Comput. Methods Appl. Mech. Engrg. 199 (2010) 334–356

Contents lists available at ScienceDirect

Comput. Methods Appl. Mech. Engrg.

journal homepage: www.elsevier.com/locate/cma

Analysis-aware modeling: Understanding quality considerations in modeling for isogeometric analysis

E. Cohen [a,*], T. Martin [a], R.M. Kirby [a,b], T. Lyche [c], R.F. Riesenfeld [a]

[a] School of Computing, University of Utah, Salt Lake City, UT 84112, USA
[b] Scientific Computing and Imaging Institute, University of Utah, Salt Lake City, UT, USA
[c] CMA and Institute of Informatics, University of Oslo, P.O. Box 1053, Blindern, 0316 Oslo, Norway

ARTICLE INFO

Article history:
Received 8 April 2009
Received in revised form 26 August 2009
Accepted 7 September 2009
Available online 13 September 2009

Keywords:
Isogeometric analysis
Solid modeling
Mesh generation
Mesh quality
Model generation
Model quality

ABSTRACT

Isogeometric analysis has been proposed as a methodology for bridging the gap between computer aided design (CAD) and finite element analysis (FEA). Although both the traditional and isogeometric pipelines rely upon the same conceptualization to solid model steps, they drastically differ in how they bring the solid model both to and through the analysis process. The isogeometric analysis process circumvents many of the meshing pitfalls experienced by the traditional pipeline by working directly within the approximation spaces used by the model representation. In this paper, we demonstrate that in a similar way as how mesh quality is used in traditional FEA to help characterize the impact of the mesh on analysis, an analogous concept of *model quality* exists within isogeometric analysis. The consequence of these observations is the need for a new area within modeling – *analysis-aware modeling* – in which model properties and parameters are selected to facilitate isogeometric analysis.

1. Introduction

The concept of isogeometric analysis was first introduced by Hughes et al. in [1] as a means of bridging the gap between computer aided engineering (CAE), including finite element analysis (FEA), and computer aided design (CAD) [2]. Those familiar with the application of FEA to CAD-based models are well-aware of the complications and frustrations which arise when one attempts to take a "solid model" (a term which we will define in the context of the modeling community in Section 2) as produced by a typical commercially available CAD system, generate a surface tessellation and corresponding volumetric representation in terms of meshing elements (e.g. triangles and quadrilaterals on the surfaces and tetrahedra and hexahedra in the volume), and run an analysis. On this "preprocessing" side prior to the actually analysis step, a large amount of effort is expended in mesh generation and optimization (in terms of mesh quality), sometimes to the point of consuming more time than what is taken by the actual analysis step. Once an analysis is run, solution refinement often requires mesh adaptation or in worst case regeneration, both of which require consultation with the original CAD-model. Isogeometric analysis claims to break this common but insidious cycle by choosing an alternative route from the solid geometric model to analysis. In isogeometric analysis, one works with the functions used to generate the model directly by using the function space used for model generation as the approximating space in which field solutions are built (hence the name isogeometric).

By circumventing many of the pitfalls that one encounters during the mesh generation process by working directly with the solid model, isogeometric analysis effectively eliminates the geometric error component of the analysis pipeline. Geometric refinement is no longer necessary; the analyst can focus attention solely on solution refinement. It is our thesis that although circumventing the mesh generation pipeline implies that one no longer needs to consider *mesh quality*, there are still issues of *model quality* that must be considered. In a similar way as mesh quality is a geometric means of assessing the impact of a mesh on the function space which it induces in the classic finite element process, *model quality* is a characterization of those properties of the representation of the model geometry that impact the representation space (or trial space) used to approximate the fields of interest. The consequence of these observations is the need for a new area within modeling – *analysis-aware modeling* – in which model properties and parameters facilitate isogeometric analysis.

1.1. Nomenclature

In this section we set up the environment for considering isogeometric analysis in the context of linear second-order partial dif-

* Corresponding author. Tel.: +1 801 583 2815; fax: +1 801 581 5843.
E-mail addresses: cohen@cs.utah.edu (E. Cohen), martin@cs.utah.edu (T. Martin), kirby@cs.utah.edu (R.M. Kirby), tom@ifi.uio.no (T. Lyche), rfr@cs.utah.edu (R.F. Riesenfeld).

0045-7825/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.cma.2009.09.010

ferential equations (PDEs) with zero Dirichlet boundary conditions. We note that there exists a straightforward extension to nonzero Dirichlet (essential) boundary conditions and also to Neumann boundary conditions. The issues we raise are quite general and will arise in using the isogeometric concept in solving many types of partial differential equations: Some examples we will consider and upon which we will comment in the examples section will have nonzero essential boundary conditions and/or more complex partial differential operators (such as those found in the modeling of linear elasticity). We do, however, set up here the nomenclature to illustrate these specific problems in an arbitrary number of space dimensions.

Although some of these terms may appear obvious to either those familiar with geometric modeling or those familiar with engineering analysis, we believe it is important for both the geometric modeling and finite element analysis communities to be overtly explicit during this time of confluence of ideas.

Let $\Omega \subset \mathbb{R}^s$ with $s \in \mathbb{N}$ be a bounded domain with boundary $\partial\Omega$. Ω is the physical domain, often called the *world space* or *physical space*.

Using the notation $D_j = D_j^1$ to denote the partial derivative with respect to the jth variable, define

$$D^\alpha := D_1^{\alpha_1} \cdots D_s^{\alpha_s}, \tag{1}$$

a mixed partial derivative of total order $|\alpha| = \alpha_1 + \cdots + \alpha_s$. Then the column vector

$$\nabla f = [D_1 f, \ldots, D_s f]^T \tag{2}$$

denotes the gradient of f. Further, let

$$H^1 = H^1(\Omega) := \{f : \Omega \to \mathbb{R} : D^\alpha f \in L_2(\Omega), |\alpha| \leqslant 1\} \tag{3}$$

$$V = H_0^1 = H_0^1(\Omega) := \{f \in H^1(\Omega) : f = 0 \text{ on } \partial\Omega\}, \tag{4}$$

denote the spaces of functions with values and first order partial derivatives in $L_2 = L_2(\Omega)$.

Let $\{\phi_i\}_{i=1}^n \subset H^1(\Omega)$ be linearly independent functions in $H^1(\Omega)$. Moreover we define

$$J := \{j \in \{1, \ldots, n\} : \phi_j \in V\} \text{ and } |J| = \#\{j : j \in J\},$$

that is, the set of indices of those ϕ_j that vanish on $\partial\Omega$. If $s = 1$ then typically $J = \{2, \ldots, n-1\}$. We define the space

$$V_h^q(\{\phi_j\}_j) = V_h^q := \left\{ \sum_{j \in J} \boldsymbol{c}_j \phi_j : \boldsymbol{c}_j \in \mathbb{R}^q, j \in J,\ q \in \mathbb{N} \right\}, \tag{5}$$

and note that $V_h = V_h^1$ is a subspace of V. The index h is a flag indicating finite dimensionality and is often a measure of element diameter. The space V_h^q forms the space in which the approximation to the solution of the differential equation is made.

Suppose $\{\psi_j\}_{j=1}^n$ is a set of real-valued linearly independent functions on a partition of the unit cube $\Theta = [0,1]^s$ in $\mathbb{R}^s$, and the functions ϕ_j are given as

$$\phi_j(\boldsymbol{x}) = \psi_j \circ \boldsymbol{F}^{-1}(\boldsymbol{x}), \quad j = 1, \ldots, n, \tag{6}$$

where $\boldsymbol{F} = (F_1, \ldots, F_s) : \Theta \to \Omega$ is a bijection. Fig. 1 illustrates a modification between the shape of a ψ and its corresponding ϕ induced by $\boldsymbol{F}^{-1}$.

