18} = 0. \quad (5.13b)$$

Also for greater number of quadrature points, the computations will lead to the same result due to the properties of the Gaussian quadrature and the constant C_i being the same at all quadrature points for this example.

Interesting is to note that the constraints (5.13) affect dofs from two different triples. This suggests that for n bending strip elements along the length of the strip i.e. $n + 1$ triples, one obtains $n + 1$ corresponding constraints when the number of quadrature points $\geq 1 \times 2$. This proves the conjecture for the case $\theta = 0$ and equal length of plates.

Arbitrary θ , one quadrature point in first parametric direction: Now we consider a general case where the angle θ can be arbitrary. For such a case, the corresponding Jacobian expression is quite complicated for large number of quadrature points and it is hard to establish the connections between the various rows to find the rank. Choosing only one quadrature point in the first direction i.e. $u_i = 0.5, \forall i$, we get

$$\mathbf{J}_{s,1,c,i} = \begin{pmatrix} (1 + \Gamma_{11}^1/2)^2 & (1 + \Gamma_{11}^1/2)(-2) & (1 + \Gamma_{11}^1/2)(1 - \Gamma_{11}^1/2) \\ (1 + \Gamma_{11}^1/2)(-2) & (-2)^2 & (-2)(1 - \Gamma_{11}^1/2) \\ (1 + \Gamma_{11}^1/2)(1 - \Gamma_{11}^1/2) & (1 - \Gamma_{11}^1/2)(-2) & (1 - \Gamma_{11}^1/2)^2 \end{pmatrix},$$

which has rank one, since each row is a multiple of the other. The covariant axes $\partial_1 \mathbf{c}_0$, $\partial_2 \mathbf{c}_0$, the curvature $\partial_{11} \mathbf{c}_0$ and the normal $\mathbf{n} = (x, 0, z)^T$ are constant for all the quadrature points having the same first coordinate due to the geometry. It implies that all $\Gamma_{11}^{1,i}$ are equal and $\Gamma_{11}^{1,i} =: \Gamma_{11}^1, \forall i$. Also, the first and the third row blocks of the full Jacobian are related by $z(\mathbf{J}_{s,1})_{1,\cdot} = x(\mathbf{J}_{s,1})_{3,\cdot}$. Hence the rank of the full Jacobian is equal to the rank of the component matrix $\mathbf{J}_{s,1,c}$, which is at maximum 2 and occurs when more than one quadrature point in the second direction is chosen.

Simplifying $\mathbf{J}_{s,1}$ for the two independent constraints, namely rows 1, 4, and applying the Gaussian quadrature allows us to take $a := v_1 = 1 - v_2 =: 1 - b$ for two quadrature points in second direction and so on, and we get

$$(\mathbf{J}_{s,1})_{i=1,4,j,\cdot} = Cw(\mathbf{J}_{s,1,c,\cdot})_{1,\cdot} \begin{pmatrix} x^2(b^2 + a^2) & x^2(2ab) & 0 & 0 & xz(b^2 + a^2) & xz(2ab) \\ x^2(2ab) & x^2(a^2 + b^2) & 0 & 0 & xz(2ab) & xz(a^2 + b^2) \end{pmatrix}.$$

The following constraints are obtained from the Jacobian

$$\begin{aligned} & (1 + \Gamma_{11}^1/2)(b^2 + a^2)xd_1 - 2(b^2 + a^2)xd_2 + (1 - \Gamma_{11}^1/2)(b^2 + a^2)xd_3 + \\ & \quad (1 + \Gamma_{11}^1/2)(2ab)xd_4 - 2(2ab)xd_5 + (1 - \Gamma_{11}^1/2)(2ab)xd_6 + \\ & (1 + \Gamma_{11}^1/2)(b^2 + a^2)zd_{13} - 2(b^2 + a^2)zd_{14} + (1 - \Gamma_{11}^1/2)(b^2 + a^2)zd_{15} + \\ & \quad (1 + \Gamma_{11}^1/2)(2ab)zd_{16} - 2(2ab)zd_{17} + (1 - \Gamma_{11}^1/2)(2ab)zd_{18} = 0, \end{aligned} \quad (5.14a)$$

$$\begin{aligned}
& (1 + \Gamma_{11}^1/2)(2ab)xd_1 - 2(2ab)xd_2 + (1 - \Gamma_{11}^1/2)(2ab)xd_3 + \\
& (1 + \Gamma_{11}^1/2)(b^2 + a^2)xd_4 - 2(b^2 + a^2)xd_5 + (1 - \Gamma_{11}^1/2)(b^2 + a^2)xd_6 + \\
& (1 + \Gamma_{11}^1/2)(2ab)zd_{13} - 2(2ab)zd_{14} + (1 - \Gamma_{11}^1/2)(2ab)zd_{15} + \\
& (1 + \Gamma_{11}^1/2)(b^2 + a^2)zd_{16} - 2(b^2 + a^2)zd_{17} + (1 - \Gamma_{11}^1/2)(b^2 + a^2)zd_{18} = 0.
\end{aligned} \tag{5.14b}$$

Simplifying the system (5.14) gives

$$\begin{aligned}
(1 + \Gamma_{11}^1/2)xd_1 - 2xd_2 + (1 - \Gamma_{11}^1/2)xd_3 + (1 + \Gamma_{11}^1/2)zd_{13} \\
- 2zd_{14} + (1 - \Gamma_{11}^1/2)zd_{15} = 0,
\end{aligned} \tag{5.15a}$$

$$\begin{aligned}
(1 + \Gamma_{11}^1/2)xd_3 - 2xd_4 + (1 - \Gamma_{11}^1/2)xd_5 + (1 + \Gamma_{11}^1/2)zd_{16} \\
- 2zd_{17} + (1 - \Gamma_{11}^1/2)zd_{18} = 0.
\end{aligned} \tag{5.15b}$$

It must be noted that the first equation involves only the dofs from first triple of control point and the second equation involves only those from second triple. As explained in Remark 5.3, one separate constraint is obtained here for each of the two triples of control points.

Taking the special case of $\theta = 0$ with equal flange lengths, we have $\Gamma_{11}^1 = 0$, $x = 0$, $z = 1$, leading to

$$\begin{aligned}
d_{13} - 2d_{14} + d_{15} &= 0, \\
d_{16} - 2d_{17} + d_{18} &= 0,
\end{aligned}$$

giving us one separate constraint each for the two triples of control points only involving z-direction, same as we had obtained previously.

Arbitrary θ , no quadrature point restriction: For other cases with more than one quadrature point in the first direction, due to complexity of the problem we have performed the analysis numerically. It has been found that for $\theta \neq 0$ and up to 3×2 quadrature points, the rank of the Jacobian comes out to be $i \times j$, where i, j represent the quadrature points in two parametric directions respectively. For 3×2 quadrature points or more, the rank is 6. Also, it has been noticed for the general case, that the 6 independent rows for quadrature points $\geq 3 \times 2$ are 1, 2, 4, 5, 14, 17. Moreover, through simplification of the numerical Jacobian, the constraints could be localized to triples of control point, leading to 3 separate constraints each for the two triples of control point. This also suggests that for n bending strip elements, the independent constraints would depend on the number of triples of control points $n + 1$ by $3(n + 1)$ and the statement of the conjecture for $\theta \neq 0$ also holds true.

Geometric validation of the numerical results: Now we compare the constraints obtained by the numerical analysis using 5×5 quadrature points for two different angles θ in our shell solver against the geometrical ones we obtain on preserving the length and the angle at the triples in the linear setting. This comes from the fact that using the bending strip method, the triples undergo a rigid body motion, see Remark 5.1, and hence the angle at the triples and the length of the flanges should be preserved. For the chosen geometry configuration, the triples of control points lie in the x-z plane. The number of such triples is $n + 1$, where n are the number of elements in the bending strip.

Example 5.1. We take the case where the angle at the triple $\theta = 90^\circ$, see Fig. 5.2. The continuous line depicts the original configuration and the dashed line depicts the deformed

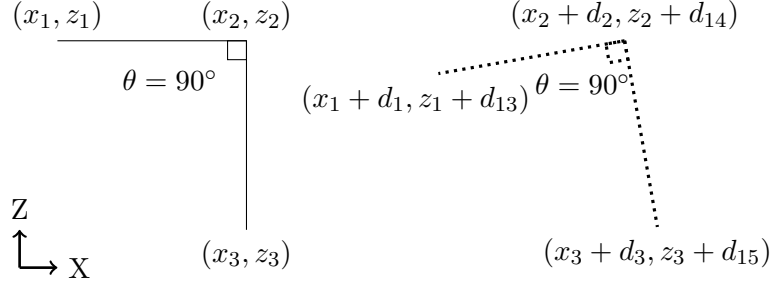


Figure 5.2: Linear deformation of a triple, $\theta = 90^\circ$ angle preserved

configuration. Numerical analysis gives for each triple of control point, the following independent bending strip constraints after the simplification of the Jacobian matrix

$$d_1 = d_2, \quad (5.16a)$$

$$d_{14} = d_{15}, \quad (5.16b)$$

$$d_1 - d_3 = d_{13} - d_{15}. \quad (5.16c)$$

We take the original configuration with flanges of unit length meeting at right angle, with $x_2 - x_1 = 1$, $z_1 = z_2 = 0$, $x_2 = x_3 = 2$, $z_2 - z_3 = 1$. The angle at the triple can be calculated by the dot product between two vectors as follows:

$$\cos \theta = \frac{\vec{\mathbf{a}} \cdot \vec{\mathbf{b}}}{\|\vec{\mathbf{a}}\| \|\vec{\mathbf{b}}\|}.$$

For our case, the vector $\vec{\mathbf{a}} = (x_1 - x_2, 0, z_1 - z_2)^T$ and $\vec{\mathbf{b}} = (x_2 - x_3, 0, z_2 - z_3)^T$ with unit lengths i.e. $\|\vec{\mathbf{a}}\| = 1$, $\|\vec{\mathbf{b}}\| = 1$, leading to

$$\cos \theta = (x_1 - x_2)(x_2 - x_3) + (z_1 - z_2)(z_2 - z_3).$$

Similarly for the deformed configuration, assuming the angle and lengths stay preserved, we get

$$\begin{aligned} \cos \theta &= (x_1 + d_1 - x_2 - d_2)(x_2 + d_2 - x_3 - d_3) \\ &\quad + (z_1 + d_{13} - z_2 - d_{14})(z_2 + d_{14} - z_3 - d_{15}) \\ &= ((x_1 - x_2) + (d_1 - d_2))((x_2 - x_3) + (d_2 - d_3)) \\ &\quad + ((z_1 - z_2) + (d_{13} - d_{14}))((z_2 - z_3) + (d_{14} - d_{15})) \\ &= (x_1 - x_2)(x_2 - x_3) + (d_2 - d_3)(x_1 - x_2) + (d_1 - d_2)(x_2 - x_3) \\ &\quad + (d_1 - d_2)(d_2 - d_3) + (z_1 - z_2)(z_2 - z_3) + (d_{14} - d_{15})(z_1 - z_2) \\ &\quad + (d_{13} - d_{14})(z_2 - z_3) + (d_{13} - d_{14})(d_{14} - d_{15}). \end{aligned}$$

Subtracting the two expressions for $\cos \theta$, we get

$$(d_2 - d_3)(x_1 - x_2) + (d_1 - d_2)(x_2 - x_3) + (d_1 - d_2)(d_2 - d_3) + (d_{14} - d_{15})(z_1 - z_2) + (d_{13} - d_{14})(z_2 - z_3) + (d_{13} - d_{14})(d_{14} - d_{15}) = 0.$$

Since $x_2 = x_3$, $z_1 = z_2$ and neglecting the quadratic terms in the displacement, we get

$$\boxed{d_2 - d_3 = d_{13} - d_{14}.} \quad (5.17)$$

To preserve the length for the first flange, we subtract the original length from the deformed length and put it equal to zero

$$\begin{aligned} 0 &= [(x_1 - x_2) + (d_1 - d_2)]^2 + [(z_1 - z_2) + (d_{13} - d_{14})]^2 - [(x_1 - x_2)^2 + (z_1 - z_2)^2] \\ &= (d_1 - d_2)^2 + 2(x_1 - x_2)(d_1 - d_2) + (d_{13} - d_{14})^2 + 2(z_1 - z_2)(d_{13} - d_{14}). \end{aligned}$$

Since $z_1 = z_2$ and neglecting the quadratic terms in the displacement, we get

$$\boxed{d_1 = d_2.} \quad (5.18)$$

Similarly for the second flange, we get

$$\begin{aligned} 0 &= [(x_2 - x_3) + (d_2 - d_3)]^2 + [(z_2 - z_3) + (d_{14} - d_{15})]^2 - [(x_2 - x_3)^2 + (z_2 - z_3)^2] \\ &= (d_2 - d_3)^2 + 2(x_2 - x_3)(d_2 - d_3) + (d_{14} - d_{15})^2 + 2(z_2 - z_3)(d_{14} - d_{15}), \end{aligned}$$

leading to

$$\boxed{d_{14} = d_{15},} \quad (5.19)$$

since $x_2 = x_3$. This shows that the system of constraints obtained numerically (5.16) matches the three geometric constraints (5.17) – (5.19) for the control point triple.

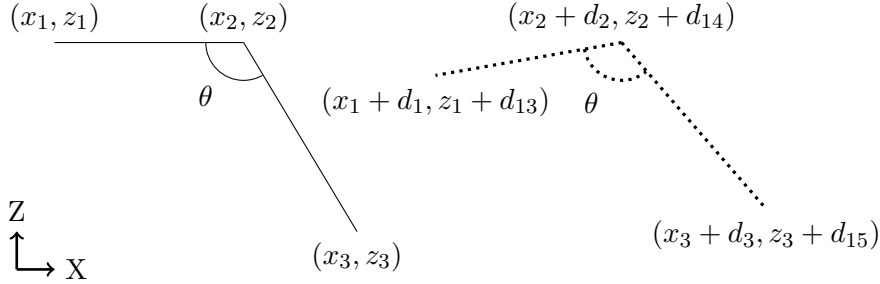


Figure 5.3: Linear deformation of a triple, $\theta \neq 0^\circ, 90^\circ$ angle preserved

Example 5.2. Next we take the general case where the angle at the triple $\theta \neq 0^\circ, 90^\circ$, see Fig. 5.3. Again the continuous line depicts the original configuration and the dashed line depicts the deformed configuration. Numerical analysis gives us the following independent bending strip constraints after the simplification of the Jacobian matrix for each triple of control points

$$d_1 = d_2, \quad (5.20a)$$

$$d_{13} - 2d_{14} + d_{15} = 0, \quad (5.20b)$$

$$d_1 - d_3 = d_{13} - d_{14}. \quad (5.20c)$$

We take the original configuration with flanges of length 1, $\sqrt{2}$ respectively meeting at an angle θ with $x_2 - x_1 = 1$, $z_1 = z_2 = 0$, $x_3 - x_2 = 1$, $z_2 - z_3 = 1$. Assuming the angle stays preserved, we get from the difference in the cosines in the two cases the expression

$$\begin{aligned} &(d_2 - d_3)(x_1 - x_2) + (d_1 - d_2)(x_2 - x_3) + (d_1 - d_2)(d_2 - d_3) + \\ &(d_{14} - d_{15})(z_1 - z_2) + (d_{13} - d_{14})(z_2 - z_3) + (d_{13} - d_{14})(d_{14} - d_{15}) = 0. \end{aligned}$$

Substituting the geometry values and neglecting the quadratic terms in the displacement, we get

$$\boxed{d_2 - d_3 = d_{13} - d_{14}}. \quad (5.21)$$

Preserving the length for the first flange, we get

$$0 = (d_1 - d_2)^2 + 2(x_1 - x_2)(d_1 - d_2) + (d_{13} - d_{14})^2 + 2(z_1 - z_2)(d_{13} - d_{14}).$$

Since $z_1 = z_2$ and neglecting the quadratic terms in the displacement, we get

$$\boxed{d_1 = d_2}. \quad (5.22)$$

Similarly for the second flange, we get

$$0 = (d_2 - d_3)^2 + 2(x_2 - x_3)(d_2 - d_3) + (d_{14} - d_{15})^2 + 2(z_2 - z_3)(d_{14} - d_{15}),$$

leading to

$$\boxed{d_2 - d_3 = d_{14} - d_{15}}, \quad (5.23)$$

since $x_3 - x_2 = 1$ and $z_2 - z_3 = 1$. This shows that the constraints obtained numerically (5.20) match the geometry constraints (5.21) – (5.23) at each triple of control points.

Remark 5.2. The above two examples are found to be consistent with the statement of Conjecture 5.1 that for $\theta \neq 0$ and quadrature points more than 3×2 , $3(n + 1)$ independent constraints are obtained, where n corresponds to the number of bending strip elements having $n + 1$ triples of control points with three independent constraints each.

Remark 5.3. An important point to note in the system (5.13) is that both constraints involve degrees of freedom coming from separate triples of control points, see Fig. 5.1. This is not surprising since the bending strip is constructed such that it only adds a unidirectional stiffness transverse to the length of the strip i.e. along the triples of control points. Absence of any other stiffness leads to no interaction between the different triples involved, although it is not easy to see straight away from the bending strip equations due to its complicated structure.

5.1.3.2 Perturbed Lagrange Method

In this section, we attempt to solve the system (5.5) with the redundant constraints by manipulating the bending strip constraint equations (5.5b) by adding additional penalty terms.

In the bending strip method, the strip has been introduced to serve a particular purpose i.e. joining the patch pairs together while maintaining the joint properties and the equations (5.5b) that it generates are relevant regardless of the redundancy. It motivates us to use the perturbed Lagrange method [ZTZ05] and introduce additional penalty terms in the constraint equations in order to make the equations non-redundant. Applying the method on a two patch system (5.5), we obtain an index-1 DAE system

$$\begin{pmatrix} \mathbf{M}_1 \mathbf{d}_1'' \\ \mathbf{M}_2 \mathbf{d}_2'' \end{pmatrix} + \begin{pmatrix} h^2 \mathbf{f}_1(\mathbf{d}_1) \\ h^2 \mathbf{f}_2(\mathbf{d}_2) \end{pmatrix} + s(\nabla \mathbf{f}_{s,1})^T \hat{\boldsymbol{\lambda}}_1 + s(\mathbf{B}_{C0,1})^T \hat{\boldsymbol{\lambda}}_2 = \begin{pmatrix} h^2 \mathbf{b}_1(t) \\ h^2 \mathbf{b}_2(t) \end{pmatrix}, \quad (5.24a)$$

$$s \mathbf{f}_{s,1}(\mathbf{d}_1, \mathbf{d}_2, 1) - \frac{s}{\alpha_1} \mathbf{C}_1 \hat{\boldsymbol{\lambda}}_1 = \mathbf{0}, \quad (5.24b)$$

$$s \mathbf{B}_{C0,1}(\mathbf{d}_1^T, \mathbf{d}_2^T)^T - \frac{s}{\alpha_2} \mathbf{C}_2 \hat{\boldsymbol{\lambda}}_2 = \mathbf{0}, \quad (5.24c)$$

where α_1, α_2 are the penalty parameters and $\mathbf{C}_1 \in \mathbb{R}^{n_{\text{DOF},s}^1 \times n_{\text{DOF},s}^1}$ and $\mathbf{C}_2 \in \mathbb{R}^{n_{\text{DOF},s}^1/3 \times n_{\text{DOF},s}^1/3}$ are chosen to be the identity matrices \mathbf{I} of the corresponding sizes. For the stationary case, the time dependent terms vanish and applying the Newton method, we obtain the Jacobian

$$\mathbf{J} = \begin{pmatrix} \left(\begin{pmatrix} \nabla \mathbf{f}_1(\mathbf{d}_1) \\ \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{0} \\ \nabla \mathbf{f}_2(\mathbf{d}_2) \end{pmatrix} \right) & (s \nabla \mathbf{f}_{s,1})^T & s(\mathbf{B}_{C0,1})^T \\ \left(\begin{matrix} s \nabla \mathbf{f}_{s,1}(\mathbf{d}_1, \mathbf{d}_2, 1) \\ s \mathbf{B}_{C0,1} \end{matrix} \right) & \begin{pmatrix} -\frac{s}{\alpha_1} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & -\frac{s}{\alpha_2} \mathbf{I} \end{pmatrix} \end{pmatrix}.$$

The presence of the term $\frac{s}{\alpha_1} \mathbf{I}$ helps remove the redundancy of the bending strip force constraints. The application of the perturbed Lagrange method to the alternative formulation gives results that compare well with the single patch and the original bending strip formulation.

