Bauhaus-Universität Weimar

Procedurally generated models for Isogeometric Analysis

(Prozedural generierte Modelle für Isogeometrische Analysen)

DISSERTATION

Zur Erlangung des akademischen Grades

Doktor - Ingenieur

an der Fakultät Bauingenieurwesen
der Bauhaus-Universität Weimar

vorgelegt von

Dipl.-Ing. Peter Stein

geboren am 12. Dezember 1983 in Dresden

Hauptberichter:	Prof. Dr.-Ing. Karl Beucke, Bauhaus-Universität Weimar
Mitberichter:	Assistant Professor Yuri Bazilevs, University of California, San Diego
Mitberichter:	Prof. Dr. rer. nat. habil. Klaus Gürlebeck, Bauhaus-Universität Weimar

Tag der Disputation: 30. 10. 2012

For my parents.

Abstract

Increasingly powerful hard- and software allows for the numerical simulation of complex physical phenomena with high levels of detail. In light of this development the definition of numerical models for the *Finite Element Method* (FEM) has become the bottleneck in the simulation process. Characteristic features of the model generation are large manual efforts and a de-coupling of geometric and numerical model. In the highly probable case of design revisions all steps of model preprocessing and mesh generation have to be repeated. This includes the idealization and approximation of a geometric model as well as the definition of boundary conditions and model parameters. Design variants leading to more resource-efficient structures might hence be disregarded due to limited budgets and constrained time frames.

A potential solution to above problem is given with the concept of *Isogeometric Analysis* (IGA). Core idea of this method is to directly employ a geometric model for numerical simulations, which allows to circumvent model transformations and the accompanying data losses. Basis for this method are geometric models described in terms of *Non-uniform rational B-Splines* (NURBS). This class of piecewise continuous rational polynomial functions is ubiquitous in computer graphics and *Computer-Aided Design* (CAD). It allows the description of a wide range of geometries using a compact mathematical representation. The shape of an object thereby results from the interpolation of a set of *control points* by means of the NURBS functions, allowing efficient representations for curves, surfaces and solid bodies alike. Existing software applications, however, only support the modeling and manipulation of the former two. The description of three-dimensional solid bodies consequently requires significant manual effort, thus essentially forbidding the setup of complex models.

This thesis proposes a procedural approach for the generation of volumetric NURBS models. That is, a model is not described in terms of its data structures but as a sequence of modeling operations applied to a simple initial shape. In a sense this describes the "evolution" of the geometric model under the sequence of operations. In order to adapt this concept to NURBS geometries, only a compact set of commands is necessary which, in turn, can be adapted from existing algorithms. A model then can be treated in terms of interpretable model parameters. This leads to an abstraction from its data structures and model variants can be set up by variation of the governing parameters.

The proposed concept complements existing template modeling approaches: templates can not only be defined in terms of modeling commands but can also serve as input geometry for said operations. Such templates, arranged in a nested hierarchy, provide an elegant model representation. They offer adaptivity on each tier of the model hierarchy and allow to create complex models from only few model parameters. This is demonstrated for volumetric fluid domains used in the simulation of vertical-axis wind turbines. Starting from a template representation of airfoil cross-sections, the complete "negative space" around the rotor blades can be described by a small set of model parameters, and model variants can be set up in a fraction of a second.

NURBS models offer a high geometric flexibility, allowing to represent a given shape in different ways. Different model instances can exhibit varying suitability for numerical

analyses. For their assessment, Finite Element mesh quality metrics are regarded. The considered metrics are based on purely geometric criteria and allow to identify model degenerations commonly used to achieve certain geometric features. They can be used to decide upon model adaptions and provide a measure for their efficacy. Unfortunately, they do not reveal a relation between mesh distortion and ill-conditioning of the equation systems resulting from the numerical model.

Keywords NURBS, Isogeometric Analysis, Procedural modeling, Templates, Mesh quality, Vertical Axis Wind Turbine

Zusammenfassung

Die zunehmende Rechenleistung moderner Hardware erlaubt, in Verbund mit effizienten Algorithmen und leistungsfähiger Software, die numerische Simulation immer komplexerer Fragestellungen. Dadurch können Modelle in höheren Detaillierungsgraden betrachtet werden. Infolge ihres hohen manuellen Aufwandes entwickelt sich dabei die Generation numerischer Modelle für die Methode der *Finiten Elemente* (FEM) zum Flaschenhals in der digitalen Prozesskette. Die im Prozess durchgeführte Idealisierung und Approximation geometrischer Modelldaten führt unweigerlich zu Informationsverlusten und zur Entkopplung von Geometrie und numerischem Modell. Im Falle von – im Ingenieurwesen durchaus häufigen – Planrevisionen müssen alle Schritte der Datenaufbereitung und der Modellvernetzung, sowie der Definition von Materialdaten und Randbedingungen neu durchgeführt werden. Das ist teuer und zeitaufwändig. Modellvarianten, die in effizienteren Strukturen resultieren könnten, werden so unter Umständen nicht in Betracht gezogen.

Das Konzept der *Isogeometrischen Analyse* (IGA) verspricht eine Vereinfachung der genannten Vorgehensweise. Kernidee des Verfahrens ist die direkte Verwendung eines geometrischen Modells für numerische Simulationen. Dadurch können Modelltransformationen umgangen und Datenverluste vermieden werden. Grundlage hierfür sind Modelle auf der Basis von *Non-uniform rational B-Splines* (NURBS), einer Klasse stückweise stetiger, gebrochenrationaler Funktionen. Diese besitzen eine weite Verbreitung im Bereich der Computergrafik und des *Computer-Aided Design* (CAD), da sie die Modellierung einer weiten Spanne geometrischer Objekte erlauben. Zudem stehen stabile und effiziente Algorithmen für ihre Manipulation zur Verfügung. Die Geometrie ergibt sich dabei aus der Interpolation einer Reihe von *Kontrollpunkten* durch ihnen zugeordnete Basisfunktionen. NURBS fanden bisher nur Anwendung zur Beschreibung von Kurven und Freiformflächen, die Methoden und Algorithmen für ihre Modellierung sind dementsprechend auf diese Formen beschränkt. Die Beschreibung dreidimensionaler Körper, so genannter *Solids*, erfordert hingegen die manuelle Definition der zugrunde liegenden Datenstrukturen. Das ist fehleranfällig und – insbesondere bei komplexeren Modellen – mühselig.

Zur Unterstützung dieses Prozesses wird in dieser Arbeit ein prozeduraler Ansatz gewählt, das heißt, volumetrische Modelle werden nicht durch ihre Datenstrukturen beschrieben, sondern als eine Sequenz von Modellierungsoperationen, die auf eine einfache Grundgeometrie angewandt werden. Man beschreibt also gewissermaßen die „Evolution“ eines geometrischen Modells unter den Modellierungsschritten. Die Übertragung dieses Konzepts auf NURBS-Geometrien erfordert nur eine geringe Menge an Basis-Operationen, zum Beispiel Koordinatentransformationen oder Netzverfeinerung. Diese sind Grundbestandteile existierender Modellierungssoftware oder lassen sich für die Modellierung volumetrischer Geometrien erweitern. Die Darstellung durch Operatoren und Templates abstrahiert von den zugrundeliegenden Datenstrukturen und erlaubt die Modellbehandlung in Form semantisch gehaltvoller Parameter. Deren Variation erlaubt die schnelle Erzeugung von Modellvarianten.

Der hier beschriebene Ansatz ergänzt bestehende Ansätze der Template-Modellierung insofern, als dass Templates nicht nur in Form von Modellierungskommandos beschrieben werden können, sondern dass sie auch als Input für Sequenzen von Operationen die-

nen können. Die hierarchische Verschachtelung von Templates erlaubt eine elegante Darstellung komplexer Geometrien schon durch wenige Modellparameter. Zudem ermöglicht dies Modelladaptionen auf jeder Stufe der Modellhierarchie, was anhand von Modellen für die Fluid-Struktur-Interaktion von Windenergieanlagen demonstriert wird. Ausgehend von Templates für die Flügelprofile kann der gesamte „Negativraum“ um die Rotorblätter durch eine Handvoll Parameter beschrieben werden. Die Auswertung der Templates benötigt dabei nur den Bruchteil einer Sekunde.

Zur Bewertung der Modellqualität der hier beschriebenen Geometrien werden eine Reihe von Netzqualitäts-Metriken aus der FE-Literatur verwendet. Diese basieren auf rein geometrischen Faktoren, zum Beispiel auf den Eigenschaften der Koordinatentransformationen die im Rahmen Isogeometrischer Analysen durchgeführt werden. Anhand einer Reihe einfacher Beispiele wird das Verhalten der Metriken sowohl quantitativ als auch qualitativ studiert. Diese identifizieren zuverlässig Netzverzerrungen die oft zum Erreichen bestimmter geometrischer Merkmale eingesetzt werden. Die Metriken bilden eine Entscheidungsgrundlage für Modelladaptionen und lassen den Effekt dieser auf die Netzgüte klar erkennen. Leider lässt sich kein Zusammenhang erkennen zwischen Netzverzerrungen und den Eigenschaften eines numerischen Modells, namentlich der Kondition der Gleichungssysteme.

Schlagworte NURBS, Isogeometrische Analyse, Prozedurale Modellierung, Templates, Modellqualität, Vertikalachswindturbine

Preface

This thesis is the result of my research at the *Research Training Group 1462* at Bauhaus-Universität Weimar from November 2008 to November 2011. The successful conclusion of these three years of intensive work would not have been possible without the support of a number of individuals.

First and foremost I would like to thank my mentor Prof. Dr.-Ing. Karl Beucke for his supervision of my research, the opportunity to work in the exciting field of Isogeometric Analysis, and for giving me the freedom to explore the ideas that came up during my work. I thank Univ.-Prof. Dr. rer. nat. habil. Klaus Gürlebeck for his remarks and suggestions, the many fruitful discussions, as well as for his review of this thesis. Thanks are also in order for Yuri Bazilevs, at the time of this writing Associate Professor at the Department of Structural Engineering of the University of California, San Diego. I highly appreciate to have had the chance to work with him and his group for a period of three months. It has been his application of wind energy turbine simulation that has given this work the last push necessary to turn into a PhD thesis.

I would like to thank my colleagues at the Research Training Group 1462, in particular Dr.-Ing. Thomas Most, Dr. rer. nat. Tom Lahmer, Dr. rer. nat. Sebastian Bock, Dr.-Ing. Toni Fröbel, Dr.-Ing. Mourad Nasser and Dr.-Ing. Markus Reuter, for their remarks, the lively discussions, and the great working spirit at our institute, as well as for our occasional ventures away from the cold, hard sciences.

I am very thankful towards my family and my friends for their support not only during the time as a PhD candidate but during my whole life.

Finally, I would like to express my gratitude towards the German Research Council DFG who provided the financial support for this work.

Contents

Symbols and abbreviations

Latin letters

$c_{i,j}$	Coefficients of the piecewise polynomial on the knot interval $[\xi_j, \xi_{j+1})$
i, j, k	(Discrete) grid coordinates, indices
x, y, z	Physical coordinates
p, q, r	Polynomial degrees
$\mathbf{P}^h$	Control point in homogeneous space
$\mathbf{P}$	Control point in physical space
$\bar{\mathbf{Q}}, \mathbf{Q}$	Point samples along airfoil profile curves
N_i^p	B-Spline function of degree p corresponding to point $\mathbf{P}_i$
A_i	Weighted B-Spline function corresponding to point $\mathbf{P}_i$
R_i	NURBS function corresponding to point $\mathbf{P}_i$
h	Homogeneous Coordinate
$\mathbb{R}^n$	n-dimensional space of real numbers
$\mathbb{P}^n$	n-dimensional homogeneous space
$\mathscr{P}$	Perspective projection
$\mathscr{S}$	Set of control points
$\mathscr{N}$	Mapping from the parametric NURBS domain to physical space
$\mathscr{G}$	Mapping from the biunit parent domain to the NURBS parameter space
$\mathbf{J}_{\mathscr{N}}$	Jacobian of the mapping $\mathscr{N}$
$\mathbf{J}_{\mathscr{G}}$	Jacobian of the mapping $\mathscr{G}$
f_{Oddy}	Oddy metric
$f_{\perp}$	Orthogonality metric
f_{det}	Determinant metric
f_{cond}	Condition metric
$\mathbb{C}_{ijkl}$	Components of the fourth-order constitutive tensor
$\mathbf{u}, u_i$	Displacement field
$\delta\mathbf{u}, \delta u_i$	Virtual displacement field
$\bar{\mathbf{t}}, \bar{t}_i$	Boundary tractions
$\mathbf{b}, b_i$	Body forces
$\mathbf{n}, n_j$	Outward normal vector
E	Young's modulus
$\mathbf{R}$	Matrix of basis functions
$\mathbf{C}$	Matrix representation of the constitutive tensor
$\mathscr{D}_k$	Operator matrix for the infinitesimal strain kinematic
$\mathbf{B}$	Strain-displacement matrix
$\mathbf{K}$	Stiffness matrix
$\mathbf{f}$	Vector of external loads
$\mathbf{x}_G$	Gauss point
s, t, u	Gauss point coordinates

Greek letters

ξ, η, ζ	Parametric coordinates
$\boldsymbol{\Xi}, \mathbf{H}, \mathbf{Z}$	Knot vectors corresponding to ξ, η, ζ
Ω^n	n-dimensional domain in physical space
Ω_e^n	n-dimensional domain of a finite element
Ω_G^d	d-dimensional biunit domain for Gauss integration
$\bar{\Omega}^m$	m-dimensional parametric domain of NURBS functions
Γ	Boundary of Ω^n
Γ_t	Neumann boundary
Γ_u	Dirichlet boundary
ϵ_{kl}	Components of the second-order infinitesimal strain tensor
$\delta\epsilon_{kl}$	Components of the virtual strain field
σ_{ij}	Components of the second-order Cauchy stress tensor
ν	Poisson's ratio
$\boldsymbol{\sigma}$	Vector representation of the stress tensor
$\boldsymbol{\epsilon}$	Vector representation of the strain tensor
$\delta\boldsymbol{\epsilon}$	Vector representation of the virtual strain
κ	Condition number of the stiffness matrix

General notation

$\|\cdot\|_2$	l^2 norm of a vector
$\|\cdot\|_F$	Frobenius norm of a matrix

Abbreviations

CAD	Computer-Aided Design
FE/FEM	Finite Element Method
IGA	Isogeometric Analysis
NURBS	Non-uniform rational B-Splines
VAWT	Vertical-Axis Wind Turbine
HAWT	Horizontal-Axis Wind Turbine

1 Introduction

Numerical simulations using the *Finite Element Method* (FEM) are a fundamental part in engineering design. Once being restricted to *a posteriori* failure analyses, this method is now applied to study the influence of model parameters already during the design process (Beall et al., 2003). This is made possible by the increasingly powerful soft- and hardware that has become available to the engineers in recent years. Fueled by the demands of the consumer electronics sector, multi-core processors, dedicated graphics chips, and large quantities of memory have become commonplace even in desktop computers. This allows for treatment of large numerical models and three-dimensional problems as well as for consideration of coupled physical effects. However, whereas the actual solution of the finite element model once represented the major part of the solution procedure it is now the model setup that has become the bottleneck of the simulation process. Current estimates state that about 80% of analysis time are spent on the generation of numerical models from *Computer-Aided Design* (CAD) geometries (Hughes et al., 2005). This is due to the fact that automatic Finite Element mesh generation is possible only for certain classes of problems. General cases, in contrast, require significant manual intervention on behalf of the engineers (Shimada, 2006).

Engineering tasks possess a predominantly geometric character. The majority of the shapes to be analyzed is nowadays provided in form of 3D CAD models (Sheffer and Üngör, 2001). These cannot be used directly for numerical simulations but have to be preprocessed and converted in multiple steps into an analysis-suitable representation (Sheffer et al., 1997). The single phases of this process are illustrated in Figure 1.1.

The first step of this transformation sequence is concerned with the removal of "dirty geometry". These inconsistencies in the geometric model are a result of data losses during model transfer and the incompatibility between the modeling software of different CAD vendors (Beall et al., 2003), in particular with respect to the internal representation and the handling of tolerances (Ibid.). As a consequence, CAD geometries often exhibit various geometric errors such as gaps and overlaps, self-intersections, inverted faces, or degenerate surfaces (Ames et al., 1997; Beall et al., 2003; White et al., 2005). Understandably, the demand for "healing software" is high.

The second step deals with geometric simplification and idealization. CAD models, in general, serve multiple purposes of which numerical simulation is but an aspect. Instead they are created with manufacturing or visualization in mind (White et al., 2005). Accordingly, their level of geometric detail far exceeds the requirements of numerical analyses (Ames et al., 1997). However, detail removal and model simplification can only be automatized to a certain degree, for instance in the identification of slender model parts by means of the *Medial-axis transform* (cf. Choi and Han, 2002, and the references therein). In addition, the identification of excessive model detail requires engineering judgment.

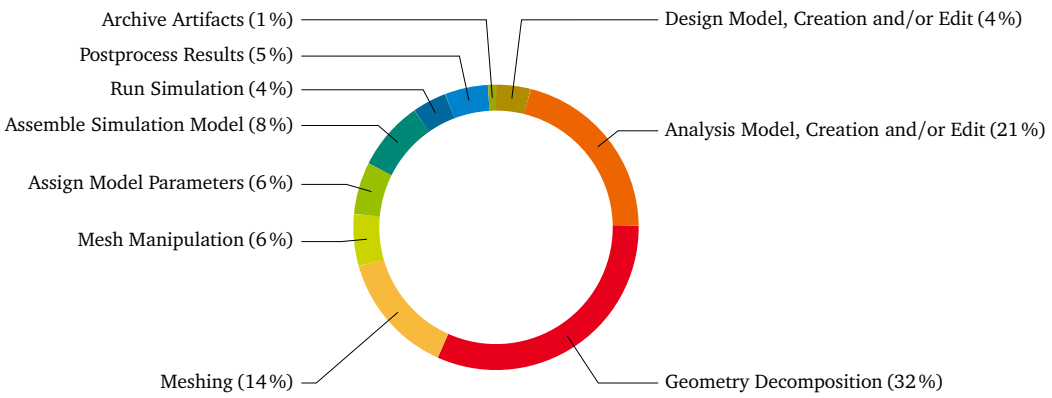

Figure 1.1: The steps of a numerical analysis and their respective fraction of the overall time and efforts spent. It can be easily recognized that the actual simulation process constitutes only a small share of the expenses whereas the overwhelming part is due to data treatment and model preprocessing. The data and the designations in above illustration originate from a survey performed by Sandia National Labs which has been cited in (Bazilevs et al., 2010).

The final step of preprocessing is the transformation of the CAD model into a form suitable for numerical simulations. CAD models commonly describe volumetric objects in terms of their boundary surfaces. Such *boundary representations* (cf. Shah and Mäntylä, 1995; Stroud, 2006) provide only incomplete information for three-dimensional analyses (Cohen et al., 2010). The geometry therefore has to be converted into a finite element mesh. From the point of computational geometry these meshes are so-called *cell decompositions* (Mäntylä, 1987), that is, sets of connected, non-overlapping shape primitives such as tetra- or hexahedra. Cell decompositions allow to represent a wide range of shapes but are merely an approximation to the original geometry, in particular as Finite Element schemes are predominantly based on multilinear elements. The approximation quality of the cell decomposition hence strongly depends on the resolution and the density of the element mesh. This can manifest itself in convergence problems in the simulation of contact problems (where the faceted representation leads to ambiguities in the detection of contact) or as spurious boundary layers in fluid flow problems (Hughes et al., 2005).

Cell decompositions lack information on the associativity of their constituents (Shah and Mäntylä, 1995). Changes in single entities thus cannot be propagated throughout a model. Consequently, the complete sequence of model repair, simplification, and meshing has to be repeated in the event of model revisions (Mäntylä, 1987; Sheffer and Üngör, 2001) since geometry and numerical model are effectively de-coupled.

Modern product design processes involve a multitude of specialized consultants for diverse domains such as marketing, recycling, costs, and energy efficiency—in addition to key areas such a design, analysis, and manufacturing. Design iterations will therefore occur with a high frequency, requiring the propagation of consistent model data to all par-

ties involved. The current simulation pipeline clearly conflicts with such iterative design processes and their frequent model revisions (Hughes et al., 2005). Given the large efforts for constructing numerical models, it must be assumed that model variants are considered seldom, if at all. That is, alternative designs that might result in more resource-efficient structures are excluded a priori due to constrained time and budget considerations.

These factors, both the geometric approximation and the lack of associativity, motivated the introduction of the concept of *Isogeometric Analysis* (IGA) (Hughes et al., 2005). IGA is an extension of FEM in which a geometric model can be directly used for the solution of a mathematical model given by partial differential equations. For IGA the geometry has to be described in terms of parametric functions. Such a representation is given, for instance, in form of *Non-uniform rational B-Splines* (NURBS) curves and surfaces that are ubiquitous in the field of computer graphics and CAD. The characteristic feature of that representation scheme is the interpolation of a set of *control points* using a set of smooth *Spline* functions. Curves accordingly use univariate splines that depend on a single variable. Geometric objects such as surfaces and solid bodies are described by bi- and trivariate functions, depending on two and three parametric variables, respectively. The degree of these functions and their continuity, together with the topological structure of the control points, lead to characteristic geometric features, allowing to describe a wide range of geometries. IGA is founded on the idea to use these basis functions as ansatz for the solution of systems of partial differential equations—defined on the domain of the geometric model. This leads to a full coupling of geometry and numerical model. NURBS are implemented in the major CAD software packages and mature algorithms are available for their manipulation and visualization. In combination with IGA they provide a uniform data structure for the complete simulation process—from design over analysis to visualization. Errors due to model transformation can thus be avoided completely (Cohen et al., 2010).

Smooth higher-order shape descriptions as given by NURBS are invaluable for problem settings that are sensitive to geometric imperfections. Accordingly, a major field of application of IGA can be found in fluid flow simulations. Applications so far include fluid-structure interaction and flow about rotating components (Bazilevs et al., 2008, 2011; Bazilevs and Hughes, 2008; Hsu et al., 2011), turbulence simulations (Akkerman et al., 2008; Bazilevs et al., 2007, 2010; Cottrell, 2007) and studies of arterial blood circulation and drug transport (Bazilevs et al., 2006, 2009; Calo et al., 2008; Zhang et al., 2007).

Being an extension of the Finite Element concept, IGA can be implemented relatively easy into existing software packages, as shown by Rypl and Patzák (2011) who deal with the integration of isogeometric concepts into an object-oriented FE-framework. The major difference between FEM and IGA are the employed basis functions whose higher continuity allows for a drastic reduction in the number of Gauss points (Hughes et al., 2010). Fundamental implementation aspects such as the evaluation routines for basis functions and the architecture of single- and multi-patch IGA code are described by Cottrell et al. (2009). A parallel implementation of IGA using the *Message Passing Interface* is described in (Calo et al., 2008).

There remains a fundamental problem, though: NURBS have so far been applied predominantly for the representation of curves and free-form surfaces. Modeling applications

therefore provide only methods and algorithms for modeling and manipulation of uni- and biparametric geometries. Consequently, IGA has found many applications for the study of thin-walled structures (cf. Benson et al., 2010, 2011; Kiendl et al., 2010, 2009). However, modeling applications for the setup of three-dimensional, volumetric NURBS models are lacking, meaning that users have to manually define numerical models. To make matters worse, volumetric models require far more data than curves or surfaces in order to describe the interior of a body. This makes the generation of *solid* NURBS models a tedious and error-prone process, effectively forbidding the setup and the simulation of complex models for Isogeometric Analyses.

Solution approach

In order to overcome these restrictions, a *procedural approach* to model generation is proposed. Instead of describing the data structures of a complex model, its "evolution" from a simple initial shape is represented. Basis for such a description is a compact set of modeling operators comprising coordinate transformations, set operations, and mesh refinement operations together with established modeling algorithms. The modeling process of an intricate object can thereby be described as a sequence of modeling operations which provides a concise and parametrized representation of geometric objects. Consistent generation of a model's data structures is ensured by the individual operations. Lists of modeling operations can further be stored as *templates* that provide a multi-level adaptive description of both single objects and complex structures.

Outline of the thesis

This thesis begins with an overview of the mathematical properties of Spline functions and their use for the description of parametric solids in Chapter 2. Building on these properties, Chapter 3 starts with a review of existing modeling approaches. It gives a detailed description of the concept of procedural modeling in computer graphics and shows its adaption to NURBS-based geometries. Therefor, a compact set of modeling operations is described. Their application to NURBS modeling is demonstrated on small sample geometries. A combination of this concept with existing template modeling approaches is elaborated. Chapter 4 serves as application example for the proposed modeling concept and demonstrates the generation of three-dimensional fluid domains for wind energy turbine simulations. Starting point for these models are discrete data sets describing common airfoil sections. Their conversion into airfoil templates involves the reduction of the initial data sets by means of two simplification algorithms. The final airfoil templates are then employed as input for a set of nested templates, resulting in the geometry of the fluid domain.[1] Chapter 5 continues with the assessment of a model's appropriateness for numerical analysis. Using the boundary value problem of linear elastostatics, the application of NURBS basis functions for Finite Element procedures is illustrated. This highlights the influence of the geometric representation scheme on the resulting numerical models. Based on the mappings between the shapes' parametric domain and physical space, a handful of *mesh quality metrics* are formulated. Their behavior is studied on several test models. Chapter 6 then concludes with a discussion of the achieved results,

[1] Parts of Chapters 3 and 4 have already been published in (Stein et al., 2012).

2 NURBS-based solid models

B-Splines[1] and NURBS provide a parametric representation of geometries. They describe an object's shape in terms of points resulting from the mapping

$$\mathcal{N} : \bar{\Omega}^m \to \Omega^n \subset \mathbb{R}^n, \tag{2.1}$$

where $\bar{\Omega}^m$ denotes a m-dimensional parameter space and Ω^n a subset of the (physical) space—the object of interest. In the scope of this work above mapping is given as follows: a collection of n-dimensional data points is interpolated using a set of m-variate functions. Such parametric representations are independent of a chosen coordinate system and allow to interpolate data of basically arbitrary dimension (Rogers, 2001). The dimension of the parameter space $\bar{\Omega}^m$ and the properties of the basis functions have a large influence on the resulting geometry, though.

A simple and straightforward approach is to use polynomial functions for the mapping (2.1) as it is done for the Bézier curves and surfaces based on the Bernstein polynomials

$$B_i^n(\xi) = \binom{n}{i} \xi^i (1-\xi)^{n-i}, \quad \xi \in [0,1], \tag{2.2}$$

where the binomial coefficients are defined as

$$\binom{n}{i} = \frac{n!}{i!\,(n-i)!}.$$

Polynomial functions such as in (2.2) are simple, but possess drawbacks that make them ill-suited for shape representation. In order to interpolate a large number of points, for instance for a detailed object representation, high-order polynomials are required which are inefficient for processing and prone to numerical instabilities (Piegl and Tiller, 1997). They also have a *non-local support*, that is, individual data points affect the shape of the complete object, making local shape control impossible (Rogers, 2001). In order to overcome these limitations, piecewise polynomial functions can be employed, of which B-Splines are but one kind.

2.1 B-Spline functions

B-Splines are defined on a one-dimensional parametric domain $\bar{\Omega}^1 := [\xi_a, \xi_b] \subset \mathbb{R}$. This domain is subdivided into segments, called *knot spans*. The values ξ_i at the respective joints between two successive segments are called *knots*. The sequence of knots is stored

[1] **B**asis-Splines

in ascending order in a *knot vector* Ξ, i.e., $\xi_i \leq \xi_{i+1}$. The knot spans $[\xi_j, \xi_{j+1})$ serve as domain for a set of polynomial segments of degree p:

$$P_j^p(\xi) = \sum_{i=0}^{p} c_{i,j}\, \xi^i, \qquad \xi \in [\xi_j, \xi_{j+1}). \tag{2.3}$$

Within each segment the spline function is a C^∞-continuous polynomial. In order to join pairs of different polynomials at a knot, the continuity of the functions at the joint has to be reduced to *at most* C^{p-1} (Bartels et al., 1987). This is related to the polynomial coefficients $c_{i,j}$ and $c_{i,j+1}$: enforcement of continuity higher than C^{p-1} would result in the pairwise equality of these coefficients and hence in the identity of the polynomials.