Moreover, we assume that

$$\boldsymbol{F}(\Theta^o) \subset \Omega^o$$

$$\sigma := \boldsymbol{F}|_{\partial\Theta} : \partial\Theta \to \partial\Omega, \tag{7}$$

i.e., $\boldsymbol{F}$ maps interior to interior and boundary to boundary. If we use the same functions ψ_j to define both the ϕ_j's and $\boldsymbol{F}$

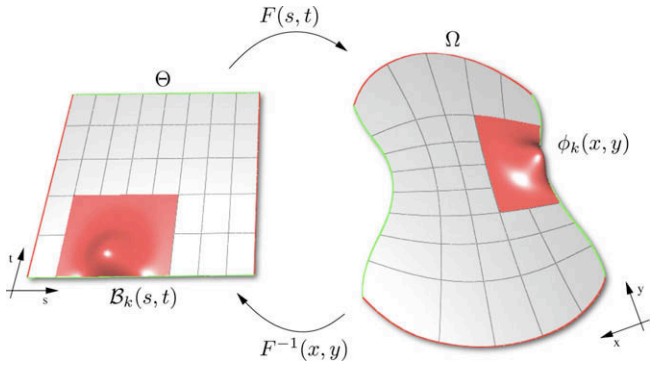

Fig. 1. F maps a point from the reference domain Θ to the physical domain Ω. Correspondingly, F^{-1} maps a point from Ω back to Θ. With F^{-1}, it is possible to define basis functions on Ω as compositions of F^{-1} with basis function defined on Θ.

$$\boldsymbol{F} = \sum_{j=1}^{n} \gamma_j \psi_j, \text{for some } \gamma_j \in \mathbb{R}^s, \quad j = 1, \ldots, n, \tag{8}$$

then the approach to solving the partial differential equation is called *isogeometric*.

Typically analysis will be carried out on models whose definition is piecewise over multiple hypercubes $\Theta = \{\Theta^i = [0,1]^s : i = 1, \ldots, K\}$ for some finite $K \in \mathbb{N}$, and $\boldsymbol{F}$ is defined piecewise in terms of the mappings from each Θ^i such that, for all $i, j, i \neq j$,

$$\boldsymbol{F}(\partial\Theta^i) \bigcap \boldsymbol{F}(\partial\Theta^j) \subset \partial\boldsymbol{F}(\Theta^i) \bigcap \partial\boldsymbol{F}(\Theta^j).$$

Further $\boldsymbol{F}$ must be continuous on every non-empty intersection. Θ is called the *parametric domain* or the *reference space*.

CAD systems typically create representations of the model that define mappings σ from $\partial\Theta$ to $\partial\Omega$, and can be written as a collection of mappings $\sigma_i : [0,1]^{s-1} \to \mathbb{R}^s, i = 1, 2 \ldots, 2s$ that agree on their shared boundaries. When $s = 3$, this representation is sometimes suitable for performing isogeometric shell analysis, but in order to perform a full volumetric analysis, the model must be *completed*. That is, the representation must be extended to completely define the interior. We use the term *model completion* to apply to completing networks of curves from boundary points, networks of surfaces from (shared) boundary curves, and networks of volumes from (shared) boundary surfaces. In each case some of the boundaries of the subpiece patches may be internal to the final geometric shape. We will discuss this further in what follows as this is often a non-trivial process.

Typically, ψ's are tensor product B-splines, Non-Uniform Rational B-splines, rectangular subdivision surfaces or T-splines.

1.2. Outline

The layout of this paper is as follows. In Section 2 we present the modeling to analysis pipeline. We briefly describe the process of going from model conceptualization to the *solid* model, and then distinguish between the classic route of surface and volumetric mesh generation as done for classic finite element analysis from the boundary representation and volumetric model generation as done for isogeometric analysis. In Section 3 we provide the mathematical and algorithmic descriptions of the isogeometric methodology employed in this work. In Section 4 we take a step back so as to appreciate the parametric modeling of geometry from the perspective of, and in the language of, isogeometric analysis. By doing so, we hope to demonstrate that in general there is not "a model" (a single ideal model) on which one does isogeometric analysis, but rather that modelers are presented with a collection of modeling choices – some of which may inadvertently impact analysis. In Sec-

tion 5 we present one-, two- and three-dimensional examples comparing different model completions and demonstrating the impact of model completion on quality of the solutions one obtains from an isogeometric analysis. In Section 6 we present some of the issues of the geometric representation, including model completion, that can affect the model quality in this new analysis paradigm. We do not seek to characterize all possible issues one might encounter, but rather to initiate a dialog between modelers and analysts concerning what is needed for analysis-aware modeling. We conclude in Section 7 with a summary of this work, some conclusions that we can draw and proposals of future work.

2. The modeling to analysis pipeline

In this section, we review the classic model to mesh to analysis pipeline for closed 3-manifold objects as appreciated by most FEA researchers, and then provide the corresponding modification to the pipeline as introduced by isogeometric analysis. Note that we pay particular attention to the use of nomenclature in this section, as the confluence of concepts from two fields (modeling and analysis) has led to misconceptions in both fields as to what is being discussed. We will use Fig. 2 as our visual guide through this process.

2.1. Conceptualization to solid model

The stages of the pipeline from conceptualization to *solid* model are denoted by the left half of Fig. 2. The designer has in mind a *concept* or *ideal* of what is to be designed, and uses a CAD modeling system to construct a collection of surfaces that are meant to represent the outer boundary of the object of interest. Note that the modeler is not working with three-dimensional manifold representations, but rather is working with surfaces sub-regions of which are intended to bound the object of interest. These surfaces are often constructed one-by-one without regard for how they will connect, intersect or overlap with other pieces, and frequently have regions that are not in the final model. The modeler then uses the CAD system to accomplish what is referred to as *trimming* – that is, cutting away portions of the original bounding surfaces that should not be part of the model. The adjacency information of the boundaries must then be incorporated to form a *water-tight* model – that is, one that clearly delineates $\mathbb{R}^3$ into three regions: inside the model, outside the model, and on the boundary of the model. This is frequently done by having the CAD system attempt to connect (or stitch) the surfaces together. Within the shape modeling community, the term *solid model* is used to characterize such a representation. When the solid model is represented using pieces of bounding 2-manifolds, the representation is called a *boundary representation* or *b-rep*. Whereas the surface representation (pre-trimming) does not necessarily faithfully represent the geometric and topological properties (such as being a water-tight surface) on the conceptual object, the newly formed solid model should. At the conclusion of this process, a solid model is output from the CAD system. Although called a solid, it is a collection of pieces of surfaces and connectivity information that define the boundary of a water-tight (closed) 3-manifold object.

Unfortunately the intersections between sculptured surface pieces that define the curves along which the pieces should be trimmed and stitched together cannot be exactly represented in the parameter spaces of the defining surfaces as discussed in [3], but rather are defined procedurally. Hence, while CAD systems have different approaches to explicitly representing these curves, the trimmed surfaces and resulting b-rep models are all approximated along the trimming edges. Frequently it may be necessary to *repair* the model to make it suitable for later processes such as analysis or fabrication.

2.2. Traditional meshing pipeline leading to analysis

The traditional meshing pipeline leading to analysis is denoted by the upper branch of the right half of Fig. 2. In the figure, we have purposefully placed quotations around "solid" to draw the reader's attention to two things. First, as previously mentioned, the solid model is not solid (in the sense of the term as used by analysts), but rather denotes the boundary of the object. Secondly, the solid model as produced by CAD systems is not always truly water-tight, but possibly only visually water-tight. This issue has been the bane of many surface tessellation efforts which have devised schemes under the assumption that the solid model formed a mathematically water-tight representation. In going from the solid model to