Theorem 5.1. *The application of the perturbed Lagrange method results in the system (5.24) that is equivalent to the scaled bending strip system (4.43).*

Proof. We consider two patches connected using a single bending strip. If we take $\mathbf{C}_1 = \mathbf{I}$, $\mathbf{C}_2 = \mathbf{0}$ and substitute $\hat{\boldsymbol{\lambda}}_1$ from (5.24b) into (5.24a), we get the system

$$\begin{pmatrix} \mathbf{M}_1 \mathbf{d}_1'' \\ \mathbf{M}_2 \mathbf{d}_2'' \end{pmatrix} + s \alpha_1 (\nabla \mathbf{f}_{s,1})^T \mathbf{f}_{s,1}(\mathbf{d}_1, \mathbf{d}_2, 1) + s(\mathbf{B}_{C0,1})^T \hat{\boldsymbol{\lambda}}_2 = \begin{pmatrix} \tilde{\mathbf{f}}_1(t) \\ \tilde{\mathbf{f}}_2(t) \end{pmatrix}, \quad (5.25a)$$

$$s \mathbf{B}_{C0,1}(\mathbf{d}_1^T, \mathbf{d}_2^T)^T = \mathbf{0}. \quad (5.25b)$$

$$\begin{pmatrix} \tilde{\mathbf{f}}_1(t) \\ \tilde{\mathbf{f}}_2(t) \end{pmatrix} = \begin{pmatrix} h^2 \mathbf{b}_1(t) \\ h^2 \mathbf{b}_2(t) \end{pmatrix} - \begin{pmatrix} h^2 \mathbf{f}_1(\mathbf{d}_1) \\ h^2 \mathbf{f}_2(\mathbf{d}_2) \end{pmatrix}.$$

This equation system is a modified version of the scaled bending strip method (4.43) with the penalty parameter β_1 replaced by $s \alpha_1$, which can be moved inside the function $\mathbf{f}_{s,1}$ due to the choice of a linear material law, see Remark 4.4, and the bending strip force Jacobian $(\nabla \mathbf{f}_{s,1})^T$ acts as an additional multiplier. \square

We conclude from Theorem 5.1 that removing the redundancy in the bending strip constraints by manipulating the equations via the application of the perturbed Lagrange approach is not a better option than the scaled bending strip method. Both the methods are equivalent. In the next section, we use linear algebra to work with independent constraints in each Newton step instead, since it does not introduce any new penalty parameters and gives the required full rank Jacobian for the constrained mechanical system.

5.1.3.3 Using Linear Algebra

The goal in this section is to present a general approach to treat redundancies in the constraint equations from the mechanical systems. The notations that we use in this section are the same as for mechanical systems from Section 2.2. The equations of the alternative formulation with the bending strips (5.5) has a similar structure as the equations of the constrained mechanical systems (2.37). Hence the technique discussed here can also be used to treat the redundancy issues of the alternative formulation.

It is customary in the literature on mechanical systems to assume the full rank condition on the constraint Jacobian from the outset, see Remark 2.1. This condition in other words

implies that the constraints $\mathbf{g}(\mathbf{q})$ should be independent. But it is often the case that the full rank condition (2.30) is not fulfilled due to various reasons such as the choice of coordinates or constraints, structure of the problem, etc. An example of the redundant constraints is equating the three components of a three dimensional unit vector, when only two are sufficient due to the unit length constraint.

Such problems with redundant constraints are not very difficult to deal with as we will see later in this section. An extensive study to deal with such problems is discussed in De Jalón and Lòpez [dJL13], where the idea is to employ techniques from linear algebra to work with the independent parts of the constraints. The techniques such as the Gaussian elimination, singular value decomposition and QR-decomposition have been used to deal with redundant constraints in mechanical systems in Wehage and Haug [WH82], Mani et al. [MHA85] and Kim and Vanderploeg [KV86] respectively. We use the Gaussian elimination technique since it is the cheapest and simplest option to decompose the constraint Jacobian in each Newton step and identify its independent rows.

We now deal with the index-3 mechanical system with redundant constraints following [dJL13]. Using a standard solution procedure, we take a multibody system with scaled equations of motion (2.58) and time integrate it using as an example the implicit Euler scheme to obtain the nonlinear system (2.61) in each time window (t_i, t_f) as in Section 2.2.4.2. This is solved using the Newton method, where in each iteration step j a linearized problem (2.62) is solved.

But in this section, we consider systems with redundancy in the constraint equations (2.58b) and therefore the system Jacobian \mathbf{J}_j from (2.63)

$$\mathbf{J}_j = \begin{pmatrix} \mathbf{M}_f - h^2 \nabla \mathbf{f}_f & s \mathbf{G}_f^T \\ s \mathbf{G}_f & \mathbf{0} \end{pmatrix}_j$$

is rank deficient, due to the degeneracy of the constraint Jacobian matrix \mathbf{G}_f i.e.

$$\text{rank}(\mathbf{G}_f) := r < n_\lambda. \quad (5.26)$$

This leads to problems in solving the linearized system $\mathbf{J}_j \Delta \mathbf{x}_j = -\mathbf{c}_j$. To correct this, we propose an algorithm, where using a technique from linear algebra we identify the independent part of the constraint Jacobian and use the information to solve an equivalent modified linearized problem in each Newton step for each time window.

In Algorithm 5.1, the *while* loop represents the Newton iteration steps. In each iteration, we apply the *Gaussian elimination with full pivoting* [TB97] to obtain the following decomposition

$$(\mathbf{P}_{n_\lambda \times n_\lambda} \mathbf{G}_{n_\lambda \times n_q} \mathbf{Q}_{n_q \times n_q})_{f,j} = \left(\mathbf{L}_{n_\lambda \times n_\lambda} \begin{bmatrix} \mathbf{U}_{r \times n_q} \\ \mathbf{0}_{(n_\lambda - r) \times n_q} \end{bmatrix} \right)_{f,j}, \quad (5.27)$$

where \mathbf{P} , \mathbf{Q} are the row and column permutation matrices respectively, \mathbf{L} is an easily invertible lower triangular matrix with 1s on the diagonal, \mathbf{U} is an upper triangular matrix with the independent information of the matrix \mathbf{G} . The subscripts mention the size of the matrices and f, j represent the evaluation at final time t_f in each time window and the Newton step j respectively.

Using this LU decomposition we make changes to the system Jacobian \mathbf{J}_j and transform it to a reduced full rank matrix $\tilde{\mathbf{J}}_j$ as mentioned in Algorithm 5.1, line 4. We also make the

Algorithm 5.1 Pseudo-code to solve mechanical systems with redundant constraints using Newton method with modifications

1: begin Newton iterations on (2.61) with $j = 0$, $\|\tilde{\mathbf{c}}_0\|_2 := \|\mathbf{c}_0\|_2$, $\|\Delta\tilde{\mathbf{x}}_{-1}\|_2 := 2tol_2$

2: **while** $\|\tilde{\mathbf{c}}_j\|_2 > tol_1$ & $\|\Delta\tilde{\mathbf{x}}_{j-1}\|_2 > tol_2$ **do**

3: Gaussian elimination $\rightarrow (\mathbf{P}_{n_\lambda \times n_\lambda} \mathbf{G}_{n_\lambda \times n_q} \mathbf{Q}_{n_q \times n_q})_{f,j} = \left(\mathbf{L}_{n_\lambda \times n_\lambda} \begin{bmatrix} \mathbf{U}_{r \times n_q} \\ \mathbf{0}_{(n_\lambda - r) \times n_q} \end{bmatrix} \right)_{f,j}$

4: solve the equivalent reduced system $\tilde{\mathbf{J}}_j \Delta\tilde{\mathbf{x}}_j = -\tilde{\mathbf{c}}_j$, with $\Delta\tilde{\mathbf{x}}_j = (\Delta\mathbf{q}_f^T, (\Delta\tilde{\boldsymbol{\lambda}}_f)_{1:r}^T)^T_{f,j}$,

$$\tilde{\mathbf{J}}_j = \begin{pmatrix} \mathbf{M}_f - h^2 \nabla \mathbf{f}_f & s \mathbf{Q}_f \mathbf{U}_f^T \\ s \mathbf{U}_f \mathbf{Q}_f^T & \mathbf{0} \end{pmatrix}_j, \quad \tilde{\mathbf{c}}_j = \begin{pmatrix} \mathbf{M}_f(\mathbf{q}_f - \mathbf{q}_i - h\mathbf{v}_i) + s \mathbf{Q}_f \mathbf{U}_f^T (\tilde{\boldsymbol{\lambda}}_f)_{1:r} - h^2 \mathbf{f}_f \\ s(\mathbf{L}_f^{-1} \mathbf{P}_f \mathbf{g}_f)_{1:r} \end{pmatrix}_j,$$

where $\tilde{\boldsymbol{\lambda}}_{f,j} = (\mathbf{L}_f^T \mathbf{P}_f \boldsymbol{\lambda}_f)_j$, $(\cdot)_{1:r}$ implies the first r vector or matrix rows

5: $\tilde{\mathbf{x}}_{j+1} = \tilde{\mathbf{x}}_j + \Delta\tilde{\mathbf{x}}_j$, $j \leftarrow j + 1$

6: **end while**

corresponding changes to the residual on the right hand side \mathbf{c}_j and the system unknowns $\Delta\mathbf{x}_j$ to obtain $\tilde{\mathbf{c}}_j$ as in the algorithm and $\Delta\tilde{\mathbf{x}}_j := (\Delta\mathbf{q}_f^T, (\Delta\tilde{\boldsymbol{\lambda}}_f)_{1:r}^T)^T_{f,j}$ respectively. The reduced system stays equivalent to the original one, see Theorem 5.2.

The reduced equivalent system $\tilde{\mathbf{J}}_j \Delta\tilde{\mathbf{x}}_j = -\tilde{\mathbf{c}}_j$ is solved for the system unknowns in each iteration step and Newton update $\tilde{\mathbf{x}}_{j+1} = \tilde{\mathbf{x}}_j + \Delta\tilde{\mathbf{x}}_j$ is done until the convergence is achieved. We assume that the tolerance for the convergence of the Newton method is tol_1 for the residual $\|\tilde{\mathbf{c}}_j\|_2$ and tol_2 for the system update $\|\Delta\tilde{\mathbf{x}}_j\|_2$. This process is repeated for each time window until the end of simulation.

It is to be noted that although the Jacobian matrix and its components that appear as a result of the isogeometric discretization are sparse, we do not utilize this sparsity in our application of the Gaussian elimination on the constraint matrix in (5.27). It is because the number of constraints involved are quite small relative to the Jacobian size for our desired applications.

Remark 5.4. *In the Algorithm 5.1, we take the case where the mechanical system is time integrated using the implicit Euler scheme from Section 2.2.4.2. The algorithm is however valid for other time integration schemes.*

Theorem 5.2. *The linearized equations $\mathbf{J}_j \Delta\mathbf{x}_j = -\mathbf{c}_j$ and $\tilde{\mathbf{J}}_j \Delta\tilde{\mathbf{x}}_j = -\tilde{\mathbf{c}}_j$ from each iteration step j in Algorithm 5.1 are equivalent.*

Proof. The proof follows on the lines of the theory in [dJL13]. To avoid lengthy expressions, we make some changes that do not affect the proof. We drop the subscripts f, j and the matrix dimensions and put the mass matrix $\mathbf{M} = \mathbf{0}$. These changes are restored at the end to get the exact terms as in the algorithm.

We now start with the original system $\mathbf{J} \Delta\mathbf{x} = -\mathbf{c}$. After applying the changes mentioned above to it, we get

$$\begin{pmatrix} -h^2 \nabla \mathbf{f} & s \mathbf{G}^T \\ s \mathbf{G} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \Delta \mathbf{q} \\ \Delta \hat{\boldsymbol{\lambda}} \end{pmatrix} = - \begin{pmatrix} s \mathbf{G}^T \hat{\boldsymbol{\lambda}} - h^2 \mathbf{f} \\ s \mathbf{g} \end{pmatrix}. \quad (5.28)$$

Using the orthogonality property of the permutation matrix \mathbf{P} gives us

$$\begin{pmatrix} -h^2\nabla\mathbf{f} & s(\mathbf{P}\mathbf{G})^T \\ s\mathbf{P}\mathbf{G} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \Delta\mathbf{q} \\ \mathbf{P}\Delta\hat{\boldsymbol{\lambda}} \end{pmatrix} = - \begin{pmatrix} s(\mathbf{P}\mathbf{G})^T(\mathbf{P}\hat{\boldsymbol{\lambda}}) - h^2\mathbf{f} \\ s\mathbf{P}\mathbf{g} \end{pmatrix}. \quad (5.29)$$

The Gaussian elimination with full pivoting (5.27) leads to

$$\begin{pmatrix} -h^2\nabla\mathbf{f} & s\mathbf{Q}(\mathbf{U}^T & \mathbf{0}^T)\mathbf{L}^T \\ s\mathbf{L} \begin{pmatrix} \mathbf{U} \\ \mathbf{0} \end{pmatrix} \mathbf{Q}^T & \mathbf{0} \end{pmatrix} \begin{pmatrix} \Delta\mathbf{q} \\ \mathbf{P}\Delta\hat{\boldsymbol{\lambda}} \end{pmatrix} = - \begin{pmatrix} s\mathbf{Q}(\mathbf{U}^T & \mathbf{0}^T)\mathbf{L}^T\mathbf{P}\hat{\boldsymbol{\lambda}} - h^2\mathbf{f} \\ s\mathbf{P}\mathbf{g} \end{pmatrix}. \quad (5.30)$$

Since the square lower triangular matrix \mathbf{L} is invertible and moreover substituting

$$\tilde{\boldsymbol{\lambda}} := \mathbf{L}^T\mathbf{P}\hat{\boldsymbol{\lambda}}, \quad (5.31)$$

leads to

$$\begin{pmatrix} -h^2\nabla\mathbf{f} & s\mathbf{Q}(\mathbf{U}^T & \mathbf{0}^T) \\ s \begin{pmatrix} \mathbf{U} \\ \mathbf{0} \end{pmatrix} \mathbf{Q}^T & \mathbf{0} \end{pmatrix} \begin{pmatrix} \Delta\mathbf{q} \\ \Delta\tilde{\boldsymbol{\lambda}} \end{pmatrix} = - \begin{pmatrix} s\mathbf{Q}(\mathbf{U}^T & \mathbf{0}^T)\tilde{\boldsymbol{\lambda}} - h^2\mathbf{f} \\ s\mathbf{L}^{-1}\mathbf{P}\mathbf{g} \end{pmatrix}. \quad (5.32)$$

Assuming the last $n_\lambda - r$ constraints \mathbf{g}_i , $i = r+1, \dots, n_\lambda$ to be compatible due to redundancy, it follows that

$$(\mathbf{L}^{-1}\mathbf{P}\mathbf{g})_{r+1} = \dots = (\mathbf{L}^{-1}\mathbf{P}\mathbf{g})_{n_\lambda} = 0. \quad (5.33)$$

We obtain hence a reduced system

$$\begin{pmatrix} -h^2\nabla\mathbf{f} & s\mathbf{Q}\mathbf{U}^T \\ s\mathbf{U}\mathbf{Q}^T & \mathbf{0} \end{pmatrix} \begin{pmatrix} \Delta\mathbf{q} \\ (\Delta\tilde{\boldsymbol{\lambda}})_{1:r} \end{pmatrix} = - \begin{pmatrix} s\mathbf{Q}\mathbf{U}^T(\tilde{\boldsymbol{\lambda}})_{1:r} - h^2\mathbf{f} \\ s(\mathbf{L}^{-1}\mathbf{P}\mathbf{g})_{1:r} \end{pmatrix}, \quad (5.34)$$

which after the re-substitution of the neglected terms and the subscripts f, j is the same as $\tilde{\mathbf{J}}_j\Delta\tilde{\mathbf{x}}_j = -\tilde{\mathbf{c}}_j$ in the Algorithm 5.1. \square

Lemma 5.2. *The Lagrange multipliers $\hat{\boldsymbol{\lambda}}$ are undetermined, nonetheless the constraint forces $\mathbf{G}^T\hat{\boldsymbol{\lambda}}$ are fully determined due to the special choice of Lagrange multipliers $\tilde{\boldsymbol{\lambda}}$.*

Proof. From (5.31), we get $\hat{\boldsymbol{\lambda}} = \mathbf{P}^T\mathbf{L}^{-T}\tilde{\boldsymbol{\lambda}}$. Since $\tilde{\lambda}_{r+1}, \tilde{\lambda}_{r+2}, \dots, \tilde{\lambda}_{n_\lambda}$ are multiplied with null entries in (5.32), they can take arbitrary values leading to $\tilde{\boldsymbol{\lambda}}$ and in turn $\hat{\boldsymbol{\lambda}}$ not being fully determined. The net constraint forces $\mathbf{G}^T\hat{\boldsymbol{\lambda}}$ can be expressed as

$$\mathbf{G}^T\hat{\boldsymbol{\lambda}} = \mathbf{G}^T\mathbf{P}^T\mathbf{L}^{-T}\tilde{\boldsymbol{\lambda}} = \mathbf{Q}(\mathbf{U}^T & \mathbf{0}^T)\mathbf{L}^T\mathbf{L}^{-T}\tilde{\boldsymbol{\lambda}} = \mathbf{Q}\mathbf{U}^T(\tilde{\boldsymbol{\lambda}})_{1:r}.$$

\square

For nonlinear constraints, in each iteration step the constraint Jacobian is newly computed and might undergo changes. It is also possible that due to the deformation, some constraints become inactive and some new ones are activated. Also, in the LU decomposition, the constraints might change their order. In each iteration step j , the first r Lagrange multipliers $\tilde{\boldsymbol{\lambda}}$ are calculated based on the LU decomposition in this step. Following this, in the next step $j+1$ the constraint forces are computed based on the Lagrange multipliers that are functions of the LU decomposition from step j and the matrices from LU decomposition from step $j+1$. For changing constraints, this can lead to instability in the residual computation and hence the overall convergence of the Newton method. Note that for linear cases, only one Newton iteration is performed and this method works well for sure.

Remark 5.5. *In context of the above mentioned issue with the nonlinear constraints, one can also use a reduced form of (5.29)*

$$\begin{pmatrix} -h^2 \nabla \mathbf{f} & s(\mathbf{P}\mathbf{G})_{1:r}^T \\ s(\mathbf{P}\mathbf{G})_{1:r} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \Delta \mathbf{q} \\ (\mathbf{P}\Delta \hat{\boldsymbol{\lambda}})_{1:r} \end{pmatrix} = - \begin{pmatrix} s(\mathbf{P}\mathbf{G})_{1:r}^T (\mathbf{P}\hat{\boldsymbol{\lambda}})_{1:r} - h^2 \mathbf{f} \\ s(\mathbf{P}\mathbf{g})_{1:r} \end{pmatrix} \quad (5.35)$$

as an alternate linearized system in Algorithm 5.1. (5.35) uses r independent rows of the permuted constraint Jacobian \mathbf{G} and Lagrange multiplier $\hat{\boldsymbol{\lambda}}$. This is on one hand very easy to implement, since only the permutation information \mathbf{P} is used from the LU decomposition to swap the rows of the constraint Jacobian, the Lagrange multiplier and the right hand side. On the other hand, it does not compute the exact constraint force $\mathbf{G}^T \hat{\boldsymbol{\lambda}}$, but instead a partial force $(\mathbf{P}\mathbf{G})_{1:r}^T (\mathbf{P}\hat{\boldsymbol{\lambda}})_{1:r}$.