The continuity of the functions might be even lower as a result of identical successive knot values, as Bartels et al. (1987) point out. Such cases can be interpreted as compacting one or several knot spans to a segment of zero length. Each knot repetition diminishes the continuity of the basis functions. A knot value ξ_i with multiplicity k therefore reduces the continuity of the basis functions to C^{p-k} (Ibid.).

Different methods are available for determining B-Spline basis functions (cf. Bartels et al., 1987; Höllig, 2003; Piegl and Tiller, 1997; Zorin and Schröder, 2000, and the references therein). Numerical implementations commonly make use of the recurrence relation of Cox-De Boor (Cox, 1972; De Boor, 1972) which provides an efficient and numerically stable routine. Given a knot vector with $n+p+2$ entries, it computes $(n+1)$ B-Spline basis functions $N_i^p(\xi)$ of degree p:

$$N_i^p(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} N_i^{p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1}^{p-1}(\xi), \tag{2.4}$$

starting from the piecewise constant functions

$$N_i^0(\xi) = \begin{cases} 1 & \text{for } \xi \in [\xi_i, \xi_{i+1}) \\ 0 & \text{else.} \end{cases} \tag{2.5}$$

Some of the linear factors in Equation (2.4) can have a zero denominator as a consequence of repeated knot values. In such cases the respective factor is defined to be zero (Piegl and Tiller, 1997).

Two types of knot vectors are used in computer-aided design, namely *periodic* and *open* knot vectors. They lead to different properties of the basis functions and hence to different features of the resulting geometries. Periodic knot vectors prove to be useful for describing closed curves or tubular/toroidal surfaces, as can be seen in the examples in (Bartels et al., 1987; Rogers, 2001). Within this work only open knot vectors are used, though. These are always of the form:

$$\Xi := \{ \underbrace{\xi_a, \ldots, \xi_a}_{p+1}, \xi_{p+1}, \ldots, \xi_n, \underbrace{\xi_b, \ldots, \xi_b}_{p+1} \}, \tag{2.6}$$

that is, both the lower boundary ξ_a and the upper boundary ξ_b of the parametric domain $\bar{\Omega}^1$ possess a multiplicity of $p+1$. Hence, $2(p+1)$ of the $n+p+2$ entries in an open knot

vector are prescribed by these “end conditions”. Open knot vectors thus encode in their form the number, the polynomial degree, and the continuity of the functions resulting from their use in the Cox-De Boor recurrence.

2.1.1 Properties of B-Spline functions

The recurrence relation (2.4), together with the use of open knot vectors, leads to characteristic properties of B-Spline functions, making them useful both for the representation of geometric objects and for Isogeometric analysis. The following list is taken from (Piegl and Tiller, 1997) which also contains proofs for these properties. Figure 2.1 illustrates some of these properties for a set of cubic basis functions.

Local support

$$N_i^p(\xi) = 0 \quad \text{for} \quad \xi \notin [\xi_i, \xi_{i+p+1}). \tag{2.7}$$

Conversely, each B-Spline is nonzero on at most $(p+1)$ knot spans, namely the knot spans

$$[\xi_i, \xi i+1),\ [\xi_{i+1}, \xi i+1),\ \ldots,\ [\xi_i, \xi_{i+p+1}).$$

The actual support might be smaller due to repeated knots. As the basis functions overlap, there are at most $(p+1)$ nonzero basis functions on the knot span $[\xi_j, \xi_{j+1})$, namely the functions $N_{j-p}^p(\xi), \ldots, N_j^p(\xi)$.

Non-negativity

$$N_i^p(\xi) \geq 0 \quad \forall i, p, \xi. \tag{2.8}$$

This property has the effect that the geometric object is fully contained in the *convex hull* of its control points (cf. Rogers, 2001), which simplifies operations such as bounding-box queries.

Partition of unity

$$\sum_{i=0}^{n} N_i^p(\xi) = 1 \quad \forall \xi \in \bar{\Omega}^1. \tag{2.9}$$

Linear independence

$$0 = \sum_{i=0}^{n} \alpha_i N_i^p(\xi) \ \Rightarrow\ \alpha_i = 0\ \forall i. \tag{2.10}$$

Both partition of unity and linear independence determine the B-Splines’ suitability as basis functions for Finite Element procedures. They ensure the convergence of solutions produced by IGA under successive mesh refinement (Cottrell et al., 2009).

Interpolation of boundaries B-Spline functions resulting from open knot vectors fulfill

$$\begin{aligned} &N_0^p\left(\xi = \xi_a\right) = 1, \\ &N_i^p\left(\xi = \xi_a\right) = 0 \quad 0 < i \leq n-1, \\ &N_n^p\left(\xi = \xi_b\right) = 1, \\ &N_i^p\left(\xi = \xi_b\right) = 0 \quad 0 \leq i < n, \end{aligned} \tag{2.11}$$

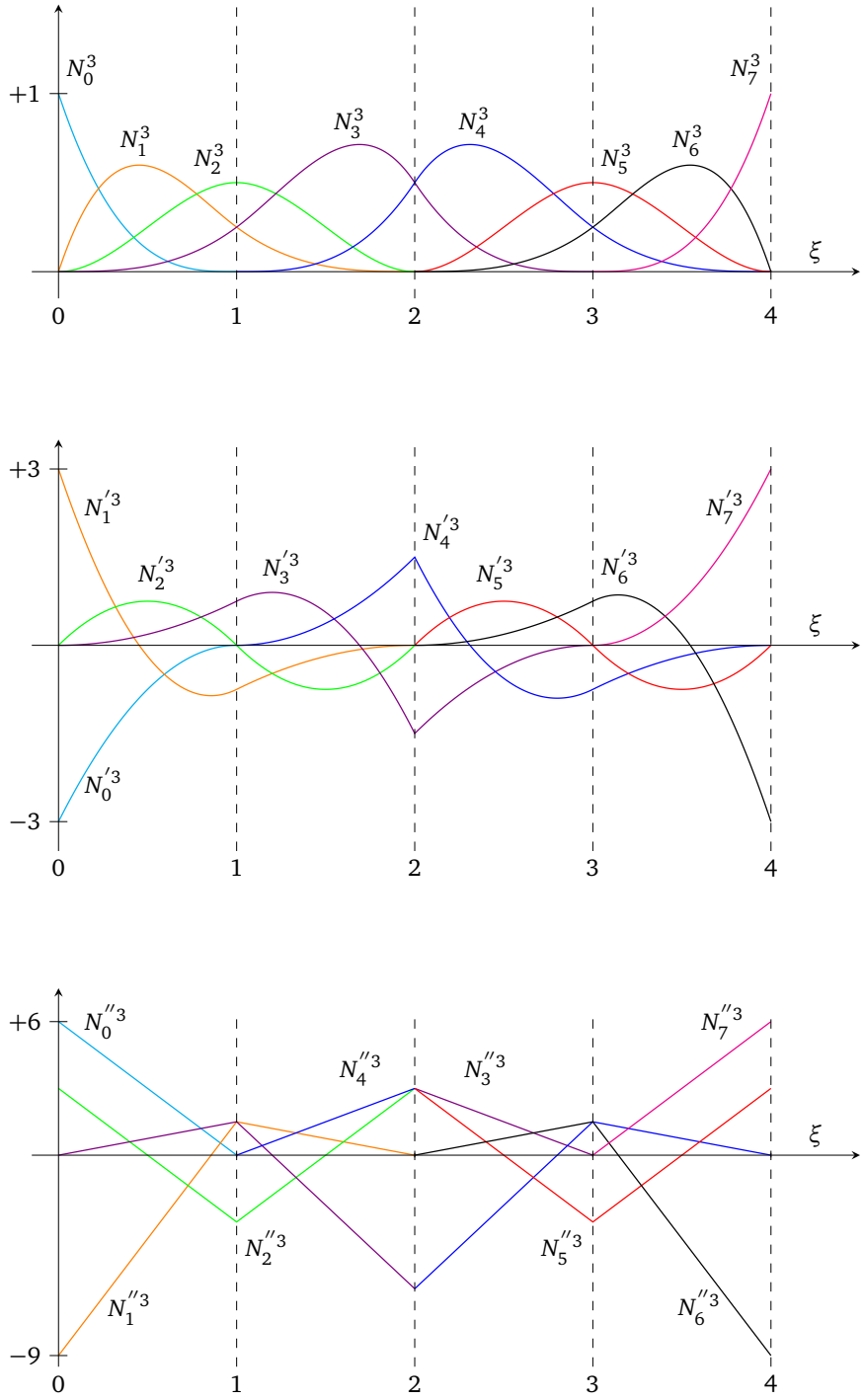

Figure 2.1: The cubic B-Spline functions $N_i^3(\xi)$ and its first and second derivatives on the domain $\bar{\Omega}^1$ defined by the knot vector $\boldsymbol{\Xi} = \{0,0,0,0,1,2,2,3,4,4,4,4\}$. The knots ξ_5 and ξ_6 are identical, reducing the continuity of the basis functions at $\xi = 2$ to C^1. This discontinuity can be observed in the jump of $N_3^{''3}$ and $N_4^{''3}$ at this position. The local support and the non-negativity of the basis functions can be readily observed, as well as the interpolative nature of the basis functions $N_i^3(\xi)$ at the boundaries of the domain.

2.1.2 Derivatives of B-Spline functions

The derivatives of the univariate B-Spline functions can be expressed in terms of basis functions of a lower degree. Piegl and Tiller (1997) give an expression that allows to compute the derivatives of B-Spline functions for higher-order derivatives:

$$N_i^p(\xi)^{(s)} = \frac{p!}{(p-s)!}\sum_{j=0}^{s}\alpha_{s,j}^{i,p}\,N_{i+j}^{p-s}(\xi), \tag{2.12}$$

where the factors $\alpha_{s,j}^{i,p}$ are computed as follows:

$$\alpha_{0,0}^{i,p}=1,\quad \alpha_{s,0}^{i,p}=\frac{\alpha_{s-1,0}^{i,p}}{\xi_{i+p+1-s}-\xi_i},\quad \alpha_{s,j}^{i,p}=\frac{\alpha_{s-1,j}^{i,p}-\alpha_{s-1,j-1}^{i,p}}{\xi_{i+p+1-s+j}-\xi_{i+j}},\quad \alpha_{s,s}^{i,p}=\frac{-\alpha_{s-1,s-1}^{i,p}}{\xi_{i+p+1}-\xi_{i+s}}.$$

2.1.3 Multivariate B-Spline functions

In order to describe volumetric bodies, trivariate basis functions are required. These can be determined from univariate B-Spline function by means of a *tensor product*:

$$N_{ijk}^{pqr}(\xi,\eta,\zeta) = N_{ijk}^{pqr}(\boldsymbol{\xi}) = N_i^p(\xi)N_j^q(\eta)N_k^r(\zeta), \tag{2.13}$$

where p,q,r denote the respective degree of the univariate basis functions. This allows to carry over the properties of the univariate B-Spline functions to their higher-dimensional extensions (Bartels et al., 1987). Consequently, the partial derivatives of the multivariate basis functions $N_{ijk}^{pqr}(\boldsymbol{\xi})$ are simply a product of the derivatives of its univariate constituents:

$$N_{ijk}^{pqr}(\boldsymbol{\xi})^{(\alpha,\beta,\gamma)} = \frac{\partial^{\alpha+\beta+\gamma}\left(N_i^p(\xi)N_j^q(\eta)N_k^r(\zeta)\right)}{\partial\xi^\alpha\,\partial\eta^\beta\,\partial\zeta^\gamma} = N_i^p(\xi)^{(\alpha)}N_j^q(\eta)^{(\beta)}N_k^r(\zeta)^{(\gamma)}. \tag{2.14}$$

The domain of the trivariate functions results from the Cartesian product of the three knot vectors $\boldsymbol{\Xi},\mathbf{H},\mathbf{Z}$ defining the respective univariate functions:

$$\bar{\Omega}^3 = \boldsymbol{\Xi}\times\mathbf{H}\times\mathbf{Z} = \left[\xi_a,\xi_b\right]\times\left[\eta_a,\eta_b\right]\times\left[\zeta_a,\zeta_b\right].$$

This yields hexahedral segments in $\bar{\Omega}^3$, as illustrated in Figure 2.2. Their image under the mapping (2.1) constitutes the volumetric Finite Element mesh used for Isogeometric analysis (Cottrell et al., 2007). *That is, the set of three open knot vectors not only uniquely determines the trivariate basis functions, but it also encodes the mesh used for the numerical simulation*.

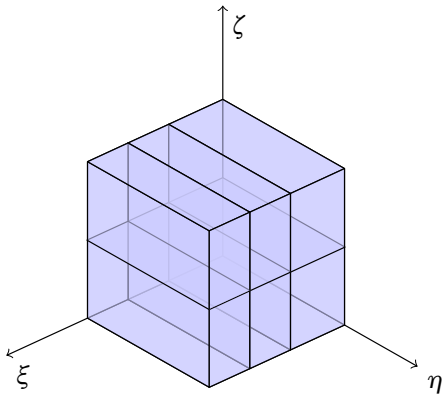

Figure 2.2: The parametric domain $\bar{\Omega}^3$ as defined by the knot vectors $\boldsymbol{\Xi} = \{\xi_a, \xi_1, \xi_2, \xi_b\}$, $\mathbf{H} = \{\eta_a, \eta_b\}$, and $\mathbf{Z} = \{\zeta_a, \zeta_1, \zeta_b\}$. The Cartesian product of the knot spans results in above hexahedral segments which constitute the volumetric mesh for the Isogeometric analysis. The knot vectors given here serve only illustrative purposes and do not fulfill the requirements for open knot vectors, i.e., the multiplicity of the exterior knots.

2.2 Control points and B-Spline solids

Given a set of trivariate basis functions, a B-Spline solid can be defined as the set of all points $\mathbf{x} = \left(x, y, z\right)^T$ resulting from the mapping

$$\mathbf{x}(\xi) = \sum_{i=0}^{n} \sum_{j=0}^{m} \sum_{k=0}^{l} N_{ijk}^{pqr}(\xi)\, \mathbf{P}_{ijk} \tag{2.15}$$

with $\mathbf{P}_{ijk} = \left(x_{ijk}, y_{ijk}, z_{ijk}\right)^T$ being the elements of a set $\mathscr{S}_{total}$ of *control points*

$$\mathscr{S}_{total} := \{\, \mathbf{P}_{ijk} \mid 0 \leq i \leq n \ \wedge \ 0 \leq j \leq m \ \wedge \ 0 \leq k \leq l \,\}. \tag{2.16}$$

As each basis function must be assigned to a point, $T = (n+1)(m+1)(l+1)$ control points are necessary to describe the mapping $\mathscr{N} : \bar{\Omega}^3 \to \Omega^3 \subset \mathbb{R}^3$. These points must be aligned—in a topological sense—in a regular grid with constant grid sizes throughout the structure. This allows to define a discrete index coordinate system; single control points can thus be addressed by their index tuple (i, j, k) that, in turn, corresponds to their alignment. Points are arranged in layers in the grid, corresponding to their k-index. Within each layer they are grouped in rows, denoted by their j-index, and within each row they are sorted by their i-index, as illustrated in Figure 2.3.

Constant grid sizes allow efficient implementations of tensor-product solids. The three-dimensional grids of control points as well as the trivariate basis functions can be stored in continuous, block-structured arrays. The resulting blockwise alignment of the geometry data in memory can be exploited by the algorithms for modeling and manipulation.

The set of control points fully determines size and shape of the resulting geometry whereas the basis functions affect the smoothness of the interpolation. Due to the tensor-product structure, different polynomial degrees as well as different numbers of control

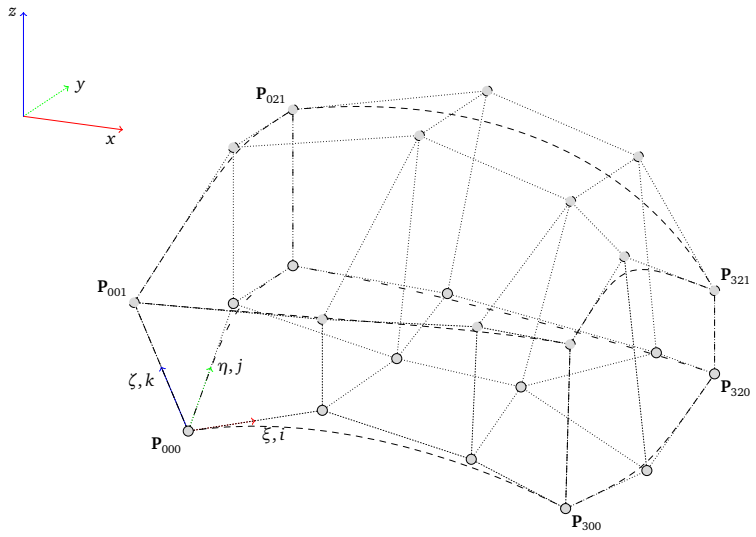

Figure 2.3: A grid of control points $\mathbf{P}_{ijk}$ necessary for describing a Spline-based solid body. The origin of the solid's parametric domain $\bar{\Omega}^3$ is mapped to the point $\mathbf{P}_{000}$ which acts as origin for the indexing scheme. By virtue of the geometric mapping the parametric coordinates (ξ, η, ζ) are overlaid with the grid coordinates (i, j, k). In above figure, lines of the same index follow the grid lines of the topology (shown in dotted lines), whereas the parametric coordinates ξ, η, ζ follow the curved boundaries of the resulting solid (shown in dashed lines).

points can be chosen for the individual grid directions. This provides a high degree of control not only for the representation of a geometric object, but also in the construction of the function spaces for the solution of partial differential equations through Isogeometric Analysis. Slender, thin-walled structures such as plates or shells can hence be modeled with an appropriate level of detail along different model axes.

2.3 NURBS solids

B-Spline solids already possess a high degree of flexibility for modeling volumetric objects. However, they lack the possibility to exactly describe *conic sections* including ellipses and circles. B-Splines can only represent approximations to such geometries, a factor which motivated the introduction of NURBS. In order to overcome this limitation, the points $\mathbf{P}_{ijk}$ are embedded in a four-dimensional projective space $\mathbb{P}^4$ with the additional *homogeneous coordinate* $h \neq 0$. The control points are represented in $\mathbb{P}^4$ as

$$\mathbf{P}^h_{ijk} = \left(h_{ijk}x_{ijk},\, h_{ijk}y_{ijk},\, h_{ijk}z_{ijk},\, h_{ijk}\right)^T, \tag{2.17}$$

where the superscript ^{h} denotes an entity in $\mathbb{P}^4$. The points' coordinates in 3-space are obtained by a *perspective projection* $\mathscr{P}$ through the origin $\mathscr{O}$ onto the hyperplane defined by $h = 1$. This corresponds to an element-wise division of the points' coordinates by the

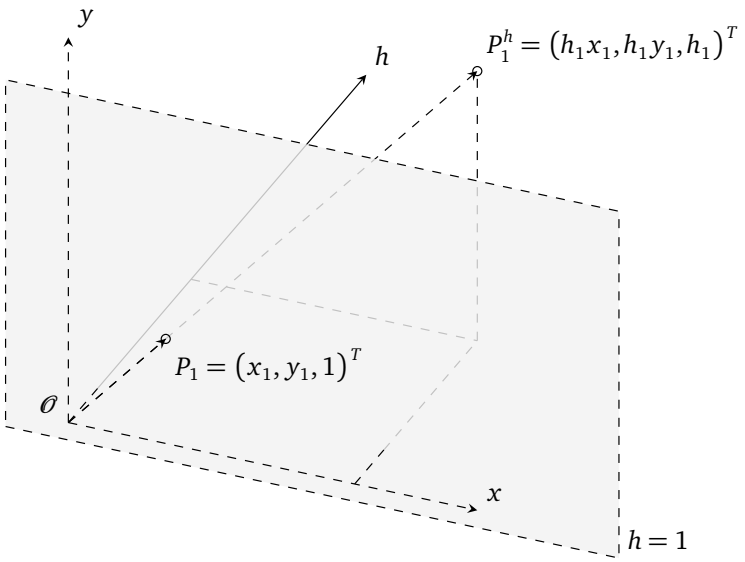

Figure 2.4: Embedding of the euclidean space $\mathbb{R}^2$ into a space $\mathbb{P}^3$ of projective coordinates. All points $\mathbf{P}_i = (x_i, y_i)^T$ are lying on the hyperplane defined by $h = 1$. They are obtained by a perspective projection of the respective points $\mathbf{P}_i^h = (h_i\mathbf{P}_i\,;\, h_i)^T$ through the origin $\mathcal{O}$ onto the hyperplane $h = 1$.

respective homogeneous coordinate (cf. Piegl and Tiller, 1997):

$$\mathbf{P}_{ijk} = \left(x_{ijk}, y_{ijk}, z_{ijk}\right)^T \equiv \left(x_{ijk}, y_{ijk}, z_{ijk}, 1\right)^T = \mathscr{P}\left\{\mathbf{P}_{ijk}^h\right\} = \frac{1}{h_{ijk}}\mathbf{P}_{ijk}^h. \tag{2.18}$$

The concept of this embedding is illustrated in Figure 2.4. A NURBS solid is accordingly defined as the set of all points $\mathbf{x}^h \in \mathbb{P}^4$ that result from the mapping

$$\mathbf{x}^h(\xi) = \sum_{i=0}^{n}\sum_{j=0}^{m}\sum_{k=0}^{l} N_{ijk}^{pqr}\left(\xi\right)\mathbf{P}_{ijk}^h = \sum_{i=0}^{n}\sum_{j=0}^{m}\sum_{k=0}^{l} h_{ijk}N_{ijk}^{pqr}\left(\xi\right)\mathbf{P}_{ijk}. \tag{2.19}$$

That is, NURBS solids are simply B-Spline solids within $\mathbb{P}^4$. Application of the perspective division by the homogeneous coordinate,

$$h(\xi) = \sum_{a=0}^{n}\sum_{b=0}^{m}\sum_{c=0}^{l} h_{abc}N_{abc}^{pqr}\left(\xi\right), \tag{2.20}$$

leads to the expression for a NURBS solid within $\mathbb{R}^3$:

$$\mathbf{x}\left(\xi\right) = \frac{\sum_{i=0}^{n}\sum_{j=0}^{m}\sum_{k=0}^{l} w_{ijk}N_{ijk}^{pqr}\left(\xi\right)\mathbf{P}_{ijk}}{h(\xi)} = \sum_{i=0}^{n}\sum_{j=0}^{m}\sum_{k=0}^{l} R_{ijk}^{pqr}\left(\xi\right)\mathbf{P}_{ijk}. \tag{2.21}$$

The projection from $\mathbb{P}^4$ into $\mathbb{R}^3$ effectively transforms the piecewise polynomial basis functions N_{ijk}^{pqr} into piecewise *rational polynomial* functions R_{ijk}^{pqr}. The homogeneous coordinates h_{ijk} are usually interpreted as weights of the respective control points, providing additional degrees of freedom for shape design. If all h_{ijk} are identical the NURBS functions degenerate to polynomial B-Spline functions.

2.3.1 Derivatives of NURBS functions

The increased flexibility of NURBS with respect to shape design comes at the price of higher efforts for determining the derivatives of the basis functions, required, for instance, for describing kinematic relationships in the simulation of the mechanical behavior of continua in Chapter 5. This section puts the focus on the derivatives with respect to a single variable, ignoring mixed partial derivatives. Following (2.21), a trivariate NURBS basis function $R_{ijk}(\boldsymbol{\xi})$ is given by

$$R_{ijk}(\boldsymbol{\xi}) = \frac{h_{ijk}\, N_{ijk}(\boldsymbol{\xi})}{h(\boldsymbol{\xi})} = \frac{A_{ijk}(\boldsymbol{\xi})}{h(\boldsymbol{\xi})}, \tag{2.22}$$

where the polynomial degrees p, q, r are tacitly assumed. The superscripts ^{pqr} are henceforth dropped from trivariate expressions for the sake of clarity. In addition, we collapse the triple index (ijk) into a single index

$$I = i + j\,(n+1) + k\,(n+1)\,(m+1)\,. \tag{2.23}$$

Following the quotient rule, the first derivatives of the basis functions with respect to ξ are then given as

$$R_I(\boldsymbol{\xi})_{,\xi} = \frac{A_I(\boldsymbol{\xi})_{,\xi}\, h(\boldsymbol{\xi}) - A_I(\boldsymbol{\xi})\, h(\boldsymbol{\xi})_{,\xi}}{h(\boldsymbol{\xi})^2} = \frac{A_I(\boldsymbol{\xi})_{,\xi} - R_I(\boldsymbol{\xi})\, h(\boldsymbol{\xi})_{,\xi}}{h(\boldsymbol{\xi})}, \tag{2.24}$$

where $_{,\xi}$ denotes the partial derivative with respect to ξ. The partial derivatives of both the weighted numerator $A_I(\boldsymbol{\xi})$ and the denominator $h(\boldsymbol{\xi})$ can be readily computed by (2.14) since the homogeneous coordinates h_I are constant with respect to the parametric coordinates $\boldsymbol{\xi}$. Therefore

$$A_I(\boldsymbol{\xi})_{,\xi} = h_I\, N_I(\boldsymbol{\xi})_{,\xi} \tag{2.25}$$

and

$$h\,(\boldsymbol{\xi})_{,\xi} = \sum_{I=0}^{T-1} h_I N_{I,\xi}\,(\boldsymbol{\xi}) = \sum_{I=0}^{T-1} A_{I,\xi}\,(\boldsymbol{\xi})\,, \tag{2.26}$$

where T is the total number of points in the control grid, defined in Section 2.2. Analogous expressions can be set up for the partial derivatives with respect to the parametric variables η and ζ.