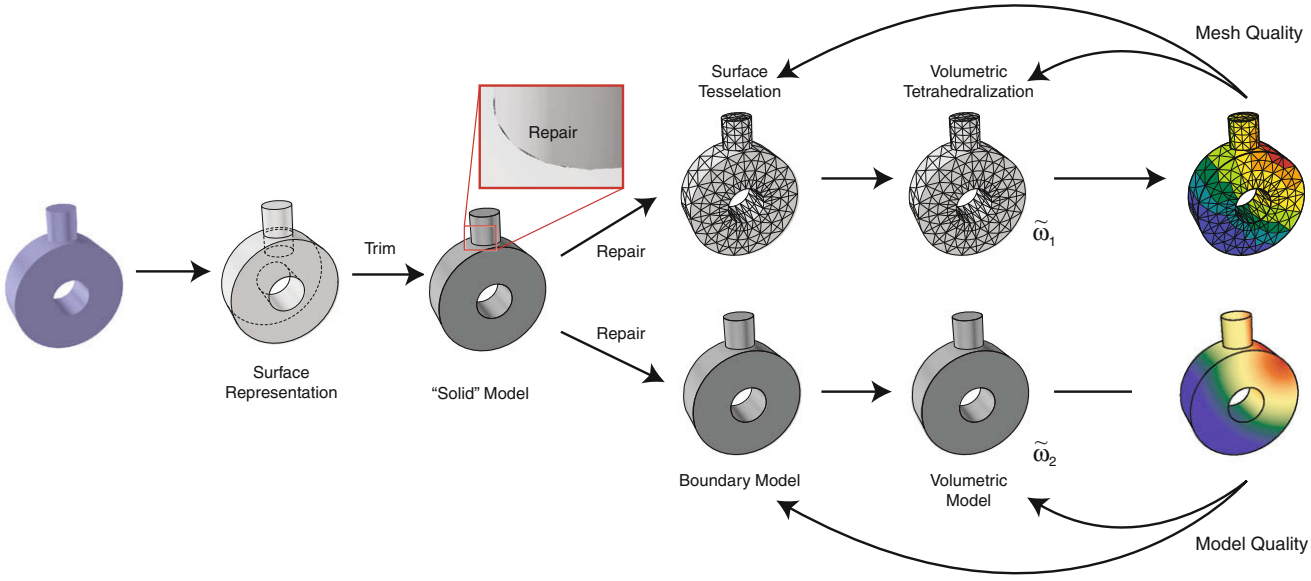

Fig. 2. Diagram presenting the classic concept-to-mesh pipeline (top branch) and the concept-to-model pipeline (bottom branch). A detailed discussion of the diagram is presented in the text.

a surface tessellation, it is often necessary to invoke a repairing procedure. We mark the repair process as being the point of deviation between the traditional pipeline and isogeometric analysis as many of the repair procedures used in traditional mesh generation assume that the target representation is a piecewise linear tessellation. The result of the repair and surface generation process is a tessellation of boundary of the object of interest. This tessellation is an approximation to the *true* geometry (in this case, the CAD-model), where approximation decisions have been made both in terms of how repairs are done and in how finely the tessellation captures the features of the original model. If a three-dimensional analysis is desired, the next step is to generate a volumetric tessellation, normally by filling in the volume with elements of the appropriate type (hexahedra or tetrahedra) for the analysis of interest.

In the classic finite element procedure, one generates a tessellation which approximates the true geometry, and then uses this tessellation to induce a function space in which approximations will be made. For instance, in classic linear finite elements over triangles, the triangular tessellation induces a piecewise linear (in total degree) space which is C^0 continuous along the edges of the triangles. As is well known by finite element practitioners, given two tessellations, both of which faithfully represent the geometry, one can get drastically different solutions due to the properties (or richness) of the approximating space that is induced.

A natural feedback loop was generated between analysts and mesh generation experts concerning the impact of meshes on solution quality. These metrics have commonly become known as *mesh quality metrics*. That is, they are geometric considerations (normally involving things like ratios of angles of elements, aspect ratios of edges, etc.) which help guide the development of meshes appropriate for analysis. Although these metrics have deep foundations within approximation theory, they are often abstracted away so that only geometric qualities of the mesh are discussed. We remind the reader, however, that maximizing mesh quality is, in its essence, an attempt to positively shape the approximating function space induced by the mesh.

2.3. Isogeometric pipeline leading to analysis

While isogeometric analysis is still a young field, the authors hypothesize that the path often taken when accomplishing isogeometric analysis is denoted by the lower branch of the right half of Fig. 2. As in the case of the traditional pipeline, repair is needed to ensure that the model being used for analysis meets the required topological constraints (such as closure) of the problem. In the case of isogeometric analysis, however, this repair process must be done keeping in mind the original and target representations. A starting point for isogeometric discussions in line with the finite element approaches is the boundary model, which should be a geometrically and topologically correct model of the bounding surfaces of the object. If a three-dimensional analysis is desired, volumetric representations must be generated prior to the analysis. The approximating space generated during an isogeometric analysis is dependent upon the boundary model (in 2D) or volumetric model (in 3D) that is used. Just as in the case of classic mesh generation, two different volumetric models generated from the same boundary model will create two different approximating spaces. Analogous to mesh quality impacting analysis, *model quality* impacts isogeometric analysis.

A different starting point for isogeometric analysis is that consideration during the shape (usually boundary) modeling process should be given to create a representation that lends itself to isogeometric analysis. There are frequently many modeling operations that lead to different representations for either the exactly same or closely related boundary shapes. Some of those representations are better suited to analysis than others, and within those groups, some are better suited to certain types of analysis than others. Similar issues have been recognized in created representations for models suitable for the multitude of computer-aided manufacturing processes and techniques. However, progress has been made in developing CAD systems that develop representations that, while suitable for design and display, are *fabrication-aware*, thus enabling a smoother, faster transition between design and fabrication.

Analysis-aware modeling in the context of isogeometric analysis may prove to be a key step towards that progression for design, engineering analysis, and simulation. Towards that end, this paper raises several important issues through a combination of analysis and demonstration in which the interaction between representation and analysis can either enhance or make the product evolution process difficult. Until such time as these issues have been quantified and embedded in analysis-aware modeling systems, the human modeler must be mindful of them.

3. Mathematical formulation

In this section, we first review the basic mathematical representational building blocks on which isogeometric analysis as well as many CAD and geometric modeling systems represent geometry. An overview of NURBS (Non-Uniform Rational B-Splines) can be found in [4]. All computational algorithms are presented there, so in the following section we discuss definitions of B-spline and NURBS functions and their combinations to define parametric mappings of global geometry. Note that this discussion provides the mathematical building blocks of modeling, but does not address how these building blocks are assembled as part of the modeling process. We will delve into the mind of the modeler in a subsequent section (Section 4).

3.1. The framework

Let $\Omega \subset \mathbb{R}^s$ with $s \in \mathbb{N}$ be a bounded domain with boundary $\partial\Omega$. The symbols D^α and ∇ are defined as in Eqs. (1) and (2), respectively. Function spaces H^1 and H^1_0 are Sobolev spaces, that is, Hilbert spaces whose norms are defined using information concerning both the function and some of its derivatives. In particular, the (Sobolev) norms for both H^1 and H^1_0 are given by

$$\|u\|_1^2 = \int_\Omega (u(x)^2 + \nabla u(x)^T \nabla u(x))\, dx.$$

Let $a : H^1_0 \times H^1_0 \to \mathbb{R}$ denote the bilinear form

$$a(u, v) := \int_\Omega \nabla u(x)^T \nabla v(x) dx, \tag{9}$$

This bilinear form is positive definite on H^1_0, and H^1_0 is a Hilbert space with inner product $a(u, v)$ and associated norm $\|u\| = \sqrt{a(u, u)}$. We let

$$(u, v) := \int_\Omega u(x) v(x) dx, \quad u, v \in L_2(\Omega),$$

be the usual L_2 inner product. For a general review of Sobolev spaces we refer the reader to [5].

Discussions in this paper will mostly be based on problems that arise in the relationship between geometry and analysis models. Most studies are focused on two prototypical mathematical model problems that arise in analysis, the Poisson problem and a corresponding eigenvalue problem, respectively.

$$-\nabla^2 u = f \text{ on } \Omega, \quad u = 0 \text{ on } \partial\Omega, \tag{10}$$

$$-\nabla^2 u = \lambda u \text{ on } \Omega, \quad u = 0 \text{ on } \partial\Omega. \tag{11}$$

Given $f \in L_2(\Omega)$ the weak form of (10) is to find $u \in V := H_0^1(\Omega)$ such that

$$a(u,v) = (f,v), \quad v \in V. \tag{12}$$

It is well known that (12) has a unique solution u, see [6].