5.2 Alternative Formulation with Continuity Constraints

In Section 5.1, the alternative formulation to the bending strip method (5.5) involves the redundant bending strip constraints (5.5b). These constraints on one hand give us the behavior of the bending strip method in the limit case, which is hard to obtain from the original method due to ill-conditioning due to the penalty parameter, but on the other hand they are redundant and require some additional treatment which in itself requires some effort.

In this section, we propose a different technique where we use continuity constraints to join shell patches as proposed in [KBH⁺10], instead of the bending strip constraints (5.5b). These constraints again require conforming discretization of the connecting patches and preserve the G^1 continuity in case of smooth intersection and the angle in case of kinks at the patch intersection.

Suppose we observe a triple $(\mathbf{P}_{n-1}^1, \mathbf{P}_n^1 = \mathbf{P}_1^2, \mathbf{P}_2^2)$ from two neighboring conforming patches 1,2 that connect to each other. For patch 1, $\mathbf{P}_n^1, \mathbf{P}_{n-1}^1$ are the control points on the boundary and one adjacent to it, whereas these are $\mathbf{P}_1^2, \mathbf{P}_2^2$ for patch 2. Since patch 1,2 are connected, $\mathbf{P}_n^1 = \mathbf{P}_1^2$.

For the G^1 continuity, the tangent planes of the connecting patches are common, which for B-Spline patches leads to the sufficient but not necessary condition

$$(\mathbf{P}_n^1 - \mathbf{P}_{n-1}^1) = c (\mathbf{P}_2^2 - \mathbf{P}_1^2), \quad (5.36)$$

where c is a scalar that can be determined from the initial configuration. (5.36) stems from the first derivative formulas in across the intersection direction similar to (4.7). Due to our assumption of conforming patches i.e. having the same parametrization along the common edge, the first derivative are the same in along the intersection direction.

In case of a kink, the angle θ is preserved through the scalar product formula

$$(\mathbf{P}_n^1 - \mathbf{P}_{n-1}^1) \cdot (\mathbf{P}_2^2 - \mathbf{P}_1^2) = |\mathbf{P}_n^1 - \mathbf{P}_{n-1}^1| |\mathbf{P}_2^2 - \mathbf{P}_1^2| \cos \theta. \quad (5.37)$$

When setting the boundary conditions for NURBS surfaces, homogeneous control variables are to be employed [FHK02]. But since in isogeometric analysis, the weights are not taken as variables, we apply here the condition as for the B-Splines. This gives us approximate constraints and the continuity might not be fulfilled exactly at any point if the weights of the involved control points are different. But since after refinement the weights of the adjacent control point approach a similar value, this error should be quite low.

Remark 5.6. For each triple across the connection, one gets 3 linear constraints (5.36) in case of G^1 continuity, whereas a single nonlinear constraint (5.37) is obtained for angle preservation.

Remark 5.7. For the pair of straight plates connecting at an angle $\theta = 0^\circ$, G^1 continuity is present and applying (5.36) to the two triples from Fig. 5.1 gives the following constraints

Triple 1:

$$d_1 - 2d_2 + d_3 = 0, \quad (5.38a)$$

$$d_7 - 2d_8 + d_9 = 0, \quad (5.38b)$$

$$d_{13} - 2d_{14} + d_{15} = 0. \quad (5.38c)$$

Triple 2:

$$d_4 - 2d_5 + d_6 = 0, \quad (5.39a)$$

$$d_{10} - 2d_{11} + d_{12} = 0, \quad (5.39b)$$

$$d_{16} - 2d_{17} + d_{18} = 0. \quad (5.39c)$$

We notice that the G^1 constraints in the z -direction (5.38c), (5.39c) correspond to the bending strip constraints (5.13a), (5.13b) respectively. It shows that although the bending strip constraints match those from the theory, not all constraints are active. The active ones are configuration dependent and in case of the plate lying in x - y plane with normal in z -direction in the linear deformation setting, they correspond to the constraints in z -direction. While applying the angle preservation using the direct continuity constraints from this section, we however apply all three constraints per triple, since they are non-redundant.

Remark 5.8. For the pair of straight plates connecting at an angle $\theta \neq 0^\circ$, the goal is to preserve the angle θ between the various triples. Hence applying (5.37) gives us one of the three constraints for each triple as presented in Example 5.1, 5.2 with the implementation of bending strips. The other two constraints per triple in the examples are obtained by preserving additionally the flange lengths. This is not a necessary condition and hence it adds extra stiffness to the system in case of the bending strips. While applying the angle preservation using the direct continuity constraints from this section, we skip the length preservation constraints.

It must be noted that in case of overlapping constraints, the continuity constraints induce over constraining in the system, see [KBH⁺10]. Therefore one has to invest extra attention at such points. On the other hand overlapping bending strips pose no trouble in a penalty approach.

5.3 Summary

We discussed in this chapter two alternative formulations in order to remove the penalty parameter dependence that appears from a penalty approach such as the bending strip method. This in turn gets rid of the ill-conditioning that appears from the penalty approach. In Section 5.1, we firstly derived an alternative formulation comparing the bending strip method to the stiff mechanical systems with an aim to tackle it as a stiff system that gives a limit case result of the bending strip method. The system consists of redundant constraints, that are appropriately dealt with. Secondly, in Section 5.2, we proposed another alternative formulation that uses directly the continuity constraints and there is no issue of redundancy in general.

To avoid redundancy issues with the first alternative formulation, in Section 5.1.3.1, we studied the bending strip constraints so as to identify the independent constraints. This is only possible in the linear setting for limited geometry settings and is a bit cumbersome.

In Section 5.1.3.2, we used the redundant alternative formulation along with the perturbed Lagrange method and as seen in Theorem 5.1, it is found to be equivalent to the scaled bending strip method up to some multiplying factor. So there is no evident benefit of using such an approach, since a new penalty parameter dependence is introduced. A third approach, from Section 5.1.3.3, that looks the most promising and general is the one that uses linear algebra in the form of the Gaussian elimination technique in order to identify the independent constraints in each Newton step.

As we see from the theory and will later see from our numerical results in Section 6.4, these alternative formulations show no penalty parameter dependence and are well-conditioned. We now study their pros and cons over the scaled bending strip method.

The advantage of the alternative formulation with bending strip constraints is that the behavior of the bending strip method carries over as it is. The application of the bending strip remains the same and only the numerical formulation changes, giving the limit case behavior of the original method. For this alternative formulation after the successful redundancy treatment and for the other one with direct continuity constraints, there is no dependence on any penalty parameter. A detailed analysis similar to that in Section 4.4.3.2 reveals that in the transient setting, the condition number of the Jacobian matrix is again independent of the time step size order. Moreover with such alternative formulations, a time step size restriction that shows up for stiff mechanical systems and is dependent on the penalty parameter value can be avoided in the transient analysis. The penalty formulations suffer from this dependence and fail to converge due to troubles with the Newton contractivity conditions as shown in [Lub93].

The disadvantages that we get with the two alternative formulations are that additional degrees of freedom in terms of new Lagrange multipliers are introduced due to the new constraints. Also, redundant constraints show up in the system for the alternative formulation with bending strip force constraints. They must be tackled in order to get a well defined system, requiring additional effort. Since we get separate constraints with the alternative formulations, there might be redundancy in constraints at places where more than two patches are joining together. This poses no trouble with a penalty formulation. Moreover for the alternative formulation with direct continuity constraints, the constraints are different for G^1 continuity and C^0 continuity with angle preservation cases, unlike with the bending strip method where both cases are solved with the same technique. Lastly we require a conforming discretization on the adjoining patches in order to connect them using the alternative formulations, which is the same case also with the bending strip method.

The main idea behind choosing the alternative formulation for multipatch shells when connecting them to multibody systems is to utilize the existing structure of the multibody system and tackle the additional constraints arising from the alternative formulation with the constraints already present in the multibody system. Important here is to generate well-conditioned, non-stiff systems, which can be efficiently time integrated without a time step size restriction. It is not always possible with a penalty approach.

Chapter 6

Applications and Examples

This chapter deals with the application of the theory discussed in this thesis. We begin with a discussion about the extraction of the NURBS data from a standard CAD data format, which contains NURBS based geometries. This information is then processed and used to describe the NURBS surface in our shell solver. We explain the implementation of our shell solver for transient multipatch shells. Later we show a technique to pointwise couple the isogeometric shell solver with a multibody system. We finally present some of the important numerical results that validate and show the benefits of using the isogeometric shell discretizations.

6.1 Extracting NURBS Surfaces from CAD Files

The idea behind implementing isogeometric analysis is to be able to use the same set of NURBS basis functions to describe the geometry as well as the system unknowns such as displacements. This gives great benefits like higher regularity and accuracy, amongst others mentioned in Section 4.2.4. To apply it on shells, we start by extracting the NURBS surface information from the CAD based NURBS description, which is used to build the input geometry in our code. The schematic to show the data flow from the standard CAD files to the input files for IGA shell solver is shown in Fig. 6.1 and explained below.

6.1.1 CAD File with NURBS Data

NURBS are commonly used in computer-aided design (CAD), manufacturing (CAM) and engineering (CAE) and are a part of numerous industry wide standards such as IGES, STEP, ACIS and PHIGS [PT97]. This implies that almost all the industrial softwares can access and work with NURBS geometries. NURBS tools are also found in various 3D modeling and animation software packages. They can be efficiently handled by the computer programs and yet allow for easy human interaction.

NURBS surfaces are functions of two parameters mapping to a surface in three dimensional space. The shape of the surface is determined by control points, corresponding weights, knot vectors and the degree of NURBS, see (4.13).

6.1.2 Make Analysis Suitable

In order to design complicated geometries the CAD designers often use surface descriptions for NURBS patches that have a lower continuity than the original part. They at times also divide

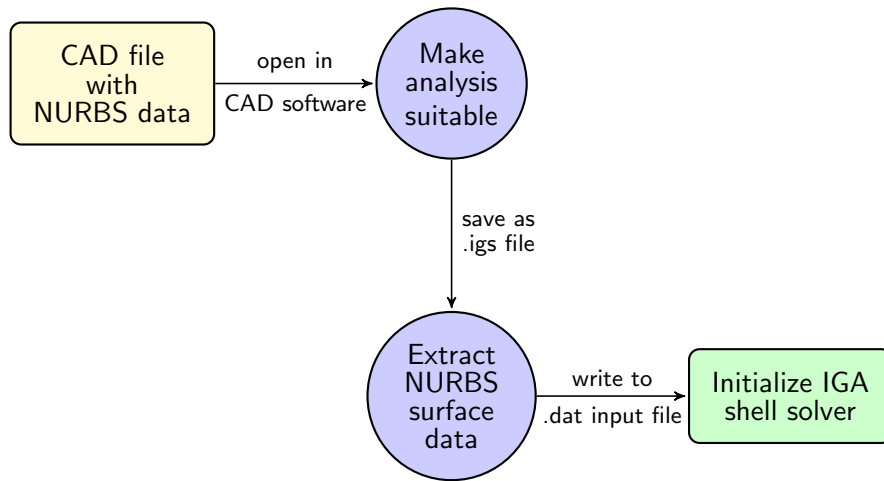


Figure 6.1: Data flow from CAD to NURBS based IGA

an otherwise smooth shape into various patches to facilitate inter-connections, which leads to a lower continuity at the patch intersections. The issue with the former setting is that the geometry might not fulfill the regularity requirements for a shell patch analysis as mentioned in Remark 3.1. For the latter scenario, this increases dofs and hence the computational expense when analyzing such models using multipatch solvers. Hence it is useful to preprocess the NURBS description to make it suitable and efficient for the application of isogeometric analysis.

For preprocessing, we load the geometry in a CAD software and rebuild it to have lesser patches with desired smoothness. After this step we save the file with an extension ".igs" in IGES file format.

6.1.3 Extract NURBS Surface Data

The NURBS surface data is listed in the ".igs" file under the entity of type 128. There are various ways to extract this information from the file. One possibility is the IGES toolbox in `Matlab` that can read the NURBS data from an ".igs" file. Another possibility is to load the file in the CAD program `Rhino` and use the `list` command to display the NURBS information.

The next step is to write the NURBS geometry information to a file which the IGA code can read. In our case, the isogeometric shell solver implemented in C++ reads a ".dat" file format created in-house by a former colleague. This file has all the relevant information needed by the IGA solver for its initialization and solution of the shell problem, see Section 6.2.2.

6.1.4 Initialize IGA Shell Solver

Based on the NURBS surface information, it is possible in our IGA shell solver, discussed in Section 6.2, to read multipatch shell geometries and automatically create a NURBS surface from it.

Below we present an example that follows the schematic of Fig. 6.1 and extracts NURBS data from a CAD geometry to serve as input for the IGA solver ".dat" file.

Example 6.1. We consider a twist beam rear axle with a complicated CAD geometry which is divided into several small patches, see Fig. 6.2. The figure depicts a ".igs" file with NURBS geometry loaded as input in the CAD software *Rhino*. It offers the possibility to display the control points and make modifications to the geometry. This marks the first step in the data flow from CAD to IGA from Fig. 6.1.

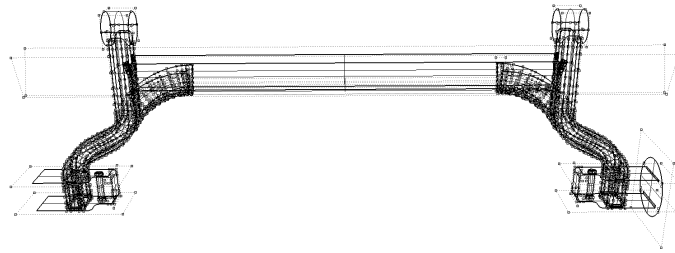


Figure 6.2: Twist beam rear axle geometry and its control points marked as dots

A closer look at the axle mid-section in Fig. 6.3 shows the breakage of a simple geometry into various small parts with lots of control points due to its interconnections in Fig. 6.2. Also the NURBS data suggested that at some points the geometry was only C^0 continuous purely due to the CAD modeling. To improve it for analysis, we approximate this geometry with a single patch geometry for the mid-section such that it is smooth and has lesser number of control points, which which directly correspond to the number of dofs for our analysis. The modifications begin by extending the edges of some surfaces so that they overlap the other smaller patches by using commands such as *EdgeSrf*, *MergeSrf*, *ExtendSrf* in *Rhino*. The overlapped patches are then removed. Once a single patch is obtained, the control points are reduced by managing the NURBS degree and the number of knots in each direction using the *Rebuild* command. This creates an approximation of the finer geometry. We rebuild until there is a good balance between the number of control points and the geometric accuracy. This step requires some experience. Finally this analysis suitable geometry, see e.g. Fig. 6.4, is saved as ".igs" file, which contains the new NURBS description.



Figure 6.3: Enlarged mid-section of twist beam rear axle and its irregular control grid

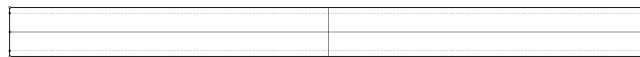


Figure 6.4: Enlarged mid-section of twist beam rear axle after modifications

The NURBS data is obtained from the ".igs" file by using the *list* command in *Rhino*, which displays the complete geometry information of the NURBS object. With this procedure we create a simplified axle model which is an approximation to the much complicated initial geometry, see Fig. 6.5, created with a goal to study majorly the effects of the mid-section. Finally we write the data to the ".dat" file, which is an input format for our IGA shell solver.

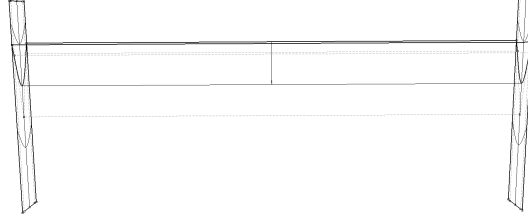


Figure 6.5: Simplified version of the twist beam rear axle used for IGA

6.2 IGA Shell Solver

In this section, we explain the functioning of our IGA shell solver that is implemented in C++. In Section 6.2.1, we comment on the underlying equations that are solved to analyze the elasticity problem on multipatch shells with a given geometry using the approach of isogeometric analysis. Then in Section 6.2.2, we present the sequence of steps that are carried out in order to obtain the desired analysis results from the shell problem.

6.2.1 Underlying Equation System

The schematic in the next section refers to the solution process of a multipatch NURBS geometry that either uses the bending strip equations (4.43) or the alternative formulation (5.5) with redundancy treatment using Algorithm 5.1 in the Newton step.