2.3.2 Interpolation of the boundaries

The trivariate basis functions' properties result in characteristic features of B-Spline solids, a thorough overview of which is given in Ma et al. (2001). A property particularly relevant for modeling purposes is the *interpolation of the boundaries*. Univariate B-Spline functions based on open knot vectors become interpolatory at the boundaries of their domain, as seen in Equation (2.11). This property carries over to multivariate B-Splines. As a result, the "corners" of the parametric domain $\bar{\Omega}^3$, that is, the tuples

$$\{\,(\xi,\eta,\zeta) \mid \xi \in \{\xi_a,\xi_b\} \wedge \eta \in \{\eta_a,\eta_b\} \wedge \zeta \in \{\zeta_a,\zeta_b\}\,\}$$

are mapped to the respective "corners" of the control grid. For example,

$$\mathbf{x}(\xi_a,\eta_a,\zeta_a) = \mathbf{P}_{000}, \qquad \mathbf{x}(\xi_a,\eta_b,\zeta_b) = \mathbf{P}_{0ml}.$$

This can also be seen in Figure 2.3. The boundary surfaces of a solid can be described by the isosurfaces

$$\begin{array}{lll} \mathbf{x}(\xi=\xi_a,\eta,\zeta) & \mathbf{x}(\xi,\eta=\eta_a,\zeta) & \mathbf{x}(\xi,\eta,\zeta=\zeta_a) \\ \mathbf{x}(\xi=\xi_b,\eta,\zeta) & \mathbf{x}(\xi,\eta=\eta_b,\zeta) & \mathbf{x}(\xi,\eta,\zeta=\zeta_b), \end{array} \tag{2.27}$$

that is, by the images of the boundary surfaces of $\bar{\Omega}^3$ under the mapping (2.1).[2] Interpolation of the boundaries by the univariate basis functions has the effect that the isosurfaces defined by (2.27) are described purely in terms of the sets of control points

$$\begin{aligned}
\mathscr{S}_{\xi_a} &:= \{\,\mathbf{P}_{ijk} \mid i=0 \wedge 0\le j\le m \wedge 0\le k\le l\,\} \quad (\text{for } \xi=\xi_a),\\
\mathscr{S}_{\xi_b} &:= \{\,\mathbf{P}_{ijk} \mid i=n \wedge 0\le j\le m \wedge 0\le k\le l\,\} \quad (\text{for } \xi=\xi_b),\\
\mathscr{S}_{\eta_a} &:= \{\,\mathbf{P}_{ijk} \mid 0\le i\le n \wedge j=0 \wedge 0\le k\le l\,\} \quad (\text{for } \eta=\eta_a),\\
\mathscr{S}_{\eta_b} &:= \{\,\mathbf{P}_{ijk} \mid 0\le i\le n \wedge j=m \wedge 0\le k\le l\,\} \quad (\text{for } \eta=\eta_a),\\
\mathscr{S}_{\zeta_a} &:= \{\,\mathbf{P}_{ijk} \mid 0\le i\le n \wedge 0\le j\le m \wedge k=0\,\} \quad (\text{for } \zeta=\zeta_a),\\
\mathscr{S}_{\zeta_b} &:= \{\,\mathbf{P}_{ijk} \mid 0\le i\le n \wedge 0\le j\le m \wedge k=l\,\} \quad (\text{for } \zeta=\zeta_b).
\end{aligned} \tag{2.28}$$

That is, only the respective outermost control points contribute to the representation of a solid's apparent shape. To put it differently: NURBS solids are bounded by NURBS surfaces which in turn are bounded by NURBS curves. Using the point sets defined in (2.28), the set of control points $\mathscr{P}_{total}$ can be decomposed into disjoint subsets $\mathscr{P}_b, \mathscr{P}_i$ containing the control points lying on the boundary of the solid, respectively within its interior:

$$\mathscr{S}_b = \mathscr{S}_{\xi_a} \cup \mathscr{S}_{\xi_b} \cup \mathscr{S}_{\eta_a} \cup \mathscr{S}_{\eta_b} \cup \mathscr{S}_{\zeta_a} \cup \mathscr{S}_{\zeta_b}, \tag{2.29a}$$

$$\mathscr{S}_i = \mathscr{S}_{total} \setminus \mathscr{S}_b. \tag{2.29b}$$

[2] In addition to the surfaces defined by (2.27), implicit surfaces can result from shape degeneracies such as self-intersections (Joy and Duchaineau, 1999). Such cases are disregarded in this thesis, though.

3 Procedural modeling of NURBS

NURBS geometries can be described by a compact set of data, allowing smooth object representations with a fraction of the data required for polygon or voxel models (Martin and Cohen, 2001). The fundamental component of NURBS-based geometries is the grid of control points carrying the majority of geometric information. Whereas the knot vectors "merely" describe the partitioned parametric domain of the basis functions, it is the set of control points that determines the position, the size, and the aspect ratios of the "isogeometric mesh". Unfortunately, solid NURBS models have to be defined manually, which is tedious and error-prone. This is worsened only by the fact that volumetric NURBS models require far more information than curves and surfaces.

A major factor is here the set of interior control points defined in Eq. (2.29b). From the standpoint of boundary-based object representations, they do not carry shape information but fulfill only topological requirements, thus incurring an overhead of data. They have an important function, though: these points affect the mapping of the volumetric segments of $\bar{\Omega}^3$ into 3-space. That is, they determine the distortion of the solid's interior and hence its usability for numerical simulations. In order to achieve regular, undistorted meshes, the interior control points' alignment should consider not only direct neighbors but also the overall structure of the shape. This forbids a (semi)-arbitrary placement of these points.

The lack of appropriate modeling tools for volumetric NURBS poses a serious problem in light of model adaptions. NURBS data structures are but unstructured lists of model *primitives*. Such models can be efficiently parsed and are suitable for numerical processing. However, they do not represent the rules and dependencies inherent to all kinds of shapes (Havemann, 2005). The design intent, namely the rationale behind a given shape, can hence not be appropriately described (Shah and Mäntylä, 1995). Furthermore, there is no indication as to which model entities have to be modified in order to achieve a desired change in an object's shape. Therefore, a more abstract treatment is necessary which allows to reflect the structure of solid NURBS models, thus facilitating their modification in case of design changes.

3.1 Modeling approaches for NURBS solids

A number of modeling approaches for NURBS solids has been proposed in the literature. Each method deals with specific classes of shapes, and most of these concepts aim at emulating the functionality of CAD applications. They can be categorized as follows:

Surface expansion Some of the most basic algorithms for NURBS geometries create surfaces from single or multiple boundary curves. Basic operations such as *sweeping*, *skinning*, *extrusion*, or *ruling* can be found in virtually every CAD package. Algorithmic details for these methods are described in (Piegl and Tiller, 1997; Rogers,

2001). The tensor-product structure of the B-Spline basis functions suggests an extension of these algorithms for the creation of solid objects.

Aigner et al. (2009) describe a variational framework for generating swept volumetric parametrizations. It allows to consider the influence of different guiding curves on the resulting shape as well as controlling different properties such as orthogonality or regularity. Martin and Cohen (2001) deal with the challenges of assigning heterogeneous volumetric attributes to NURBS solids. Each NURBS volume is assigned additional attribute fields that are independent of the control grid but that share the parametric domain with the underlying solid. This allows to maintain different resolutions for both the shape and the properties of an object. Consistent transfer of properties can then be ensured by applying the geometric modeling operations to both the attributes and the shape. Ma et al. (2001) employ NURBS volumes for rapid-prototyping applications. They construct complex objects from solid NURBS objects that are created by the aforementioned operations as well as by an additional operation which they denote as "shrinking". A volume is thereby generated by ruling (i.e. interpolating) a NURBS surface with another surface being degenerated into a point or into a line segment. Assembly of the NURBS components into the final model is achieved in a two-step process. To that end, individual NURBS volumes are converted into voxel representations which are subsequently combined by Boolean operations. Zhou and Lu (2005) apply "shrinking" for the modeling and biomechanical simulation of muscle tissue. The boundary surfaces of a muscle strand are then generated by skinning the muscle's cross-sections from sectioned medical image data. These surfaces are subsequently shrunk onto the median axis of the strand, yielding solid NURBS geometries.

Constructive modeling An alternative to boundary-based representations of solid bodies is given by the concept of *Constructive Solid Geometry* (CSG). An object is thereby constructed from parametrized shape primitives such as cylinders or hexahedra by means of *regularized Boolean operations* (cf. Shapiro, 2002). The major challenge for NURBS in this approach is either to ensure the correct alignment of the different meshes of the shape primitives or to provide solution strategies that cope with nonconforming and overlapping meshes. This idea has been implemented by Natekar et al. (2004) who describe the concept of *Constructive Solid Analysis*. They employ NURBS for the description of both the shape primitives and the field quantities. By employing a *Meshless Finite Element* formulation they can construct a boundary-value problem from the solution fields defined on the shape primitives.

Transfinite interpolation A common class of free-form surfaces, the so-called *Gordon-Coons patches*, results from the bilinear interpolation of a set of boundary curves. Shih et al. (2005) extend this concept for the interpolation of boundary surfaces that allows them to create solid NURBS bodies with appropriate interior control points. The overhead of the grid specification can thus be avoided completely.

Constrained modeling An important issue in computer-aided geometric design is the consideration of shape constraints. A corresponding question in the context of free-

form surfaces might be how a given surface has to be modified in order to pass through a specified point. Hu et al. (2001) turn this into a constrained optimization problem. This allows them to regard positional constraints as well constraints on the curvature and the normal of a given surface. A similar course of action is taken by Xu et al. (2010, 2011a,b). In their objective function they consider both orthogonality and regularity of the mesh; constraint conditions then ensure non-overlapping meshes.

Extraction of parametrizations A common problem in geometric modeling is to find a parametrization for a given set of discrete data. Eck and Hoppe (1996) describe a method for reconstructing tensor product spline surfaces from scanned 3D point sets. Surfaces of arbitrary topology are thereby decomposed into a network of connected patches. To that end, the given point samples are used to set up an initial polygon mesh which serves as input for several (re-)parametrization steps. Once a suitable parametrization has been found, the free-form surface is reconstructed by a fitting algorithm and refined as necessary. Martin et al. (2009) present a framework for modeling B-Spline volumes from a given exterior surface and additional interior boundaries. These boundaries are supplied as triangle meshes describing the outer surface and possible interior layers.

The quality of the input data is probably the most important factor in these approaches as geometry originating from reverse engineering is sensitive towards measurement noise. Data originating from medical imaging similarly depends on chosen imaging thresholds. This can lead to oscillations in the resulting geometries (Martin et al., 2009). These methods can furthermore produce excessive model data, making model simplification highly necessary. Wang and Zhang (2010) describe a method for simplification and fairing of NURBS geometries that is based on a wavelet transform of the NURBS geometry. High-frequency noise can thus be removed from the model data while the geometric properties, in particular the continuity, is retained.

Template modeling Many problem settings in engineering exhibit a high geometric similarity to previous tasks. In addition, a certain regularity can be observed for man-made technical objects (Mehra et al., 2009). It results either from functional considerations and technical constraints (Mitra and Pauly, 2008) or from aesthetic principles such as proportion and symmetry (Havemann, 2005). Consequently, components of technical systems can often be characterized by a handful of parameters. This allows their compilation in catalogs or codes. The compact shape representation of NURBS favors such representations. Fundamental patterns of control points and basis functions can hence be defined as *templates* encoding basic shapes such as circular arcs. They may be combined in order to obtain more complex shapes, as shown by Cottrell et al. (2009): here, a pipe segment is modeled from the combination of several simple templates describing circular arcs and annular segments. Zhang et al. (2007) describe a template-based approach for the analysis of cardiovascular blood flow and its fluid-structure interaction (FSI) with the arterial walls. Templates are employed here to capture specific branching configurations of the

blood vessels. Models of patient-specific vasculature are constructed by fitting the template models to medical image data. Bazilevs et al. (2011) developed a set of templates for the modeling of Horizontal-Axis Wind Turbine (HAWT) rotors. The rotor blades for off-shore wind turbines are created by skinning cross-sectional curves given at regular intervals along the blade's length. Starting from the blade surface, the *negative space* around a rotor is constructed by "imprinting" the blade's control point patterns into a sufficiently refined portion of a volumetric domain segment.

Templates are an invaluable aid in the setup of highly complex models. They have the potential to drastically reduce the efforts for creating model variants in case they provide a parametrized representation of a geometric model. However, they still have to be defined manually which can be a major hurdle for intricate models such as three-dimensional fluid domains. At their core they are but predefined lists of model primitives and they suffer, at least internally, from inefficient low-level modeling (Shah and Mäntylä, 1995). In order to overcome this limitation a *procedural* approach is chosen by which the low-level data structures are described in terms of more abstract modeling operations. The following section gives an introduction to this modeling concept; its application to (solid) NURBS modeling is then detailed in Section 3.3.

3.2 Procedural modeling in computer graphics

Procedural modeling is a concept originating in computer graphics whereby "*code segments or algorithms . . . specify some characteristic of a computer-generated model or effect.*" (Ebert et al., 2003). That is, a complex model is not described in terms of its raw data. Instead, a deterministic algorithm generates these data from a small set of model parameters. The determinism of the employed algorithms is of utmost importance as it ensures that identical input data yield identical results. One can then create the final model from the recorded input parameters—the "data seed"—whenever it is necessary. This feature, denoted in the literature as *data amplification*, allows tremendous savings in model size. It is invaluable for systems with limited memory. Accordingly, one of the first applications of procedural synthesis can be found in early video games such as *Elite* (Acornsoft, 1984) where an evaluation routine turned sets of parameters into the game data. Game "content" such as whole star systems or planets could hence be regarded despite the strong limitations imposed by the hardware. Another field of application for procedural generation lies in the so-called *Demo scene* (cf. Reunanen, 2010; Shor and Eyal, 2004). This branch of the *hacker subculture* frequently employs procedural content generation techniques in the creation of "demos". These small programs must be considered to be pieces of art, both with respect to their content and their realization. Upon execution, a computer graphics animation of several minutes length, including music and sound effects, is computed in real-time on the host computer. The sizes of these executables range mostly between 4 kB and 64 kB, which can only be achieved by using compression techniques and procedural generation.

The fields of computer graphics, animation and geometric modeling have experienced a steady increase in model complexity. Main driving force of this development is the en-

tertainment industry with its objective of photo-realistic imagery, either in the form of video games or as visual effects in films. This has lead to the availability of powerful dedicated hardware such as GPUs[1] (Havemann, 2005). Once being restricted to computing visual effects in games, they are now increasingly used in the simulation of physical phenomena, providing scientists and engineers with powerful yet affordable tools. Downside of this development are steadily rising development costs in the entertainment industry: realism is linked with the complexity of an image, both in terms of the number and the diversity of the image's elements (Ebert et al., 2003)—but each visual effect, each animation, and each texture must not only be modeled by an artist but also be integrated into the overall product. Model creation has therefore become the bottleneck in 3D computer graphics (Aliaga et al., 2007; Bokeloh et al., 2010; Kelly and McCabe, 2006)—which closely resembles the model generation in the simulation sciences with its large fraction of manual contribution.

Procedural synthesis offers an opportunity to manage the increased complexity of models and effects. It allows to "*generate plausible details of a model without much or any user interaction*" (Aliaga et al., 2007). It is hence possible to deal with models whose manual setup is either impractical or impossible. This can be illustrated with the simulation of urban systems, whose application is not restricted to video games and virtual reality, but which are also employed for navigation, emergency response training, studies of urban development, or traffic analyses (Aschwanden et al., 2009; Vanegas et al., 2009). Cities are systems of intimidating visual and functional complexity. They are the result of a development spanning several hundred years under the influence of various socio-political factors (Kelly and McCabe, 2006). The sheer amount of data necessary to describe them clearly forbids the manual setup of a model. Procedural techniques, however, render this task feasible, as shown by Parish and Müller (2001). They describe a system for modeling a complete city using a set of statistical and geographical input data. These are provided as 2D image maps and describe, for instance, population density, elevation data, or land use. Their system comprises a pipeline in which each generation step uses the output of prior steps as input data. Basis for the generation of the street layouts and the building geometry are extended *L-systems*, that is, systems for handling of symbol strings which have initially been devised for algorithmic modeling of plants (cf. Prusinkiewicz and Lindenmayer, 1990). Using information on the population density, the system connects highly populated centers using a network of highways. These centers are "meshed" with a network of streets according to a superimposed street pattern. Different characteristic patterns are described in terms of road generation rules leading, for instance, to orthogonal or radial grids. Areas in between the roads are subdivided into convex, regularly shaped allotments. These serve, in turn, as input for a second L-system dealing with the generation of the buildings' geometry. This second system operates on simple geometric shapes and considers not only the ground area of the current allotment but also additional data such as land use, population density, or zoning rules.

Systems such as the described can be nested even further, allowing not only to create building models (Müller et al., 2006) or intricate facades (Haegler et al., 2010), but also

[1] Graphics Processing Unit, i.e. dedicated processors for computer graphics operations.

to fill in large amounts of model details. Cities can hence be modeled in tremendous detail—down to a level that even considers furniture (Germer and Schwarz, 2009). This illustrates two properties of procedural systems: the generation of data from data and the feedback of the results of one generation step to further "data generators".

Outside the field of computer graphics, procedural generation has found only scarce attention. A noteworthy exception is (Häfner et al., 2006) where this concept has been applied for modeling the mesoscale structure of concrete. Thereby, aggregates of varying size are aligned within the confines of a given specimen geometry, while contact checks prevent overlaps of individual particles. Models with varying size and shape distributions of the aggregates can thus be readily generated and are subsequently "fed" into a Finite Element mesh generator.

Another, more recent application can be found in the work of Whiting et al. (2009) who combine procedural modeling with physical constraints. Their object of research is the procedural generation of physically sound masonry structures. To that end they construct an inverse statics problem: given a set of constraints and a building's topology, an appropriate shape is derived by means of a nonlinear optimization procedure. This highlights the possible connections of procedural modeling techniques with other areas of research, such as mathematical optimization (cf. Togelius et al., 2010), stressing the need for an abstract model representation.

Procedural systems provide an abstraction layer from the raw data structures of a computer model. Instead of designing a single instance, a whole class of objects is described in terms of parametrized construction steps and interpretable parameters (Berndt et al., 2004). Each operation is basically a rule of how the current state of the model is transformed into the next state. This is in remarkable agreement with the perception of shape, as put forth by Leyton (2001). He argues that the human perceptual system is structured as a sequence of transformations applied to groups of objects. A simple square is hence observed as the four-fold translation and rotation of a line segment which, in turn, is perceived as a swept point. He further formalizes this as a sequence of *unfolding groups* that expand a complex shape from its structural core, its *alignment kernel* (Leyton, 2001). This kernel describes the final object in its most compact form; it is successively unfolded by a sequence of operations. Minimization of the "data seed" then emphasizes the structure of the model, which, firstly, provides a basis for understanding and communicating shape (Havemann, 2005) and which, secondly, makes it possible to capture the "*essence of a model*" (Ebert et al., 2003).

The fundamental element in formulating procedural models is the inference of the rules that determine the desired shape, as well as their translation into an algorithm for shape generation (Kelly and McCabe, 2006). This includes the definition of the model entities, their connectivity, and the constraints between the components and parameters (Shah and Mäntylä, 1995). The evaluation algorithms then stepwise resolve the given constraints and assign values to model parameters. Model variations are achieved by adapting the input parameters, followed by re-evaluation of the algorithm. Consequently, the setup of design variants becomes trivial (Berndt et al., 2004). This is invaluable for engineering practice with its frequent design changes and the geometric similarity of problem settings.

Procedural systems can be implemented relatively straightforward. They express only a small range of model variations, though, and the fixed model evaluation sequence complicates subsequent addition of constraints (Shah and Mäntylä, 1995). Expression of a model in terms of modeling commands puts the focus on the selection of appropriate model parameters and operations. This can prove to be a difficult task as it may result in discontinuous data generation (Aliaga et al., 2007). That is, small variations in the model parameters can, depending on the evaluation algorithm, lead to large differences in the output.

3.3 Operator-based modeling of NURBS geometries

NURBS geometries are commonly modeled in an interactive manner within a CAD application. A simple initial object, for instance a flat surface or a cube, is the starting point from which complex models are created. Therefor the initial geometry is deformed by successive transformation of groups of control points. Fine-grained shape control is provided by mesh refinement operations that introduce new rows of control points into the object's data structures while maintaining its shape. The newly created points are used for further manipulations of the control points. The modeling process is hence but a sequence of interleaved point transformations and mesh refinements where each modeling operation can be conceived as a transformation of the current model into a new, possibly more complex state. This corresponds to the view of (Leyton, 2001). That is, the initial object acts as the "alignment kernel" which is successively transformed into the final shape.

3.3.1 Atomic NURBS modeling operations

In order to implement procedural modeling for solid NURBS geometries, only a surprisingly small set of modeling operations has to be provided. These operations comprise the "surface expansion" operations such as *sweeping*, *skinning*, *revolution*, and *extrusion* as mentioned in Section 3.1. In addition, the following methods are necessary:

Transformation of control points The coordinates of a solid's control points can be modified by means of *affine transformations* such as translation, rotation, and scaling. They are formulated in the space $\mathbb{P}^4$ of homogeneous coordinates which allows their uniform representation as (4×4) matrices. These matrices can be easily combined for the description of complex, chained coordinate transformations (Details can be found in Akenine-Möller et al., 2008; Rogers and Adams, 1990). NURBS-based geometry has the property of "affine covariance" (Hughes et al., 2005). That is, the uniform transformation of all control points leads to rigid-body transformations. By restricting the transformation to subsets of control points, however, the object under consideration is deformed.

Selection operations Selection of such subsets of points can be done either on a patch-wise basis by means of the control points' index coordinates (i, j, k), or in a global manner by using their Cartesian coordinates. The results can be stored in point sets,

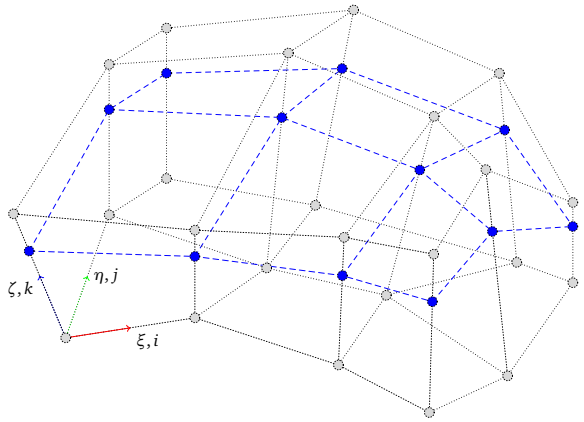

Figure 3.1: The effect of a knot insertion algorithm on the control point grid $\mathbf{P}_{ijk}$ of the solid shown in Figure 2.3. By inserting a new knot value into the knot vector $\mathbf{Z}$ the control grid $\mathbf{P}_{ijk}$ is augmented by a complete subgrid of new points (shown in blue).

allowing the application of set operations such as *union* or *intersection*. Selection of point ranges or combination of different point sets can thus be trivially implemented.

Point sets provide a mechanism for labeling of nodes: a fundamental problem of parametric and procedural modeling schemes is that entities of the final model are only implicitly described (Hoffmann and Joan-Arinyo, 2002). Model components are only instantiated during model evaluation. Application of boundary conditions or forces, however, requires that the affected entities are explicitly known. This problem can be circumvented by node labels where significant control points are assigned model-specific labels, denoting, for example, that they represent a certain model support.

Mesh refinement A major advantage of NURBS over "monolithic" polynomial representations is their high flexibility. Other than polynomial schemes, the order of the basis functions constitutes only a lower bound on the number of control points that can be regarded. Additional degrees for shape design can be readily added to an existing model without affecting the overall geometry. This is achieved by refinement operations such as *knot insertion/refinement* or *degree elevation* which are probably the most fundamental algorithms for the manipulation of Spline-based geometries (Piegl and Tiller, 1997).

During knot refinement, the one-dimensional parameter space described by a knot vector is subdivided along new knot values. The number of corresponding univariate basis functions increases by one for each new knot value. In order to maintain the topology of the geometry, new rows of control points have to be introduced into the grid of points. These are the result of linear combinations of existing points. The algorithm is described in (Piegl and Tiller, 1997) for uni- and bivariate NURBS objects but it can be easily extended to solids as a consequence of the regular grid topology of the control points.

Knot refinement neither changes the geometry of a NURBS object nor the properties of the geometric mapping (2.1). Only the object's basis functions—and hence its internal representation—are modified (Piegl and Tiller, 1997). The recombination of existing control points does not introduce geometric information into the model. However, the newly created points are consistently inserted in between existing points. This makes knot refinement an invaluable part of an operator-based model representation. Instead of storing a fully refined model, it suffices to maintain a list of knots to be inserted into its coarse representation. Depending on the size of the control grid, tremendous savings in model size can be achieved. This is illustrated in Figure 3.1 where a single knot insertion leads to the creation of twelve control points. These efficiency gains depend only on the respective grid sizes and they become increasingly economic for larger grids.

The second refinement operation, degree elevation, increases the order of one set of univariate basis functions. To that end, the NURBS object is split by successive knot refinement along the specified parametric direction into a series of *Bézier segments*. The continuity of the basis functions is thereby reduced to C^0 at each knot value in the corresponding knot vector. The segments' polynomial degree is then raised individually whereby new control points are introduced. The modified segments are eventually joined and excess knot values removed from the affected knot vector. Similar to knot refinement only the internal representation of the geometry is modified.

Component extraction NURBS geometries interpolate their boundaries, as shown in Section 2.3.2. The control points defining a boundary surface can hence be readily extracted from the solid's data structure along with the corresponding knot vectors. These surfaces can serve as input for further modeling operations such as extrusion, sweeping, or revolution. In addition to point sets, this facilitates the definition of boundary conditions for the numerical model as the extracted objects can be used for the prescription of primary unknowns or for the integration of flux quantities.

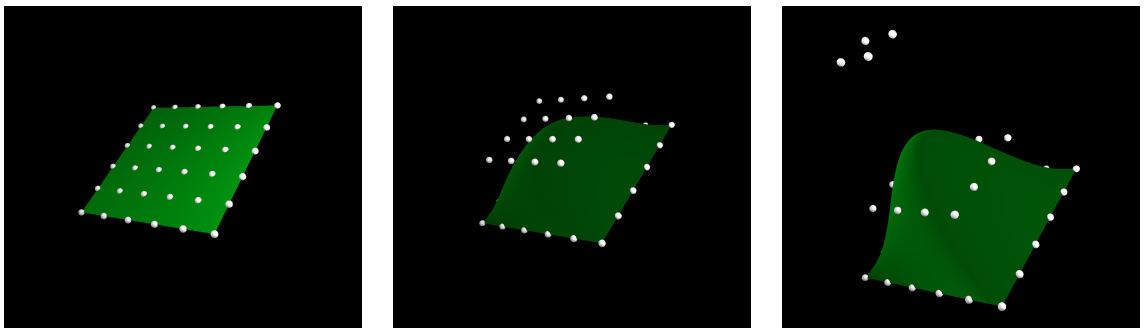

Figure 3.2: Application of a sequence of modeling operators to a planar surface. The initial surface on the left-hand side is the result of a *Surface* template defining appropriate knot vectors and control points for user-defined grid sizes. The planar surface is deformed using a combination of selection and translation operations. The first selection comprises the points $\mathbf{P}_{ij}$ with indices i, j in the range $[1, 4]$. This set of points is then translated uniformly, giving the shape in the center panel. This selection is narrowed down by reselection of the points with indices i, j in the range $[2, 3]$, which are translated by an additional offset and yield the final shape shown in the panel on the right-hand side.

3.3.2 Operator-based modeling of NURBS geometries

Following the concept of procedural modeling, the aforementioned modeling commands are simply stored in a list, where each command refers to the set of global parameters. This allows, first and foremost, to define complex modeling operations from the combination of simpler commands: finite sets of modeling commands cannot fulfill all requirements that can come up for model generation (Berndt et al., 2004). User-defined "macros" provide here the basis to define complex commands and to overcome the limitations of predefined modeling operations.

The list of modeling commands can be conceived as an algorithm to turn a given initial object into the final geometry. Families of fundamentally similar shapes can thus be trivially generated—where the resulting objects nonetheless share an identical model structure. This puts the focus on a handful of semantically strong parameters; it provides an abstraction from the model's raw data structures. The generation of model variants accordingly leads back to suitably adapting the model's parameters. This property is demonstrated in Figure 3.2 where a planar NURBS surface is deformed by combined selections and point transformations. Model variants of the shown surface patch can be realized by using different selection criteria and/or translation offsets.

The code used for the creation of the shape that is shown in Figure 3.2 is given in Listings 3.1 and 3.2. One can easily recognize the *database amplification*, that is, the significant reduction of the model's size by means of the operator representation: the grid of 36 control points and their topology, as well as the knot vectors describing the basis functions, can be represented by essentially nine lines of code. This does not only allow to recognize the model's structure, but it also simplifies the exchange of model data. Instead of transmitting the complete set of raw model data, it suffices to transmit the smaller set of model parameters. It is clear that this requires the availability of the modeling operations

```
// create a surface described by a (6 by 6)-grid
Surface surfaceTemplate(6, 6);

// instantiate an actual NURBS patch from the template
Patch *surface = surfaceTemplate.createGeometry();

// select all points with index i = [1,4]
surface->select(NEW, RANGE, I, 1, 4);

// restrict the selection to the points with index j = [1,4]
surface->select(RESELECT, RANGE, J, 1, 4);

// uniform translation of the currently selected points
// by "offset1"
surface->transform(Translation(Z_AXIS, offset1));

// further restriction of the selection and translation
// by "offset2"
surface->select(RESELECT, RANGE, I, 2, 3);
surface->select(RESELECT, RANGE, J, 2, 3);
surface->transform(Translation(Z_AXIS, offset2));

// reset selection state
surface->select(NEW, ALL);
```

Listing 3.1: The code for describing the bell-shaped surface patch in Figure 3.2.

for both parties of a model exchange. An implementation of the operator-based modeling concept requires, as shown before, only a small set of commands. They can be compiled into a compact kernel for modeling of NURBS solids that is made available to all sides. Model transfer is then reduced to the exchange of model parameters along with custom modeling procedures.