The weak form of (11) is to find $\lambda \in \mathbb{R}$ and a nonzero $u \in V$ such that

$$a(u,v) = \lambda(u,v), \quad v \in V. \tag{13}$$

Since this involves finding the eigenvalues and eigenfunctions of a compact, symmetric operator in the Hilbert space $(V, a(\cdot,\cdot))$ there exists an increasing sequence of strictly positive eigenvalues

$$0 < \lambda_1 \leqslant \lambda_2 \leqslant \cdots \leqslant \lambda_k \leqslant \cdots$$

with $\lim \lambda_k = \infty$ and associated eigenfunctions u_k, which can be orthogonalized so that

$$a(u_j, u_k) = \lambda_k \delta_{j,k}, j, \quad k \geqslant 1.$$

Moreover, the eigenfunctions form a complete system, i.e. the set of all linear combinations is dense both in V and $L_2(\Omega)$, again see [6].

For V_h^q defined as in Eq. (5), the *Galerkin method* for (12) consists in finding $u_h \in V_h^q$ such that

$$a(u_h, v_h) = (f, v_h), \quad v_h \in V_h.$$

Writing $u_h = \sum_{j\in J} c_j \phi_j$, we obtain a linear system for the unknown coefficients c.

$$\boldsymbol{S}\boldsymbol{c} = \boldsymbol{f}, \boldsymbol{S} = (a(\phi_i, \phi_j))_{i,j\in J}, \quad \boldsymbol{f} = ((f, \phi_i))_{i\in J}. \tag{14}$$

Since the ϕ's are linearly independent and vanish on $\partial\Omega$ the *stiffness matrix* $\boldsymbol{S}$ is a symmetric positive definite matrix and (14) has a unique solution that is amendable to iterative methods like conjugate gradient [7].

The *Rayleigh–Ritz method* for (13) consists of finding λ and a nonzero $u_h \in V_h^q$ such that

$$a(u_h, v_h) = \lambda(u_h, v_h), \quad v_h \in V_h.$$

Writing $u_h = \sum_{j\in J} c_j \phi_j$ as before we obtain a generalized eigenvalue problem

$$\boldsymbol{S}\boldsymbol{c} = \lambda \boldsymbol{M}\boldsymbol{c}, \boldsymbol{S} = (a(\phi_i, \phi_j))_{i,j\in J}, \quad \boldsymbol{M} = ((\phi_i, \phi_j))_{i,j\in J}, \tag{15}$$

where both $\boldsymbol{S}$ and the *mass matrix* $\boldsymbol{M}$ are positive definite.

Thus the eigenvalues λ_{kh} of (15) that approximate the exact eigenvalues of (13) are positive

$$0 < \lambda_{1h} \leqslant \cdots \leqslant \lambda_{|J|h}$$

and the eigenfunctions u_h can be chosen to be orthogonalized so that

$$a(u_{jh}, u_{kh}) = \lambda_{kh}\delta_{j,k}.$$

Moreover $\lim_{h\to 0}\lambda_{kh} = \lambda_k$ for $1 \leqslant k \leqslant |J|$, provided $\lim_{h\to 0}\inf_{v_h\in V_h} \|u_k - v_h\| = 0$ for $1 \leqslant k \leqslant |J|$.

3.2. Definition of isogeometric finite element analysis

Suppose the basis functions $\phi_j(x)$ are given as in Eq. (6). Moreover, we assume that (7) holds i.e., $\boldsymbol{F}$ maps interior to interior and boundary to boundary, and that $\boldsymbol{F}$ is defined as in Eq. (8). Then the Galerkin and Rayleigh–Ritz methods for (12) or (13) are called *isogeometric*.

The elements s_{ij} of the stiffness matrix $\boldsymbol{S}$ can be expressed in terms of the gradients of the ψ_j basis functions. Let

$$\boldsymbol{J} = \boldsymbol{J}_{\boldsymbol{F}} := \begin{bmatrix} D_1F_1 & \cdots & D_sF_1 \\ \vdots & & \vdots \\ D_1F_s & \cdots & D_sF_s \end{bmatrix} = \begin{bmatrix} \nabla F_1^T \\ \vdots \\ \nabla F_s^T \end{bmatrix} \tag{16}$$

be the Jacobian of $\boldsymbol{F}$. Note that the elements of $\boldsymbol{J}$ are functions defined on Θ. We assume that $\boldsymbol{J}(t)$ is non-singular for all $t \in \Theta$. Then

$$s_{ij} = \int_\Omega \nabla\phi_i(x)^T \nabla\phi_j(x)dx = \int_\Theta \nabla\psi_i(t)^T \boldsymbol{N}(t) \nabla\psi_j(t)dt, \quad i,j = 1,\ldots,n, \tag{17}$$

where

$$\boldsymbol{N} = |\det(\boldsymbol{J})|\boldsymbol{J}^{-T}\boldsymbol{J}^{-1}. \tag{18}$$

Note that $\boldsymbol{N}(t)$ is positive definite for all $t \in \Theta$. Explicitly, for $s = 1$,

$$N(t) = \frac{1}{|F'(t)|},$$

and for $s = 2$

$$N = \frac{1}{|\det(\boldsymbol{J})|}\begin{bmatrix} \|\nabla F_2\|_2^2 & -\nabla F_1^T F_2 \\ -\nabla F_1^T F_2 & \|\nabla F_1\|_2^2 \end{bmatrix}.$$

If

$$\boldsymbol{K} := |\det(\boldsymbol{J})|^{-1/2}\boldsymbol{J} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T, \quad \boldsymbol{\Sigma} = \operatorname{diag}(\sigma_1,\ldots,\sigma_s), \quad \boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{V}^T\boldsymbol{V} = \boldsymbol{I},$$

with $\sigma_1 \geqslant \sigma_2 \geqslant \cdots \geqslant \sigma_s > 0$ is the singular value decomposition of $\boldsymbol{K}$ then

$$\boldsymbol{N} = \boldsymbol{U}\boldsymbol{\Sigma}^{-2}\boldsymbol{U}^T$$

is the spectral decomposition of $\boldsymbol{N}$. Since $\boldsymbol{N}$ is positive definite the eigenvalues of $\boldsymbol{N}$ are the inverse square of the singular values of $\boldsymbol{K}$ and the orthonormal eigenvectors of $\boldsymbol{N}$ are the right singular vectors of $\boldsymbol{K}$.

3.3. B-splines

For integers $n \geqslant 1$ and $d \geqslant 0$ let $\tau = \{\tau_i\}$ be a non-decreasing finite sequence of real numbers. We refer to τ as a *knot vector* and its components as *knots*. On τ we can recursively define degree d B-splines $B_{j,d} = B_{j,d,\tau} : \mathbb{R} \to \mathbb{R}$ by

$$B_{j,d}(t) = \frac{t-\tau_j}{\tau_{j+d}-\tau_j}B_{j,d-1}(t) + \frac{\tau_{j+d+1}-t}{\tau_{j+d+1}-\tau_{j+1}}B_{j+1,d-1}(t), \quad t \in \mathbb{R}, \tag{19}$$

starting with

$$B_{j,0}(t) = \begin{cases} 1, & \text{if } \tau_j \leqslant t < \tau_{j+1}, \\ 0, & \text{otherwise.} \end{cases}$$

Here we use the convention that terms with zero denominator are defined to be zero. We let $\mathbb{B}_{d,\tau} = \{B_{j,d,\tau}\}_j$.

A B-spline $B_{j,d}$ of degree d has the following properties:

1. It depends only on knots $\tau_j,\ldots,\tau_{j+d+1}$ and is identically zero if $\tau_{j+d+1} = \tau_j$.
2. For $t \in (\tau_j, \tau_{j+d+1}), 0 < B_{j,d}(t) \leqslant 1$ and $B_{j,d}(t) = 0$ otherwise. The interval $[\tau_j, \tau_{j+d+1}]$ is called the *support* of $B_{j,d}$.
3. Its derivative is
$$DB_{j,d}(t) = d\left(\frac{B_{j,d-1}(t)}{\tau_{j+d}-\tau_j} - \frac{B_{j+1,d-1}(t)}{\tau_{j+d+1}-\tau_{j+1}}\right),$$
again with the convention that terms with 0 denominator are set to 0.
4. If m of the $\tau_j,\ldots,\tau_{j+d_1}$ are equal to one value z, then $D^r B_{j,d}$ is continuous at z for $r = 0,\ldots,d-m$ and $D^{d-m+1}B_{j,d}$ is discontinuous at z.
5. Its integral is
$$\int_{\tau_j}^{\tau_{j+d+1}} B_{j,d}(t)dt = \frac{\tau_{j+d+1}-\tau_j}{d+1}.$$

6. It is affine invariant, i.e., for $u, v, t \in \mathbb{R} B_{j,d,u\tau+v}(ut+v) = B_{j,d,\tau}(t)$, where $u\tau + v := (u\tau_j + v)_j$.