We consider in this case the scaled two patch system (5.5), although we follow a general procedure which can solve a wider class of shell problems with arbitrary number of patches. Taking (5.5)

$$\begin{aligned} \begin{pmatrix} \mathbf{M}_1 \mathbf{d}_1'' \\ \mathbf{M}_2 \mathbf{d}_2'' \end{pmatrix} + \begin{pmatrix} h^2 \mathbf{f}_1(\mathbf{d}_1) \\ h^2 \mathbf{f}_2(\mathbf{d}_2) \end{pmatrix} + s(\nabla \mathbf{f}_{s,1})^T \hat{\boldsymbol{\lambda}}_1 + s(\mathbf{B}_{C0,1})^T \hat{\boldsymbol{\lambda}}_2 &= \begin{pmatrix} h^2 \mathbf{b}_1(t) \\ h^2 \mathbf{b}_2(t) \end{pmatrix}, \\ s \mathbf{f}_{s,1}(\mathbf{d}_1, \mathbf{d}_2, 1) &= \mathbf{0}, \\ s \mathbf{B}_{C0,1}(\mathbf{d}_1^T, \mathbf{d}_2^T)^T &= \mathbf{0}, \end{aligned}$$

we solve the system at the normalized times $\tau_0 < \tau_1 < \dots < \tau_N$, where $N+1$ are the number of discretized time points. We assume here a fixed time interval $\tau_{n+1} - \tau_n = k$, $n = 0, \dots, N-1$. The shell problem is an initial value problem of the kind (2.11) with known initial values of the unknown variables $\mathbf{d}_1, \mathbf{d}_2, \hat{\boldsymbol{\lambda}}_1, \hat{\boldsymbol{\lambda}}_2$ at time τ_0 . Applying the implicit Euler method as the time integration scheme for our example, we get for each time window $(\tau_n, \tau_{n+1}), 0 \leq n \leq N-1$, the time discretized system

$$\begin{aligned} \begin{pmatrix} \mathbf{M}_1(\mathbf{d}_{1,n+1} - \mathbf{d}_{1,n} - k\mathbf{v}_{1,n}) \\ \mathbf{M}_2(\mathbf{d}_{2,n+1} - \mathbf{d}_{2,n} - k\mathbf{v}_{2,n}) \end{pmatrix} + \begin{pmatrix} k^2 h^2 \mathbf{f}_1(\mathbf{d}_{1,n+1}) \\ k^2 h^2 \mathbf{f}_2(\mathbf{d}_{2,n+1}) \end{pmatrix} \\ + k^2 s(\nabla \mathbf{f}_{s,1})^T \hat{\boldsymbol{\lambda}}_{1,n+1} + k^2 s(\mathbf{B}_{C0,1})^T \hat{\boldsymbol{\lambda}}_{2,n+1} &= \begin{pmatrix} k^2 h^2 \mathbf{b}_1(\tau_{n+1}) \\ k^2 h^2 \mathbf{b}_2(\tau_{n+1}) \end{pmatrix}, \end{aligned} \quad (6.2a)$$

$$s \mathbf{f}_{s,1}(\mathbf{d}_{1,n+1}, \mathbf{d}_{2,n+1}, 1) = \mathbf{0}, \quad (6.2b)$$

$$s \mathbf{B}_{C0,1}(\mathbf{d}_{1,n+1}^T, \mathbf{d}_{2,n+1}^T)^T = \mathbf{0}. \quad (6.2c)$$

The discretized system (6.2) is nonlinear with respect to the displacements $\mathbf{d}_{1,n+1}$, $\mathbf{d}_{2,n+1}$ at time τ_{n+1} . The discretized velocities are given by $\mathbf{v}_{\alpha,n} = \frac{\mathbf{d}_{\alpha,n} - \mathbf{d}_{\alpha,n-1}}{k}$, $\alpha = 1, 2$. To solve this nonlinear system, we apply the Newton method and solve in each Newton step i the linearized system

$$\mathbf{J}_{n+1}^i \Delta \mathbf{x}_{n+1}^i = -\mathbf{c}_{n+1}^i, \quad \text{where} \quad (6.3)$$

$$\mathbf{J}_{n+1}^i = \begin{pmatrix} \left(\begin{array}{cc} \mathbf{M}_1 + k^2 h^2 \nabla \mathbf{f}_1(\mathbf{d}_{1,n+1}^i) & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_2 + k^2 h^2 \nabla \mathbf{f}_2(\mathbf{d}_{2,n+1}^i) \end{array} \right) & k^2 s \nabla \mathbf{f}_{s,1}^T & k^2 s \mathbf{B}_{C0,1}^T \\ s \nabla \mathbf{f}_{s,1}(\mathbf{d}_{1,n+1}^i, \mathbf{d}_{2,n+1}^i, 1) & \mathbf{0} & \mathbf{0} \\ s \nabla \mathbf{B}_{C0,1} & \mathbf{0} & \mathbf{0} \end{pmatrix},$$

\mathbf{c}_{n+1}^i is the residual of (6.2) evaluated at

$$\mathbf{x}_{n+1}^i = ((\mathbf{d}_{1,n+1}^i)^T, (\mathbf{d}_{2,n+1}^i)^T, (\hat{\boldsymbol{\lambda}}_{1,n+1}^i)^T, (\hat{\boldsymbol{\lambda}}_{2,n+1}^i)^T)^T$$

and update the unknown

$$\mathbf{x}_{n+1}^{i+1} = \mathbf{x}_{n+1}^i + \Delta \mathbf{x}_{n+1}^i. \quad (6.4)$$

Solving (6.3) and updating (6.4) is done until the norm of the residual reaches below a tolerance *tol* or the maximum number of Newton steps i_{\max} are exceeded. The process is repeated for each time step until the end of simulation.

6.2.2 Information Flow

In order to solve a multipatch shell elasticity problem on a computer, the input information is collected and based on this data, the geometry is setup and the solver is prepared so that it can deliver the desired results. The schematic in Fig. 6.6 shows us the step by step information flow to solve an elasticity problem.

The first step is to collect the input data regarding the geometry, material, boundary conditions, solver, quadrature rule and plot information. This can be done in C++ using various ways such as through command line arguments, step by step input through a separately programmed graphical user interface, reading an input file, etc. Our code is setup to read an input ".dat" file. This file gives a complete problem description based on which the code runs.

The geometry of the shell is constructed with the help of the NURBS information read from the input ".dat" file. In case of multiple patches, we read multiple ".dat" files each defining the geometry of an individual patch. Then surface objects corresponding to each patch are created using additionally the material properties such as E-modulus, Poisson ratio, shell thickness, etc. Also refinement information is provided in the input file, which defines the spatial discretization of the surfaces for the analysis. The refinement is realized by applying a k-refinement procedure as explained in Section 4.2.2.

In the next step to solve (5.5), a pre-evaluation of the relevant data in the initial configuration is carried out. This includes the evaluation of the components like basis functions, initial centerline position, slopes, etc., which assist in the computation of the strains in (3.11), (3.12) and therefrom the stresses and their derivatives. On the one hand it saves the time and effort of having to compute all the initial data repeatedly at each time point for each quadrature point in each element for each patch. On the other hand it is taxing for the system memory, since all this data is stored up in the system RAM during the program run time. One has to therefore strike a balance between the speed of the program and the RAM consumption.

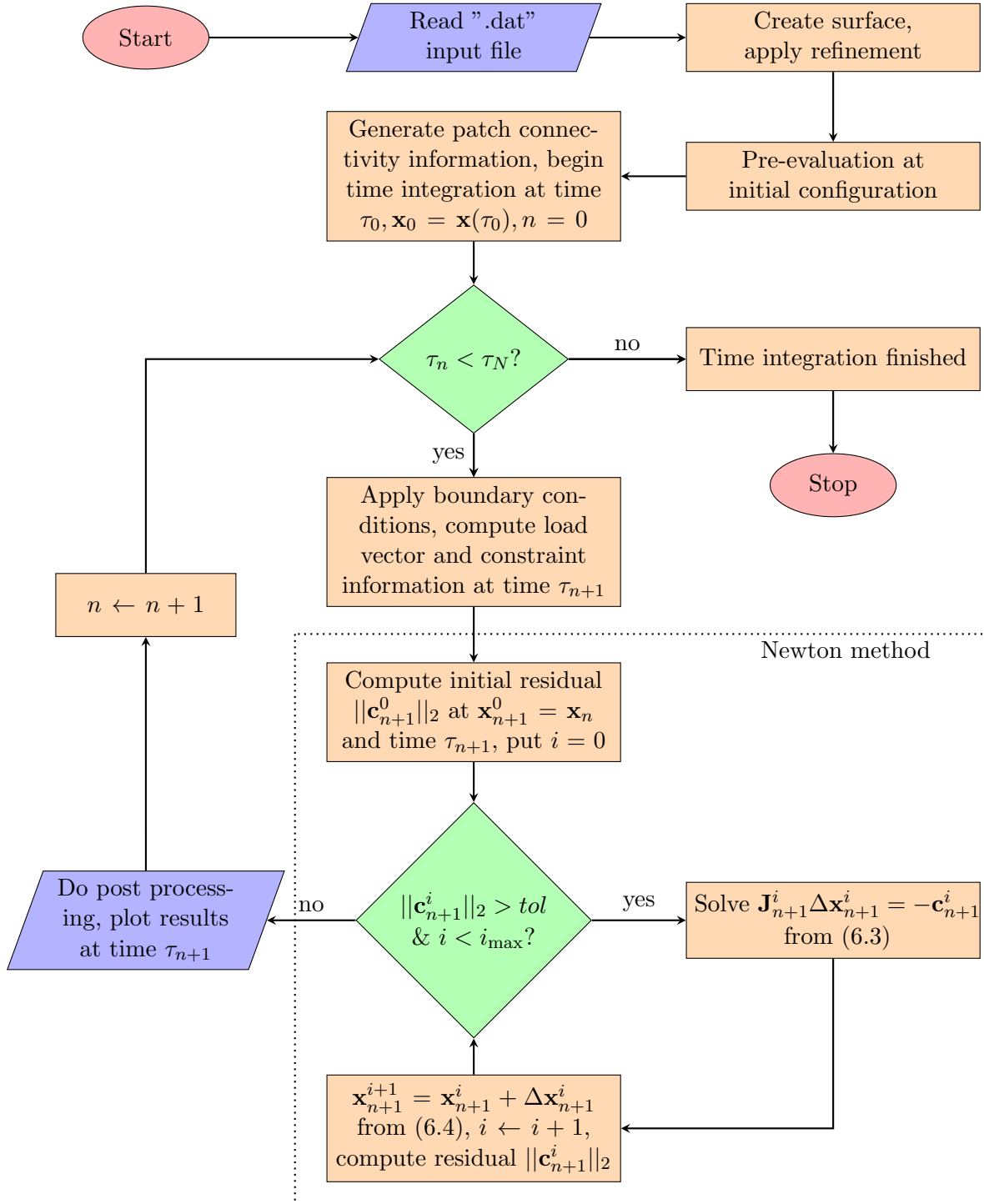


Figure 6.6: Schematic of the IGA shell solver

The patch connectivity information is then created by observing the boundary control points of the patches. Those patches which have the same control points on a boundary are joined with each other and the connectivity information such as the triples of control points as explained in Fig. 4.7 and the joint properties are saved for the implementation of the various multipatch approaches discussed in the thesis.

Next the system (5.5) is time integrated for normalized times $\tau_0 < \tau_1 < \dots < \tau_N$, where $N + 1$ are the number of discretized time points. This is an initial value problem with a pre-specified initial value $\mathbf{x}_0 = \mathbf{x}(\tau_0)$. Also we initialize the time integration step counter $n = 0$.

As an example, we apply the implicit Euler method for the time integration and obtain the equations (6.2) that are solved at times τ_{n+1} , $n = 0, \dots, N - 1$ for the unknowns

$$\mathbf{x}_{n+1} = ((\mathbf{d}_{1,n+1})^T, (\mathbf{d}_{2,n+1})^T, (\hat{\boldsymbol{\lambda}}_{1,n+1})^T, (\hat{\boldsymbol{\lambda}}_{2,n+1})^T)^T.$$

The boundary conditions computed at time τ_{n+1} are applied in order to define the current constraints and the load vector in each time step. This is done by appending the Lagrangian with the corresponding contributions from the Dirichlet conditions (3.37) and the external energy (3.30). This defines our complete problem where the initial values and the current loading is known. We then compute the corresponding unknowns \mathbf{x}_{n+1} in (6.2) using the Newton method. This is marked inside the dotted lines in the schematic from Fig. 6.6.

To begin the Newton iterations for the nonlinear problem (6.2) at time τ_{n+1} , we compute the initial residual $\|\mathbf{c}_{n+1}^0\|_2$ of (6.2) using the converged value from the last time integration step i.e. $\mathbf{x}_{n+1}^0 = \mathbf{x}_n$ such that $\mathbf{x}_n := \mathbf{x}_n^i$, where the residual is below the tolerance $\|\mathbf{c}_n^i\|_2 = \|\mathbf{c}(\mathbf{x}_n^i)\|_2 < tol$. The Newton iteration counter is set to $i = 0$. If the residual is more than the tolerance and the maximum number of Newton iterations i_{\max} are not exceeded, then in each step i a linearized problem (6.3) is solved to determine \mathbf{x}_{n+1}^i using (6.4). It is important to note that the linearized system (6.3) might not be well defined in some cases, such as the bending strip constraints. For such cases, we incorporate the Algorithm 5.1 in the Newton method. After the update (6.4), the residual is again computed, checked and the steps repeated until the convergence of the Newton method to obtain $\mathbf{x}_{n+1} := \mathbf{x}_{n+1}^i$ such that $\|\mathbf{c}(\mathbf{x}_{n+1}^i)\|_2 < tol$.

Once the Newton method converges, the displacement results are known. A post processing stage then follows to compute other physical quantities such as strains and stresses following Section 3.4. Then the results are written to a *.vtk* file, which is a standard file format that can be loaded in an open source program like **Paraview**. The desired plots can thus be viewed in Paraview for this time step.

The next time step is then observed in a similar fashion as above until the time $\tau_n = \tau_N$. This completes the simulation, which solves the elasticity problem for a multipatch shell system using the theory of IGA.

6.3 MBS-Shell Coupling

In this section, we explain a procedure to analyze isogeometric shells coupled with multibody systems. Firstly in Section 6.3.1 we describe the pointwise coupling, followed by a brief description of the simulation techniques one can use to solve a multibody system containing flexible isogeometric shells in Section 6.3.2.

6.3.1 The Coupling

A flexible multibody system has non-coupled multibody and elastic body equations (2.68), in case the elastic body dofs \mathbf{d} include its elastic as well as rigid motion. Since this is also the case with the higher order nonlinear NURBS in isogeometric analysis, we employ the system (2.68) to define the MBS with isogeometric shells as flexible structures.

In our context it is particularly important to extend the isogeometric approach in such a way that point forces and moments can be applied easily and that the displacement data available in terms of the variables \mathbf{d} is transformed to nodal data on the shell surface such that force laws can be evaluated in the usual way. More specifically we introduce the function $\mathbf{w}(\mathbf{d})$ that computes the motion of a set of given points on the shell surface by means of (4.20).

We now discuss the nodal interaction of the isogeometric shell structure within a flexible MBS framework. Fig. 6.7 shows the pointwise interaction at time t between a rigid body of the MBS at point \mathbf{x}_i with a flexible shell structure at point \mathbf{x}_j . We define two inertial axes I_{MBS} and I_{Shell} with respect to which the absolute state of the MBS and shell is calculated respectively. The vector $\mathbf{r}_R(t)$ and the matrix $\mathbf{A}_R(t)$ represent the translation and rotation respectively of the rigid body at time t , whereas \mathbf{r}_E and \mathbf{A}_E represent that of I_{Shell} with respect to I_{MBS} . The interaction point \mathbf{x}_i on the rigid body has the absolute position

$$\mathbf{x}_i(t) = \mathbf{r}_R(t) + \mathbf{A}_R(t)\mathbf{x}_f, \quad (6.5)$$

where the translation vector \mathbf{r}_R represents the position of a reference point \mathbf{x}_0 and \mathbf{x}_f is an orientation vector of the rigid body.

The absolute position of the interaction point \mathbf{x}_j on the shell centerline at time t with respect to I_{MBS} is

$$\mathbf{x}_j(t) = \mathbf{r}_E + \mathbf{A}_E\mathbf{c}(\vartheta_{1,j}, \vartheta_{2,j}, t), \quad (6.6)$$

where $\mathbf{c}(\vartheta_{1,j}, \vartheta_{2,j}, t) = \mathbf{c}_0(\vartheta_{1,j}, \vartheta_{2,j}) + \mathbf{u}(\vartheta_{1,j}, \vartheta_{2,j}, t)$ from (3.4) is the position of the point \mathbf{x}_j with respect to I_{Shell} and $(\vartheta_{1,j}, \vartheta_{2,j})$ is the parametric coordinate of \mathbf{x}_j .

The interaction force $\mathbf{f}_{ij}(\zeta_{ij}, \dot{\zeta}_{ij}, t)$ acts between the MBS and the shell structure, where $\zeta_{ij}(t) = \|\mathbf{x}_j(t) - \mathbf{x}_i(t)\|_2$ is the distance between \mathbf{x}_i and \mathbf{x}_j at time t . A clamped end or prescribed motion is imparted to the shell through the boundary condition $\mathbf{u}_\Gamma = \mathbf{u}_0(\mathbf{q}, t)$, where \mathbf{q} is the position vector of the MBS.

Since the elastic degrees of freedom \mathbf{d} include the rigid body motion, we obtain from equations (2.68), (2.69) the following system

$$\mathbf{M}_r(\mathbf{q})\ddot{\mathbf{q}} = \mathbf{f}_r(\mathbf{q}, \mathbf{w}(\mathbf{d}), \dot{\mathbf{q}}, \dot{\mathbf{w}}(\mathbf{d}), t) - \frac{\partial \mathbf{g}}{\partial \mathbf{q}}(\mathbf{q}, \mathbf{d})^T \boldsymbol{\lambda}, \quad (6.7a)$$

$$\mathbf{M}_h\ddot{\mathbf{d}} = -\mathbf{K}(\mathbf{d})\mathbf{d} - \mathbf{D}\dot{\mathbf{d}} + \mathbf{b}(\mathbf{d}, \dot{\mathbf{d}}, \mathbf{q}, \dot{\mathbf{q}}, t) - \frac{\partial \mathbf{g}}{\partial \mathbf{d}}(\mathbf{q}, \mathbf{d})^T \boldsymbol{\lambda}, \quad (6.7b)$$

$$\mathbf{0} = \mathbf{g}(\mathbf{q}, \mathbf{d}). \quad (6.7c)$$

The net force on the MBS and the net external force on shell are given as

$$\mathbf{f}_r(\mathbf{q}, \mathbf{w}(\mathbf{d}), \dot{\mathbf{q}}, \dot{\mathbf{w}}(\mathbf{d}), t) = \mathbf{f}_{MBS}(\mathbf{q}, \dot{\mathbf{q}}, t) + \sum_{i,j} \mathbf{f}_{ij,r}(\zeta_{ij}, \dot{\zeta}_{ij}, t), \quad (6.8)$$

$$\mathbf{b}(\mathbf{d}, \dot{\mathbf{d}}, \mathbf{q}, \dot{\mathbf{q}}, t) = \mathbf{p}(\mathbf{d}, \dot{\mathbf{d}}, t) + \sum_{i,j} \mathbf{f}_{ij,h}(\zeta_{ij}, \dot{\zeta}_{ij}, t), \quad (6.9)$$

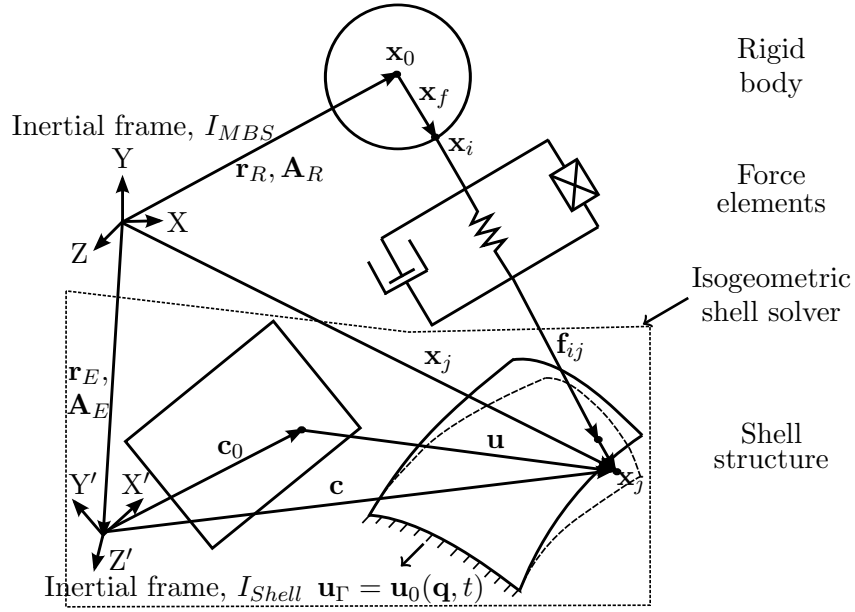


Figure 6.7: Flexible MBS with shell structure interaction

with \mathbf{f}_{MBS} denoting MBS forces, $\mathbf{f}_{ij,r}$ load vector with interaction forces \mathbf{f}_{ij} between the point pairs $(\mathbf{x}_i(\mathbf{q}), \mathbf{x}_j(\mathbf{w}(\mathbf{d})))$, \mathbf{p} the external loads on the shell and $\mathbf{f}_{ij,h}$ the load vector of interaction forces \mathbf{f}_{ij} for the shell. Moreover

$$\mathbf{g}(\mathbf{q}, \mathbf{d}) = \begin{pmatrix} \mathbf{g}_r(\mathbf{q}) \\ \mathbf{u}_\Gamma(\mathbf{d}) - \mathbf{u}_0(\mathbf{q}, t) \end{pmatrix} \quad (6.10)$$

are the system constraints comprising of the MBS constraints \mathbf{g}_r and the boundary conditions \mathbf{u}_Γ on the shell.