3.4 Templates for NURBS geometries

Operator-based model representations excel at describing objects of technical origin which exhibit a certain geometric structure. There exists, however, a broad range of objects whose geometry cannot be expressed in terms of modeling operations. Their shape is either subject to different objectives or its design intent has been lost during model and data transformations. The former case can be found with shapes being the result of functional requirements, for example airfoils that are designed to meet aerodynamic criteria. The latter case is at hand with models resulting from reverse engineering or from image processing. Albeit the geometry might be captured with high fidelity, the semantic of a geometry, its structure, cannot be recovered by scans and data transformations (Havemann,

```
// "size1" and "size2" are the respective grid sizes for the
// indices i,j
unsigned total = size1 * size2;
Point4D** lst = new Point4D*[total];
unsigned index = 0;
// create a regular grid of points with a spacing of 1.0
   units
for(unsigned j = 0, n = size2; j < n; ++j) {
  for(unsigned i = 0, m = size1; i < m; ++i) {
    lst[index++] = new Point4D(i, j, 0.0, 1.0);
  }
}
unsigned sizes[] = { size1, size2 };
// allocate a two-dimensional control grid with the
// given sizes
PointGrid *grid = new PointGrid(sizes, 2, lst);
KnotVector *kVecs[2];
// create open knot vectors for the basis functions; the
// degree of interpolation is the respective number of
// functions minus one, yielding a Bezier basis
kVecs[0] = new KnotVector((size1 - 1), size1);
kVecs[1] = new KnotVector((size2 - 1), size2);
NURBSBasis *basis = new NURBSBasis(kVecs, 2);
return new Patch(grid, basis);
```

Listing 3.2: The code generating the control points and knot vectors for a planar NURBS surface. It is called by the `createGeometry()` method in Listing 3.1.

2005). Such "structure-less" shapes can, however, be described as predefined object templates whereby the compact shape representation provided by NURBS plays an important role. The predefined data structures are instantiated only upon template evaluation to yield the desired shape. Templates can therefore be considered to be commands that do not modify but that create geometry. The resulting shape is, at first, only static. A certain degree of flexibility can be added, however, by the affine transformations described in Section 3.3.1, in particular by means of *nonuniform scaling*.

Template- and operator-based modeling complement each other. Templates can provide, on one hand, the input to a list of modeling commands, thus representing the "alignment kernel" of the model (cf. Leyton, 2001). This is shown in Figure 3.3 where some sample geometries, described in terms of templates, have been combined with sequences of modeling commands. The modeling operators can, on the other, hand be used in the definition of the templates. In this case it is possible to extend their range of representable shapes beyond static geometry. *When defined in terms of modeling operations, templates can become fully parametrized objects themselves*.

The geometry resulting from a given template may be used as intermediate data within other templates. This establishes a hierarchy of model components and leads to a system

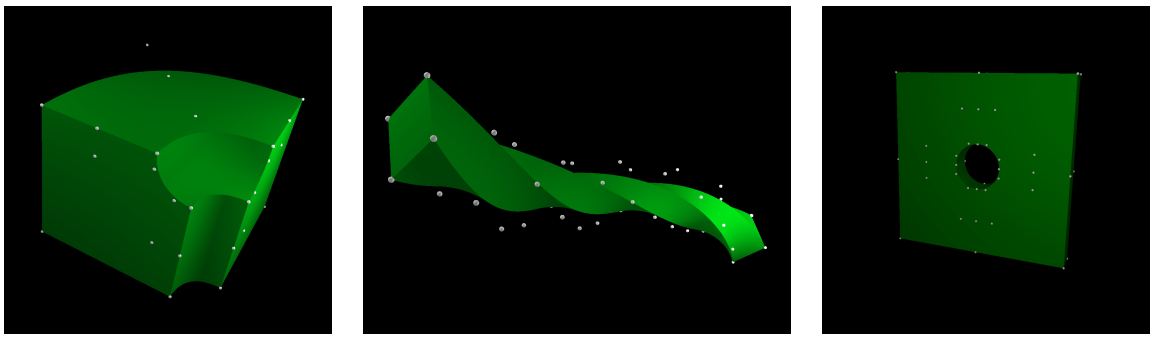

Figure 3.3: Sample geometries that result from the combination of different templates with modeling operators. The leftmost frame shows a panel with a circular cut-out after a *Revolution* operation (i.e. a circular sweep). The cross-section is described as a template. The center frame shows a thin rod whose control points have been rotated such as to describe the twisted geometry. Each $2 \times 2 \times 1$-subset of control points in longitudinal direction is rotated by a multiple of 36 degrees. The rightmost figure shows a thin plate with a circular hole. This plate is composed from four instances of the panel template that has also been used in the leftmost frame. These instances are aligned using affine transformations and are extruded with the desired panel thickness.

of "cascading" templates. Complex models may thus be described in terms of previously defined and *reusable* components, where each template encodes the rules by which its parameters are translated into the resulting shape. Nesting then allows to establish dependencies among the components' parameters. Constraints between the parts of a complex model can be ensured by propagation of suitable parameters from a parent template to its children. This increases the effect of data amplification and allows to efficiently describe complex models. The most important effect, however, is that this provides model adaptivity on multiple levels. Parameters can be adapted on all levels of the model hierarchy while the encoded constraints ensure consistency of the model. An extensive example for this is given in Chapter 4.

4 Template models for NURBS solids

An excellent field of application for NURBS-based geometries and Isogeometric Analysis are fluid flow simulations. These problems are highly sensitive to imperfections in the boundaries of their domain and hence gain enormously from smooth shape representations (Hughes et al., 2005). Of these, the simulation of wind energy turbines is particularly interesting. It comprises a rotating large-scale system of smooth rotor blades, together with the surrounding body of air. The definition of volumetric numerical models constitutes here a large part of the problem.

Two basic design types can be distinguished for wind energy turbines, based on their respective axis of rotation. Horizontal-axis wind turbines (HAWT) are the popular, well-known concept. They can be found on all orders of magnitude, from child's toys to large offshore installations. Their axis of rotation is parallel to the direction of fluid flow which lets the wind impact the rotor blades at more or less constant angles. This is commonly taken as a basis for a dimensional reduction of the problem. Fully volumetric simulations such as performed in (Bazilevs et al., 2011; Hsu et al., 2011) allow to recognize turbulence or fluid-structure interaction effects, though, which indicate the contrary.

The second design type is the Vertical-axis wind turbine (VAWT). Other than HAWT, these wind turbines are more self-contained and fit into close spaces, making them valuable for application in urban areas. Their axis of rotation is perpendicular to the direction of fluid flow. Accordingly, the impact angle of the fluid flow is changing steadily which forbids an exploitation of the rotational symmetry for a model simplification. VAWT geometries nonetheless possess a certain structure, making them eligible for procedural representations.

An integral part of modeling the "negative space" around the wind turbine is the description of the rotor blades' geometry. It is clear that their fidelity directly affects the quality of the fluid flow solution. Airfoil profiles are, unfortunately, one of the cases that cannot be handled purely in terms of modeling operators. They result, similar to aircraft fuselage and ship hulls, from optimization processes and significant experimental work, and their shape follows aerodynamic requirements rather than geometric considerations. template representations of the rotor blades hence become necessary. Their generation from given profile curves is shown in the following section. The application of these templates to the setup of the fluid domain is then described in Section 4.2.

4.1 Simplification of airfoil profiles

Airfoil cross-sections are typically described in terms of analytical functions or as sets of sample points derived from the former. Figure 4.1 shows a selection of these profile curves. This work assumes that the profiles are given in terms of the point sets $\mathbf{Q} :=$

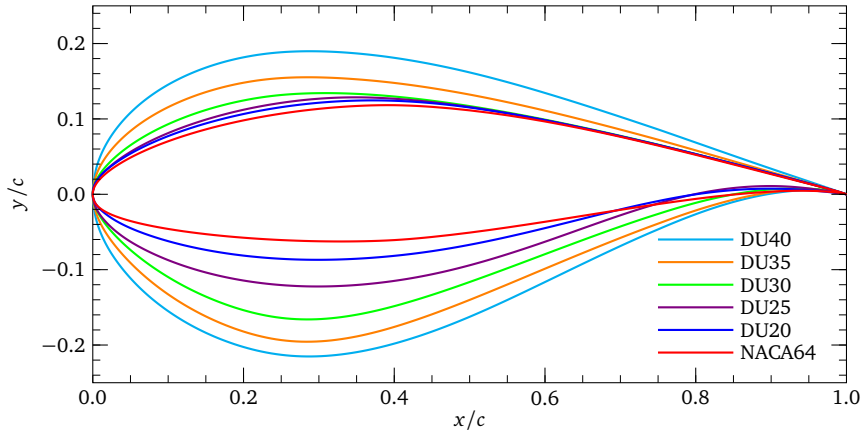

Figure 4.1: Cross sections of various airfoil profiles as used in (Bazilevs et al., 2011). The point samples comprising the profiles are commonly normalized with respect to the *chord length* c.

$\{\mathbf{Q}_i = (x_i, y_i)\}_{i=0}^{m}$. They can be found in publicly accessible databases.[1] These samples can be converted into spline curves by means of the algorithms described in (Piegl and Tiller, 1997) which determine appropriate control points and parametrizations through systems of linear equations. These procedures yield curves that either interpolate the given samples or that approximate these points in a least-squares fashion. In this work a *global interpolation* algorithm has been chosen which introduces a control point for each initial point sample. This is independent of the order of interpolation which merely affects the bandwidth of the employed equation system. Profile curves usually consist of several hundreds of data points in order to accurately capture their geometry. This would lead to an overhead of point data, contradicting the conciseness of a template representation. In order to obtain manageable sets of points, data reductions are required. To that end, two simplification algorithms are used. Their main objective is a reduction of the number of control points in the final curve representation while, at the same time, its geometric character is preserved. Therefor, the sets of points describing the profile curves are split into two subsets, corresponding to the lower and the upper portion of the profile, respectively. Application of the global interpolation algorithm then yields two independent B-Spline curves that are joined C^0-continuously at the leading edge of the profile.

4.1.1 Subset selection

The first reduction approach determines a subset $\bar{\mathbf{Q}} \subset \mathbf{Q}$ from the full set of discrete curve samples. To that end a *divide-and-conquer* approach is chosen. The set of discrete curve samples is therefor split into subsets of points. The first and last points $\bar{\mathbf{Q}}_{low}$ and $\bar{\mathbf{Q}}_{upp}$ of each set, based on their x-coordinates, are used for a linear approximation of the set. This

[1] For example the UIUC Airfoil Coordinates Database at `http://www.ae.illinois.edu/m-selig/ads/coord_database.html`.

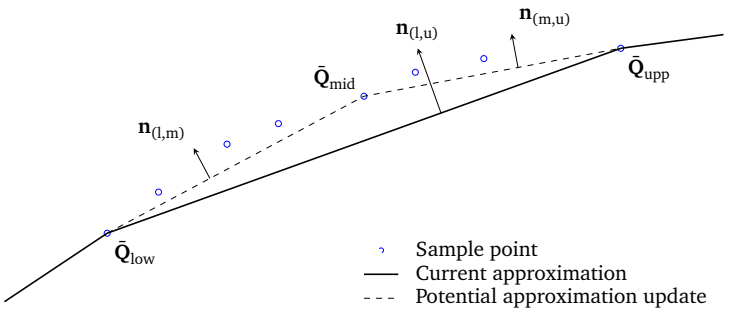

Figure 4.2: Recursive selection of points from a full set of samples. $\bar{\mathbf{Q}}_{\text{low}}$ and $\bar{\mathbf{Q}}_{\text{upp}}$ denote the respective lower and the upper elements in a range of point samples (shown in blue). $\bar{\mathbf{Q}}_{\text{mid}}$ is the element in the middle of the point range. The line segment $\left(\bar{\mathbf{Q}}_{\text{low}}, \bar{\mathbf{Q}}_{\text{upp}}\right)$ is the current approximation to the polygon of sample points with the normal vector $\mathbf{n}_{(\text{l,u})}$. The criterion for subdivision is based on the new line segments $\bar{\mathbf{Q}}_{\text{low}} - \bar{\mathbf{Q}}_{\text{mid}}$ and $\bar{\mathbf{Q}}_{\text{mid}} - \bar{\mathbf{Q}}_{\text{upp}}$ with the normal vectors $\mathbf{n}_{(\text{l,m})}$ and $\mathbf{n}_{(\text{m,u})}$.

is illustrated in Figure 4.2. A point $\bar{\mathbf{Q}}_{mid}$ is chosen from the remaining points. It is either located at the midpoint of the set or it is chosen simply by the cardinality of the set. This point is used to set up the segments $\bar{\mathbf{Q}}_{mid} - \bar{\mathbf{Q}}_{low}$ and $\bar{\mathbf{Q}}_{upp} - \bar{\mathbf{Q}}_{mid}$.

The criterion for further subdivision of the point sets is adapted from the *one-edge normal test* as described in (Espino et al., 2003). It is based on the normal vectors $\mathbf{n}_{(\text{mid,low})}$ and $\mathbf{n}_{(\text{upp,mid})}$. If the relative difference of these vectors from the normal vector $\mathbf{n}_{(\text{upp,low})}$ exceeds a user-defined threshold then the point $\bar{\mathbf{Q}}_{\text{mid}}$ is included in the subset $\bar{\mathbf{Q}}$ and the selection scheme is repeated recursively for each sub-range of points $\left[\bar{\mathbf{Q}}_{\text{low}}, \bar{\mathbf{Q}}_{\text{mid}}\right]$ and $\left[\bar{\mathbf{Q}}_{\text{mid}}, \bar{\mathbf{Q}}_{\text{upp}}\right]$. This algorithm continues either until all line segments have been checked or until the normal deviation stays below the defined error level.

4.1.2 Adaptive sampling

The second simplification method is based on a resampling of the "original" profile curve. That is, a completely new set of points, independent from the given samples, is computed for use in the interpolation scheme. Therefor the complete set of initial data points is used for a global interpolation. This results in a one-to-one translation of samples to control points. Starting point of this approach is a relatively coarse set of samples taken along the arc length of this curve which is successively refined throughout the procedure.

The initial coarse sampling defines a piecewise linear approximation of the full curve where each segment is defined by a pair of samples $\bar{\mathbf{Q}}_{low} = \mathbf{x}(\xi_{low})$ and $\bar{\mathbf{Q}}_{upp} = \mathbf{x}(\xi_{upp})$, as illustrated in Figure 4.3. In order to determine whether it is necessary to subdivide this segment, the average of $\bar{\mathbf{Q}}_{low}$ and $\bar{\mathbf{Q}}_{upp}$ is compared with the sample $\bar{\mathbf{Q}}_{mid}$. This sample is taken at the average of the parametric coordinates ξ_{low}, ξ_{upp}:

$$\xi_{mid} = \frac{\xi_{low} + \xi_{upp}}{2}.$$

To put it differently: *the sample taken at the average of the points' parametric coordinates is compared to the points' average physical coordinates*. If that distance exceeds a given threshold then the new sample $\bar{\mathbf{Q}}_{mid}$ is added to the set $\bar{\mathbf{Q}}$. The subdivision is then recursively repeated for the segments $\bar{\mathbf{Q}}_{low} - \bar{\mathbf{Q}}_{mid}$ and $\bar{\mathbf{Q}}_{mid} - \bar{\mathbf{Q}}_{upp}$ as long as the coordinate deviation exceeds the set error level.

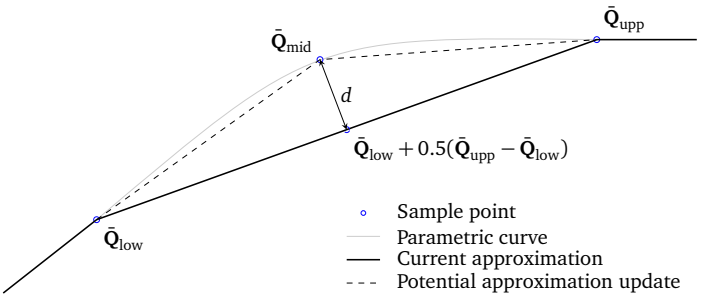

Figure 4.3: Adaptive sampling of a profile curve describing an airfoil. $\bar{\mathbf{Q}}_{low}$ and $\bar{\mathbf{Q}}_{upp}$ are two sample points along a curve that results from considering all initial data points. Each of these samples is assigned to a parametric coordinate ξ_{low} and ξ_{upp}, respectively. The sample point $\bar{\mathbf{Q}}_{mid}$ is evaluated at the parametric coordinate $\xi_{mid} = {}^1/_2(\xi_{low} + \xi_{upp})$. The criterion for subdivision is based on its deviation from the average of $\bar{\mathbf{Q}}_{low}$ and $\bar{\mathbf{Q}}_{upp}$.

4.1.3 Error analysis

The sets of point data that originate from the simplification routine are then used in the global interpolation scheme. The resulting curves represent a good approximation to the original, uncompressed curves, as can be seen in Figure 4.4 that compares a simplified NACA64-618 profile (using a subset selection approach) with its original. It can be readily confirmed that the simplification retains the geometric character of the curve—despite a reduction to one third of the original control points. Both the control points and the images of the knot values exhibit a regular spacing along the curve's arc length. This helps to ensure a nearly uniform element size distribution in the fluid domain mesh based on this geometry. Similar compression rates can be attained for the other profiles and for both simplification algorithms. It is important to note, however, that the two methods perform differently for the considered profile curves. That is, there seems to be no optimal compression method for the general case.

Table 4.1 lists the results of the reduction schemes applied to different airfoil profiles. Apart from the NACA0018 section, all airfoils are asymmetric. Nevertheless, the section curves are defined by the same number of initial sample points. For the shown thresholds they are reduced to the same number of points for both the upper and the lower portion.

It is clear that the simplification of the airfoil profiles introduces errors into the geometric representation. In order to determine the magnitude and the distribution of this error, the simplified curves are compared with the original sections resulting from the full set of point samples $\{\mathbf{Q}_i\}$. In order to determine the error, both curves are sampled along

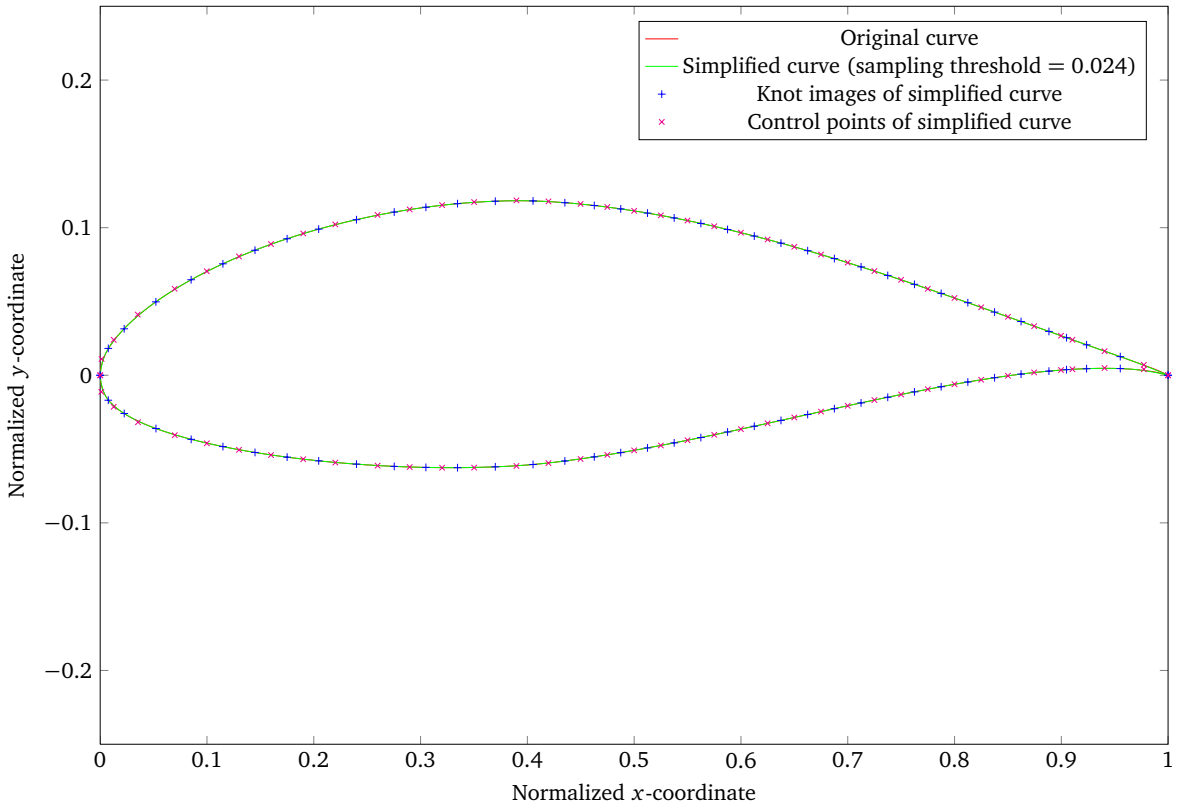

Figure 4.4: Comparison of a simplified NACA64-618 profile curve (resulting from the subset selection scheme) with the curve given by a full set of samples. Both the control points and the knot images of the curve exhibit a regular spacing, hinting at a good mesh quality for a later application of the curve in the modeling of fluid domains.

their arc length with a predefined number of samples. However, the geometric mapping from the parameter space $\bar{\Omega}^1$ of the curve to physical coordinates is in general nonlinear. That is, samples ξ_i that are regularly spaced in $\bar{\Omega}^1$ are not necessarily mapped to points $\mathbf{x}(\xi_i)$ that are regularly spaced in $\mathbb{R}^2$. These mappings can differ even between curves resulting from the same set of point samples but from different simplification methods/selection thresholds. It is hence necessary to uniformly distribute the sample points along the curves' arc length l_{arc}. This length can be approximately determined by summing up the respective distances between successive mapped samples:

$$l_{arc} = \sum_{i=0}^{m-1} \| \mathbf{x}(\xi_{i+1}) - \mathbf{x}(\xi_i) \|_2 \tag{4.1}$$

The relative position $l_{rel}(\xi_i)$ of the sample $\mathbf{x}(\xi_i)$ along the arc length can then be computed by:

$$l_{rel}(\xi_i) = \frac{1}{l_{arc}} \sum_{j=0}^{i-1} \| \mathbf{x}(\xi_{j+1}) - \mathbf{x}(\xi_j) \|_2, \tag{4.2}$$

Profile	Threshold (Method)	Segment	Max. error	Avg. error	Initial points	Final points
NACA64	0.024 (Subset)	upper lower	$3.385 \cdot 10^{-4}$ $2.887 \cdot 10^{-4}$	$5.312 \cdot 10^{-5}$ $4.216 \cdot 10^{-5}$	102	39
NACA0018	0.022 (Adaptive)	upper lower	$2.713 \cdot 10^{-4}$ $2.713 \cdot 10^{-4}$	$4.713 \cdot 10^{-5}$ $4.713 \cdot 10^{-5}$	50	33
DU21	0.026 (Subset)	upper lower	$8.442 \cdot 10^{-4}$ $8.246 \cdot 10^{-4}$	$1.488 \cdot 10^{-4}$ $1.599 \cdot 10^{-4}$	200	33
DU25	0.022 (Adaptive)	upper lower	$8.187 \cdot 10^{-4}$ $4.041 \cdot 10^{-4}$	$1.311 \cdot 10^{-4}$ $5.808 \cdot 10^{-5}$	200	33
DU30	0.032 (Subset)	upper lower	$7.876 \cdot 10^{-4}$ $6.691 \cdot 10^{-4}$	$1.208 \cdot 10^{-4}$ $1.052 \cdot 10^{-4}$	200	27
DU35	0.036 (Subset)	upper lower	$3.620 \cdot 10^{-4}$ $4.182 \cdot 10^{-4}$	$5.821 \cdot 10^{-5}$ $7.734 \cdot 10^{-5}$	200	26
DU40	0.065 (Adaptive)	upper lower	$4.693 \cdot 10^{-4}$ $5.448 \cdot 10^{-4}$	$6.408 \cdot 10^{-5}$ $9.193 \cdot 10^{-5}$	200	17

Table 4.1: Overview of the compression rates and approximation errors for different airfoil profiles. Each entry states the simplification method that achieved the best reduction together with the employed selection threshold and the resulting approximation error. With the NACA0018 profile being the exception, all airfoils are asymmetric. The approximation error is therefore given for both the respective upper and the lower segments of each profile. The two rightmost columns show the initial number of control points of the respective profile curves together with the number of points defining the simplified curves. Both the upper and lower profile segments of each curve are described by the same number of control points—in spite of the profiles' asymmetry.

where $\|\cdot\|_2$ denotes the l^2 vector norm. For a vector $\mathbf{a}$ with components a_i it is defined as

$$\|\mathbf{a}\|_2 = \sqrt{\sum_{i=1}^{3} |a_i|^2}.$$

Linear interpolation of the samples' positions allows to determine the parametric coordinates leading to regularly spaced sample points along the curve's arc length. For each of these points both the geometric error

$$e_{geom} = \| \mathbf{x}_{orig}(\xi) - \mathbf{x}_{red}(\xi) \|_2 \tag{4.3}$$

and the tangential error

$$e_{tang} = \frac{\| \mathbf{x}_{orig,\xi}(\xi) - \mathbf{x}_{red,\xi}(\xi) \|_2}{\| \mathbf{x}_{orig,\xi} \|_2} \tag{4.4}$$

are determined with respect to the original curve. The following error analysis is based on 200 samples taken along the arc length of the upper respectively the lower portion of the airfoil sections.