Now suppose n, d are integers with $0 < d < n$. We say that $\tau = \{\tau_i\}_{i=1}^{n+d+1}$ is a $(d+1)$ *extended* knot vector on an interval $[a,b]$ if

$$a = \tau_{d+1} < \tau_{d+2}, \tau_n < \tau_{n+1} = b, \tau_{i+d+1} > \tau_i, \quad i = 1, \dots, n.$$

It is $(d+1)$-*regular* or $(d+1)$-*open* if in addition $\tau_1 = a$ and $\tau_{n+d+1} = b$; it is $(d+1)$-*regular uniform* or $(d+1)$-*open uniform* if $\tau_{i+1} - \tau_i = h$ for $i = d+1, \dots, n$ and $h > 0$. The knot vector is *uniform* if $\tau_{i+1} - \tau_i = h > 0$ for $i = 1, \dots, n+d$.

On the knot vector $\tau = \{\tau_i\}_{i=1}^{n+d+1}$ we can define n B-splines of degree d. The linear space of all linear combinations of B-splines is the *spline space* defined by

$$\mathbb{S}_{d,\tau}^q := \left\{ \sum_{j=1}^n \boldsymbol{c}_j B_{j,d} | \boldsymbol{c}_j \in \mathbb{R}^q \text{ for } 1 \leqslant j \leqslant n \right\}, \quad d \geqslant 0, \quad q \geqslant 1.$$

An element $f = \sum_{j=1}^n c_j B_{j,d}$ of $\mathbb{S}_{d,\tau} = \mathbb{S}_{d,\tau}^1$ is called a *spline function* if $q = 1$ or just a *spline* of degree d with knots τ, and $(c_j)_{j=1}^n$ are called the *B-spline coefficients* of f. For $q > 1$ the combination $\boldsymbol{f} = \sum_{j=1}^n \boldsymbol{c}_j B_{j,d}$ is a *spline curve*.

Suppose $\tau = \{\tau_i\}_{i=1}^{n+d+1}$ is a $(d+1)$-open knot vector on $[a,b]$. A spline $f : [a,b] \to \mathbb{R}$ is by definition continuous from the right. We define $f(b)$ by taking limits from the left. Let $\boldsymbol{f} = \sum_{j=1}^n \boldsymbol{c}_j B_{j,d}$. Then the following properties hold:

(1) B-splines $(B_{j,d})_{j=1}^n$ are *linearly independent* on $[a,b]$ and therefore a basis for $\mathbb{S}_{d,\tau}^q$.
(2) *Partition of unity*: $\sum_{j=1}^n B_{j,d}(t) = 1, \quad t \in [a,b]$.
(3) *Convex hull property*: $\boldsymbol{f}(t), t \in [a,b]$ lies in the convex hull of $[\boldsymbol{c}_1, \dots, \boldsymbol{c}_n]$.
(4) *Smoothness*: If z occurs m times in τ then f has continuous derivatives of order $0, \dots, d-m$ at z.
(5) *Locality*: If t is in the interval $[\tau_k, \tau_{k+1})$ for some k in the range $d+1 \leqslant k \leqslant n$ then

$$\boldsymbol{f}(t) = \sum_{j=k-d}^k \boldsymbol{c}_j B_{j,d}(t), \tag{20}$$

(6) *Affine invariance*: If $u\tau + v := (u\tau_j + v)_{j=1}^{n+d+1}$ then

$$\sum_{j=1}^n \boldsymbol{c}_j B_{j,d,u\tau+v}(ut+v) = \sum_{j=1}^n \boldsymbol{c}_j B_{j,d,\tau}(t), \ t \in [a,b], u, \ v \in \mathbb{R}. \tag{21}$$

(7) *Derivative of a spline*:

$$D\boldsymbol{f}(t) = d \sum_{j=2}^n \frac{\boldsymbol{c}_j - \boldsymbol{c}_{j-1}}{\tau_{j+d} - \tau_j} B_{j,d-1}(t), \quad t \in [a,b],$$

where terms with 0 valued denominator are set to 0.
(8) *Integral of a spline*:

$$\int_{\tau_1}^{\tau_{n+d+1}} \boldsymbol{f}(t) dt = \sum_{j=1}^n \frac{\tau_{j+d+1} - \tau_j}{d+1} \boldsymbol{c}_j. \tag{22}$$

(9) If $z = \tau_{j+1} = \cdots = \tau_{j+d} < \tau_{j+d+1}$ for $1 \leqslant j \leqslant n$ then $f(z) = c_j$.
(10) *Marsden's identity*:

$$(s-t)^d = \sum_{j=1}^n \prod_{i=j+1}^{j+d} (s - \tau_i) B_{j,d}(t), t \in [a,b], \quad s \in \mathbb{R}. \tag{23}$$

(11) *Nodal representation*:

$$t = \sum_{j=1}^n \tau_{j,d}^* B_{j,d}(t), \tau_{j,d}^* = \frac{\tau_{j+1} + \cdots + \tau_{j+d}}{d}, \quad t \in [a,b].$$

The nodal representation follows from Marsden's identity.

3.4. Knot insertion and degree raising (h- and p-refinement)

Suppose τ is a knot vector. Since two or more knots in τ can have the same value, we need to distinguish between the knot vector and the position of the knots. The distinct knot values in τ are called break points. We define the *multiplicity* of z in τ as

$$\mu_\tau(z) = \#\{\tau_j \in \tau : \tau_j = z\}, \quad z \in \mathbb{R}.$$

Notice that $\mu_\tau(z) = 0$ if z is not equal to one of the knots in τ. For $k \geqslant 0$ we define the knot vector $\tau^{(k)}$ to have the same break points as τ, and

$$\mu_{\tau^{(k)}}(\xi) = \mu_\tau(\xi) + k \text{ for all } \xi \in \tau.$$

Thus we increase the multiplicity of each break point in τ by k.

If $\boldsymbol{t}$ is another knot vector then we say that $\tau \subset \boldsymbol{t}$ if each break point ξ in τ is also a break point in $\boldsymbol{t}$ and $\mu_\tau(\xi) \leqslant \mu_{\boldsymbol{t}}(\xi)$.

Let d, e be integers, $0 \leqslant d \leqslant e$, let $\tau = (\tau_j)_{j=1}^{n+d+1}$ be $(d+1)$ extended on $[a,b]$ and let $\boldsymbol{t} = (t_i)_{i=1}^{m+e+1}$ be an $(e+1)$ extended knot vector on the same interval $[a,b]$. If $\tau^{(e-d)} \subset \boldsymbol{t}$ then $\mathbb{S}_{d,\tau} \subset \mathbb{S}_{e,\boldsymbol{t}}$, and there is a matrix $\boldsymbol{A} \in \mathbb{R}^{m,n}$ transforming the B-splines in $\mathbb{S}_{d,\tau}$ into the B-splines in $\mathbb{S}_{e,\boldsymbol{t}}$. Thus

$$B_{j,d,\tau} = \sum_{i=1}^m a_{ij} B_{i,e,\boldsymbol{t}}, j = 1, \dots, n, \text{ or } \boldsymbol{B}_{d,\tau}^T = \boldsymbol{B}_{e,\boldsymbol{t}}^T \boldsymbol{A},$$

where $\boldsymbol{B}_{d,\tau}^T = [B_{1,d,\tau}, \dots, B_{n,d,\tau}]$ and $\boldsymbol{B}_{e,\boldsymbol{t}}^T = [B_{1,e,\boldsymbol{t}}, \dots, B_{m,e,\boldsymbol{t}}]$ are row vectors.