The load vector of interaction forces for the shell $\mathbf{f}_{ij,h}$ is obtained through (3.29). The point \mathbf{x}_j in the physical domain is first approximated by a point $(\vartheta_{1,j}, \vartheta_{2,j})$ on the parametric domain using point projection as explained in Remark 3.7 and then the external energy is calculated using (3.29) by evaluating only the active basis functions at point $(\vartheta_{1,j}, \vartheta_{2,j})$.

There are two important points to keep in mind. One is that we always work with the displacement \mathbf{u} , which includes the rigid body motion as well as the elastic motion. This is crucial since this directly includes the effects of rigid motion in the inertia terms. Another point is that the shell solver requires the boundary condition \mathbf{u}_Γ and the input loads to be specified with respect to I_{Shell} . Since all the interactions are considered with respect to I_{MBS} , a conversion step is needed to convert the boundary condition and the loads to I_{Shell} . E.g., \mathbf{u}_Γ with respect to I_{Shell} is calculated as

$$\mathbf{u}_\Gamma|_{I_{Shell}} = \mathbf{A}_E^{-1}(\mathbf{u}_\Gamma - \mathbf{r}_E) \quad (6.11)$$

and the force \mathbf{f} is given by

$$\mathbf{f}|_{I_{Shell}} = \mathbf{A}_E^{-1}\mathbf{f}. \quad (6.12)$$

In practice one can always avoid the distinction between the two axes I_{MBS} and I_{Shell} . To do this, the control points \mathbf{P} defined with respect to I_{Shell} need to be shifted to the I_{MBS}

using the relation

$$\mathbf{P}|_{I_{MBS}} = \mathbf{r}_E + \mathbf{A}_E \mathbf{P}|_{I_{Shell}}. \quad (6.13)$$

After this, for all calculations, the translation $\mathbf{r}_E = \mathbf{0}$ and rotation $\mathbf{A}_E = \mathbf{I}_3$.

6.3.2 The Simulation

To solve the flexible multibody systems (6.7), one can use the simulation techniques such as the co-simulation, where the time integration of the structural part and of the MBS is performed in independent, specialized codes. Alternatively the monolithic approach can be used, where the time integration takes place in a single computational framework. A regular exchange of data between both the multibody system and the elastic body solvers needs to be maintained.

A serial co-simulation methodology with a force-displacement interaction following [Goy11] is shown in Fig. 6.8. At each data exchange time t_k , based on the state \mathbf{q}^k the MBS provides the boundary condition $\mathbf{u}_\Gamma(\mathbf{q}^k, t_k)$ and applies loads $\mathbf{f}_k := \mathbf{f}_{ij,h}(\zeta_{ij}, \dot{\zeta}_{ij}, t_k)$ at some user specified nodes of the elastic body. The node displacement information $\mathbf{w}(\mathbf{d}^k)$ based on the new states \mathbf{d}^k is then returned to the MBS after a time integration of the elastic body. The MBS is then itself integrated to compute the new states \mathbf{q}^{k+1} , boundary conditions and loads at time t_{k+1} . This procedure is repeated until the end of simulation time.

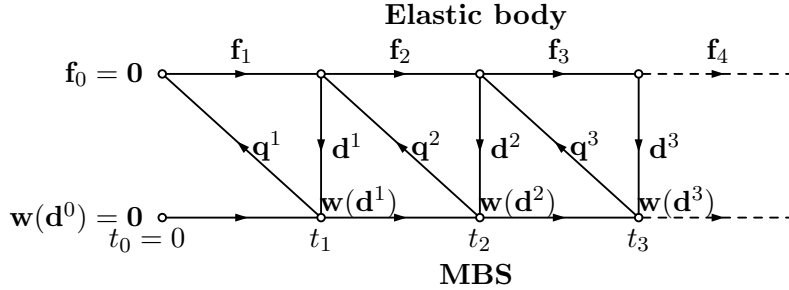


Figure 6.8: Co-simulation between MBS and elastic body via series data exchange

For the monolithic approach, at the start of each time integration step, the right hand side of equations (6.7) is evaluated for the whole system based on the latest values of the variables \mathbf{q} , \mathbf{d} and $\boldsymbol{\lambda}$ from both the MBS and IGA solver. The time integration to calculate the new states is then carried out in a single solver, e.g. that of the MBS.

During simulation, one might run into some troubles if the mechanical system is stiff. Lubich [Lub93] claims that stiff mechanical systems are the ones that include stiff force elements like stiff springs or elastic bodies that are almost rigid or both. Almost rigid elastic bodies are elastic bodies that have a high E-modulus. Such systems behave similar to index-3 DAEs and undergo order reduction [Lub93, Sim98]. To obtain an asymptotic result, one must use techniques such as h-scaling [SS02] or reducing the velocity tolerance. Guidelines to efficiently solve stiff mechanical systems are discussed in the paper by Simeon [Sim01].

6.4 Numerical Results

This section shows some examples that validate our isogeometric shell solver. The results demonstrate the gain in terms of dofs that can be expected from an isogeometric discretization.

The results have been divided into four main parts. First we present the single patch results for validation, followed by the multipatch results based on the various multipatch approaches from Section 4.4 and Chapter 5 comparing them against each other and validating with single patch results. Then we show some results for the viscoelastic case from Section 3.5, where due to the material damping unnecessary oscillations in the multibody framework can be avoided. Lastly we show the results of the coupling between our solver and the MBS code Simpack from Section 6.3.2, where we replace the rigid connecting rod of an engine mechanism with a flexible shell plate.

6.4.1 Single Patch

We focus on our single patch results from [GDSV13] that are aimed at validating and showing the benefits of the isogeometric approach. Some results are restricted to a stationary setting and refer to popular benchmarks in shell analysis. The numerical tests compare the errors of different solutions obtained with variable mesh sizes and polynomial degree. The tests are conducted on both geometrically linear as well as nonlinear shells.

6.4.1.1 Nonlinear Plates

The first example is a nonlinear cantilever plate subjected to different moments at the right-most edge in Fig. 6.9, with E-modulus $E = 1.2 \times 10^6$, Poisson's number $\nu = 0$, length $L = 12$, width $b = 1$, thickness $t = 0.1$. A pointwise error plot between the exact and the calculated end x-displacement versus the NURBS element size h is shown. The results are analyzed for a 180° bending case. The exact displacements at the tip are taken from [SLL04],

$$\begin{aligned} U_{tip,exact} &= L \left(\frac{M_0}{M} \sin \left(\frac{M}{M_0} \right) - 1 \right), \\ W_{tip,exact} &= L \frac{M_0}{M} \left(1 - \cos \left(\frac{M}{M_0} \right) \right), \end{aligned} \quad (6.14)$$

where $M_0 = \frac{EI}{L}$, $M = \theta M_0$ and θ is the bending angle in radians.

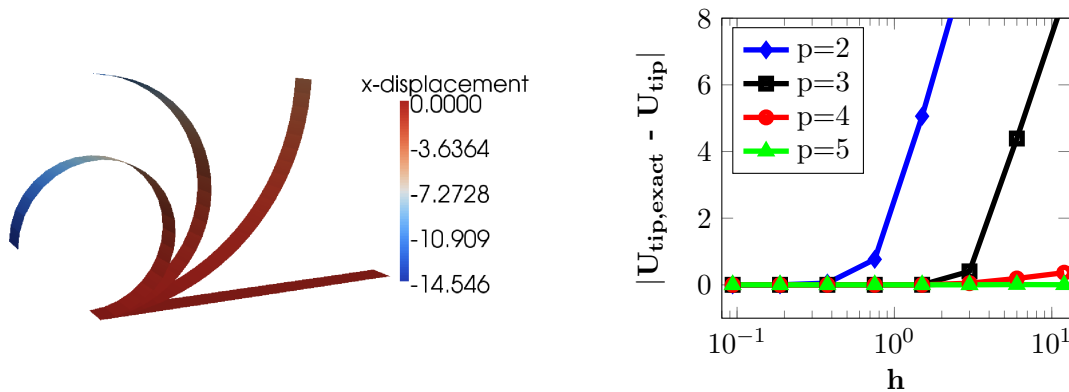


Figure 6.9: Cantilever plate with applied moment (180° bending) and its pointwise error at tip

It is observed that taking higher degree NURBS as basis functions shows good convergence even with a coarse mesh. On the other hand, quadratic NURBS which correspond to standard

isoparametric finite elements for shells require quite fine meshing to produce good results. Similar conclusions can be drawn from the plots for plates, $L = 20$ and $L = 10$ respectively, under different loading conditions, Fig. 6.10. The plots show the L^2 error, which is a global error measure. The error is calculated using the result of a highly refined simulation as reference solution.

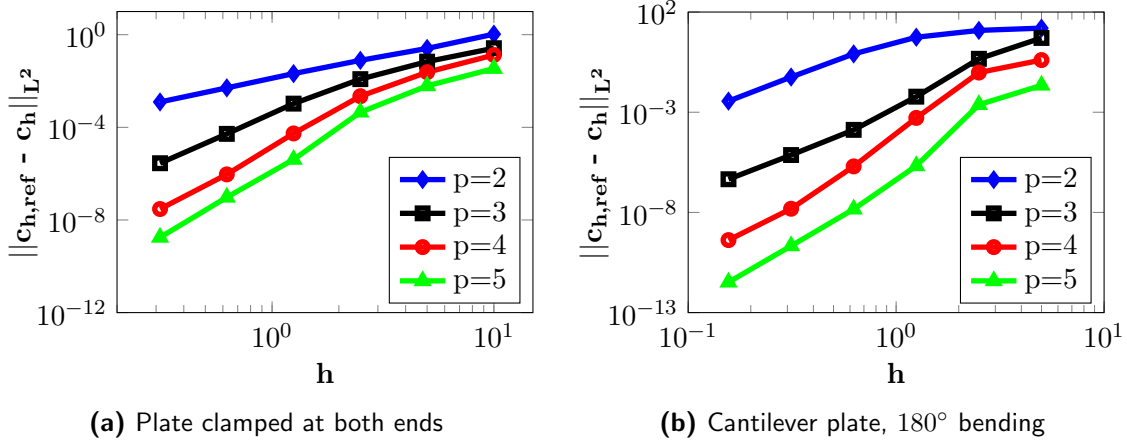


Figure 6.10: L^2 -convergence of nonlinear plate under different loading conditions

6.4.1.2 Scordelis Lo Roof

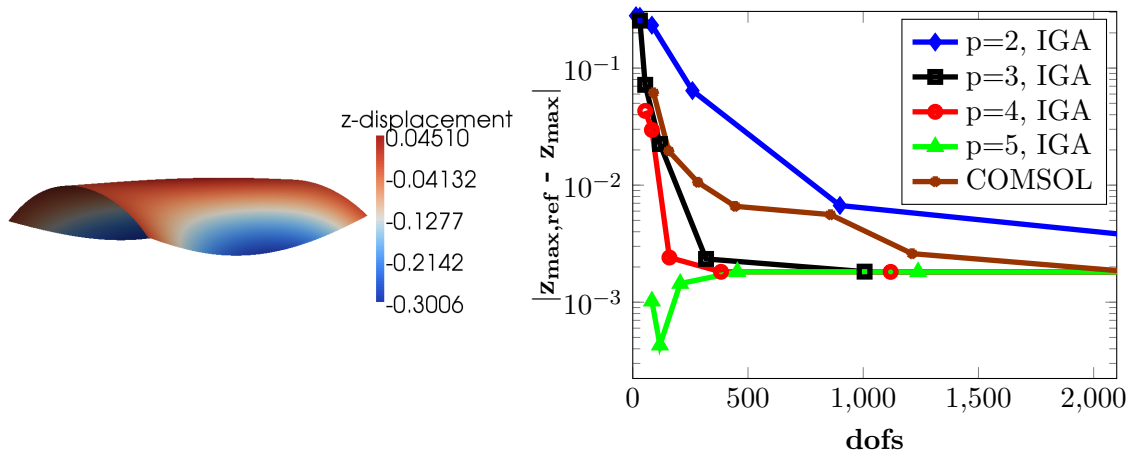


Figure 6.11: z-displacement error of linear Scordelis Lo Roof with constant body load of 90 N/m^2

In Fig. 6.11, a popular benchmark from the shell obstacle course namely the *Scordelis lo roof* is analyzed, with E-modulus $E = 4.32 \times 10^8$, Poisson's number $\nu = 0$, length $L = 50$, radius $r = 25$, thickness $t = 0.25$. It is a section of a cylindrical shell, which is supported by rigid diaphragms at the left and right edges and the other two edges are free. It is subjected to a constant body load of 90 per unit area. We consider the small displacement or linear regime and study the error with respect to the z-displacement. The maximum vertical displacement $z_{max,ref} = 0.3024$ from [BSL⁺85] is taken as reference. It is observed that for quartic and



Figure 6.12: Membrane S_{m-x} , bending S_{f-x} stress resultants in the x-direction for Scordelis Lo Roof

quintic NURBS in both parametric directions, small errors are obtained even with a small number of degrees of freedom which is in contrast to the shell analysis results obtained from a commercial FE tool Comsol or with lower order NURBS.

In Fig. 6.12, we obtain the membrane $\hat{\mathbf{S}}^m$ (or S_{m-x}) and bending $\hat{\mathbf{S}}^f$ (or S_{f-x}) stress resultants in the x-direction of the local Cartesian axis (along the length) using (3.47) from Section 3.4. The plots are obtained for degree $p = 5$ and on refining by inserting 7 equally spaced knots in both parametric directions. They are found to be in good accordance with [Kie11]. Note that the computed stresses are smooth functions and do not suffer from jumps between element boundaries, something that is not possible with C^0 continuous finite elements.

6.4.1.3 Pinched Cylinder

We take another shell obstacle problem, the *pinched cylinder* in the linear deformation regime, with E-modulus $E = 3.0 \times 10^6$, Poisson's number $\nu = 0.3$, length $L = 600$, radius $r = 300$, thickness $t = 3.0$. The cylinder is supported by rigid diaphragms at the ends and subjected to two opposite point loads in the middle, as shown in Fig. 6.13. For the analysis, only one eighth of the geometry is considered due to the symmetry, using appropriate boundary conditions. The displacement under the point load $z_{min,ref} = -1.8248 \times 10^{-5}$ from [BSL⁺85] is taken as reference. Again higher order NURBS perform better in terms of the dofs.

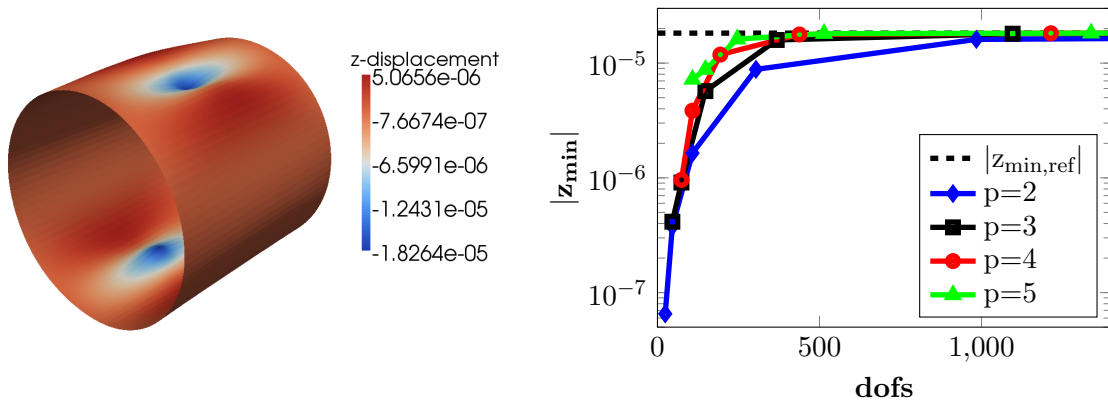


Figure 6.13: z-displacement of linear pinched cylinder with two unit point loads

6.4.1.4 Pinched Hemisphere

Another well-known shell obstacle problem is the *pinched hemisphere* in the linear deformation regime, with E-modulus $E = 6.825 \times 10^7$, Poisson's number $\nu = 0.3$, radius $r = 10$, thickness $t = 0.04$. The hemisphere in Fig. 6.14 is fixed at the top and the circumferential edge is free. It is subjected to two diametrically opposite loads of magnitude 2. Due to symmetry, only one fourth of the geometry is analyzed, using appropriate boundary conditions. The reference solution is the maximum displacement at the point on the circumference where the load is applied, $y_{min,ref} = -0.0924$ from [BSL⁺85]. Our solver converges to the reference solution.

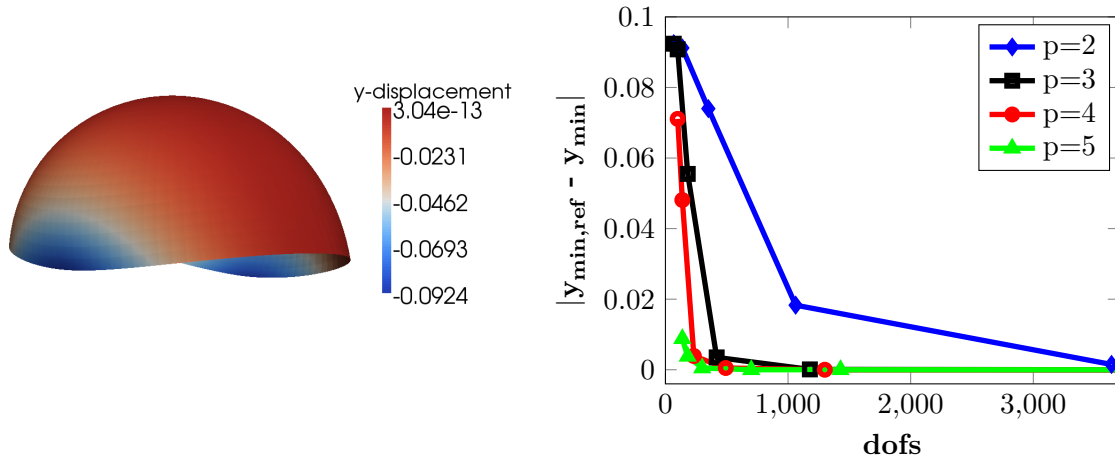


Figure 6.14: y-displacement error of linear pinched hemisphere with two diametrically opposite loads of magnitude 2

6.4.1.5 Nonlinear Cantilever with Transient Loading

In Fig. 6.15, we observe the error propagation in time of a cantilever plate loaded with the force $f(t)$ in the z-direction at the two extreme points of its free edge.

$$f(t) = \begin{cases} 2 \sin(\pi t/5) & \text{if } t \leq 10 \\ 2 & \text{if } t > 10 \end{cases} \quad (6.15)$$

The shell solver used the implicit Euler scheme (2.20) to integrate from time 0 to 70 with a constant time stepping of 0.1. The error of different degree NURBS solutions having element size $h = 2.5$ in the length direction is compared with the highly refined solution at $p = 5$ and $h = 0.078125$. Again it is observed that with increasing NURBS degree we get considerably lesser errors for the same element size h . This implies that for the same error, lesser dofs are required with higher order NURBS.

6.4.2 Multipatch

In this section, we present some examples dealing with multipatch geometries using the various approaches mentioned in the thesis. The results are validated by comparing them against the reference single patch solutions from Section 6.4.1. The various multipatch methods, namely the scaled bending strip method (4.43), the scaled alternative formulation with independent

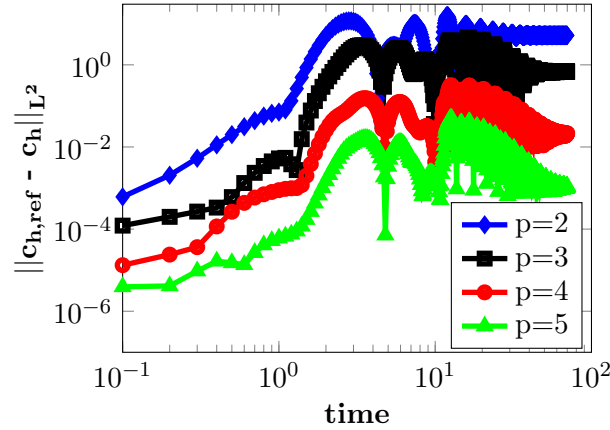


Figure 6.15: L^2 - error propagation of nonlinear cantilever plate under time dependent loading at tip

bending strip constraints, where the system is treated using our proposed algorithm 5.1 and the scaled alternative formulation with direct continuity constraints from Section 5.2 are abbreviated as *SBSM*, *AF-BS* and *AF-D* respectively.