The two simplification methods lead to characteristic error distributions along the arc length of the profile, shown in Figure 4.5 for a NACA64-618 airfoil. Similar distributions

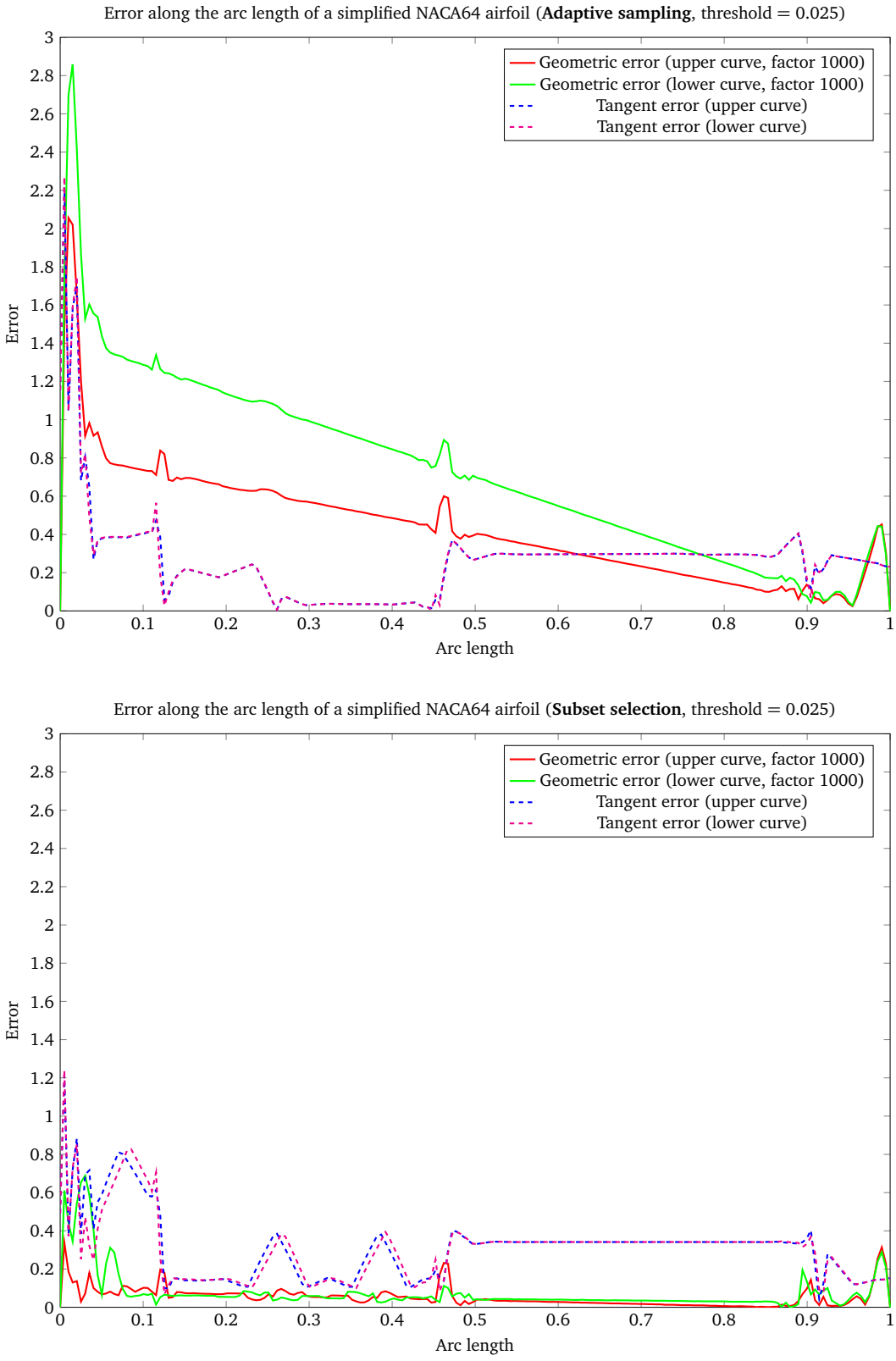

Figure 4.5: The distribution of the geometric and the tangent error along the arc length of the NACA64-618 profile for the selection threshold 0.025. One can clearly recognize the lower error levels of the model produced by the subset selection scheme, shown on the bottom. Errors are shown for both portions of the asymmetric profile, and the geometric error is scaled up by a factor of 10^3.

Threshold		Subset selection			Adaptive Sampling		
		Max. error	Avg. error	Points	Max. error	Avg. error	Points
0.015	upper	$2.332 \cdot 10^{-4}$	$5.593 \cdot 10^{-5}$	56	$2.574 \cdot 10^{-3}$	$5.156 \cdot 10^{-4}$	50
	lower	$1.644 \cdot 10^{-4}$	$4.486 \cdot 10^{-5}$	56	$3.491 \cdot 10^{-3}$	$8.569 \cdot 10^{-4}$	50
0.020	upper	$3.501 \cdot 10^{-4}$	$6.797 \cdot 10^{-5}$	54	$2.052 \cdot 10^{-3}$	$4.345 \cdot 10^{-4}$	36
	lower	$2.670 \cdot 10^{-4}$	$6.398 \cdot 10^{-5}$	54	$2.856 \cdot 10^{-3}$	$7.339 \cdot 10^{-4}$	35
0.025	upper	$3.385 \cdot 10^{-4}$	$5.312 \cdot 10^{-5}$	39	$2.056 \cdot 10^{-3}$	$4.385 \cdot 10^{-4}$	30
	lower	$6.884 \cdot 10^{-4}$	$7.559 \cdot 10^{-5}$	38	$2.858 \cdot 10^{-3}$	$7.315 \cdot 10^{-4}$	30
0.030	upper	$4.968 \cdot 10^{-4}$	$6.792 \cdot 10^{-5}$	31	$2.056 \cdot 10^{-3}$	$4.385 \cdot 10^{-4}$	26
	lower	$6.919 \cdot 10^{-4}$	$8.921 \cdot 10^{-5}$	31	$2.859 \cdot 10^{-3}$	$7.302 \cdot 10^{-4}$	26
0.035	upper	$5.587 \cdot 10^{-4}$	$8.677 \cdot 10^{-5}$	27	$7.126 \cdot 10^{-3}$	$1.782 \cdot 10^{-3}$	23
	lower	$8.209 \cdot 10^{-4}$	$1.370 \cdot 10^{-4}$	27	$1.114 \cdot 10^{-2}$	$3.007 \cdot 10^{-3}$	24
0.040	upper	$5.587 \cdot 10^{-4}$	$7.862 \cdot 10^{-5}$	26	$7.124 \cdot 10^{-5}$	$1.781 \cdot 10^{-3}$	17
	lower	$4.477 \cdot 10^{-3}$	$1.254 \cdot 10^{-3}$	25	$9.807 \cdot 10^{-3}$	$2.788 \cdot 10^{-3}$	17

Table 4.2: Comparison of the approximation error of the NACA64 profile for both simplification methods and different selection thresholds. Each entry shows the maximum and the average error together with the number of control points defining the resulting curve. The error is specified for both segments of the asymmetric profile.

arise for the other profiles shown in Figure 4.1. The most salient effect is caused by the split of the profile curve into two halves. Thereby the continuity of the curve is reduced to C^0 at the leading edge of the airfoil, corresponding to an arc length of zero. The tangent error accordingly exhibits a large increase in this region. A second region of large tangential error can be identified for the arc length range $[0.5, 0.9]$. Comparison with Figure 4.4 shows that this region corresponds to the smooth slope towards the trailing edge of the profile curve. This portion possesses a relatively low curvature compared to the overall airfoil. Both simplification methods select data points based on a linear approximation to the original shape. Regions with low curvature can therefore be approximated by few samples, confirmed by the low geometric error for this region.

The adaptive sampling scheme appears to be sensitive to profile curves with largely varying curvature. This is the case for the NACA64-618 profile that exhibits large portions of low curvature towards the trailing edge of the airfoil whereas the leading edge is, in comparison, strongly curved. In order to capture the latter, a relatively low error threshold has to be specified. This, however, leads to the introduction of excessive sample points in the regions of low curvature. These regions can already be faithfully described by few data points. This explains the low geometric error in the rear parts of the profile curve. At the same time a large error can be observed at the leading edge of the profile. One must assume, however, that the split of the profile curve into two halves contributes to this error spike.

Figure 4.6 illustrates the evolution of the error for different selection thresholds for both simplification methods. It should be noted that the principal distribution of the error, as shown in Figure 4.5, does not change with varying thresholds, only the magnitude of the error is affected. Therefore, only the mean geometric error is considered. A quantitative

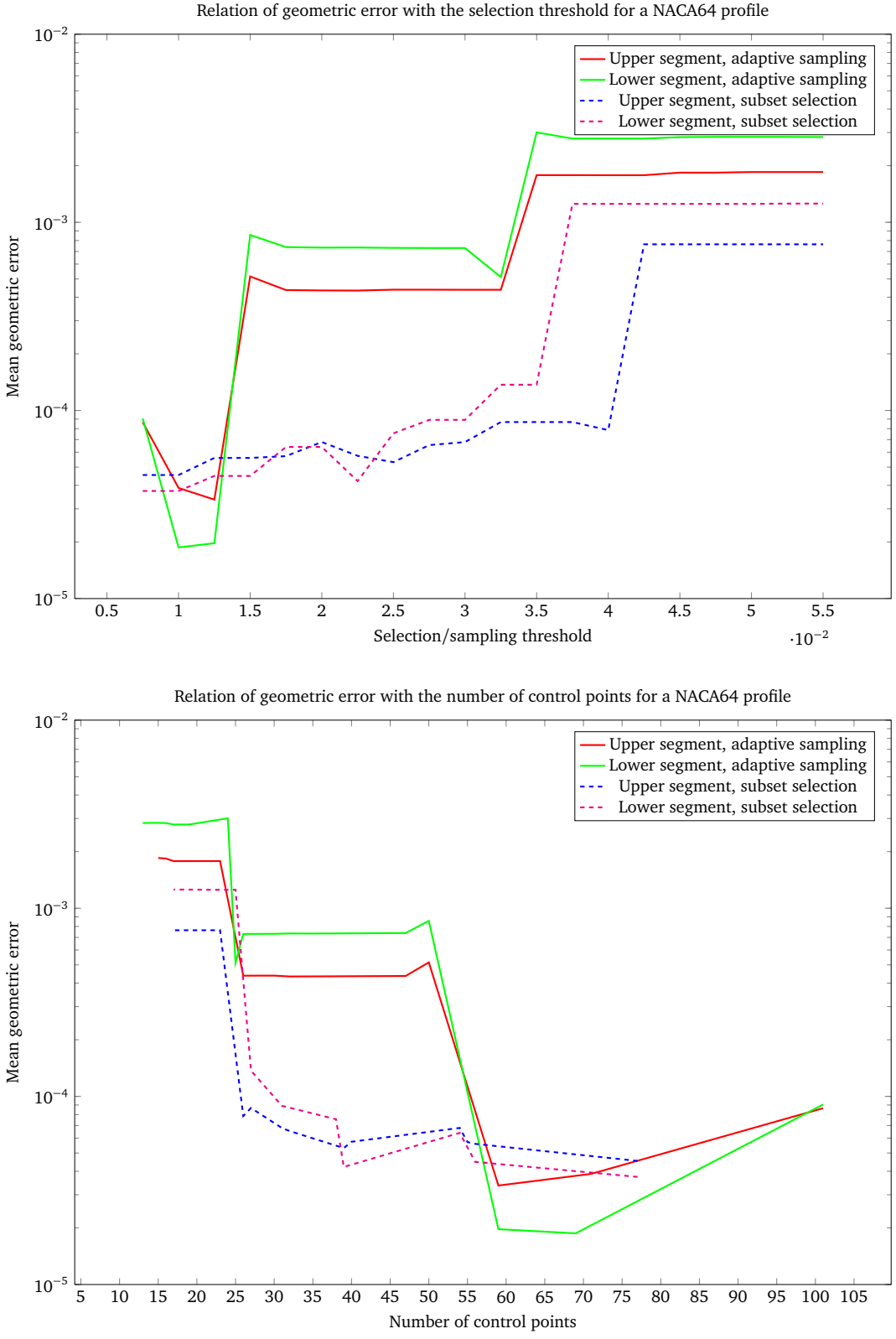

Figure 4.6: The size of the geometric error of the simplified profile curves and its relation to the size of the curve's representation. The left-hand image shows the mean geometric error over a profile curve for different selection/sampling thresholds for the two simplification schemes. The right-hand image illustrates the attainable error levels in relation to the number of control points being used for the curve representation.

comparison of the error levels resulting for different selection thresholds is given in Table 4.2 for the NACA64-618 section. It considers, in addition to the mean geometric error, the magnitude of the error spike at the leading edge of the simplified profile.

Interestingly, the geometric error exhibits, for both algorithms, characteristic error plateaus. For certain "critical thresholds" jumps in the error level can be observed. This can be attributed, at least for the subset selection scheme, to the initial distribution of the point samples. Their spacing is highly irregular, both with respect to their x-coordinates and to their distances. The leading and the trailing edge of the profile exhibit a much higher density of data points than other parts of the curve. This leads to a certain instability with respect to the user-defined selection thresholds. That is, a slight variation in the threshold can result in different point sets being regarded for the global interpolation. This dependency can be reduced by using regularly spaced samples as a starting point for the simplification routines.

The adaptive sampling scheme is independent from the initial data. Instead, it selects its sample points from the "uncompressed" curve, defined by the full set of initial data. This indirect relation of the adaptive selection method from the initial data points can have a detrimental effect: new sample points are taken with respect to the interpolated profile, irrespective of the number of initial data points. For very small selection thresholds this method can actually result in *more control points than initially given*. New samples are taken, by definition, at the midpoints between existing samples. These are compared with a point at the average parametric coordinate of the existing samples. Due to the nonlinear mapping, these points do not necessarily coincide with the midpoint between the given points along the curve's arc length. Instead, this point is slightly shifted. This effects a sensitivity to the selection thresholds similar to that of the subset selection scheme. It introduces, in addition, a kind of "minimum error" into these error studies. Corresponding sample points along different curves can therefore exhibit slight offsets which can be recognized by local oscillations in the error—although visual inspection indicates good agreement of the curves being compared. This can be reduced by a higher number of sample points for the error analysis in combination with a more faithful computation of the samples' parametric coordinates.

4.2 Templates for VAWT fluid domains

Given a selection threshold, the aforementioned simplification algorithms return a compact set of control points along with an appropriate parametrization. The corresponding profile curves provide a faithful representation of the original airfoil sections, as confirmed by Figure 4.4. Their template implementation is straightforward. A curve's control points are, similar to the initial point data, normalized with respect to the chord length of the considered airfoil. In addition to the chord length, further parameters are relevant for airfoil shapes. These determine an airfoil's alignment and orientation and are depicted in Figure 4.7. Together with the chord length, they establish a natural parametrization for the airfoil geometry.

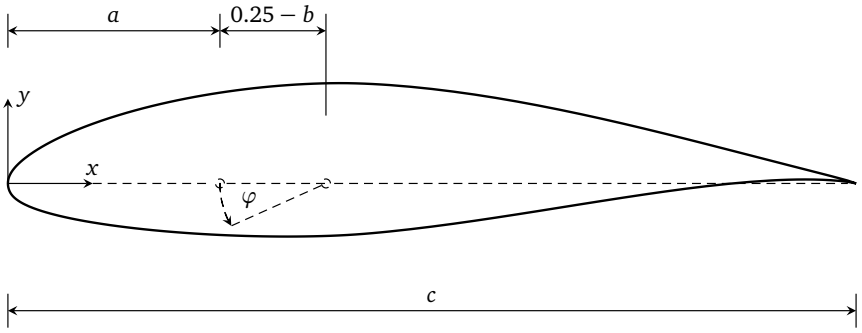

Figure 4.7: Parametrization of an airfoil as used in (Bazilevs et al., 2011). The chord length of the profile is denoted by L. The aerodynamic center is located at a units from the leading edge. The profile is twisted by an angle φ around the z-axis. The axis of revolution passes through a point that is located on the median axis (dashed) at $(0.25 - b)$ units from the aerodynamic center.

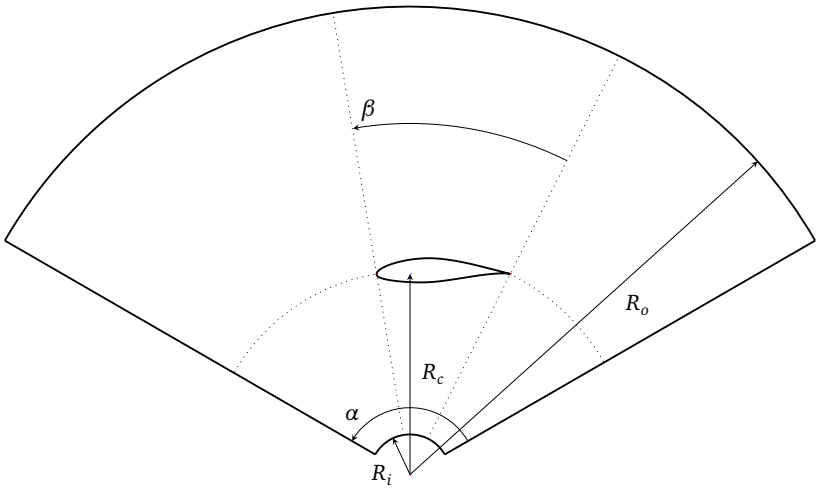

Figure 4.8: Parametrization of a section of the fluid domain. The section can be completely defined by the inner radius r_i, the outer radius r_o, the arc length α and the distance r_c of the profile's median axis to the center of revolution. The angle β denotes the arc length that corresponds to the chord length of the profile. The section cannot be modeled using a single NURBS patch only. Instead the domain is subdivided into six patches whose boundaries are shown in dotted lines.

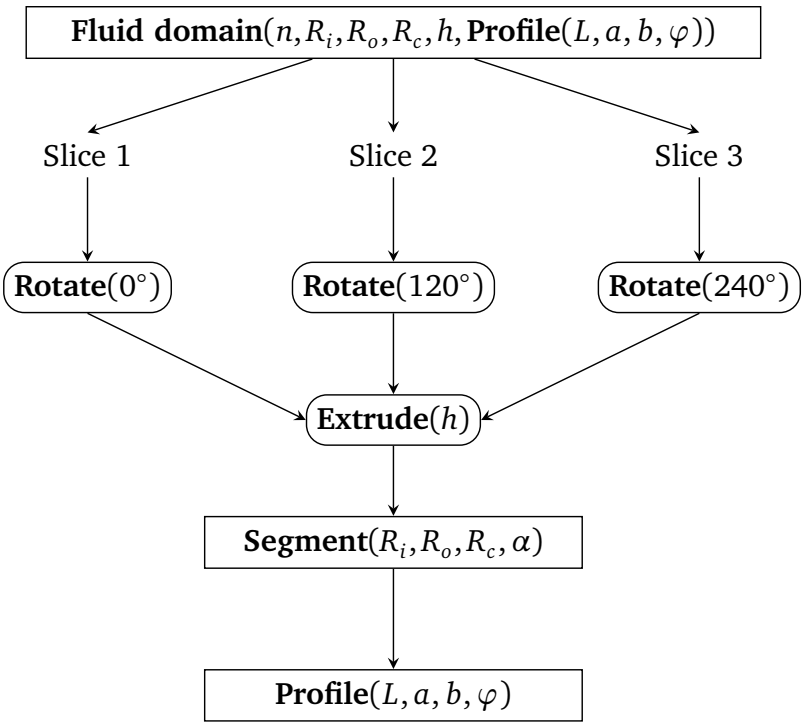

Figure 4.9: Template hierarchy for the negative space around the Vertical-Axis Wind Turbine.

Due to an ever-changing angle of attack of the fluid flow on the rotor blades, the rotational symmetry of the model cannot be exploited for a dimensional reduction of the problem. It can nonetheless be made use of in the definition of the system's geometry. That is, the negative space around the blades only has to be modeled for a single section enclosing a single blade. This space can be readily described by a template. Similar to the airfoil template, the fluid domain section is determined by a handful of parameters. These are: the inner and the outer radii R_i respectively R_o, the distance R_c of the airfoil to the axis of rotation, and the arc length α of the section which depends on the number of rotor blades in the design of the wind turbine. These parameters are illustrated in Figure 4.8 along with the decomposition of the section into individual patches. This decomposition is necessary as a consequence of the simple topology of NURBS patches; it is also the motivation for the split of the airfoil profiles into two separate curves.

Using the simplified airfoil profiles as input, the template of the fluid domain can be fully described in terms of modeling operations. A section of the fluid domain can accordingly be described in terms of eight model parameters that are translated into the required data structures. The instantiation of the template geometry proceeds as follows: at first the section curves of the airfoil are instantiated using the specified chord length and a possible pitch angle. The arc length β results from the center radius R_c and the arc length α. It is used to describe the respective inner and outer arcs delimiting the fluid domain in radial direction. These arcs are themselves provided as templates that, given an arc length, return a quadratic NURBS curve. Affine transformations are used for proper

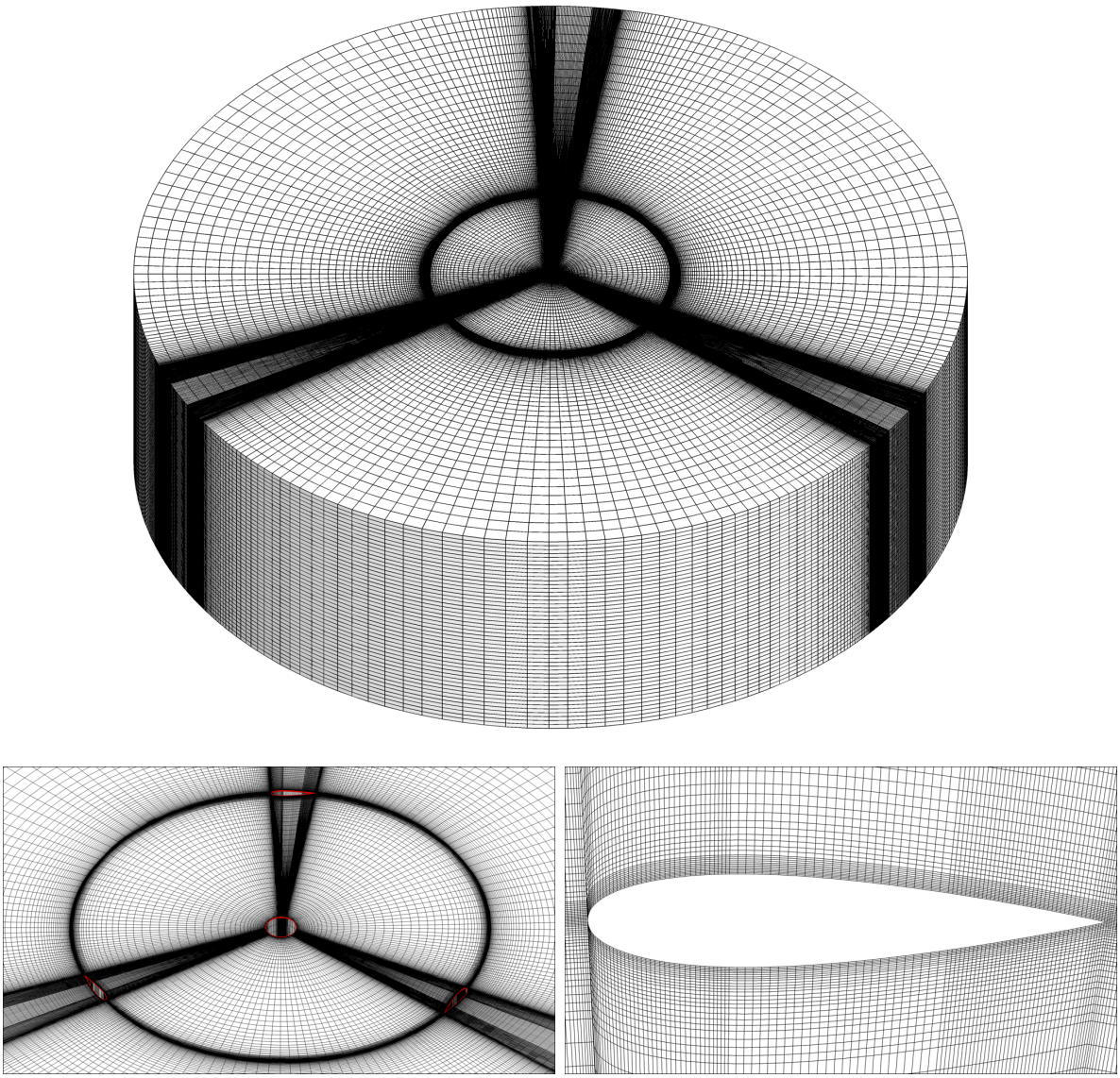

Figure 4.10: The isogeometric finite element mesh of the fluid domain after full refinement. For the purpose of visualization, each quadratic NURBS element is interpolated with $2 \times 2 \times 2$ bilinear elements. Images courtesy of Y. Bazilevs.

alignment of these arcs with the sections of the airfoil. Knot refinement is used to produce conformal parametrizations of the arcs and the portions of the airfoil. Eventually, the two patches around the airfoil are generated by a *ruling* operation. It effects, first and foremost, a linear interpolation of the two given boundary curves in radial direction; higher orders of interpolation can be readily achieved, though. The lateral sides of the resulting patches are extracted subsequently and serve as input for radial sweeps. This results in six independent patches that possess conformal point grids and parametrizations along their respective shared edges. Extrusion of these patches along the turbine's vertical axis yields a segment of the volumetric fluid domain. This resulting segment can be copied and put into place using further transformations, producing a mesh for a wind turbine with straight blades, a so-called *giro-mill* design. More complex designs such as the *Darrieus* rotors involve helical or outwardly bent blades. Such designs are disregarded here; the current template can, however, act as a basis for their implementation.

The complete isogeometric mesh for a three-blade design is described by 18 solid patches comprising, depending on the employed airfoil profile and selection threshold, between 88 and 288 control points. The individual patches share consistent interfaces. Their template hierarchy is illustrated in Figure 4.9. Evaluation of the templates can be readily handled by a desktop computer in fractions of a second.

It is important to note that this model only describes the coarsest mesh for this configuration. In order to allow convergence of the numerical simulation, further mesh refinement is necessary—which can be equally well formulated as a list of operations. Accordingly, different versions of an isogeometric mesh can be kept by storing a coarse basic model along with different refinement macros. A fully refined mesh of the volumetric fluid domain is shown in Figure 4.10. Due to the construction steps it is possible to faithfully capture the boundary layers along the rotor blades. At the same time one can observe the high regularity of the mesh in the bulk of the domain.

5 Analysis-suitability of NURBS models for Isogeometric analysis

Operator-based modeling allows an abstract treatment of NURBS models and simplifies the definition and adaption of geometric models. According to the isogeometric concept the geometric and the numerical model are identical. That is, the proposed procedural modeling approach is simultaneously a basis for geometric modeling and for the description and modification of numerical models. NURBS are inherently a higher-order, higher-continuity approach. At least quadratic basis functions are necessary in order to describe the full range of possible geometries (Hughes et al., 2005). They are suitable as ansatz for Finite Element models (Bazilevs et al., 2006). As shown by Cottrell et al. (2007), their implicit higher continuity is a benefit for the numerical solution scheme as it allows, for instance, to limit the influence of model singularities by adjusting the continuity of the basis functions. In the context of structural vibrations the higher continuity effects a disappearance of spurious effects such as *optical branches* from the model spectra (Cottrell et al., 2006). IGA models furthermore exhibit only a third of the degrees of freedom compared to "classical" FEM discretizations (Uhm et al., 2008). They are consequently of high value for structural optimization problems (cf. Cho and Ha, 2008; Nagy et al., 2010; Seo et al., 2010a,b; Wall et al., 2008).

NURBS solids are the result of mapping a simple hexahedral parametric domain into a (curved) geometry. Many geometric features, for example kinks or sharp edges in an otherwise smooth shape, may only be achieved by distortion of this domain. This can adversely affect the representation of the solution field of the mathematical model and might even invalidate its solution (Lipton et al., 2010). Different geometric models can furthermore exhibit varying suitability for numerical analyses. This manifests itself by widely varying error levels between different parametrizations of the same model (Cohen et al., 2010; Xu et al., 2010). NURBS provide a highly flexible, *non-unique* representation scheme (cf. Mäntylä, 1987). That is, different parametrizations and control point configurations can, in general, be used for describing the same object. It is hence vital to be able to detect *already at the modeling stage* a possibly detrimental influence of the chosen representation on the properties of the numerical model.

Isogeometric Analysis is an extension of Finite Element procedures. Likewise, it constructs approximate solutions to systems of partial differential equations by Galerkin's method. The following section illustrates the features of the isogeometric concept using the boundary-value problem of linear elastostatics. Based on these, a set of mesh quality metrics is formulated and studied in Section 5.2.

5.1 Isogeometric analysis of linear elasticity problems

Consider a three-dimensional body $\Omega \subset \mathbb{R}^3$, bounded by its surface Γ which consists of the subsets Γ_u and Γ_t such that $\Gamma = \Gamma_u \cup \Gamma_t$, $\Gamma_u \cap \Gamma_t = \varnothing$. This body is subject to volumetric body loads b_i, for instance gravity, and to tractions $\bar{t}_i$ acting on Γ_t. It is supported on Γ_u by prescribed displacements $\bar{u}_i$ such that rigid-body motions of the object are prevented. Primary variable of this problem is the displacement field u_i describing the deformation of the body under the specified loads. Assuming geometrically linear behavior, the strains corresponding to the displacement field are described by the symmetric, second-order *infinitesimal strain tensor* with the components

$$\epsilon_{kl} = \frac{1}{2}\left(\frac{\partial u_k}{\partial x_l} + \frac{\partial u_l}{\partial x_k}\right) = \frac{1}{2}\left(u_{k,l} + u_{l,k}\right), \tag{5.1}$$

where $(\cdot)_{,i}$ denotes a partial derivative with respect to the i-th coordinate. The stress state that evolves in the body as a consequence of the external loads and the constrained deformations is denoted by the symmetric, second-order *Cauchy stress tensor* with the components σ_{ij}. Stresses and strains are linked by the generalized Hooke's law

$$\sigma_{ij} = \mathbb{C}_{ijkl}\,\epsilon_{kl} \tag{5.2}$$

where the $\mathbb{C}_{ijkl}$ are the components of the fourth-order constitutive tensor describing the material behavior. This section makes use of the summation convention, that is, repetition of an index in a given term implies summation over this index.