If $f = \sum_{j=1}^n c_j B_{j,d,\tau}$ then $f = \sum_{i=1}^m b_i B_{i,e,\boldsymbol{t}}$, where

$$\boldsymbol{b} = \boldsymbol{A}\boldsymbol{c}, \quad \boldsymbol{c} = [c_1, \dots, c_n]^T, \quad \boldsymbol{b} = [b_1 \dots, b_m]^T. \tag{24}$$

The case where $e = d$ is called *knot insertion* and corresponds to *h*-refinement in the finite element literature. The situation where $e > d$ and $\boldsymbol{t} = \tau^{(e-d)}$ is called *degree raising* or *degree elevation* and corresponds to what is commonly known as *p*-refinement or *p*-enrichment [5,8,9]. In the general case where $\tau^{(e-d)}$ is a proper subset of $\boldsymbol{t}$ both knot insertion and degree raising occur. When this transformation is carried out with degree raising followed by knot insertion, Hughes et al. [1] introduced the term *k*-refinement to the isogeometric literature. Although it is possible to do the transformation in opposite order, i.e. a knot insertion followed by a degree raising, as observed in [1] in their discussion of *k*-refinement, this ordering leads to more coefficients and less smooth functions.

There are two algorithms for knot insertion. In Boehm's algorithm one knot at a time is inserted. In particular, if z is inserted in τ say between τ_k and τ_{k+1} so that $\tau_k \leqslant z < \tau_{k+1}$, then we obtain (24) with

$$b_i = \begin{cases} c_i & i = 1, \dots, k-d, \\ \frac{z-\tau_i}{\tau_{i+d}-\tau_i} c_i + \frac{\tau_{i+d}-z}{\tau_{i+d}-\tau_i} c_{i-1}, & i = k-d+1, \dots, k, \\ c_{i-1} & i = k+1, \dots, n+1. \end{cases} \tag{25}$$

Alternatively, using the Oslo Algorithms [10] we can insert all knots simultaneously and compute the elements of $\boldsymbol{A}$ row by row. Suppose t_i is located between τ_k and τ_{k+1}, *i.e.* $\tau_k \leqslant t_i < \tau_{k+1}$, then for row i,

$$\alpha_{j,r}(i) = \frac{t_{i+r} - \tau_j}{\tau_{j+r} - \tau_j} \alpha_{j,r-1}(i) + \frac{\tau_{j+r+1} - t_{i+r}}{\tau_{j+r+1} - \tau_{j+1}} \alpha_{j+1,r-1}(i),$$
$$j = k - r + 1, \dots, k, \quad r = 1, \dots, d, \tag{26}$$

starting with $\alpha_{j,k} = \delta_{j,k}$. Then we obtain Eq. (24) with $a_{i,j} = \alpha_{j,d}(i)$ for $j = k-d, \dots, k$ and $a_{i,j} = 0$ for other values of j.

The recurrence relation in (26) bears a strong resemblance to the one for B-splines given in (19). Since the numbers $\alpha_{j,d}(i)$ also have rather similar properties to $B_{j,d}(t)$, they are called *discrete B-splines*. For example

$$\alpha_{j,d}(i) \geqslant 0, \quad j = 1, \dots, n, \quad \sum_{j=1}^n \alpha_{j,d}(i) = 1, \quad i = 1, \dots, m.$$

For degree raising we also compute the transformation matrix $\boldsymbol{A}$ row by row [11]. Suppose as for knot insertion that $\tau_k \leqslant t_i < \tau_{k+1}$ and set $\Lambda_{j,0,r}(i) = \delta_{j,k}$ for $0 \leqslant r \leqslant e$, $\Lambda_{j,\ell,r}(i) = 0$ for all j if $0 \leqslant r < \ell$, and $\Lambda_{j,\ell,r}(i) = 0$ for all ℓ, r if $j < k - \ell$ or $j > k$. If we compute

$$\Lambda_{j,\ell,r}(i) = \frac{\ell}{r}\left(\frac{t_{i+r} - \tau_j}{\tau_{j+r} - \tau_j}\Lambda_{j,\ell-1,r-1}(i) + \frac{\tau_{j+r+1} - t_{i+r}}{\tau_{j+r+1} - \tau_{j+1}}\Lambda_{j+1,\ell-1,r-1}(i)\right) + \frac{r-\ell}{r}\Lambda_{j,\ell,r-1}(i), \tag{27}$$

for $\ell = 1, \ldots, d, r = \ell, \ldots, e$ and $j = k - \ell, \ldots, k$ then $a_{i,j} = \Lambda_{j,d,e}(i)$ for $j = k - d, \ldots, k$ and 0 for other values of j. Again terms with 0 denominator are set to 0. For $e = d$ we only need to compute $\Lambda_{j,\ell,\ell}(i)$ in (27) and we see that $\Lambda_{j,\ell,\ell}(i) = \alpha_{j,\ell}(i)$ for all j, r. It is shown in [12] that $\boldsymbol{A}$ is a non-negative stochastic matrix:

$$\Lambda_{j,d,e}(i) \geqslant 0, \quad j = 1, \ldots, n, \quad \sum_{j=1}^{n} \Lambda_{j,d,e}(i) = 1, \quad i = 1, \ldots, m,$$

$$0 \leqslant d \leqslant e.$$

An algorithm that is a literal implementation of (27) has complexity $O(de^2m)$; however, it is possible to derive faster algorithms for this kind of conversion. The main advantages are that it is quite stable and simple to implement.

Alternatively, degree raising can be carried out by converting the representation to examine individual polynomial pieces, performing degree raising on them, and then converting back to the B-spline form.

3.5. Nurbs

Suppose $\tau = \{\tau_i\}_{i=1}^{n+d+1}$ is a $(d+1)$-regular knot vector on $[a, b]$. Given positive numbers $\boldsymbol{w} = \{w_i\}_{i=1}^{n}$, we define the associated NURBS-basis of degree d by

$$R_{j,d}(t) := \frac{w_j B_{j,d}(t)}{\sum_{i=1}^{n} w_i B_{i,d}(t)}, \quad t \in [a, b], \quad j = 1, \ldots, n, \tag{28}$$

where $B_{i,d}$ is the B-spline of degree d with knots $\tau_i, \ldots, \tau_{i+d+1}$. Given $\boldsymbol{c}_j \in \mathbb{R}^q$ the sum

$$\boldsymbol{f} = \sum_{j=1}^{n} \boldsymbol{c}_j R_{j,d} \tag{29}$$

is called a *NURBS function* if $q = 1$ and a *NURBS curve* if $q > 1$. We have $R_{i,d} = B_{i,d}$ when $w_i = 1$ for all i. NURBS curves retains many of the desirable properties of splines curves. Moreover,

(1) NURBS can represent conic sections exactly.
(2) $R_{i,d}$ has the same local support and smoothness properties as $B_{i,d}$.
(3) NURBS-basis functions are non-negative and form a partition of unity, hence the convex hull property holds.
(4) $\{R_{1,d}, \ldots, R_{n,d}\}$ is linearly independent on $[a, b]$.
(5) A NURBS curve is affine invariant.

The exact properties of these functions depend on $\boldsymbol{w}$ as well as the knot vector τ and degree.

3.6. Tensor product splines

Using multi-index notation, an s-variate tensor product B-spline has the form

$$B_{\boldsymbol{j},\boldsymbol{d},\boldsymbol{T}}(\boldsymbol{t}) = \prod_{i=1}^{s} B_{j_i,d_i,\tau_i}(t_i), \quad \text{where} \quad B_{j_i,d_i,\tau_i} \in \mathbb{B}_{d_i,\tau_i},$$

where $\boldsymbol{d} = (d_1, \ldots, d_s), \boldsymbol{T} = (\tau_1, \ldots, \tau_s)$, and $\boldsymbol{j} = (j_1, \ldots, j_s)$. Define $\mathbb{B}_{\boldsymbol{d},\boldsymbol{T}}$ to be the set of all possible such s-variate combinations. The s-variate tensor product spline space is defined by

$$\mathbb{S}^q_{\boldsymbol{d},\boldsymbol{T}} := \left\{ \sum_{1 \leqslant \boldsymbol{j} \leqslant \boldsymbol{n}} \boldsymbol{c}_{\boldsymbol{j}} B_{\boldsymbol{j},\boldsymbol{d},\boldsymbol{T}} | \boldsymbol{c}_{\boldsymbol{j}} \in \mathbb{R}^q \text{ over all } B_{\boldsymbol{j},\boldsymbol{d},\boldsymbol{T}} \in \mathbb{B}_{\boldsymbol{d},\boldsymbol{T}} \right\}, \quad d \geqslant 0, \quad q \geqslant 1.$$

The definition for the s-variate rational is extended analogously.