6.4.2.1 Cantilever Plate Bent to Circle

To validate and see the effects of our different approaches, we consider a nonlinear cantilever plate with same material properties and dimensions as the plate in Section 6.4.1.1. It is clamped at the left edge and subjected to an end moment of $M = 2\pi M_0 = -52.36$ from (6.14) at the right edge that bends it to a circle [SLL04]. We model the plate as a two-patch geometry by cutting the plate in the middle along its length. Fig. 6.16 shows the deformed configuration of the plate joined with a bending strip. The minimum x-displacement $x_{min} = -11.995$ is quite close to the reference result $x_{min,ref} = -12$ from [SLL04].

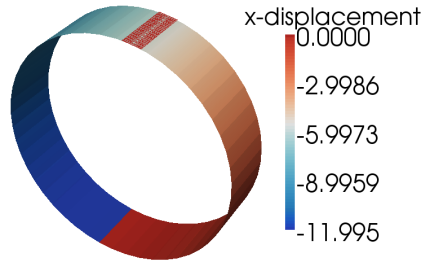
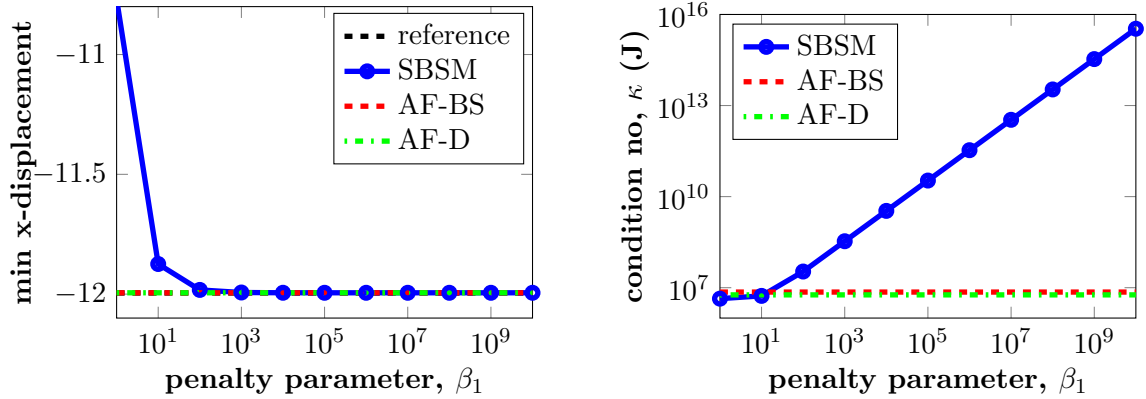


Figure 6.16: Two-patch nonlinear cantilever plate bent to circle, joined using a bending strip

For studying the behavior of the cantilever plate as a function of penalty parameter, in Fig. 6.17, we take quintic NURBS in first parametric direction and linear in the second. Also we add 7 equally spaced knots for h-refinement in the first parametric direction. In Fig. 6.17a, the minimum x-displacement is studied with the change in the penalty parameter β_1 . For the SBSM, it is observed that the displacement converges towards the reference result for increasing β_1 values. Also from Fig. 6.17b, we notice that the condition number steadily deteriorates with increasing β_1 for the SBSM. To remove this penalty parameter dependence, we use the alternative formulations and for *AF-BS* and *AF-D*, we observe that

the x-displacement matches the reference result and the condition numbers in both cases are independent of the penalty parameter. The SBSM uses 154 dofs, followed by the AF-BS which uses 156 dofs, i.e. 2 more independent dofs belonging to each of the two triples following (5.13a), (5.13b). The most, 160 dofs are used by AF-D, which adds 6 G^1 continuity constraints (5.38), (5.39) to the system, as in Remark 5.7.



(a) Minimum x-displacement (b) Conditioning of Jacobian

Figure 6.17: Cantilever plate behavior as function of penalty parameter

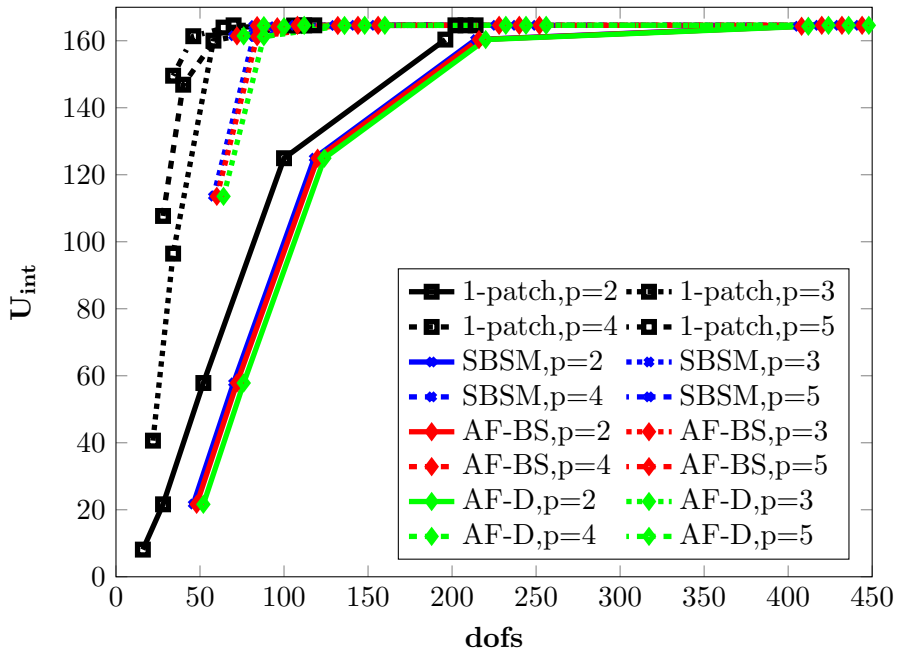


Figure 6.18: Energy convergence plot

A convergence analysis of the plate in case of different methods shows that all the methods converge to the same energy value in Fig. 6.18. Moreover it is seen for all the methods that the higher order NURBS require quite less dofs as compared to lower order NURBS, a feature of NURBS which is frequently exploited in isogeometric analysis. In Fig. 6.19, we further observe the influence of varying the number of patches on the convergence. It is seen that it

is favorable to use as less number of patches as possible to converge fastest in terms of the dofs regardless of the order of the NURBS used.

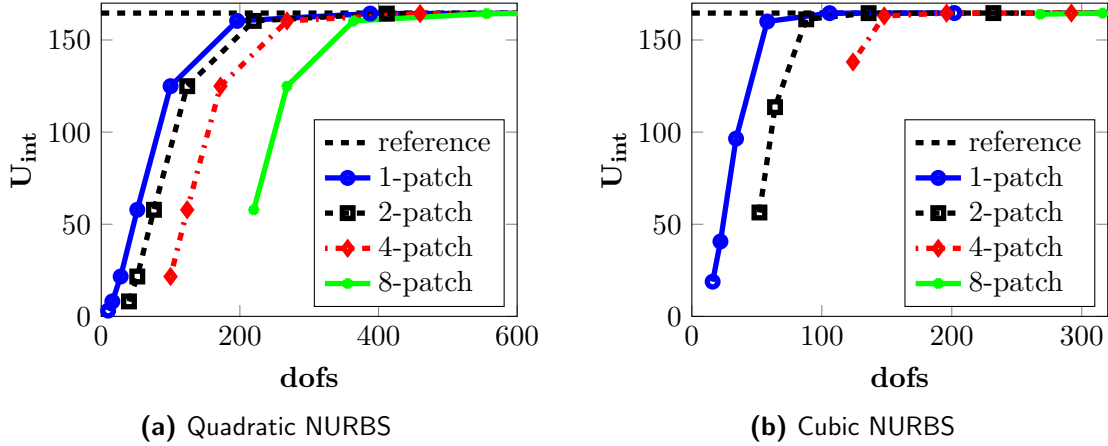


Figure 6.19: Influence of varying number of patches

6.4.2.2 L-shaped Cantilever Plate

This example presents a linear two patch L-shaped plate, where the patches are joined with a C^0 continuity at right angles to each other, see Fig. 6.20a. The two patches have the same length $L = 10$ and thickness $t = 0.1$. The widths for the two patches are $b = 1, 3$ respectively. The material properties are E-modulus $E = 1.2 \times 10^6$ and Poisson's number $\nu = 0$. The two patch plate is clamped at the back edges and a point load of 3 in the downward direction is applied at the top flange on the front left extremity. A connection with just the C^0 continuity in Fig. 6.20b finds the plate connection acting as a hinge and no bending moment is getting transferred. To transfer the bending moment, the bending strip method or direct angle preservation constraints are applied and in the Fig. 6.20c, the right angle seems to be preserved during the deflection of the plate.

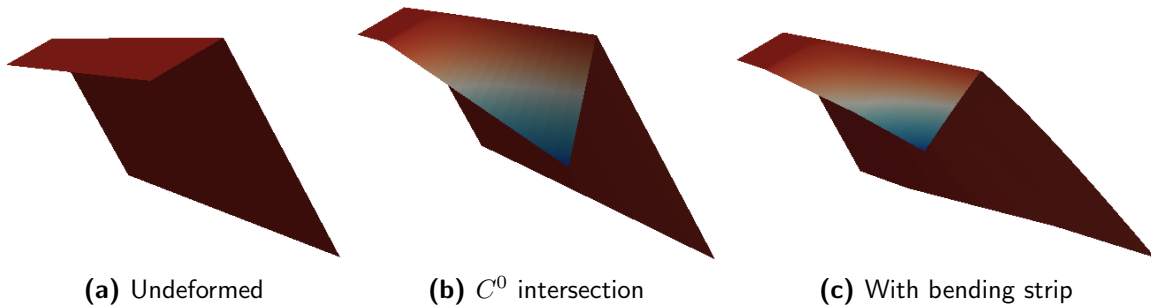


Figure 6.20: Configurations of the two patch linear L-shaped plate

For the analysis of the L-shaped plate as a function of penalty parameter, see Fig. 6.21, we take cubic NURBS with 3 equally spaced knots in both the parametric directions. This gives 5 triples of control points at the patch intersection. In Fig. 6.21a, the absolute value of the relative error in the angle of the plate $|\theta - \theta_0|/\theta_0$, with $\theta_0 = \pi/2$ at the front intersection point is studied with the change in the penalty parameter β_1 . For the SBSM, it is observed

that the absolute relative error in the angle decreases towards zero for increasing β_1 value. This is consistent with what is expected from the theory as the constraint violations are more strictly penalized with increasing penalty parameter values. Also from Fig. 6.21b, we notice that the condition number steadily deteriorates with increasing β_1 for the SBSM. To remove this penalty parameter dependence, we again use the alternative formulations and for AF-BS and AF-D, we observe that a zero absolute relative error is obtained with the condition numbers in both cases being independent of the penalty parameter. The SBSM uses 121 dofs, followed by the AF-D which uses 125 dofs and has additionally 4 angle preservation constraints (5.37). One angle preservation constraint is lost since one of the triple falls on the Dirichlet boundary. The most, 132 dofs are used by AF-BS, which has additionally 4 angle as well as 7 flange length preservation constraints. One obtains 2 length constraints per triple, amounting to 10, but 3 length constraints that fall on the clamped Dirichlet boundary are removed.

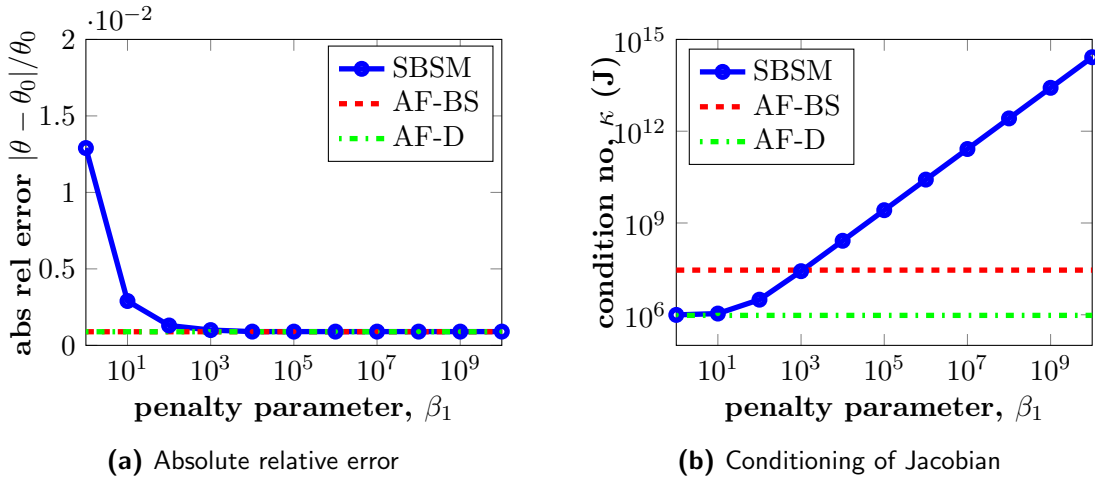


Figure 6.21: Two patch L-shaped plate behavior as function of penalty parameter

In our analysis setup, we take 4×4 elements in the two parametric directions. According to the Conjecture 5.1 from Section 5.1.3.1, the number of constraints for two plates connecting at an angle $\theta \neq 0$ should be $3(n+1)$ for sufficient number of quadrature points, where n is the number of elements along the strip, which is 4 in this case. Hence we should have $3(4+1) = 15$ constraints. We have only 11 constraints for AF-BS, since due to the clamping of the plate at the end of the strip, all 3 constraints from triple on the boundary and 1 z-constraint from the triple adjacent to it must be removed due to prescribed boundary condition.

6.4.2.3 Scordelis Lo Roof

We again study the Scordelis lo roof, this time in a multipatch setting. The configuration of the roof is the same as explained in Section 6.4.1.2. We model it as a two patch geometry by dividing the original roof in the middle along its length. A two patch roof joined by a bending strip is shown in Fig. 6.22. The maximum displacement at the mid point of the side of the roof is taken as the reference value, $z_{max,ref} = -0.302$ from [BSL⁺85]. A displacement $z_{max} = -0.301$ is obtained by the single patch shell solver with degree $p = 5$ and refined by adding 7 additional equally spaced knots in each parametric direction with 454 dofs, which is in good accordance with the reference value.

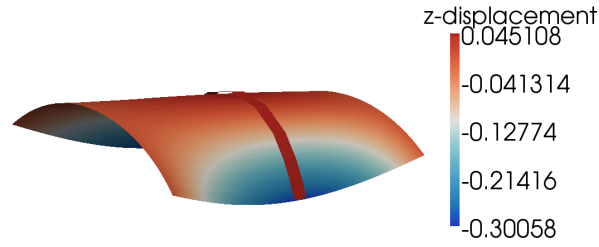


Figure 6.22: Two patch linear Scordelis lo Roof

The same result is obtained from the multipatch solvers for SBSM, AF-BS, AF-D, using quintic NURBS and refining for each patch by adding 3,7 knots in the two parametric directions respectively. 687 dofs were used with SBSM, followed by 711 dofs with the AF-BS, due to 24 additional active G^1 conditions at the intersection. The most, 726 dofs are used by the AF-D, due to the 39 additional constraints as a results of 3 constraints (5.36) each for each of 13 triples. Again for the SBSM, the behavior of the roof as seen from Fig. 6.23 is similar to the previous example, where with increasing penalty parameter β_1 value, the maximum displacement converges to the reference result, but after some point gives wrong results due to ill-conditioning, which can be seen from Fig. 6.23b. For the alternative formulations AF-BS and AF-D, the displacements matched the reference result and the condition numbers are constant with β_1 .

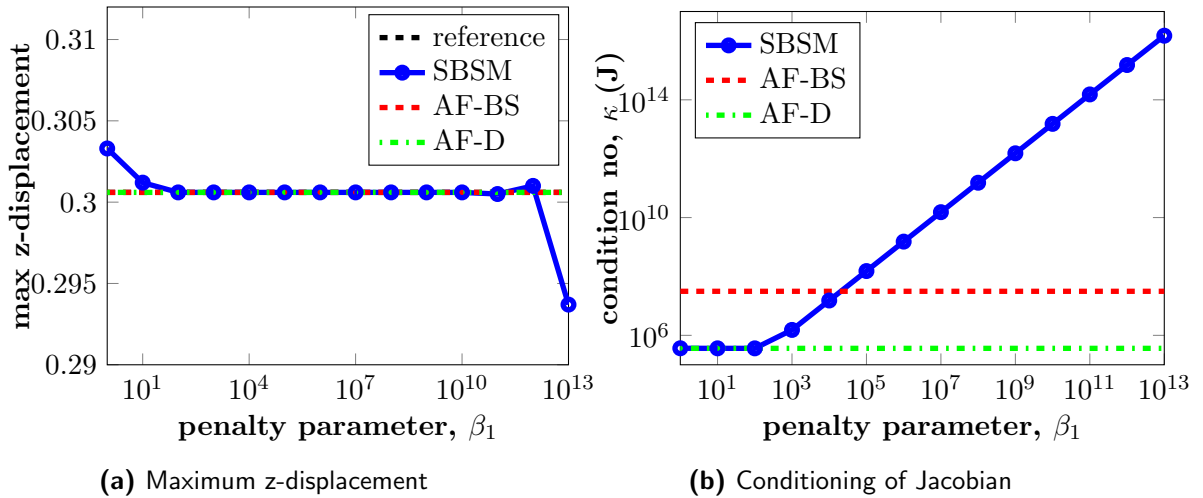


Figure 6.23: Two patch roof behavior as function of penalty parameter

6.4.2.4 Simplified Twist Beam Rear Axle

Now we want to study a twist beam rear axle of small to mid-ranged vehicle, see Fig. 6.24, for which we use multipatches due to its complex structure. With a goal to analyze the mid-section of the axle, we consider a simplified 5-patch geometry constructed following the steps from Example 6.1, with the patches numbered as in Fig. 6.25a. The left and the right edges of patch 2 are fixed and a moment of $1e8$ is applied at the two end points of the right edge of patch 1, which connects to patches 3 and 5. A nonlinear theory is required since the axle undergoes large deformations as can be seen from the Fig. 6.25b.