The pointwise equilibrium of forces acting on a material point subject to body loads is described by

$$\sigma_{ij,j} + b_i = 0 \quad \forall x_i \in \Omega, \tag{5.3}$$

with the boundary conditions

$$\sigma_{ij} n_j = \bar{t}_i \quad \forall x_i \in \Gamma_t, \tag{5.4a}$$

$$u_i = \bar{u}_i \quad \forall x_i \in \Gamma_u, \tag{5.4b}$$

where n_j denotes the outward normal vector to the body's boundary. Closed solutions for the partial differential equations (5.3) subject to (5.4) can only be found for simple domains (Bathe, 1996). Approximate solutions can be constructed for arbitrary geometries, though. To that end above strong, point-wise conditions are transformed into the so-called "weak form", in which the equilibrium is only fulfilled in an integral sense. Therefor, Equation (5.3) is multiplied by the *virtual displacement field* δu_i and integrated over the whole domain. Partial integration and application of the divergence theorem (cf. Bathe, 1996; Hughes, 2000; Zienkiewicz et al., 2005) eventually yields the *principle of virtual work*:

$$\int_\Omega \delta\epsilon_{ij}\sigma_{ij}\,\mathrm{dV} = \int_\Omega \delta u_i b_i\,\mathrm{dV} + \int_{\Gamma_t} \delta u_i \bar{t}_i\,\mathrm{dA} \tag{5.5}$$

where $\delta\epsilon_{ij}$ is the virtual strain field corresponding to the virtual displacements. The symmetries of stress, strain and constitutive tensor can be exploited so as to introduce

a vector representation that simplifies the notation and that closely corresponds to the implementation of FEM, respectively IGA. Subsequently, both vector quantities and the vector representation of tensors are printed in boldface letters. The components of the stress tensor are thus arranged in the stress vector as follows:

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_{11} & \sigma_{22} & \sigma_{33} & \sigma_{12} & \sigma_{23} & \sigma_{13} \end{bmatrix}^T.$$

Likewise, we arrange the strain tensor's components in the strain vector

$$\boldsymbol{\epsilon} = \begin{bmatrix} \epsilon_{11} & \epsilon_{22} & \epsilon_{33} & 2\,\epsilon_{12} & 2\,\epsilon_{23} & 2\,\epsilon_{13} \end{bmatrix}^T.$$

The constitutive relations (5.2) accordingly read

$$\boldsymbol{\sigma} = \mathbf{C}\,\boldsymbol{\epsilon}$$

with $\mathbf{C}$ being the 6×6 matrix representation of the constitutive tensor. For linear elastic, isotropic material behavior this matrix can be fully expressed in terms of Young's modulus E and Poisson's ratio ν:

$$\mathbf{C} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} (1-\nu) & \nu & \nu & 0 & 0 & 0 \\ \nu & (1-\nu) & \nu & 0 & 0 & 0 \\ \nu & \nu & (1-\nu) & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} \end{bmatrix}$$

In this vector representation, the weak form (5.5) reads

$$\int_\Omega \delta\boldsymbol{\epsilon}^T \boldsymbol{\sigma}\,\mathrm{dV} = \int_\Omega \delta\boldsymbol{\epsilon}^T \mathbf{C}\,\boldsymbol{\epsilon}\,\mathrm{dV} = \int_\Omega \delta\mathbf{u}^T \mathbf{b}\,\mathrm{dV} + \int_{\Gamma_t} \delta\mathbf{u}^T \bar{\mathbf{t}}\,\mathrm{dA}, \tag{5.6}$$

where $\mathbf{u}$, $\mathbf{b}$ and $\bar{\mathbf{t}}$ are the displacement vector, the vector of body forces, and the surface traction vector, respectively. Furthermore, $\delta\mathbf{u}$ and $\delta\boldsymbol{\epsilon}$ are the so-called *virtual displacements* and the corresponding virtual strains.

5.1.1 Discretization with NURBS

Both the Finite Element Method and Isogeometric Analysis start from (5.6) in order to obtain approximate solutions fulfilling (5.3). To that end, they construct an approximate solution

$$\mathbf{u} \approx \sum_i \phi_i\,\mathbf{c}_i$$

from a set of trial functions ϕ_i together with a set of coefficients $\mathbf{c}_i$. Using the commonly employed isoparametric concept of FEM, the displacement field $\mathbf{u}(\xi)$ is described within

a given NURBS patch by interpolating the displacements $\hat{\mathbf{u}}_A = (u_A, v_A, w_A)^T$ of the solid's control points:

$$\mathbf{u}(\xi) = \sum_{A=0}^{T-1} R_A(\xi)\,\hat{\mathbf{u}}_A, \tag{5.7}$$

where T denotes the total number of control points in a given patch (cf. Section 2.2) and the R_A are the corresponding NURBS basis functions. This can be formulated as a matrix-vector product

$$\mathbf{u}(\xi) = \mathbf{R}(\xi)\,\hat{\mathbf{u}} \tag{5.8}$$

with the control point displacements $\hat{\mathbf{u}} = \left[\hat{\mathbf{u}}_0, \hat{\mathbf{u}}_1, \ldots, \hat{\mathbf{u}}_T\right]^T$ and the matrix of basis functions

$$\mathbf{R}(\xi) = \begin{bmatrix} R_0(\xi) & 0 & 0 & R_1(\xi) & 0 & 0 & \ldots & R_T(\xi) & 0 & 0 \\ 0 & R_0(\xi) & 0 & 0 & R_1(\xi) & 0 & \ldots & 0 & R_T(\xi) & 0 \\ 0 & 0 & R_0(\xi) & 0 & 0 & R_1(\xi) & \ldots & 0 & 0 & R_T(\xi) \end{bmatrix}.$$

In vector representation, the kinematic relations (5.1) can be written as

$$\boldsymbol{\epsilon} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 \\ 0 & \frac{\partial}{\partial y} & 0 \\ 0 & 0 & \frac{\partial}{\partial z} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial x} \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \mathscr{D}_k \mathbf{u} = \mathscr{D}_k \mathbf{R}(\xi)\,\hat{\mathbf{u}} = \mathbf{B}(\xi)\,\hat{\mathbf{u}} \tag{5.9}$$

with the *strain-displacement matrix*

$$\mathbf{B}(\xi) = \begin{bmatrix} R_{0,x}(\xi) & 0 & 0 & \ldots & R_{T,x}(\xi) & 0 & 0 \\ 0 & R_{0,y}(\xi) & 0 & \ldots & 0 & R_{T,y}(\xi) & 0 \\ 0 & 0 & R_{0,z}(\xi) & \ldots & 0 & 0 & R_{T,z}(\xi) \\ R_{0,y}(\xi) & R_{0,x}(\xi) & 0 & \ldots & R_{T,y}(\xi) & R_{T,x}(\xi) & 0 \\ 0 & R_{0,z}(\xi) & R_{0,y}(\xi) & \ldots & 0 & R_{T,z}(\xi) & R_{T,y}(\xi) \\ R_{0,z}(\xi) & 0 & R_{0,x}(\xi) & \ldots & R_{T,z}(\xi) & 0 & R_{T,x}(\xi) \end{bmatrix}$$

Analogous expressions can be set up for the virtual displacements $\delta\mathbf{u}$ and the virtual strains $\delta\boldsymbol{\epsilon}$. This transforms Equation (5.6) into

$$\delta\hat{\mathbf{u}}^T \int_\Omega \mathbf{B}(\xi)^T \mathbf{C}\,\mathbf{B}(\xi)\,\hat{\mathbf{u}}\,\mathrm{dV} = \delta\hat{\mathbf{u}}^T \int_\Omega \mathbf{R}(\xi)^T \mathbf{b}\,\mathrm{dV} + \delta\hat{\mathbf{u}}^T \int_{\Gamma_N} \mathbf{R}(\xi)^T \bar{\mathbf{t}}\,\mathrm{dA}. \tag{5.10}$$

This must be fulfilled for arbitrary virtual displacements $\delta\hat{\mathbf{u}}$. Above equation can hence be written as a system of linear equations:

$$\mathbf{K}\hat{\mathbf{u}} = \mathbf{f}$$

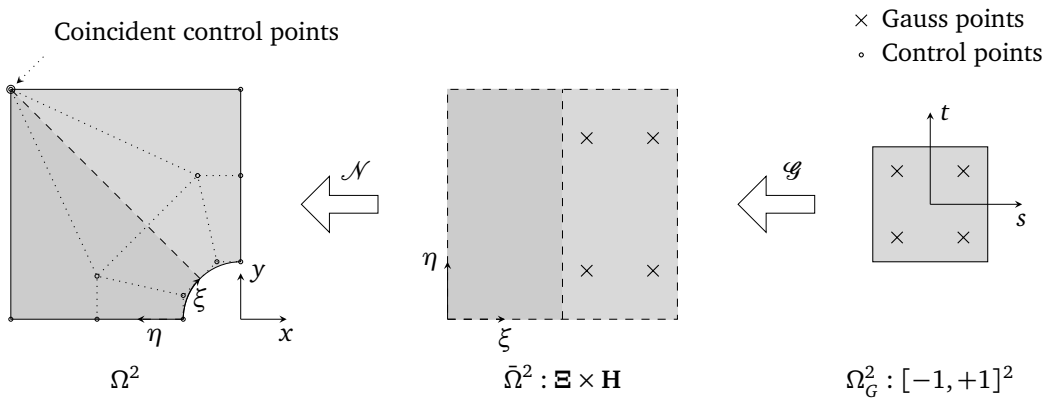

Figure 5.1: The different spaces that have to be considered during numerical integration for IGA, shown for the two-dimensional case. The actual quadrature is performed on the biunit parent domain Ω_G^2. The integrands are evaluated at appropriate $\boldsymbol{\xi}(\mathbf{x}_G)$, $\mathbf{x}_G \in \Omega_G^d$, $\boldsymbol{\xi} \in \bar{\Omega}^d$, that are determined through the mapping $\mathcal{G}$. The integrands comprise partial derivatives with respect to physical coordinates which are determined by means of the geometric mapping $\mathcal{N}$, respectively its inverse.

with the stiffness matrix

$$\mathbf{K} = \int_\Omega \mathbf{B}(\boldsymbol{\xi})^T \mathbf{C}\, \mathbf{B}(\boldsymbol{\xi})\, \mathrm{dV} \tag{5.11}$$

and the load vector

$$\mathbf{f} = \int_\Omega \mathbf{R}(\boldsymbol{\xi})^T \mathbf{b}\, dV + \int_{\Gamma_t} \mathbf{R}(\boldsymbol{\xi})^T \bar{\mathbf{t}}\, \mathrm{dA}. \tag{5.12}$$

5.1.2 Numerical integration

Integrals (5.11) and (5.12) are evaluated by Gaussian integration which replaces the integrals by a weighted sum of the integrand's values taken at discrete points $\mathbf{x}_G$, the so-called *Gauss points*. Therefor, Finite Element schemes compute a cell decomposition of the original geometry. That is, the domain Ω^d is subdivided into a set of disjoint elements Ω_e^d, each of which can be mapped to the biunit domain $\Omega_G^d = [-1,+1]^d$. The actual integration is then performed for each element on this biunit domain by means of a change of variables. In that respect Isogeometric Analysis differs from isoparametric Finite Element methods: NURBS geometries are based on a parametric domain $\bar{\Omega}^d$, as illustrated in Figure 5.1. This domain consists of rectangular/hexahedral segments that can be mapped directly to Ω_G^d—rendering the aforementioned cell decomposition obsolete. In other words, the parametric domain can be used for numerical integration without preprocessing. It is this factor which allows to carry over the geometric fidelity from the source model to the numerical model.

NURBS basis functions are rational polynomial functions. Gaussian integration therefore results, allows, speaking, only approximate integration of the stiffness and load inte-

grals. Hughes et al. (2005) argue, however, that the NURBS basis functions converge to polynomial B-Spline functions, given sufficient mesh refinement. This is due to the affine combinations during knot refinement operations that "smoothen out" the difference in the control points' weights. To put it differently: while Gaussian integration remains insufficient on the coarsest mesh, its use can be justified on meshes that have been sufficiently refined so as to provide reasonable, smooth solutions (Cottrell, 2007). Accordingly, the number of integration points can be determined from the polynomial degrees of the respective univariate B-Spline functions constituting the NURBS basis.

Two coordinate transformations are necessary for the evaluation of (5.11) and (5.12): the integrands are expressions of the parametric coordinates $\boldsymbol{\xi}$ and they contain partial derivatives with respect to the physical coordinates $\mathbf{x}$. As a consequence of the discretization, the latter reduce to derivatives of the employed shape functions. These can, in turn, be computed by

$$\begin{bmatrix} R_{I,x} \\ R_{I,y} \\ R_{I,z} \end{bmatrix} = \begin{bmatrix} \xi_{,x} & \eta_{,x} & \zeta_{,x} \\ \xi_{,y} & \eta_{,y} & \zeta_{,y} \\ \xi_{,z} & \eta_{,z} & \zeta_{,z} \end{bmatrix} \begin{bmatrix} R_{I,\xi} \\ R_{I,\eta} \\ R_{I,\zeta} \end{bmatrix} = \mathbf{J}_{\mathcal{N}}^{-1} \begin{bmatrix} R_{I,\xi} \\ R_{I,\eta} \\ R_{I,\zeta} \end{bmatrix}, \tag{5.13}$$

which requires inversion of the *Jacobian matrix* of the geometric mapping

$$\mathcal{N} : \bar{\Omega}^d \rightarrow \Omega^d, \tag{5.14}$$

that, in turn, is defined by

$$\mathbf{J}_{\mathcal{N}} = \begin{bmatrix} x_{,\xi} & y_{,\xi} & z_{,\xi} \\ x_{,\eta} & y_{,\eta} & z_{,\eta} \\ x_{,\zeta} & y_{,\zeta} & z_{,\zeta} \end{bmatrix}, \tag{5.15}$$

with the row vectors simply being the tangent vectors of the NURBS mapping at a given parametric coordinate $\boldsymbol{\xi}$:

$$\begin{aligned} \left[x_{,\xi}, y_{,\xi}, z_{,\xi}\right]^T &= \sum_{I=0}^{T-1} R_{I,\xi} \mathbf{P}_I, \\ \left[x_{,\eta}, y_{,\eta}, z_{,\eta}\right]^T &= \sum_{I=0}^{T-1} R_{I,\eta} \mathbf{P}_I, \\ \left[x_{,\zeta}, y_{,\zeta}, z_{,\zeta}\right]^T &= \sum_{I=0}^{T-1} R_{I,\zeta} \mathbf{P}_I. \end{aligned} \tag{5.16}$$

In order to obtain appropriate parametric coordinates $\boldsymbol{\xi}$ for the evaluation of basis functions and derivatives, the coordinates of the Gauss points $\mathbf{x}_G$ are mapped to the current *nonzero* knot span $[\xi_i, \xi_{i+1}) \times [\eta_j, \eta_{j+1}) \times [\zeta_k, \zeta_{k+1})$ through the mapping $\mathcal{G} : \Omega_G^d \rightarrow \bar{\Omega}^d$:

$$\mathcal{G}: \quad \boldsymbol{\xi} = \frac{1}{2} \begin{bmatrix} (\xi_{i+1} - \xi_i)\, s + (\xi_i + \xi_{i+1}) \\ (\eta_{j+1} - \eta_j)\, t + (\eta_j + \eta_{j+1}) \\ (\zeta_{k+1} - \zeta_k)\, u + (\zeta_k + \zeta_{k+1}) \end{bmatrix},$$

where s, t, u denote the Gauss point coordinates within Ω_G^d. This mapping has the Jacobian

$$\mathbf{J}_{\mathscr{G}} = \begin{bmatrix} \xi_{,s} & \eta_{,s} & \zeta_{,s} \\ \xi_{,t} & \eta_{,t} & \zeta_{,t} \\ \xi_{,u} & \eta_{,u} & \zeta_{,u} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} (\xi_{i+1} - \xi_i) & 0 & 0 \\ 0 & (\eta_{j+1} - \eta_j) & 0 \\ 0 & 0 & (\zeta_{k+1} - \zeta_k) \end{bmatrix}.$$

The combined change of variables transforms the stiffness matrix integral into

$$\mathbf{K} = \int_{\Omega_G} \mathbf{B}(\xi(\mathbf{x}_G))^T \mathbf{C}\, \mathbf{B}(\xi(\mathbf{x}_G)) \det(\mathbf{J}_{\mathscr{N}} \mathbf{J}_{\mathscr{G}})\, \mathrm{dV}. \tag{5.17}$$

An analog expression results for the integral of the load vector.

5.2 Quality metrics for isogeometric models

In order for the stiffness integral (5.17) and the corresponding load integral to make sense, the transformation relations (5.14) must be well-defined and unique. That is, the bijectivity of the NURBS mapping $\mathscr{N}$ must be guaranteed (Xu et al., 2011b). A simple test for this property is proposed in (Joy and Duchaineau, 1999; Xu et al., 2010, 2011a,b). To that end the tangent vectors of the NURBS mapping, (5.16), are approximated in terms of difference vectors $\mathbf{P}_I - \mathbf{P}_J$, $I \neq J$. These vectors are combined into so-called tangent cones corresponding to the respective partial derivatives, and overlapping cones indicate self-overlap of the mesh. This criterion, however, allows only patch-global statements on possible mesh distortions. It does neither allow to locate the source of this disturbance nor is it suitable for a comparison of non-degenerate meshes. Furthermore, it might lead to the rejection of a model even when the mesh distortion is only localized or when it could be removed. As a consequence, mesh quality indicators considering local mesh properties are necessary. In what follows, these indicators will be referred to as *metrics*[1]. They are intended to reflect the properties of a NURBS model with respect to quantities such as smoothness, regularity, or element aspect ratios.

Finite element procedures offer various criteria for evaluating the quality of a mesh. These criteria assess the interpolation quality of an element based on its shape, for instance on its aspect ratios or its interior angles (Cohen et al., 2010). Thresholds are imposed for these quantities in order to ensure well-defined transformations from the elements' parent domains to physical space. Unfortunately, these criteria are designed for multilinear shape approximations. NURBS geometries in contrast offer smooth, higher-order shape descriptions; the NURBS equivalent of the element mesh can hence contain curved elements. It is consequently doubtful whether such "linear" mesh quality estimators can be applied for the assessment of NURBS models.

Geometric criteria are only a part of what determines the quality of a mesh for numerical simulation. Even the Jacobian matrix $\mathbf{J}_{\mathscr{N}}$ is only one factor that affects the quality of an isogeometric mesh: models that are deemed optimal—judged by the Jacobian

[1] This designation should not be confused with the mathematical concept of a "metric" that is used to measure distances and angles in metric spaces.

determinant—can perform differently for a given task (Cohen et al., 2010). The quality of a mesh depends, in addition to the shape of its elements, on the considered mathematical model, the characteristics of the desired solution, and on the employed discretization method (Berzins, 1998; Knupp, 2007). Individual elements nevertheless have a large influence as they affect the error in the gradients of the solution field and the condition number of the resulting stiffness equations (Shewchuck, 2002). A suitable mesh must therefore be considered to be a sufficient condition for meaningful simulation results. That is, a poor mesh can at least complicate the solution process (Knupp, 2007). It is consequently of great interest whether existing mesh quality criteria can be employed for the assessment of NURBS models, and which magnitudes of the respective criteria indicate an ill-suited model.

5.2.1 Formulation of metrics

Mappings between different spaces are the essence of both NURBS-based shape representations and Isogeometric Analysis, as could be seen in Subsection 5.1.2. In particular the properties of these mappings should therefore serve as the basis for isogeometric mesh quality indicators. Fundamental element in these mappings is the Jacobian matrix of the NURBS mapping, (5.15), respectively its determinant, which is therefore chosen as a basic metric:

$$f_{det} = \det(\mathbf{J}_{\mathcal{N}}). \tag{5.18}$$

This quantity reflects the transformation of the parametric domain $\bar{\Omega}^d$ under the geometric mapping and is hence affected by the steps taken during creation of the geometric model. This metric becomes zero for singular points, where the bijectivity of the mapping is lost. Negative values likewise indicate inverted elements with a negative volume—both cases indicate invalid regions of a mesh.

The Jacobian can further be used to construct a series of metrics such as given in (Knupp, 2000). Of these, the condition number of the Jacobian,

$$f_{cond} = \kappa(\mathbf{J}_{\mathcal{N}}) = \|\mathbf{J}_{\mathcal{N}}\|_F \cdot \|\mathbf{J}_{\mathcal{N}}^{-1}\|_F, \tag{5.19}$$

seems to be the most suitable as it provides a non-dimensional, symmetric, and scale-independent criterion (Knupp, 2000), ensuring applicability to models on different size scales. For its evaluation the *Frobenius norm* is employed,

$$\|\mathbf{J}_{\mathcal{N}}\|_F = \sqrt{\sum_{i=1}^{3}\sum_{j=1}^{3}|J_{ij}|^2},$$

where the J_{ij} denote the entries of the Jacobian matrix. A similarly promising metric derived from the Jacobian is described by Oddy et al. (1988). It is based on an analogy between mesh distortions and mechanical strains. To that end, the Jacobian matrix is used to set up the *right Cauchy-Green deformation tensor*. Starting from this tensor, Oddy et al. determine the infinitesimal strain tensor, from which the deviatoric strains can be

extracted. The second tensor invariant of the result yields the metric. In the form given in (Knupp, 2000), the "Oddy metric" reads

$$f_{Oddy}(\mathbf{J}_{\mathcal{N}}) = \det(\mathbf{J}_{\mathcal{N}})^{-4/3} \left\{ \|\mathbf{J}_{\mathcal{N}}^T \mathbf{J}_{\mathcal{N}}\|_F^2 - \frac{1}{3}\|\mathbf{J}_{\mathcal{N}}\|_F^4 \right\}. \tag{5.20}$$

Brakhage and Lamby (2008) mention, in addition, the orthogonality of the isogeometric mesh lines as a beneficial mesh property in fluid flow problems. This claim is supported by Cohen et al. (2010) who observed notable differences in the model response for meshes that violate this criterion to different extents. They do not, however, specify a metric for this criterion. For its construction, the scalar products of the normalized tangent vectors (5.16) are employed:

$$f_{\perp} = \left(1 - \left|\frac{\mathbf{x}_{,\xi}}{\|\mathbf{x}_{,\xi}\|_2} \cdot \frac{\mathbf{x}_{,\eta}}{\|\mathbf{x}_{,\eta}\|_2}\right|\right) \times \left(1 - \left|\frac{\mathbf{x}_{,\xi}}{\|\mathbf{x}_{,\xi}\|_2} \cdot \frac{\mathbf{x}_{,\zeta}}{\|\mathbf{x}_{,\zeta}\|_2}\right|\right) \times \left(1 - \left|\frac{\mathbf{x}_{,\eta}}{\|\mathbf{x}_{,\eta}\|_2} \cdot \frac{\mathbf{x}_{,\zeta}}{\|\mathbf{x}_{,\zeta}\|_2}\right|\right). \tag{5.21}$$

Expressions (5.18) to (5.21) allow to recognize the values that the metrics take on for both undistorted and invalid meshes, the former case being given by the identity transformation $\mathbf{J}_{\mathcal{N}} = \mathbf{I}$. The Jacobian determinant accordingly becomes one. It goes to zero for singular mappings but can otherwise take on any value—positive and negative—for general meshes. The condition of the Jacobian of an ideal, undistorted mesh is 3 as a consequence of the employed Frobenius norm. This metric goes to infinity at singular points, the same as the Oddy metric which, in turn, becomes zero for ideal, undistorted meshes. The orthogonality criterion takes on values between zero and one for degenerated and undistorted meshes, respectively.

In the presence of singularities some of the tangent vectors in (5.21) might become undefined, in which case only the Jacobian determinant can be determined; the other metrics then cannot be evaluated numerically. For visualization purposes this problem can be overcome by taking regular samples over the domain of the geometry. Upon encounter of a singular point, the metrics can be simply set to their respective optimal values or, in the case of the orthogonality metric, to zero. The presence of the singularity will nonetheless be indicated by neighbor samples. This is equivalent to "cutting off" the metrics at the nearest point to a singularity in the geometric mapping.

Two strategies can be pursued for the evaluation of NURBS models, the reasons for which becoming apparent in the following subsections. One of these strategies is a qualitative analysis that employs the aforementioned regular sampling over the model's domain. It will be described in the following subsection where a set of sample models illustrate the metrics' general behavior. Furthermore, their application for the comparison of model variants is highlighted. An analysis of the numerical properties, however, relies on the evaluation of the metrics at the model's Gauss points. This will be done in Subsection 5.2.3 where an attempt is made to link given values of these metrics to the properties of the numerical model.

5.2.2 Qualitative mesh quality analysis

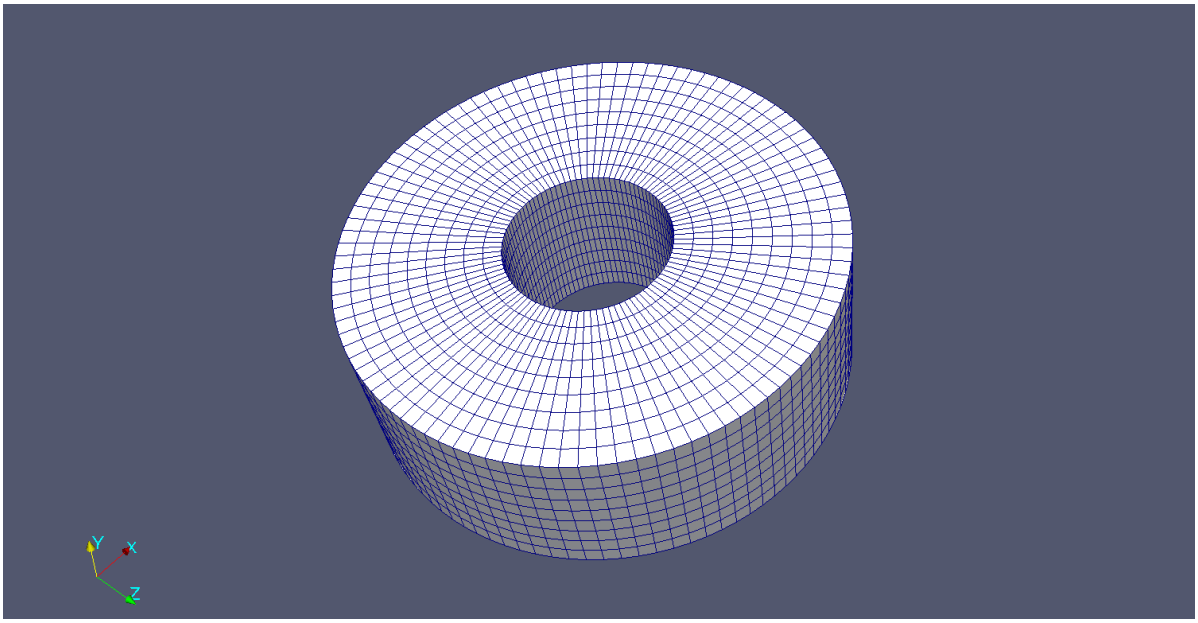

Figure 5.2: Isogeometric mesh of a circular annulus with rectangular cross-section. The mesh is the result of revolving a rectangular surface lying in the xy-plane around the z-axis by 360°. The coarsest model (which nonetheless *exactly* describes this geometry) requires a mere 36 control points. The parametric domain is described by the knot vectors $\Xi = \mathbf{H} = \{0,0,1,1\}$, and $\mathbf{Z} = \{0,0,0,1,1,2,2,3,3,4,4,4\}$. That is, the mesh is comprised in circumferential direction of four parametric segments that are joined with C^1-continuity.