Let $F \in \mathbb{S}^s_{\boldsymbol{d},\boldsymbol{T}}$, and fix the ith coordinate to be an element of the knot vector in that dimension. The $(s-1)$ free variables in Θ form an $(s-1)$-dimensional unit cube, $\ell_{i,j}(\boldsymbol{t}) = (t_1, \ldots, t_{i-1}, \tau^i_j, t_{i+1}, \ldots, t_s)$ for $\boldsymbol{t} = (t_1, \ldots, t_{i-1}, t_{i+1}, \ldots, t_s) \in [0, 1]^{s-1}$, and $F(\ell_{i,j}(\boldsymbol{t}))$ is called a *generalized knot-line*.

3.7. NURBS elements

Define $\theta_{\boldsymbol{i}}$ to be an s-dimensional rectangular parallelepiped (Cartesian product):

$$\theta_{\boldsymbol{i}} = [\boldsymbol{T}_{\boldsymbol{i}}, \boldsymbol{T}_{\boldsymbol{i}+1}] := \left[\tau^1_{i_1}, \tau^1_{i_1+1}\right] \times \cdots \times \left[\tau^s_{i_s}, \tau^s_{i_s+1}\right].$$

Each nonzero function in $\mathbb{S}_{\boldsymbol{d},\boldsymbol{T}}$ is a single multivariate polynomial over the interior of $\theta_{\boldsymbol{i}}$, and $\bigcup_{\boldsymbol{i}} \theta_{\boldsymbol{i}} = \Theta$. Then $\Omega_{\boldsymbol{i}} = \boldsymbol{F}(\theta_{\boldsymbol{i}})$ is an *element* or a *patch* in the physical space. The generalized knot-lines form the boundaries between the patches.

Sometimes a measure of behavior of the system as a whole is gauged by the behaviors of the collection of *localized* stiffness matrices, one for each element. Denote by $\boldsymbol{S}^{\boldsymbol{i}}$ the matrix formed by integrating over a just the $\boldsymbol{i}$th element. That is

$$a^{\boldsymbol{i}}(u, v) := \int_{\Omega_{\boldsymbol{i}}} \nabla u(x)^T \nabla v(x) dx$$

is used in computing the elements of $\boldsymbol{S}^{\boldsymbol{i}}$. If $s = 3$ and $\boldsymbol{d} = (3, 3, 3)$, then there are 64 $\phi_{\boldsymbol{j}}$ that are nonzero over $\Omega_{\boldsymbol{i}}$, and $\boldsymbol{S}^{\boldsymbol{i}}$ is (only) a 64×64 matrix.

Define the d-extended rectangular parallelepiped

$$\theta_{\boldsymbol{i},\boldsymbol{d}} := [\boldsymbol{T}_{\boldsymbol{i}-\boldsymbol{d}+1}, \boldsymbol{T}_{\boldsymbol{i}+\boldsymbol{d}}].$$

The significance of this set is that the value of a polynomial spline on θ_i only depends on the knots in $\theta_{\boldsymbol{i},\boldsymbol{d}}$.

4. Parametric representation of geometry

In this section we now focus on portraying a modeler's view of defining parametric representations of geometry, what is often called *modeling* in the CAD world. In general, the creator of the shape model is not the person who performs the analysis. Although many systems have analysis modules, the subsystem to create the shape is focused solely on shape. Furthermore, it would miss the point to create a system devoted exclusively to *design for analysis*, because the created design must be for shape, for analysis, for manufacturing and fabrication, for assembly analysis, versioning, and more. Hence *analysis-aware* modeling that exploits the currently available modeling flexibility of existing systems should be aimed at supporting the designer to make intelligent decisions about modeling that result in models $\boldsymbol{F}$ (defined in Section 1.1) that are better able to support analysis while preserving the capability of supporting the other important facets of the production process. For the most part, we present the discussion in the context of dimensional models to illustrate the points in the studies of Section 5.

While the analyst begins the process with a shape, the designer works towards a shape *representation* that meets the design specifications as the end goal. Hence the process for attaining the modeling goal varies with the design discipline, the individual designer, and the CAD system environment. For certain types of design, feature-lines establish key characteristics of shape. The subsequent surface must be generated around those features. Surfaces are assembled into models along surface edge curves, matching them carefully. In another style of modeling four boundary curve ele-

ments define the essence of a surface whose interior representation must be conformally generated. Networks of such regions form the model. Still other styles of design create reference curves that define surfaces through operations such as surface of revolution, extrusion, sweep, and the like. Another style employs named standard feature objects like hole, boss, fillet, etc., to describe shape, from which the designer or CAD system can generate the surfaces. Prevalent practice in design engineering delineates planar regions by bounding curves. The CAD process requires that a tensor product surface description is then created for computational purposes, typically a difficult process unless a non-tensor product representation is added to the general representation. Typically, the model will be bounded by many surfaces, so a volume model will not be able to be represented as the mapping of a single cube.

It is important to point out that current CAD modeling typically focuses on constructing of the boundaries to define an object. Although it is clearly the case that the boundary representations alone are sufficient for some types of analysis like shell analysis, they are not sufficient for all types of analysis. In particular, it is important to appreciate that modeling systems have nurtured a modeling mindset focused on generating surface representations, not on full volumetric representations. However, it is necessary to create a fully specified volumetric representation $\boldsymbol{F}$ so that the space V_h can be defined and used in the analysis. Creating $\boldsymbol{F}$ from its boundaries is called *model completion*. To be suitable for a full volume analysis, but unnecessary for most design and fabrication requirements, the interior of the bounded region should have a representation as well, i.e., it should be a volumetric model.

However, issues of modeling volumes or completing a boundary model to a full volume model have not been the subject of broad research focus other than a few scattered efforts [13–16]. We first examine some of the challenges of model completion.

4.1. Completion

There has been significant modeling research on the issue of completing a surface given boundaries, for both boundary curves in 2D (our case) and 3D. The generic problem is formidable, especially for a non-convex bounded region. Indeed, for complicated planar regions, and even more so for 3D curved boundaries, generating representations for smooth completions is still an area of research. Using a tensor product form requires the existence of four bounding curves, as described below. Complicated regions do not naturally lend themselves to this form (cf. Fig. 3), in much the same way that complicated volumes cannot be straightforwardly represented as a single mapping of the unit cube. So it is necessary to decompose the model into multiple regions, each one of which is the mapping of a cube. The process for attaining the decomposition is not well defined. Thus, it will be better understood if the modeling process can incorporate, without undue effort on the part of the designer, intrinsically volumetric design operators. We believe simpler cases will illustrate the issues that arise in creating representations $\boldsymbol{F}$ that defines Ω and the reference space when starting with a boundary representation.

Putting aside the details of how to effect this, assume that Ω has a theoretical decomposition, and the current concern is creating a single mapping from Θ with whatever partition is necessary. That is, there are $2s$ boundary faces, and opposite boundary faces share the same degree and knot vectors. The studies that we present are characterized by $\Theta, \Omega \subset \mathbb{R}^s, s = 1, 2, 3$. As remarked earlier, $\partial\Theta$ has $2s$ bounding faces, each an $s-1$ manifold. The mappings from each face in $\partial\Theta$ to $\partial\Omega$ are designated by the coordinate held constant over the face. Hence, the face labeled $i = 2(j-1) + (\ell), j = 1, \ldots, s$ and $\ell = 1, 2$ corresponds to the face that holds the j–th coordinate constant to value $\ell - 1$, and $\sigma_i = F|_{(\partial\Theta)_i}$. Considered separately, $\sigma_i : [0,1]^{s-1} \to \Omega$.