We study the max z-displacement of the axle for the various approaches against the penalty

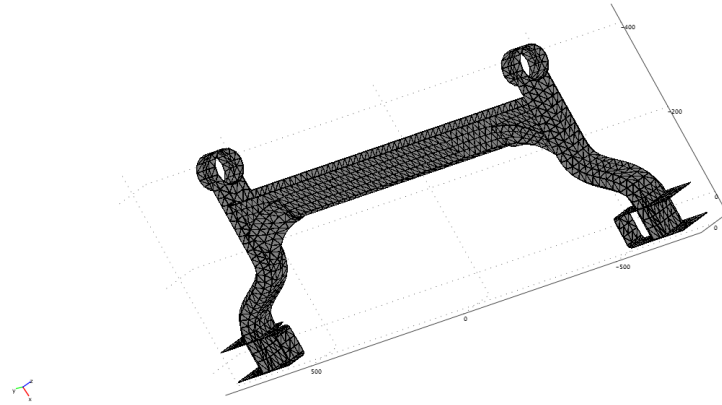


Figure 6.24: FE mesh of a Twist Beam Rear Axle

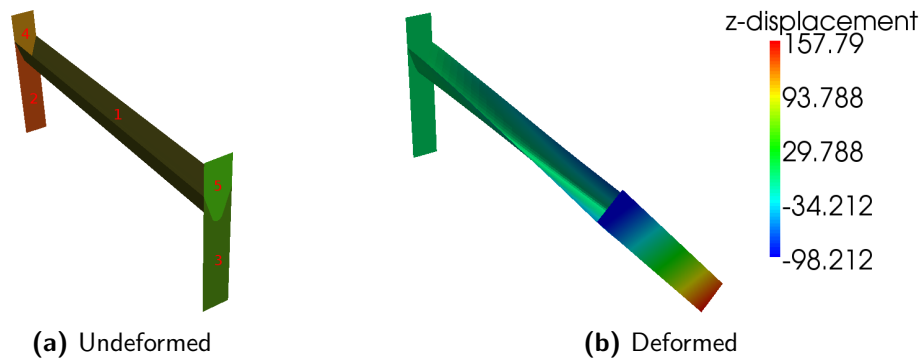


Figure 6.25: 5-patch nonlinear Twist Beam Rear Axle

parameter β_1 . With the SBSM, the z-displacement starts to converge with increasing penalty parameter, but further computations could not be done due to ill-conditioning of the system. The penalty free solver AF-BS shows no penalty parameter dependence and gives directly the result that one expects as the limit case of the SBSM. This is good since we get the best result without any ill-conditioning. In this case, we get a different result with AF-D, since Remark 5.8 states that the bending strip constraints are stiffer than just applying the angle preservation constraints, which are sufficient for this example. Due to this effect, we notice a smaller displacement with the AF-BS as compared to AF-D.

6.4.2.5 Hinged Plate in Transient Setting

Lastly we take a nonlinear hinged plate in the transient setting. We observe here the negative effects of the stiff bending strip system on the time integration and show that a non-stiff alternative formulation performs better.

We take the plate divided into two equal halves along the length with the material properties and dimensions as in Section 6.4.2.1. The plate is hinged on the left edge and simulated for a time $t \in [0, 10]$. We use for the time integration the Generalized- α method for DAEs from Section 2.1.3.2 with the dissipation parameter $\rho_\infty = 0.5$ and a fixed step size h . The plate is released from an initial moment

$$M_x = \begin{cases} (7, 0, 0)^T & \text{if } t = 0 \\ (0, 0, 0)^T & \text{else} \end{cases}$$

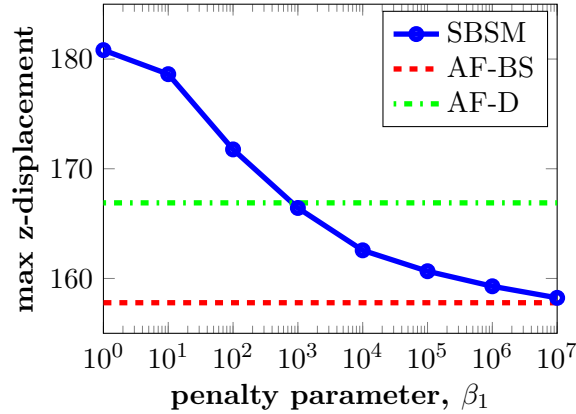


Figure 6.26: Maximum z-displacement of axle as function of penalty parameter

applied at the mid point of the free edge to twist the plate about its length and a ramp function is used to apply the force

$$F_z = \begin{cases} (0, 0, -2)^T \times t & \text{if } t < 1 \\ (0, 0, -2)^T & \text{else} \end{cases}$$

at the same point. The plate in the initial position deforms as in Fig. 6.27 due to the initial moment.

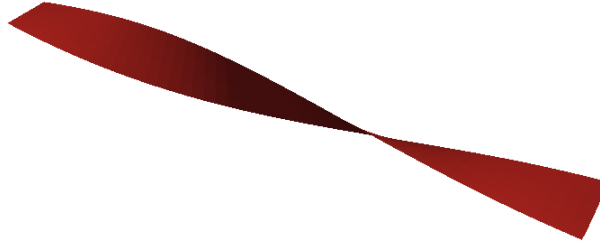


Figure 6.27: Hinged plate in the initial position

We tabulate in Table 6.1 the number of Newton iteration steps per time step and the condition number magnitude order against the various time step sizes h for the penalty approach SBSM and the alternative formulation AF-D, and compare them against the single patch results. It is noticed that the behavior of the Newton iteration number and the condition number order are quite similar in the single patch case and the penalty free formulation AF-D, whereas for the penalty approach SBSM, the higher the penalty parameter value, the worse is the convergence of the Newton step and the conditioning of the system Jacobian. The Newton iterations do not converge for the bigger step sizes $h = 10^{-1}$ and we observe a kind of step size restriction influenced by the penalty parameter value analogous to [Lub93]. This shows the benefits of a penalty free formulation over the one with a penalty approach for transient cases.

h	1-patch (168 dofs)		2-patch, SBSM (312 dofs)						2-patch AF-D (324 dofs)	
			$\beta_1 = 10^3$		$\beta_1 = 10^4$		$\beta_1 = 10^5$			
10^{-1}	35.840	10^4	–	10^6	–	10^7	–	10^8	41.780	10^4
10^{-2}	4.071	10^3	4.061	10^5	4.374	10^6	–	10^7	4.047	10^3
10^{-3}	3.008	10^3	3.007	10^4	3.405	10^5	–	10^6	3.001	10^3

Table 6.1: Newton iterations per step | condition no. for different step sizes h

6.4.3 Viscoelasticity

In this section, we present some results of the Kelvin-Voigt kind of viscoelasticity applied to our shell models following the theory from Section 3.5 and compare them against results from other sources. We show in particular nonlinear beam examples, for which good reference results for comparison were found. Standard methods like Rayleigh damping are not helpful for such examples, since they are only well understood for a linear deformation regime.

6.4.3.1 Beam Released after Initial Loading

The first example that we take is a cantilever beam that is released after an initial tangential loading $F_0 = 0.05 N$, see Fig. 6.28. The beam is studied in the linear deformation setting, with the material properties: E-modulus $E = 1 MPa$, Poisson's number $\nu = 0.3$ and the density $\rho = 1000 kg/m^3$. The beam has length $L = 0.3m$, width $b = 0.01m$ and thickness $t = 0.01m$.

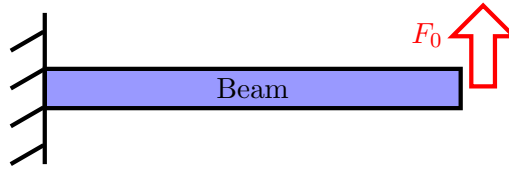


Figure 6.28: Beam released after initial loading

We model the beam as an isogeometric shell. The example is taken from the paper on Kelvin-Voigt kinds viscoelasticity for Cosserat rods by Linn et al. [LLT13]. In Fig. 6.29, we observe the displacement behavior of the beam tip with respect to time for different values of the damping factor η . We observe that for zero damping $\eta = 0$, the maximum z-amplitude remains constant and there is no dissipation. With increasing η , the dissipation also increases correspondingly. In Fig. 6.30, we have plotted the damping ratio, ζ_k against the oscillation number k for different η values. The damping ratio, ζ_k is determined using the relation

$$\frac{\zeta_k}{\sqrt{1 - \zeta_k^2}} := \frac{\delta_k}{2\pi}, \quad \text{where} \quad \delta_k = \ln \left(\frac{z_k}{z_{k+1}} \right).$$

An oscillation is the tip movement from rest, going up and down and back to initial position. It can be seen that in each case, the damping ratio increases until the fifth or sixth oscillation nonlinearly and then becomes constant. The constant part marks the linear deformation where the amplitude reduces exponentially. The damping in the nonlinear regime is lower

than in the linear regime as is reflected by the damping ratio. The behavior of the isogeometric shell is analogous to the results presented in [LLT13].

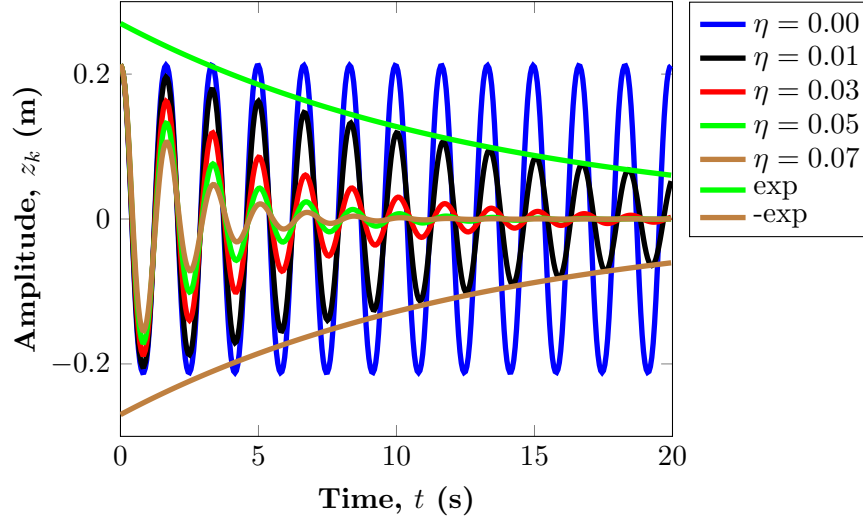


Figure 6.29: Amplitude vs time

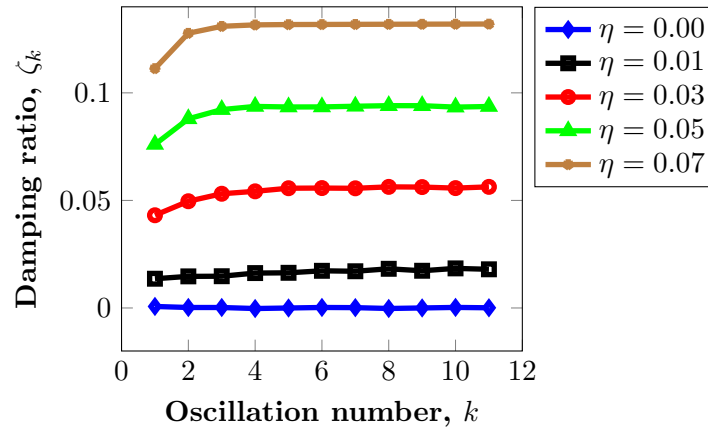


Figure 6.30: Damping ratio vs oscillation number

6.4.3.2 Beam under Periodic Loading

The next example is the same beam as in the Section 6.4.3.1 but this time under periodic tangential loading, see Fig. 6.31,

$$F(t) = F_0 \sin(\omega t), \quad F_0 = 5 \times 10^{-4} \text{ N}, \quad \omega = 3.39 \text{ rad/s}.$$

We compare the results at the steady state, where the beam also undergoes a sinusoidal deformation similar to the loading but with a phase shift. We plot the max z -amplitude and phase shift between the loading and the deformation cycles against different damping factors η . We compare the results from our isogeometric shell solver and from the application of the

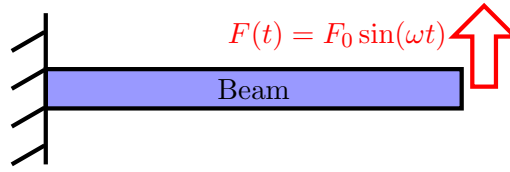


Figure 6.31: Beam under periodic loading

Harmonic Balance Method (HBM) to an isogeometric solid model developed for steady state frequency response analysis from [WWS13]. We notice from Fig. 6.32 that both results are in very good accordance with each other, although our technique performs a time integration to come to the result and the HBM achieves this without it.

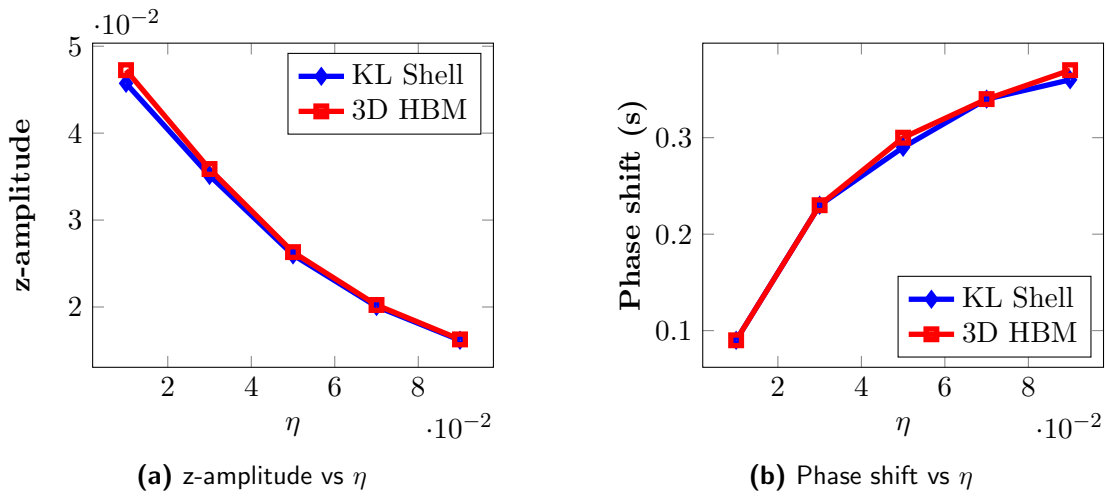


Figure 6.32: Beam behavior under different damping factor η

6.4.4 Coupled MBS

An application of our isogeometric shell theory is its use in flexible multibody dynamics. The various benefits of NURBS based IGA on elastic shells mentioned in Section 4.2.4 make it a very good choice for its coupling with multibody systems. Its major advantages are the properties such as the affine invariance to rigid motion, good results with lesser dofs as seen from Section 6.4.1 and having exact geometry even with coarse meshes for geometrically sensitive shell structures.

Due to the presence of mostly dense matrices in multibody dynamics, there is no provision to utilize the sparsity of the system matrices that arise from the elastic discretizations. In this setting, a monolithic simulation in the MBS code with an IGA shell discretization is more favorable as compared to standard finite elements. This is because IGA uses for the same accuracy lesser dofs as compared to standard finite elements. Hence the matrices involved will be smaller and the size of the full system with IGA will not increase as much as with the standard finite elements. In such a setting even though the IGA matrices are a little denser as compared to standard finite elements, it does not have a bad influence on the simulation performance.

6.4.4.1 Flexible Slider Crank Mechanism

In Fig. 6.33, we take a slider crank mechanism in the commercial MBS code Simpack and replace the rigid connecting rod with a flexible one. On the left is a piston, that gives the translatory motion to the piston rod and in turn to the slider block. This slider is connected to one end of the flexible connecting rod modeled using various beam and equivalent shell models. The connecting rod at the other end connects to the crank and rotates it about a shaft on which the wheel is mounted.

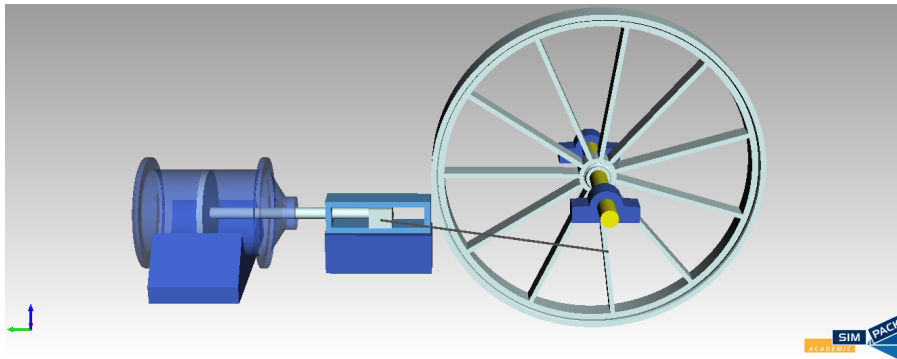


Figure 6.33: Slider Crank with flexible connecting rod in Simpack

The elastic motion of the connecting rod is in the small deformation or linear regime, but the nonlinearity in the motion stems from the rigid motion of the rod. The choice of the *floating frame of reference* [Sha98], where the elastic body has a separate reference frame which moves with the body, separates the rigid and elastic motion. Another choice is the *absolute nodal coordinates* [BS09], where the reference frame is inertial. The coupling between the MBS and the IGA shell solver is done using the theory from Section 6.3 and uses absolute nodal coordinates. A nodal interaction is established between our isogeometric shell solver and the MBS tool SIMPACK via an interface [SDT⁺12]. We obtained the results below with a monolithic simulation.

To validate the correctness of our coupling, we compared the results for various elastic connecting rod models, namely a plate model in Abaqus, Cosserat rod model, Simpack beam (Simbeam) model, our isogeometric shell models with and without some artificial Rayleigh damping, included in the slider crank mechanism. In Fig. 6.34, we have plotted the slider displacement and velocity as a function of time for a 5 second simulation time. Due to the choice of floating frame of reference in case of Abaqus plate and Simbeam, linear models were used for the same. In order to study the coupling under the nonlinear deformation setting, we employed the absolute nodal coordinates for the Cosserat rod and isogeometric shell models.

The displacements of all the models matched, except the Simbeam which started to diverge at one point. This can be explained by the fact that there were oscillations in the velocities at each zero crossing of the velocity, which the Simbeam model could not handle. For the Rayleigh damped Abaqus plate model, these oscillations were damped out quite fast. Also the Cosserat rod model had some material damping inside the model and showed no oscillations. The isogeometric model without damping showed huge oscillations that damped out slowly, but the model with an artificial Rayleigh damping showed much lesser oscillations.

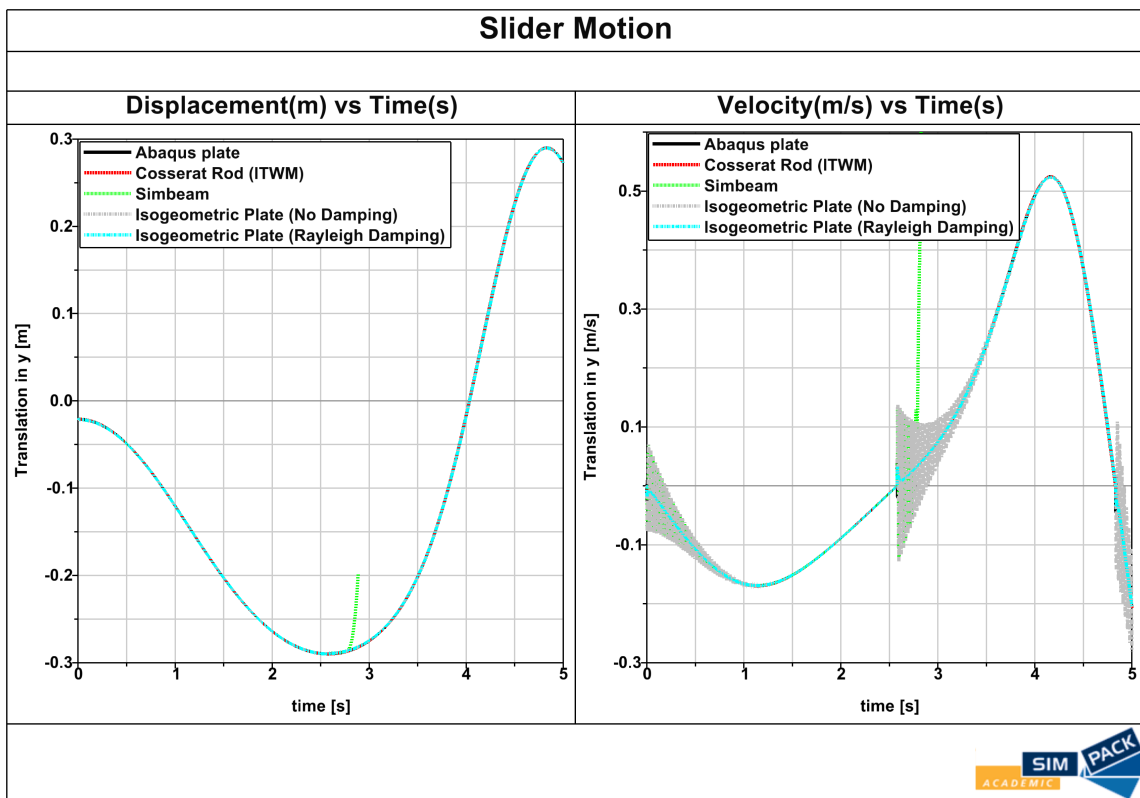


Figure 6.34: Slider Motion on coupling with different flexible connecting rod models

Chapter 7

Conclusion

As the title of the thesis suggests, in this work we developed isogeometric shell discretizations for flexible multibody dynamics. We discussed the underlying aspects such as describing the shell theory, defining flexible multibody systems and applying the novel concept of isogeometric analysis to shells. Furthermore, a coupling of the isogeometric shells with the multibody systems was presented that finally brought us in the regime of flexible multibody dynamics.