According to Finite Element heuristics, meshes with regular, uniform element size distribution are well-suited for numerical applications. As Isogeometric Analysis shares its mathematical foundations with classical FEM schemes, the same behavior should hold, accordingly, for isogeometric meshes. To that end, the annulus model illustrated in Figure 5.2 is taken as the first example for the application of the mesh quality metrics. The results are depicted in Figure 5.3, which clearly confirms the impression of a high-quality mesh: each of the shown metrics takes on values at or near their respective optima and the mesh lines are orthogonal throughout the whole model.

NURBS geometries provide, in general, smooth shape representations for a wide range of geometries. In order to obtain features such as kinks or sharp edges, the geometric mapping from the parametric to the physical domain has to be suitably modified. A prominent example for this is rectangular panel with a circular cut-out, illustrated in Figure 5.4. This model represents a common benchmark in structural mechanics, namely an infinite plane with a traction-free, circular interior boundary. The mesh data for this model is described in (Hughes et al., 2005). It is used as a landmark example for demonstrating the flexibility of isogeometric meshes: the geometry of this model can be described exactly by a single NURBS patch. The corresponding mapping of its rectangular parametric domain

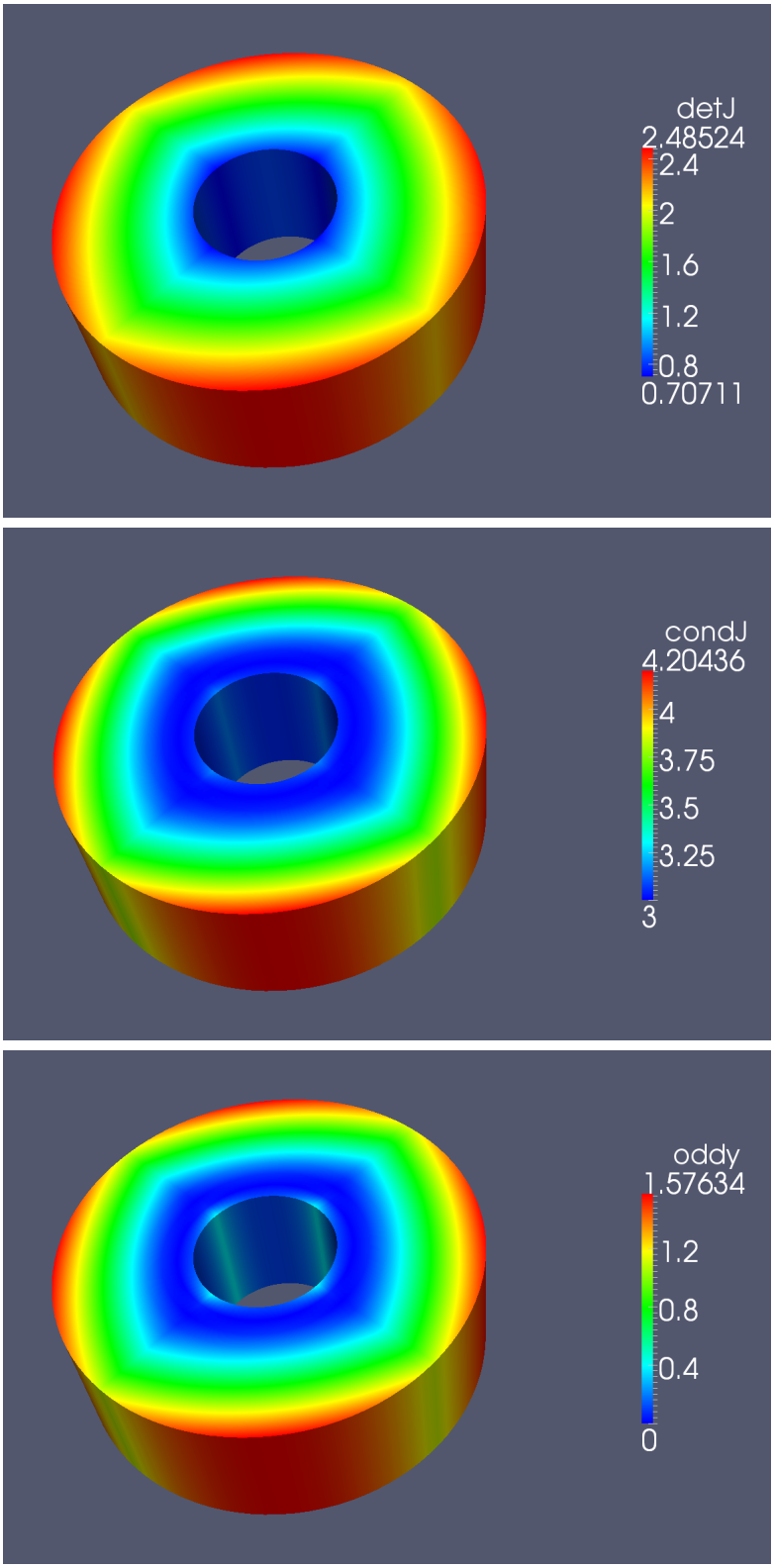

Figure 5.3: Mesh quality metrics for the circular annulus in Figure 5.2. Only three out of the four metrics are illustrated since the mesh is fully orthogonal throughout the domain. The shown metrics deviate only slightly from their optimal values, thereby confirming the good quality of the mesh. Within each segment the mesh is slightly widened, indicated by growing metric values on the exterior boundary surfaces. This allows to recognize the combination of the mesh from four segments.

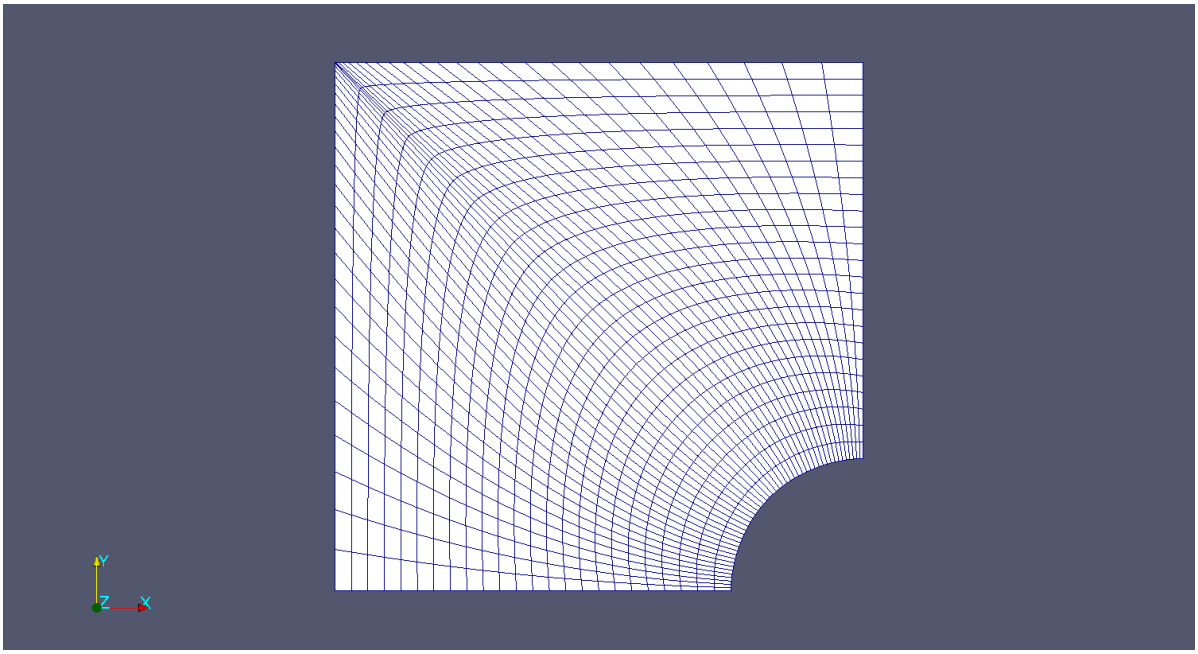

Figure 5.4: Isogeometric mesh of a rectangular panel with circular hole. The parametric domain is described by the knot vectors $\Xi = \{0,0,0,1,2,2,2\}$, and $\mathbf{H} = \{0,0,0,1,1,1\}$.

into physical space is illustrated in Figure 5.1 on page 46. In order to obtain the corner in the upper left-hand corner, the corresponding control points are made to coincide, thereby introducing a singularity into the NURBS mapping. As will be seen in Figure 5.6, this singularity "pollutes" the surrounding mesh, despite being restricted to the very corner point of the panel.

Said singularity can be removed by a split of the panel into two individual patches. NURBS geometries possess, as demonstrated before, only a simple topology. In order to describe complex geometries without excessive mesh distortions, a decomposition into components is necessary. Therefor, two approaches are at hand, namely a *top-down* and a *bottom-up* approach. The first method attempts to subdivide the given object into components for which a parametrization might be found more easily. The latter ansatz, however, attempts to determine components with a known parametrization from which the desired shape can be composed. The template modeling concept described in Chapter 3 turns out to be well-suited for such a course of action as it allows to define the components of a complex model in a compact form—for which, in turn, the mesh quality can be ensured more easily than for a global model (Kolšek et al., 2003).

Decomposition of the panel can be achieved by knot insertion into the knot vector Ξ which reduces the continuity from C^1 to C^0 across the isoparametric line $\xi = 1$. The resulting control point configuration then allows the setup of two individual patches, leading to the mesh shown in Figure 5.5. The effect of this decomposition can be readily observed in 5.6, where one can recognize the metrics' shift towards their respective optimal values.

Meshes with singular points reveal, regardless of these observations, a problem with

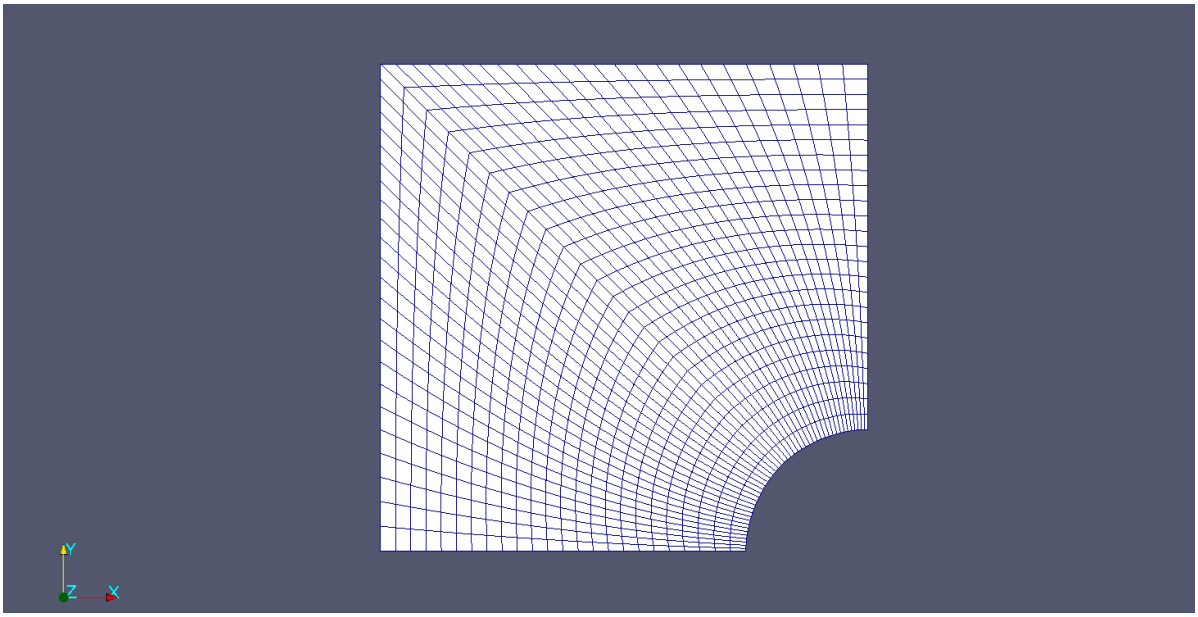

Figure 5.5: Isogeometric mesh of the rectangular panel with circular hole, split into two separate patches. Within the patches' respective domains the basis functions are C^1-continuous biquadratic functions. The patches are visibly joined with C^0-continuity along the diagonal.

the unbounded metrics f_{cond} and f_{Oddy}. Both metrics grow to infinity near singular points in the mesh. Plots for the visual inspection of a model's quality can be produced, however, only with a finite step size. Depending on the sample points' distance to the singularity, basically arbitrary values for these metrics can be produced, which clearly runs contrary to the purpose of a metric. Such singular values blur, in addition, the information on the mesh quality in other regions of the model. In particular the Oddy metric is prone to this effect as can be seen in Figure 5.6. A qualitative model analysis nonetheless gives information on the location of a singularity, even in a complex mesh. This can be recognized in Figure 5.7 that shows a segment of the fluid domain modeled in Chapter 4 that is located along the outer portion of the airfoil. Here, the metrics indicate an *almost* singular point at the leading edge of the profile. The cause of this singularity can be recognized in Figure 4.10 on page 40. It is the linear interpolation of the section curve with the outer arc of the fluid domain template that introduces a so-called *mesh folding*, a (near) self-overlap of the isogeometric mesh. This observation gives solid proof that even seemingly good meshes can exhibit regions that might be causing numerical problems. The effect of this mesh degeneration on the behavior of the numerical model is so far unknown; however, the preliminary results of the model's application in (Stein et al., 2012) are promising, hinting at only a minor influence.

5.2.3 Quantitative mesh quality analysis

Lipton et al. (2010) observed a higher robustness of isogeometric meshes against distor-

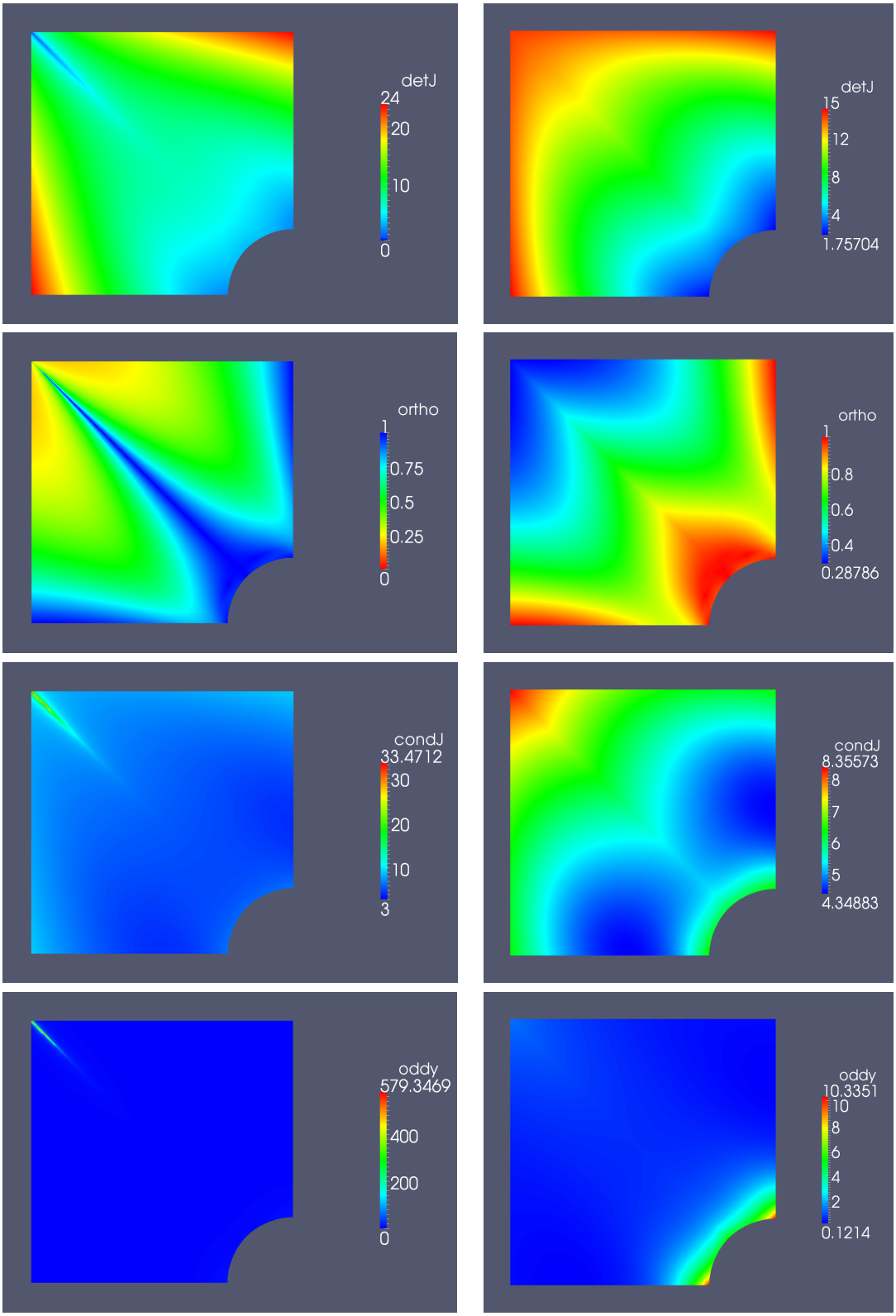

Figure 5.6: Comparison of the mesh quality metrics for the monolithic and the split rectangular panel, shown in the left-hand and the right-hand column, respectively. The corresponding meshes are depicted in Figures 5.4 and 5.5. The color grading of the uppermost left-hand image is slightly adapted in order to highlight that along the circular boundary $\det(\mathbf{J}_{\mathcal{N}}) \approx 2$.

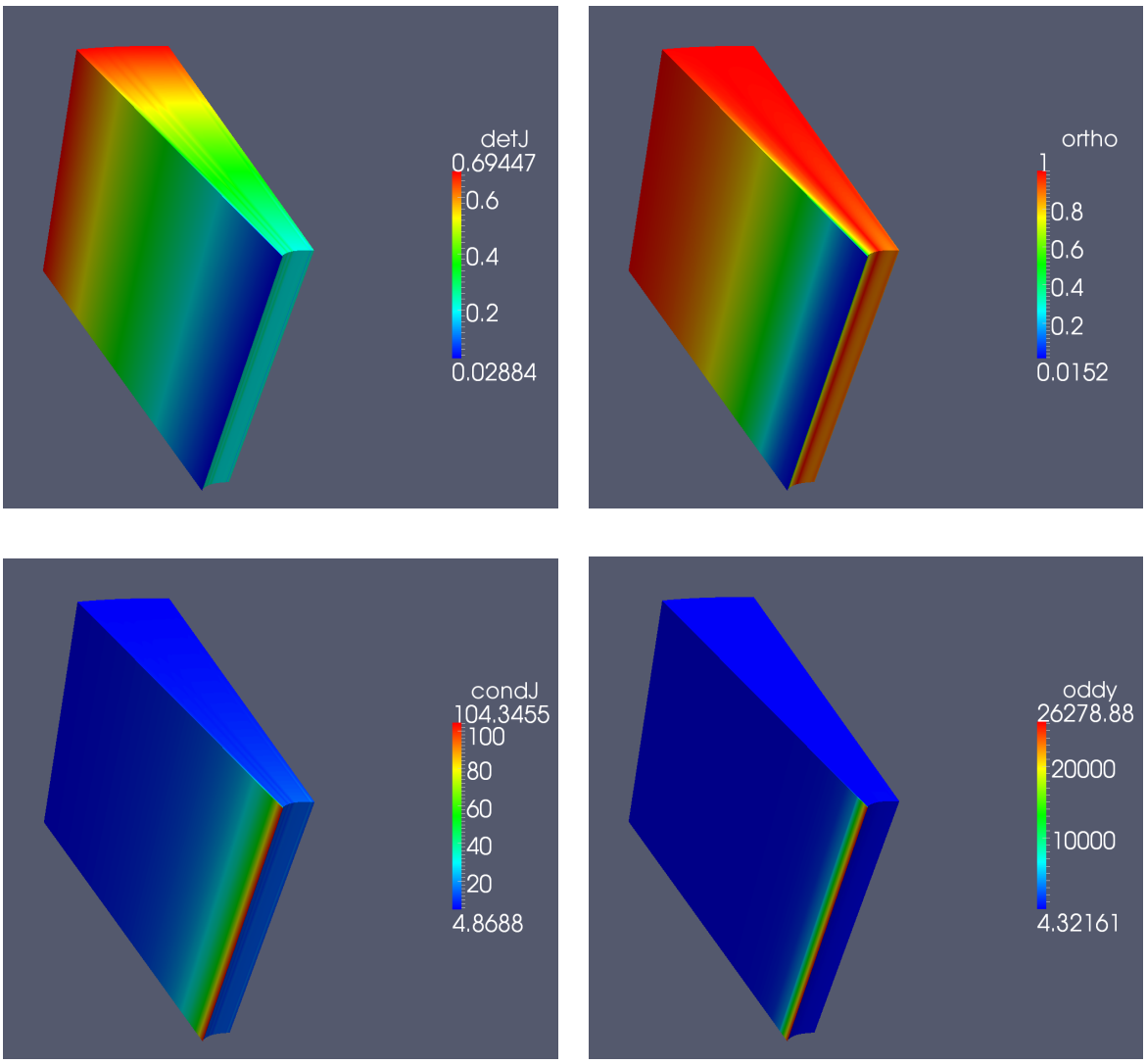

Figure 5.7: The mesh quality metrics plotted over a section of the fluid domain that has been described in Chapter 4. The metrics highlight a "mesh folding" effect at the leading edge of the profile. Different simplification levels of the rotor blade profile fail to produce a salutary effect. On the contrary, coarser approximations of the airfoil's geometry even lead to self-overlaps of the mesh.

tions (compared with lower-order Finite Element schemes), the effect of which increasing with the degree of the underlying basis functions. They attribute this to the *variation-diminishing* property of NURBS. That is, noisy geometry data and mesh irregularities are smoothened out by the interpolation in the geometric mapping. As a result, Isogeometric Analysis can give meaningful prognoses despite severe mesh distortions in the underlying NURBS models. Considering the mesh depicted in Figure 5.7, the following question arises: which magnitude of the respective metrics can be observed in a distorted mesh that is still suitable for numerical simulation?

In order to establish such a relation, a linear-elastic *patch test* as described in (Lipton et al., 2010) is used. Purpose of such tests is to ascertain that a given Finite element is able to reproduce constant strain states evolving under predefined boundary conditions, which in turn allows to make statements about the convergence of the numerical solution.

The employed geometric model is a simple unit cube with an isotropic ansatz; for this test, linear to cubic basis functions are used. Suitable knot insertion ensures a uniform split of the mesh into four elements along each parametric direction. The displacements of the control points at the bottom of the cube are kept fixed whereas the points comprising the top surface are subject to a uniform displacement in vertical direction, resulting in a linearly varying displacement field over the cube's height. Consequently, a constant strain state must evolve along the vertical direction.

The stiffness matrix and the load vector of the numerical model are set up in accordance with Section 5.1. Therefor a Young's modulus of $E = 10^4$ MPa is employed along with a Poisson's ration of $\nu = 0.2$. The quantity of interest in this study is the *condition number* $\kappa(\mathbf{K})$ of the stiffness matrix. It reflects the sensitivity of the equation system with respect to perturbations in the input data, namely the load vector. The condition number further affects the accuracy of the equations' solution as well as the speed of convergence for iterative equation solvers. This number depends on the contributions of the individual Gauss points to the stiffness integral and is directly affected by mesh distortions through the Jacobian matrix. It is hence taken as a representation for the quality of a numerical model originating from a given mesh. For the considered problem, and given appropriate boundary conditions, the stiffness matrix is symmetric and positive definite. As a result, the stiffness matrix possesses positive real eigenvalues and its condition number can be determined by the ratio of its largest to its smallest eigenvalues λ_n and λ_1, respectively:

$$\kappa(\mathbf{K}) = \frac{\lambda_n}{\lambda_1}.$$

The test model, including the geometry and the boundary conditions, is formulated in terms of a template. Mesh properties and the condition number can hence be readily determined for various degrees of mesh degeneration—which are basically model variants for the initial, undistorted unit cube. The mesh distortion is achieved by collapsing a subset of control points onto a single point. Therefor a $2 \times 2 \times 2$-set of control points is selected from the grid of control points. The center of the bounding box encompassing the selection is chosen as "singular point" $\mathbf{x}_c$. The distortion is applied to the selected control points by means of a combined affine transformation

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & x_c \\ 0 & 1 & 0 & y_c \\ 0 & 0 & 1 & z_c \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} (1-d) & 0 & 0 & 0 \\ 0 & (1-d) & 0 & 0 \\ 0 & 0 & (1-d) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & -x_c \\ 0 & 1 & 0 & -y_c \\ 0 & 0 & 1 & -z_c \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where $d \in [0, 1]$ denotes the degree of degeneration. This transformation scales the coordinates of the selected control points relative to the point $\mathbf{x}_c$.

d	f_{dist}	$f_{det,min}$	$f_{det,max}$	$f_{cond,max}$	$f_{\perp,min}$	$f_{Oddy,max}$	$\kappa(\mathbf{K})$
0.0	0.000	1.000	1.000	3.000	1.0000	0.0000	183.519
0.1	0.004	0.729	1.118	3.030	0.9142	0.0665	182.803
0.2	0.008	0.512	1.236	3.127	0.8348	0.3166	183.068
0.3	0.012	0.343	1.354	3.307	0.7621	0.8756	184.127
0.4	0.017	0.216	1.472	3.598	0.6960	1.9910	185.888
0.5	0.021	0.125	1.590	4.047	0.6363	4.1870	188.209
0.6	0.025	0.064	1.709	4.735	0.5825	8.6860	191.149
0.7	0.029	0.027	1.827	5.826	0.5342	18.7700	194.887
0.8	0.034	0.008	1.945	7.681	0.4908	45.1500	199.855
0.9	0.038	0.001	2.063	11.300	0.4519	136.5000	207.229
1.0	0.042	0.000	2.181	20.830	0.4168	∞	220.944

Table 5.1: Overview of the mesh quality metrics for the patch test using a trilinear basis.

The mesh metrics are evaluated at the Gauss points of the isogeometric elements—contrary to the qualitative analysis of the prior section that is based on regularly spaced sample points. Whereas the latter yields vital information for modeling purposes, it produces an overhead of data for numerical simulation. That is, it is not the distribution of the mesh properties that is of interest but the deformation level at which the metrics possibly indicate an invalid numerical model. For each model variant and distortion level d both the condition number and the Gauss point values of the metrics are computed, of which the respective extremal values are extracted.

In addition to these values, the geometric deviation is computed for each model instance as the distortion level d provides only an abstract quantity. To that end, the coordinates $\mathbf{x}(\xi)$ are evaluated for each model at regular sample points and are stored contiguously in a vector $\mathbf{s}$. These points are compared with the points corresponding to an undistorted mesh. The relative error

$$f_{dist} = \frac{||\mathbf{s}_{orig} - \mathbf{s}_{distorted}||_2}{||\mathbf{s}_{orig}||_2} \tag{5.22}$$

between the samples of different models then describes the deviation between the "original" and the distorted mesh.

The results of the study are given in Tables 5.1-5.3 for the respective order of the basis functions. They allow the following observations:

1. Both the distortion measure f_{dist} and the maximum Jacobian determinant grow linearly for all three orders of basis functions. The higher the ansatz of the multivariate basis the lower the increase in mesh deformation.

2. The amount of mesh distortion is surprisingly small: in the linear case the distortion metric indicates a mere 4% deviation from the original mesh, despite the collapse of eight control points onto one and the introduction of singular points. Likewise,

d	f_{dist}	$f_{det,min}$	$f_{det,max}$	$f_{cond,max}$	$f_{\perp,min}$	$f_{Oddy,max}$	$\kappa(\mathbf{K})$
0.0	0.0000	1.0000	1.000	3.000	1.0000	0.0000	224.238
0.1	0.0022	0.7735	1.049	3.012	0.9413	0.0253	223.114
0.2	0.0044	0.5841	1.096	3.045	0.8873	0.1026	222.343
0.3	0.0066	0.4284	1.141	3.097	0.8378	0.2355	221.788
0.4	0.0088	0.3032	1.185	3.167	0.7925	0.4350	221.375
0.5	0.0110	0.2051	1.226	3.272	0.7508	0.7646	221.057
0.6	0.0132	0.1308	1.266	3.409	0.7069	1.2550	220.809
0.7	0.0154	0.0771	1.304	3.585	0.6605	1.9760	220.613
0.8	0.0176	0.0405	1.340	3.810	0.6154	3.0360	220.457
0.9	0.0198	0.0178	1.375	4.094	0.5715	4.6120	220.334
1.0	0.0220	0.0057	1.407	4.557	0.5291	7.7320	220.238

Table 5.2: Overview of the mesh quality metrics for the patch test using a triquadratic basis.

d	f_{dist}	$f_{det,min}$	$f_{det,max}$	$f_{cond,max}$	$f_{\perp,min}$	$f_{Oddy,max}$	$\kappa(\mathbf{K})$
0.0	0.0000	1.0000	1.000	3.000	1.0000	0.0000	1550.905
0.1	0.0006	0.8986	1.027	3.005	0.9637	0.0100	1550.549
0.2	0.0013	0.8043	1.053	3.019	0.9286	0.0403	1550.946
0.3	0.0020	0.7168	1.079	3.041	0.8946	0.0917	1551.680
0.4	0.0027	0.6359	1.105	3.070	0.8618	0.1652	1552.524
0.5	0.0034	0.5613	1.130	3.107	0.8300	0.2621	1553.370
0.6	0.0041	0.4927	1.154	3.150	0.7994	0.3843	1554.166
0.7	0.0048	0.4299	1.179	3.201	0.7700	0.5337	1554.883
0.8	0.0055	0.3727	1.203	3.257	0.7416	0.7122	1555.509
0.9	0.0062	0.3208	1.226	3.319	0.7143	0.9224	1556.036
1.0	0.0069	0.2738	1.249	3.387	0.6881	1.1670	1556.461

Table 5.3: Overview of the mesh quality metrics for the patch test using a tricubic basis.

one can observe a deviation of 2% and 0.69% in the quadratic and cubic models, respectively.

3. A significant change in the condition number $\kappa(\mathbf{K})$ of the stiffness matrix can only be observed in the linear case, where κ grows by approximately 20 %. Whereas the condition number remains basically equal for the cubic basis functions, it decreases for the quadratic basis.

4. With increasing distortion both the f_{cond} and the f_{Oddy} metric grow rapidly in the linear model whereas in the models with quadratic and cubic ansatz functions the remain relatively small.

5. Even in the fully collapsed linear model the mesh retains some orthogonality, as

indicated by $f_{\perp}$. Likewise, in the quadratic model the tangent vectors seem to retain their orthogonality to a large part.

6. In the quadratic and cubic model the mesh metrics' values remain close to their optimal values. Significant changes can only be observed with $f_{det,min}$. Neither of these models exhibits Gauss points with a singular geometric mapping.

Based on these observations it has to be concluded that the initial hypothesis is wrong. That is, the mesh distortions are not reflected in the condition number of the patch tests' stiffness matrices. Instead, the condition numbers remain stable and within ranges that usually do not suggest ill-posed numerical models. Their variation must therefore depend on other, so far unknown factors. This warrants further studies, for instance with respect to the degree of the ansatz functions.

The orthogonality metric $f_{\perp}$ retains a nonzero value even in the fully degenerated linear model. This might be attributed to an error in the evaluation routines. However, this metric behaves similar in the quadratic and cubic models which indicates otherwise. Considering that only the linear model fails the patch test, this can only mean that this metric possesses only a minor prognostic value for problems of linear elastostatics. The $f_{det,min}$, $f_{cond,max}$ and f_{Oddy} metrics, in contrast, appropriately reflect the higher robustness of isogeometric models with respect to mesh distortions, as noted by Lipton et al. (2010). Their application to the fluid domain segment shown in Figure 5.7 explains the fact that the model has been applied successfully for fluid flow simulations: as in all Finite Element schemes, the Gauss points are slightly offset from the boundaries of a given element. That is, they do not overlap with the mesh folding; the corresponding mesh metrics hence take on values near their optimal range.

6 Discussion

This work contains an approach for the generation of numerical models from geometry. It is based on the concept of Isogeometric Analysis and makes use of NURBS models. These provide a common data structure for the various stages of Computer-Aided Engineering, from geometric modeling over numerical simulation to visualization. It makes them invaluable in overcoming the limitations of the current design-through-analysis pipeline.

Key idea of the proposed modeling approach is a procedural representation of the modeling process. That is, instead of describing the raw model data, the evolution of the geometry from a simple initial shape is represented in terms of modeling operations. This provides an abstraction from low-level data structures and allows a model treatment through meaningful parameters. The proposed concept integrates existing modeling approaches such as template modeling, surface expansion, and parameter extraction. Of these, the concept of template modeling proves to be complementary to the operator-based representations. That is, templates can not only serve as input data to sequences of modeling commands, but can also be fully defined in terms of modeling operations. Complex, multi-level adaptive and reusable geometries can then be realized by nesting templates such that the result of one template is used as input for others.

Operator- and template-based model representations lead to a trade-off between model size and evaluation time. In order to employ procedurally described NURBS models, they have to be evaluated first. However, the evaluation times observed so far are insignificant: the fluid domain model of Chapter 4 could be generated in the fraction of a second on a desktop computer. In face of increasingly powerful hardware this should also hold true for deeply nested template hierarchies. Further improvements might even be possible by parallelizing the evaluation of independent components.

NURBS models exhibit some topological restrictions that limit their refinability and the range of shapes that can be modeled by means of a single patch. A common method to circumvent these problems is the decomposition of a shape into simpler components. This requires additional coupling conditions in the equation system in order to ensure consistent behavior of the numerical model. Current approaches require (semi-)conformal parametrizations of the components along their interfaces. In order to allow greater flexibility for shape modeling and model adaption concepts such as *mortar methods* or *weak coupling conditions* should be investigated.

Alternatively, one could overcome the limitations of NURBS by means of *T-Spline* representations which have been described in (Bazilevs et al., 2010; Sederberg et al., 2003). It is hence of great interest whether the concepts described herein can be extended such as to allow generation of volumetric T-Spline geometries.

Procedural representations possess, despite their numerous advantages, restrictions concerning the order of model evaluation and the addition of constraints among parameters (cf. Shah and Mäntylä, 1995). A possible solution might here be given with feature-

based model representations and their adaption to NURBS or T-Spline geometries.

Geometric design for Isogeometric Analysis requires tools for the assessment of a model's suitability for numerical analyses. To that end a series of mesh quality metrics have been studied in this thesis. These reflect the properties of the geometric mapping underlying the NURBS geometry and are therefore suitable for model comparison. Furthermore, they allow to identify and to locate problematic regions within complex models. Using this information, model adaptions can be initiated and evaluated with respect to their efficacy. However, a link between these metrics and the analysis-suitability of a given model, represented by the condition number of its stiffness matrix, could not be established. Here, further studies regarding the numerical behavior of NURBS basis functions and isogeometric discretizations are required.

List of Figures and Tables

Bibliography

Aigner, M., C. Heinrich, B. Jüttler, E. Pilgerstorfer, B. Simeon, and A.-V. Vuong (2009). Swept Volume Parameterization for Isogeometric Analysis. In E. Hancock and R. Martin (Eds.), *The Mathematics of Surfaces XIII*, Volume 5654 of *Lecture Notes in Computer Science*, pp. 19–44. Springer.

Akenine-Möller, T., E. Haines, and N. Hoffman (2008). *Real-Time Rendering* (Third ed.). A K Peters.

Akkerman, I., Y. Bazilevs, V. Calo, T. Hughes, and S. Hulshoff (2008). The role of continuity in residual-based variational multiscale modeling of turbulence. *Comput. Mech. 41*, 371–378.

Aliaga, D. G., P. A. Rosen, and D. R. Bekins (2007). Style Grammars for Interactive Visualization of Architecture. *IEEE Transactions on Visualization and Computer Graphics 13*(4), 786–797.

Ames, A. L., J. J. Rivera, A. J. Webb, and D. M. Hensinger (1997). Solid Model Design Simplification. Technical Report SAND97-3141, Sandia National Laboratories.

Aschwanden, G., S. Haegler, J. Halatsch, R. Jecker, G. Schmitt, and L. Van Gool (2009). Evaluation of 3D City Models Using Automatic Placed Urban Agents. In X. Wang and N. Gu (Eds.), *CONVR 2009 Proceedings of the 9th International Conference on Construction Applications of Virtual Reality*, Sidney, Australia, pp. 165–176.

Bartels, R. H., J. C. Beatty, and B. A. Barsky (1987). *An Introduction to Splines for use in Computer Graphics & Geometric Modeling*. Morgan Kaufman Publishers, Inc.

Bathe, K.-J. (1996). *Finite Element Procedures*. Prentice Hall.

Bazilevs, Y., L. Beirão da Vega, J. Cottrell, T. Hughes, and G. Sangalli (2006). Isogeometric analysis: Approximation, stability and error estimates for h-refined meshes. *Mathematical Models and Methods in Applied Sciences 16*(7), 1031–1090.

Bazilevs, Y., V. Calo, J. Cottrell, J. Evans, T. Hughes, S. Lipton, M. Scott, and T. Sederberg (2010). Isogeometric analysis using t-splines. *Comput. Methods Appl. Mech. Engrg. 199*, 229–263.

Bazilevs, Y., V. Calo, J. Cottrell, T. Hughes, A. Reali, and G. Scovazzi (2007). Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows. *Comput. Methods Appl. Mech. Engrg. 197*, 173–201.

Bazilevs, Y., V. Calo, T. Hughes, and Y. Zhang (2008). Isogeometric fluid-structure interaction: theory, algorithms, and computations. *Comput. Mech. 43*, 3–37.

Bazilevs, Y., V. Calo, Y. Zhang, and T. Hughes (2006). Isogeometric fluid-structure interaction analysis with applications to arterial blood flow. *Comput. Mech. 38*, 310–322.

Bazilevs, Y., J. Gohean, T. Hughes, R. Moser, and Y. Zhang (2009). Patient-specific isogeometric fluid–structure interaction analysis of thoracic aortic blood flow due to implantation of the Jarvik 2000 left ventricular assist device. *Comput. Methods Appl. Mech. Engrg. 1998*, 3534–3550.

Bazilevs, Y., M.-C. Hsu, I. Akkerman, S. Wright, K. Takizawa, B. Henicke, T. Spielman, and T. T.E. (2011). 3D simulation of wind turbine rotors at full scale. Part I: Geometry modeling and aerodynamics. *Int. J. Numer. Meth. Fluids 65*, 207 – 235.

Bazilevs, Y. and T. Hughes (2008). NURBS-based isogeometric analysis for the computation of flows about rotating components. *Comput. Mech. 43*, 143–150.

Bazilevs, Y., C. Michler, V. Calo, and T. Hughes (2010). Isogeometric variational multiscale modeling of wall-bounded turbulent flows with weakly enforced boundary conditions on unstretched meshes. *Comput. Methods Appl. Mech. Engrg. 199*, 780–790.

Beall, M. W., J. Walsh, and M. S. Shephard (2003). Accessing CAD geometry for mesh generation. In *Proceedings of the 12th International Meshing Roundtable*, Santa Fe, New Mexico, pp. 33–42.

Benson, D., Y. Bazilevs, M.-C. Hsu, and T. Hughes (2010). Isogeometric shell analysis: The Reissner-Mindlin shell. *Comput. Methods Appl. Mech. Engrg. 199*, 276–289.

Benson, D., Y. Bazilevs, M.-C. Hsu, and T. Hughes (2011). A large deformation, rotation-free, isogeometric shell. *Comput. Methods Appl. Mech. Engrg. 200*, 1367–1378.

Berndt, R., D. W. Fellner, and S. Havemann (2004). Generative 3D Models: A Key to More Information within Less Bandwidth at Higher Quality. Technical Report TUBS-CG-2004-08, Institute of Computer Graphics, University of Technology, Braunschweig.

Berzins, M. (1998). Mesh quality: a function of geometry, error estimates or both. In *Seventh International Meshing Roundtable*, pp. 229–238.

Bokeloh, M., M. Wand, and H.-P. Seidel (2010). A connection between partial symmetry and inverse procedural modeling. *ACM Trans. Graph. 29*(4), 104:1–104:10.

Brakhage, K.-H. and P. Lamby (2008). Application of B-Spline Techniques to the Modeling of Airplane Wings and Numerical Grid Generation. *Comput. Aided Geom. Design 25*, 738–750.

Calo, V., N. Brasher, Y. Bazilevs, and T. Hughes (2008). Multiphysics model for blood flow and drug transport with application to patient-specific coronary artery flow. *Comput. Mech. 43*, 161–177.

Calo, V., H. Gomez, Y. Bazilevs, G. Johnson, and T. Hughes (2008). Simulation of Engineering Applications using Isogeometric Analysis. In *TeraGrid 08*, Las Vegas, Nevada.

Cho, S. and S.-H. Ha (2008). Isogeometric shape design optimization: exact geometry and enhanced sensitivity. *Structural and Multidisciplinary Optimization 38*(1), 53–70.

Choi, H. I. and C. Y. Han (2002). The Medial Axis Transform. In G. Farin, J. Hoschek, and M.-S. Kim (Eds.), *Handbook of Computer Aided Geometric Design*, pp. 451–472. North-Holland/Elsevier.

Cohen, E., T. Martin, R. M. Kirby, T. Lyche, and R. Riesenfeld (2010). Analysis-aware modeling: Understanding quality considerations in modeling for isogeometric analysis. *Comput. Methods Appl. Mech. Engrg. 199*(5-8), 334–356.

Cottrell, J. (2007). *Isogeometric analysis and numerical modeling of the fine scales within the variational multiscale method*. Ph. D. thesis, University of Texas, Austin.

Cottrell, J., T. Hughes, and A. Reali (2007). Studies of refinement and continuity in isogeometric structural analysis. *Comput. Methods Appl. Mech. Engrg. 196*, 4610–4183.

Cottrell, J., T. J. Hughes, and Y. Bazilevs (2009). *Isogeometric Analysis - Toward Integration of CAD and FEA*. Wiley.

Cottrell, J., A. Reali, Y. Bazilevs, and T. Hughes (2006). Isogeometric analysis of structural vibrations. *Comput. Methods Appl. Mech. Engrg. 195*, 5257–5296.

Cox, M. (1972). The numerical evaluation of B-Splines. *J. Inst. Maths Applics 10*, 134–149.

De Boor, C. (1972). On calculating with B-Splines. *Journal of Approximation Theory 6*, 50–62.

Ebert, D. S., F. K. Musgrave, D. Peachey, K. Perlin, and S. Worley (2003). *Texturing & Modeling: A Procedural Approach* (Third ed.). Morgan Kaufman Publishers.

Eck, M. and H. Hoppe (1996). Automatic reconstruction of B-spline surfaces of arbitrary topological type. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pp. 325–334.

Espino, F., M. Bóo, M. Amor, and J. Bruguera (2003). Adaptive tessellation of NURBS surfaces. *Journal of WSCG 11*, 2003.

Germer, T. and M. Schwarz (2009). Procedural Arrangement of Furniture for Real-Time Walkthroughs. *Computer Graphics Forum 28*(8), 2068–2078.

Haegler, S., P. Wonka, S. Müller Arisona, L. Van Gool, and P. Müller (2010). Grammar-based Encoding of Facades. In J. Lawrence and M. Stamminger (Eds.), *Eurographics Symposium on Rendering*, Volume 29.

Häfner, S., S. Eckardt, T. Luther, and C. Könke (2006). Mesoscale modeling of concrete: Geometry and numerics. *Computers and Structures 84*, 450–461.

Havemann, S. (2005). *Generative Mesh Modeling*. Ph. D. thesis, Technische Universität Braunschweig.

Hoffmann, C. M. and R. Joan-Arinyo (2002). Parametric Modeling. In G. Farin, J. Hoschek, and M.-S. Kim (Eds.), *Handbook of Computer Aided Geometric Design*, pp. 519–541. North-Holland/Elsevier.

Höllig, K. (2003). *Finite Element Methods with B-Splines*. Society for Industrial and Applied Mathematics.

Hsu, M.-C., I. Akkerman, and Y. Bazilevs (2011). High-performance computing of wind turbine aerodynamics using isogeometric analysis. *Computers & Fluids -*, –. doi:10.1016/j.compfluid.2011.05.002.

Hu, S.-M., Y.-F. Li, T. Ju, and X. Zhu (2001). Modifying the shape of NURBS surfaces with geometric constraints. *Computer-Aided Design 33*, 903–912.

Hughes, T., A. Reali, and G. Sangalli (2010). Efficient quadrature for NURBS-based isogeometric analysis. *Comput. Meth. Appl. Mech. Engrg. 1999*, 301–313.

Hughes, T. J. (2000). *The Finite Element Method, Linear Static and Dynamic Finite Element Analysis*. Dover Publications.

Hughes, T. J. R., J. A. Cottrell, and Y. Bazilevs (2005). Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Comput. Methods Appl. Mech. Engrg. 194*, 4135–4195.

Joy, K. I. and M. A. Duchaineau (1999). Boundary Determination for Trivariate Solids. In *Proceedings of the 7th Pacific Conference on Computer Graphics and Applications*, PG '99, pp. 82–91.

Kelly, G. and H. McCabe (2006). A survey of procedural techniques for city generation. *ITB Journal 14*, 87–130.

Kiendl, J., Y. Bazilevs, M.-C. Hsu, R. Wüchner, and K.-U. Bletzinger (2010). The bending strip method for isogeometric analysis of Kirchhoff-Love shell structures comprised of multiple patches. *Comput. Methods Appl. Mech. Engrg. 199*, 2403–2416.

Kiendl, J., K.-U. Bletzinger, J. Linhard, and R. Wüchner (2009). Isogeometric shell analysis with Kirchhoff-Love elements. *Comput. Methods Appl. Mech. Engrg. 198*, 3902–3914.

Knupp, P. M. (2000). Achieving finite element mesh quality via optimization of the Jacobian matrix norm and associated quantities. Part II—A framework for volume mesh optimization and the condition number of the Jacobian matrix. *Int J. Numer. Meth. Engng. 48*, 1165–1185.

Knupp, P. M. (2007). Remarks on mesh quality. In *45th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, NV.

Kolšek, T., M. Šubelj, and J. Duhovnik (2003). Generation of block-structured grids in complex computational domains using templates. *Finite Elements in Analysis and Design 39*(12), 1139–1154.

Leyton, M. (2001). *A Generative Theory of Shape*, Volume 2145 of *Lecture Notes in Computer Science*. Berlin/Heidelberg: Springer.

Lipton, S., J. Evans, Y. Bazilevs, T. Elguedj, and T. Hughes (2010). Robustness of isogeometric structural discretizations under severe mesh distortion. *Comput. Methods Appl. Mech. Engrg. 199*(5-8), 357 – 373.

Ma, D., F. Lin, and C. K. Chua (2001). Rapid Prototyping Applications in Medicine. Part 1: NURBS-Based Volume Modelling. *The International Journal of Advanced Manufacturing Technology 18*, 103–117.

Mäntylä, M. (1987). *An introduction to solid modeling*. New York, NY, USA: Computer Science Press, Inc.

Martin, T., E. Cohen, and R. M. Kirby (2009). Volumetric parameterization and trivariate B-spline fitting using harmonic functions. *Computer Aided Geometric Design 26*, 648–664.

Martin, W. and E. Cohen (2001). Representation and Extraction of Volumetric Attributes Using Trivariate Splines: A Mathematical Framework. In *Proceedings of the 6th ACM Symposium on Solid modeling and applications*.

Mehra, R., Q. Zhou, J. Long, A. Sheffer, A. Gooch, and M. N. J. (2009). Abstraction of Man-Made Shapes. *ACM Trans. Graph. 28*(5), 137:1–137:10.

Mitra, N. and M. Pauly (2008). Symmetry for Architectural Design. In *Advances in Architectural Geometry*, pp. 13–16.

Müller, P., P. Wonka, S. Haegler, A. Ulmer, and L. Van Gool (2006). Procedural modeling of buildings. *ACM Trans. Graph. 25*, 614–623.

Nagy, A. P., M. M. Abdalla, and Z. Gürdal (2010). Isogeometric sizing and shape optimisation of beam structures. *Comput. Methods Appl. Mech. Engrg. 199*, 1216–1230.

Natekar, D., X. Zhang, and G. Subbarayan (2004). Constructive solid analysis: a hierchical, geometry-based meshless analysis procedure for integrated design and analysis. *Computer Aided Design 36*, 473–486.

Oddy, A., J. Goldak, M. McDill, and M. Bibby (1988). A distortion metric for isoparametric finite elements. *Transactions of the CSME 12*(4), 213–217.

Parish, Y. I. H. and P. Müller (2001). Procedural modeling of cities. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '01, New York, NY, USA, pp. 301–308. ACM.

Piegl, L. and W. Tiller (1997). *The NURBS Book* (Second ed.). Berlin; Heidelberg: Springer Verlag.

Prusinkiewicz, P. and A. Lindenmayer (1990). *The algorithmic beauty of plants*. Springer Verlag.

Reunanen, M. (2010). Computer demos–what makes them tick? Licentiate thesis, Aalto University.

Rogers, D. F. (2001). *An Introduction to NURBS: With Historical Perspective*. Morgan Kaufman Publishers.

Rogers, D. F. and J. A. Adams (1990). *Mathematical Elements for Computer Graphics* (Second ed.). McGraw-Hill.

Rypl, D. and B. Patzák (2011). From the finite element analysis to the isogeometric analysis in an object oriented computing environment. *Advances in Engineering Software (Article in press)*, –. doi:10.1016/j.advengsoft.2011.05.032.

Sederberg, T. W., J. Zheng, A. Bakenov, and A. Nasri (2003). T-splines and T-NURCCs. *ACM Transactions on Graphics 22*(3), 477–484.

Seo, Y.-D., H.-J. Kim, and S.-K. Youn (2010a). Isogeometric topology optimization using trimmed spline surfaces. *Comput. Methods Appl. Mech. Engrg. 199*, 3270–3296.

Seo, Y.-D., H.-J. Kim, and S.-K. Youn (2010b). Shape optimization and its extension to topological design based on isogeometric analysis. *International Journal of Solids and Structures 47*, 1618–1640.

Shah, J. J. and M. Mäntylä (1995). *Parametric and Feature-Based CAD/CAM*. John Wiley & Sons, Inc.

Shapiro, V. (2002). Solid modeling. In G. Farin, J. Hoschek, and M.-S. Kim (Eds.), *Handbook of Computer Aided Geometric Design*, pp. 473–518. North-Holland/Elsevier.

Sheffer, A., T. Blacker, J. Clements, and M. Bercovier (1997). Virtual Topology Operators for Meshing. In *Proceedings of the 6th International Meshing Roundtable*, pp. 49–66.

Sheffer, A. and A. Üngör (2001). Efficient adaptive meshing of parametric models. In *Proceedings of the sixth ACM symposium on Solid modeling and applications*, pp. 59–70.

Shewchuck, J. R. (2002). What Is a Good Linear Element? Interpolation, Conditioning, and Quality Measures. In *Eleventh International Meshing Roundtable*, Ithaca, New York, pp. 115–126.

Shih, A. M., T.-Y. Yu, S. Gopalsamy, Y. Ito, and B. Soni (2005). Geometry and mesh generation for high fidelity computational simulations using non-uniform rational B-splines. *Applied Numerical Mathematics 55*, 368–381.

Shimada, k. (2006). Current Trends and Issues in Automatic Mesh Generation. *Computer-Aided Design & Applications 3*(6), 741–750.

Shor, S. and A. Eyal (2004). DEMOing: an emerging art form or just another digital craft? *Intelligent Agent Magazine 4*(1), –.

Stein, P., M.-C. Hsu, Y. Bazilevs, and K. Beucke (2012). Operator- and template-based modeling of solid geometry for Isogeometric Analysis with application to Vertical Axis Wind Turbine simulation. *Comput. Methods Appl. Mech. Engrg. 213*, 71–83.

Stroud, I. (2006). *Boundary Representation Modelling Techniques*. Springer.

Togelius, J., G. N. Yannakakis, K. O. Stanley, and C. Browne (2010). Search-based procedural content generation. In C. Di Chio (Ed.), *Applications of evolutionary computations*, Number 6025 in Lecture Notes in Computer Science. Springer.

Uhm, T.-K., K.-S. Kim, Y.-D. Seo, and S.-K. Youn (2008). A locally refinable T-spline finite element method for CAD/CAE integration. *Structural Engineering and Mechanics 30*(2), 225–245.

Vanegas, C. A., D. G. Aliaga, P. Wonka, P. Müller, P. Waddell, and B. Watson (2009). Modeling the Appearance and Behavior of Urban Spaces. *Computer Graphics Forum 28*(2), 1–18.

Wall, W. A., M. A. Frenzel, and C. Cyron (2008). Isogeometric structural shape optimization. *Comput. Methods Appl. Mech. Engrg. 197*, 2976–2988.

Wang, W. and Y. Zhang (2010). Wavelets-based NURBS simplification and fairing. *Comput. Methods Appl. Mech. Engrg. 199*, 290–300.

White, D. R., S. Saigal, and S. J. Owen (2005). Meshing complexity: predicting mesh difficulty for single part CAD models. *Engineering with Computers 21*, 76–90.

Whiting, E., J. Ochsendorf, and F. Durand (2009). Procedural modeling of structurally-sound masonry buildings. *ACM Trans. Graph. 28*(5), 112:1–112:9.

Xu, G., B. Mourrain, R. Duvigneau, and A. Galligo (2010). Optimal Analysis-Aware Parameterization of Computational Domain in Isogeometric Analysis. In B. Mourrain, S. Schaefer, and G. Xu (Eds.), *Advances in Geometric Modeling and Processing*, Volume 6130 of *Lecture Notes in Computer Science*, pp. 236–254. Berlin/Heidelberg: Springer.

Xu, G., B. Mourrain, R. Duvigneau, and A. Galligo (2011a). Optimal analysis-aware parameterization of computational domain in 3D isogeometric analysis. *Computer-Aided Design (Article in Press)*, –. doi:10.1016/j.cad.2011.05.007.

Xu, G., B. Mourrain, R. Duvigneau, and A. Galligo (2011b). Parameterization of computational domain in isogeometric analysis: Methods and comparison. *Comput. Methods Appl. Mech. Engrg. 200*, 2021–2031.

Zhang, Y., Y. Bazilevs, S. Goswami, C. Bajaj, and T. J. R. Hughes (2007). Patient-specific vascular NURBS modeling for isogeometric analysis of blood flow. *Comput. Methods Appl. Mech. Engrg. 196*, 2943–2959.

Zhou, X. and J. Lu (2005). Nurbs-based Galerkin method and application to skeletal muscle modeling. In *Proceedings of the 2005 ACM symposium on Solid and physical modeling*, pp. 71–78.

Zienkiewicz, O. C., R. L. Taylor, and J. Z. Zhu (2005). *The Finite Element Method: Its Basis & Fundamentals* (Sixth ed.). Elsevier Butterworth-Heinemann.

Zorin, D. and P. Schröder (2000). Subdivision for Modeling and Animation. SIGGRAPH Course Notes.