This work in particular demonstrated that the possibility to create smooth NURBS makes the isogeometric approach a very suitable choice for analyzing the Kirchhoff-Love shells, which require at least C^1 continuity that is difficult to achieve with standard finite elements. The numerical results demonstrated a good convergence behavior for higher order NURBS with a coarse mesh and fewer degrees of freedom. It shows that our approach is a promising way to include geometrically exact, nonlinear elastic shell structures in multibody dynamics without considerably increasing the computational expense. It is particularly crucial for shell structures that are very sensitive to geometry imperfections.

Another important aspect discussed is the handling of multipatch shells. We analyzed the penalty approach with the bending strips [KBH⁺10] and identified the ill-conditioning sources that can lead to trouble. The bending strip systems due to their penalty parameter dependence were found to be analogous to the stiff mechanical systems, which for transient cases show a time step size restriction that depends on the penalty parameter [Lub93]. Some correction techniques such as scaling and development of alternative formulations were applied to the bending strip method, similar to the treatment of stiff mechanical systems. We proposed the conversion of the penalty formulation to the alternative differential algebraic form, thereby eliminating the penalty parameter dependence and its ill effects. The idea is to identify the stiff forces and include them as separate constraints. It is a quite general approach that can be applied to all penalty approaches fulfilling some mild regularity requirements from [Lub93]. The resulting bending strip constraints were found to be redundant. Following [dJL13], we presented an algorithm to deal with redundant constraints in general flexible multibody systems. Our alternative formulation with bending strips falls under this class and can be treated in the same way.

The numerical results validated the presented theory and depicted the benefits of isogeometric shell discretizations, especially for their applications in the flexible multibody dynamics. We compared the results from the various multipatch approaches we discussed and observed that the alternative formulations yield penalty parameter independent formulations, which give well-conditioned system Jacobians and show no time step size restrictions for the

transient case. We also presented a technique to couple our isogeometric shells with multi-body systems using a pointwise interaction, which is solved both monolithically and via co-simulation.

Bibliography

- [AB07] Arnold, M.; Brüls, O.: Convergence of the generalized- α scheme for constrained mechanical systems, *Multibody System Dynamics* 18(2), 185–202 (2013)
- [ASWB14] Apostolatos, A.; Schmidt, R.; Wüchner, R.; Bletzinger, K.-U.: A Nitsche-type formulation and comparison of the most common domain decomposition methods in isogeometric analysis, *International Journal for Numerical Methods in Engineering* 97(7), 473–504 (2014)
- [Bau72] Baumgarte, J.: Stabilization of constraints and integrals of motion in dynamical systems, *Computer Methods in Applied Mechanics and Engineering* 1(1), 1–16 (1972)
- [Bau11] Bauchau, O.A.: *Flexible multibody dynamics*, Vol. 176, Springer Science+ Business Media (2011)
- [Ber12] Bernstein, S. N.: Démonstration du théorème de Weierstrass fondée sur le calcul des probabilités, *Commun. Soc. Math. Kharkov* 12(2), 1–2 (1912)
- [Bez72] Bézier, P. E.: *Numerical Control: Mathematics and Applications*, John Wiley, New York (1972)
- [Boe80] Boehm, W.: Inserting new knots into B-spline curves, *Computer Aided Design* 12(4), 199–201 (1980)
- [Boe85] Boehm, W.: On the efficiency of knot insertion algorithms, *Computer Aided Geometric Design* 2(1), 141–143 (1985)
- [Bra01] Braess, D.: *Finite elements: Theory, fast solvers, and applications in solid mechanics*, Cambridge University Press (2001)
- [Bre90] Brezzi, F.: A discourse on the stability conditions for mixed finite element formulations, *Computer Methods in Applied Mechanics and Engineering* 82, 27–57 (1990)
- [BBC07] Bottasso, C.L.; Bauchau, O.A.; Cardona, A.: Time-step-size-independent conditioning and sensitivity to perturbations in the numerical solution of index three differential algebraic equations, *SIAM Journal on Scientific Computing* 29(1), 397–414 (2007)
- [BCP96] Brenan, K.E.; Campbell, S.L.; Petzold, L.R.: *The Numerical Solution of Initial Value Problems in Differential Algebraic Equations*, Classics in Applied Mathematics, SIAM (1996)

- [BEB07] Bauchau, O.A.; Epple, A.; Bottasso, C.L.: Scaling of constraints and augmented Lagrangian formulations in multibody dynamics simulations, *Journal of computational and nonlinear dynamics* 4(2), 021007-1 (2009)
- [BGL05] Benzi, M.; Golub, G.H.; Liesen, J.: Numerical solution of saddle point problems, *Acta numerica* 14(1), 1–137 (2005)
- [BHB⁺13] Benson, D. J.; Hartmann, S.; Bazilevs, Y.; Hsu, M. C.; Hughes, T. J. R.: Blended isogeometric shells, *Computer Methods in Applied Mechanics and Engineering* 255, 133–146 (2013)
- [BP92] Bremer, H.; Pfeiffer, F.: *Elastische Mehrkörpersysteme*, Teubner, Stuttgart (1992)
- [BS09] Betsch, P.; Sängler, N.: On the use of geometrically exact shells in a conserving framework for flexible multibody dynamics, *Computer Methods in Applied Mechanics and Engineering* 198, 1609–1630 (2009)
- [BSL⁺85] Belytschko, T.; Stolarski, H.; Liu, W.K.; Carpenter, N.; Ong, J.S.-J: Stress projection for membrane and shear locking in shell finite elements, *Computer Methods in Applied Mechanics and Engineering* 51, 221–258 (1985)
- [BW00] Basar, Y.; Weichert, D.: *Nonlinear continuum mechanics of solids*, Springer-Verlag, Berlin, Heidelberg, New York (2000)
- [BWBR04] Bischoff, M.; Wall, W.A.; Bletzinger, K.-U.; Ramm, E.: Models and Finite Elements for Thin-walled Structures, In *Encyclopedia of Computational Mechanics 2, Solids, Structures and Coupled Problems*, Chapter 3, Wiley (2004)
- [Cia78] Ciarlet, P.G.: *The finite element method for elliptic problems*, Elsevier (1978)
- [Cia05] Ciarlet, P.G.: *An Introduction to Differential Geometry with Applications to Elasticity*, Springer-Verlag, Heidelberg (2005)
- [Cox72] Cox, M.G.: The numerical evaluation of B-splines, *IMA Journal of Applied Mathematics* 10(2), 134–149 (1972)
- [CH93] Chung, J.; Hulbert, G.M.: A Time Integration Algorithm for Structural Dynamics with Improved Numerical Dissipation: The Generalized- α Method, *Journal of Applied Mechanics* 60(2), 371–375 (1993)
- [CH09] Cho, S.; Ha, S. H.: Isogeometric shape design optimization: exact geometry and enhanced sensitivity, *Structural and Multidisciplinary Optimization* 38(1), 53–70 (2009)
- [CHB09] Cottrell, J.A.; Hughes, T.J.R.; Bazilevs, Y.: *Isogeometric Analysis: Toward Integration of CAD and FEA*, John Wiley & Sons Ltd, New York (2009)
- [CK06] Craig, R.; Kurdila, A.: *Fundamentals of Structural Dynamics*, John Wiley, Englewood-Cliffs (2006)
- [dB72] De Boor, C.: On calculating with B-splines, *Journal of Approximation Theory* 6(1), 50–62 (1972)

- [dJL13] De Jalòn, J. G.; Gutiérrez-Lòpez, M. D.: Multibody dynamics with redundant constraints and singular mass matrix: existence, uniqueness and determination of solutions for accelerations and constraint forces, *Multibody System Dynamics* 30, 311–341 (2013)
- [DB02] Deuffhard, P.; Bornemann, F.: *Scientific Computing with ODEs*, Springer-Verlag, New York Berlin Heidelberg (2002)
- [DKS12] Dornisch, W.; Klinkel, S.; Simeon, B.: Isogeometric Reissner-Mindlin shell analysis with exactly computed director vectors, *Computer Methods in Applied Mechanics and Engineering* 253, 491–504 (2013)
- [EF98] Eich-Söllner, E.; Führer, C.: *Numerical Methods in Multibody Dynamics*, Teubner Verlag, Stuttgart (1998)
- [Far96] Farin, G.E.: *Curves and Surfaces for Computer-Aided Geometric Design*, 4th ed., Academic Press, Orlando, FL (1996)
- [FHK02] Farin, G.E.; Hoschek, J.; Kim, M.S.: *Handbook of computer aided geometric design*, Elsevier (2002)
- [Goy11] Goyal, A.: *Numerical Analysis of the Multibody System coupled with Rubber Components*, Master's Thesis, TU Kaiserslautern (2011)
- [GDSV13] Goyal, A.; Dörfel, M. R.; Simeon, B.; Vuong, A. V.: Isogeometric shell discretizations for flexible multibody dynamics, *Multibody System Dynamics* 30, 139–151 (2013)
- [Hug00] Hughes, T.J.R.: *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*, Dover Publications, Mineola, NY (2000)
- [HCB05] Hughes, T.J.R.; Cottrell, J.A.; Bazilevs, Y.: Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement, *Computer Methods in Applied Mechanics and Engineering* 194, 4135–4195 (2005)
- [HLR89] Hairer, E.; Lubich, C.; Roche, M.: *The Numerical Solution of Differential-Algebraic Systems By Runge-Kutta Methods*, Springer, New York (1989)
- [HW88] Hairer, E.; Wanner, G.: *Radau5 - an implicit Runge-Kutta code*, Report, Universet  de Gen ve, Dept. de math matiques, Gen ve (1988)
- [HW96] Hairer, E.; Wanner, G.: *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*, 2nd ed., Springer, Berlin (1996)
- [JKD14] Johannessen, K. A.; Kvamsdal, T.; Dokken, T.: Isogeometric analysis using LR B-splines, *Computer Methods in Applied Mechanics and Engineering* 269, 471–514 (2014)
- [Kie11] Kiendl, J.: *Isogeometric Analysis and Shape Optimal Design of Shell Structures*, Ph.D. Dissertation, Lehrstuhl f r Statik, TU M nchen (2011)
- [Koi66] Koiter, W. T.: On the Nonlinear Theory of Thin Elastic Shells, *Proceedings Koninklijke Nederlandse Akademie van Wetenschappen* 69(1), 1–54 (1966)

- [KBH⁺10] Kiendl, J.; Bazilevs, Y.; Hsu, M.-C.; Wüchner, R.; Bletzinger, K.-U.: The bending strip method for isogeometric analysis of Kirchhoff-Love shell structures comprised of multiple patches, *Computer Methods in Applied Mechanics and Engineering* 199, 2403–2416 (2010)
- [KBLW09] Kiendl, J.; Bletzinger, K.-U.; Linhard, J.; Wüchner, R.: Isogeometric shell analysis with Kirchhoff-Love elements, *Computer Methods in Applied Mechanics and Engineering* 198, 3902–3914 (2009)
- [KM06] Kunkel, P.; Mehrmann, V.: *Differential-Algebraic Equations - Analysis and Numerical Solution*, EMS Publishing House, Zürich (2006)
- [KPJT12] Kleiss, S. K.; Pechstein, C.; Jüttler, B.; Tomar, S.: IETI-isogeometric tearing and interconnecting, *Computer Methods in Applied Mechanics and Engineering* 247, 201–215 (2012)
- [KV86] Kim, S.S.; Vanderploeg, M.J.: QR decomposition for state space representation of constrained mechanical dynamic systems, *Journal of Mechanisms, Transmissions and Automation in Design* 108, 183–188 (1986)
- [Lor86] Lorentz, G.G.: *Bernstein Polynomials*, New York: Chelsea Publishing Co. (1986)
- [Lub93] Lubich, Ch.: Integration of stiff mechanical systems by Runge-Kutta methods, *Z angew Math Phys* 44, 1022–1053 (1993)
- [LC90] Lemaitre, J.; Chaboche, J. L.: *Mechanics of solid materials*, Cambridge university press (1990)
- [LLT13] Linn, J.; Lang, H.; Tuganov, A.: Geometrically exact Cosserat rods with Kelvin-Voigt type viscous damping, *Mechanical Sciences* 4, 79–96 (2013)
- [MHA85] Mani, N.K.; Atkinson, K.E.; Haug, E.J.: Application of singular value decomposition for analysis of mechanical system dynamics, *Journal of Mechanisms, Transmissions and Automation in Design* 107, 82–87 (1985)
- [Pet82] Petzold, L.R.: Description of DASSL: a differential/algebraic system solver, *Scientific Computing*, R.S. Stepleman et al., eds., North Holland, Amsterdam, 65–68 (1983)
- [Pie91] Piegl, L.: On NURBS: a survey, *IEEE Computer Graphics and Applications* 11(1), 55–71 (1991)
- [Pra84] Prautzsch, H.: Degree elevation of B-spline curves, *Computer Aided Geometric Design* 1(2), 193–198 (1984)
- [PT97] Piegl, L.; Tiller, W.: *The NURBS Book*, second ed., Springer-Verlag, New York (1997)
- [Rog01] Rogers, D.F.: *An Introduction to NURBS With Historical Perspective*, Academic Press, San Diego, CA (2001)
- [RA90] Rogers, D.F.; Adams, J.A.: *Mathematical elements for computer graphics*, second ed., McGraw-Hill, New York (1990)

- [Sha98] Shabana, A.A.: Dynamics of Multibody Systems, Cambridge University Press, Cambridge (1998)
- [Sim98] Simeon, B.: Order reduction of stiff solvers at elastic multibody systems Applied Numerical Mathematics 28, 459–475 (1998)
- [Sim00] Simeon, B.: Numerische Simulation gekoppelter Systeme von partiellen und differential-algebraischen Gleichungen in der Mehrkörperdynamik, VDI-Verlag, Düsseldorf (2000)
- [Sim01] Simeon, B.: Numerical Analysis of Flexible Multibody Systems, Multibody System Dynamics 6, 305–325 (2001)
- [Sim13] Simeon, B.: Computational Flexible Multibody Dynamics, Springer-Verlag, Heidelberg New York Dordrecht London (2013)
- [SDT⁺12] Schulze, M.; Dietz, S.; Tuganov, A.; Lang, H., Linn, J.: Integration of nonlinear models of flexible body deformation in Multibody System Dynamics, Proceedings of the IMSD2012 - The 2nd Joint International Conference on Multibody System Dynamics, May 29 - June 1, 2012, Stuttgart, Germany, P. Eberhard and P. Ziegler Eds. (2012)
- [SLL04] Sze, K.Y.; Liu, X.H.; Lo, S.H.: Popular benchmark problems for geometric nonlinear analysis of shells, Finite Elements in Analysis and Design 40, 1551–1569 (2004)
- [SLSH12] Scott, M.A.; Li, X.; Sederberg, T.W.; Hughes, T.J.R.: Local refinement of analysis-suitable T-splines Computer Methods in Applied Mechanics and Engineering 213–216, 206–222 (2012)
- [SS02] Schaub, M.; Simeon, B.: Automatic h-Scaling for the Efficient Time Integration of Flexible Multibody Systems, Multibody System Dynamics 8, 329–345 (2002)
- [SWB12] Schmidt, R.; Wüchner, R.; Bletzinger, K.-U.: Isogeometric analysis of trimmed NURBS geometries, Computer Methods in Applied Mechanics and Engineering 241–244, 93–111 (2012)
- [SW99] Schwertassek, R.; Wallrapp, O.: Dynamik Flexibler Mehrkörpersysteme, Vieweg Verlag, Braunschweig (1999)
- [TB97] Trefethen, L. N., Bau III, D.: Numerical linear algebra, Society for Industrial and Applied Mathematics 50 (1997)
- [VdP26] Van der Pol, B.: LXXXVIII. On "relaxation-oscillations", The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science 2(11), 978–992 (1926)
- [Ver75] Versprille, K.J.: Computer-aided design applications of the rational B-spline approximation form, Ph.D. Dissertation, Syracuse Univ. (1975)
- [VGJS11] 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 200, 3554–3567 (2011)
- [VHS10] Vuong, A.-V.; Heinrich, Ch.; Simeon, B.: ISOGAT: A 2D tutorial MATLAB code for Isogeometric Analysis, Computer Aided Geometric Design 27(8), 644–655 (2010)

- [VS99] Von Schwerin, R.: MultiBody System SIMulation: Numerical Methods, Algorithms, and Software, Lecture Notes in Computational Science and Engineering 7, Springer-Verlag (1999)
- [Wal91] Wallrapp, O.: Linearized flexible multibody dynamics including geometric stiffening effects, *Mechanics of Structures and Machines* 19, 385–409 (1991)
- [WFC08] Wall, W. A.; Frenzel, M. A.; Cyron, C.: Isogeometric structural shape optimization, *Computer Methods in Applied Mechanics and Engineering* 197(33), 2976–2988 (2008)
- [WH82] Wehage, R.; Haug, E.J.: Generalized coordinate partitioning for dimension reduction in analysis of constrained mechanical systems, *Journal of Mechanical Design* 104, 247–255 (1982)
- [WWS13] Weeger, O.; Wever, U.; Simeon, B.: Isogeometric analysis of nonlinear Euler-Bernoulli beam vibrations, *Nonlinear Dynamics* 72(4), 813–835 (2013)
- [ZTZ05] Zienkiewicz, O. C.; Taylor, R. L.; Zhu, J. Z.: *The finite element method: its basis and fundamentals*, Butterworth-Heinemann, Vol. 1 (2005)

Akademischer Lebenslauf

Schulbildung:

1991 – 2004: Ramjas Public School, Delhi

Studium:

Aug 2004 – Mai 2008: Maschinenbau - Bachelor of Engineering,
Delhi College of Engineering

Okt 2008 – Mai 2011: Mathematik - Master of Science,
Technische Universität Kaiserslautern

Akademische Anstellungen:

seit Sept 2011: Wissenschaftlicher Mitarbeiter,
Technische Universität Kaiserslautern

Academic Curriculum Vitae

School Education:

1991 – 2004: Ramjas Public School, Delhi

Higher Education:

Aug 2004 – May 2008: Mechanical Engineering - Bachelor of Engineering,
Delhi College of Engineering

Oct 2008 – May 2011: Mathematics - Master of Science,
University of Kaiserslautern

Academic Employments:

since Sept 2011: Research Associate,
University of Kaiserslautern