CAD-models are generally represented only in terms of the boundary, that is, as a collection of mappings $\mathscr{A} = \{\alpha_p\}_p$, $\alpha_p : [0,1]^2 \to \mathbb{R}^3$, that have not be created with any considerations for analysis. Simply designing and representing the model can be a major challenge. Resulting representations have the characteristic that:

- $\bigcup_p \alpha_p([0,1]^2)$ form a closed region of space $\partial\Omega$.
- Two surface pieces can meet only along a boundary curve, which is either identical or entirely disjoint.
- Arbitrarily many surfaces pieces can define a boundary.
- Arbitrarily many surface pieces can meet at a point.

4.2. Representing a line segment

A line segment may be considered the parametric completion of its boundary, namely, the two endpoints. Consider points $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$. Viewed as a B-spline curve, the linear parameterization of the line segment joining them is $\gamma(t) = \boldsymbol{P}_1 B_{1,1}(t) + \boldsymbol{P}_2 B_{2,1}(t)$, where the corresponding knot vector is $\tau = [0,0,1,1]$. Using the degree raising algorithms (p-refinement) this can be represented as a higher order curve γ_d. Since $\gamma_d(t) = \gamma(t)$ for all t, the curve exhibits constant velocity. Using knot insertion to refine the higher degree curve, perhaps non-uniformly, we obtain a curve $\tilde{\gamma}_d(t)$, that is still the same curve, but written in a different representation.

It is possible to write the same line with different, seemingly rather arbitrary nonlinear parameterizations. Now, we create several representations for later use in Section 5.

Let $\boldsymbol{P}_1 = (0,0)$ and $\boldsymbol{P}_2 = (1,0)$. We can just consider the mapping from $[0,1] \to [0,1]$, since the second coordinate is 0. Let $d = 3$, and consider two different knot vectors to complete the interior of the interval. Let τ_1, be the open uniform knot vector,

$$\tau_1 = [0,0,0,0,h,2h,\ldots,(n-4)h,1,1,1,1], \quad h = 1/(n-3),$$

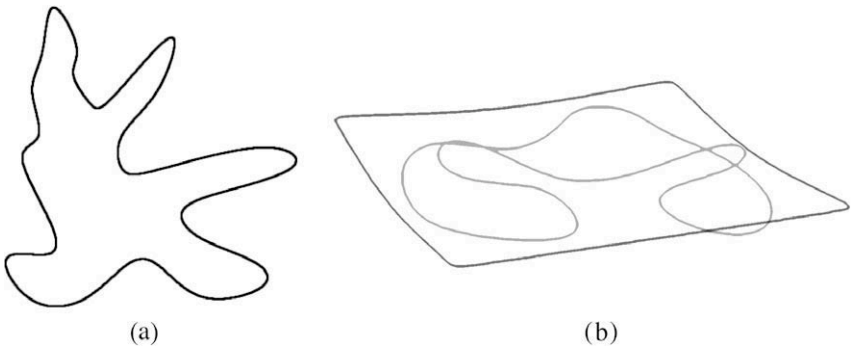

Fig. 3. (a) Boundary curves in 2D and (b) boundary curves in 3D.

and let for $0 < a < b < 1/2, n > 8$, and $\delta = (1-2b)/(n-7)$,

$$\tau_2 = [0,0,0,0,a,b,b+\delta,b+2\delta,\ldots,b+(n-8)\delta,1-b,1-a,1,1,1,1], \tag{30}$$

where the value of δ is chosen so that $b, b+\delta, b+2\delta, \ldots, b+(n-8)\delta, 1-b$ is a uniform partition of $[b, 1-b]$. By the nodal representation property (Section 3.3),

$$x = \mathscr{I}_k(x) := \sum_{j=1}^{n} \tau^*_{kj} B_{j,3,\tau_k}(x), \quad k = 1,2, \tag{31}$$

where $\tau^*_{kj} = (\tau_{kj+1} + \tau_{kj+2} + \tau_{kj+3})/3, j = 1,\ldots,n$, so that,

$$\tau^*_1 = \left(\tau^*_{1j}\right)_{j=1}^{n} = \left[0, \frac{1}{3}h, h, 2h, \ldots, 1-h, 1-\frac{1}{3}h, 1\right], \quad h = 1/(n-3).$$

Define a nonlinear parameterization of the unit interval with uniformly spaced coefficients given by

$$U_k(x) = \sum_{i=1}^{n} \frac{i-1}{n-1} B_{i,3,\tau_k}(x), \quad k = 1,2. \tag{32}$$

Notice that $\mathscr{I}_k$ is the identity and U_k stretches the two knot intervals near both endpoints, $k = 1,2$. This process can be extended to higher degree, in which case the $d-1$ knot intervals near both endpoints are stretched.

Applying the derivative formula reveals that on the first two knot intervals and the last two knot intervals the derivative changes quadratically, but on the rest of the interior knot intervals, it is constant. Hence U_k is linear on all but the two boundary knot intervals near the ends. An additional application of the derivative formula reveals that the second derivative is negative on the first two intervals and positive on the last two intervals, so the curve is concave on the first two intervals, and convex on the last two intervals. This mapping effects a stretch of the two knot intervals at each end and preserves it as constant in the middle (cf. Fig. 4).

We explore the effects of control polygon degeneracy on the knot intervals by creating $\boldsymbol{c}_1$, corresponding to map M_1, to have a cluster of two identical sequential control points ($c_{1,n/2} = c_{1,n/2+1}$), and $\boldsymbol{c}_2$, corresponding to M_2, to have one cluster of three identical sequential control points (at the corresponding center of the control point range), both are defined over a uniform open knot vector that yields n basis functions, where n is even for M_1 and odd (1 more) for M_2, and for which the remaining control points are uniformly spaced. So for $n = 10$,

Ordered control points for $\boldsymbol{c}_1$

$$= [0, 1/8, 1/4, 3/8, 1/2, 1/2, 5/8, 3/4, 7/8, 1] \tag{33}$$

Ordered control points for $\boldsymbol{c}_2$

$$= [0, 1/8, 1/4, 3/8, 1/2, 1/2, 1/2, 5/8, 3/4, 7/8, 1]. \tag{34}$$

In Section 5.1 we investigate the interactions between the knot vectors and the mappings $\mathscr{I}$ and U that act as the map from reference space to physical space in the case of longitudinal vibrations along a string. Cottrell et al. [17] studied this problem for $\mathscr{I}_1$ and U_1. We investigate more general cases and consider how interactions of mappings and knot vectors change the V_h and affect the eigenstructures.

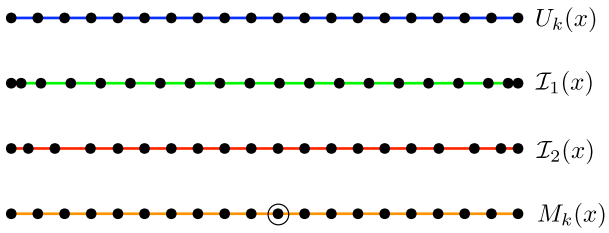

Fig. 4. Coefficients of the geometry maps. The top image portrays the evenly-spaced coefficients of $U_k(x)$, $k = 1,2$. The second and third rows are coefficients of $\mathscr{I}_k(x)$, the identity, for $k = 1,2$. They depend on the knot vectors. The fourth row shows the coefficients of M_1 are mostly the same as the coefficients of U_1, except for the double control point.

4.3. Completing surface regions bounded by curves

Given a curvilinear rectangular mesh of curves, there has been significant work on techniques to complete the representations to an implied smooth surface, including early work by Coons [18] and Gordon [19] in representing shape objects. However, when a single boundary has no straightforward decomposition into four boundary curves, most of these methods cannot be applied directly. Rather, the user must decompose the boundary into pieces amenable to patch fitting, and then work to guarantee that the patches join smoothly, a process sometimes called *skinning*. Finding a suitable solution can be facilitated by processes called *fairing*, which attempt to perform constrained optimization on the representation. Guaranteeing that the interior boundaries are identical can be a challenge when bounding curves are nonlinear.

Research in [15] for the planar case and [16] for the non-planar case seem to generate reasonable surfaces and parameterizations for modeling, but they have not been tested for suitability in analysis. Fig. 21 shows completions of the bounding curves in Fig. 3 formed using these methods, respectively.

First we investigate some specific representations of simple geometries for curves and their surface completions that are used in our studies in Section 5.

4.3.1. Representing a circular Arc

Used to represent part of a boundary, circular arcs appear ubiquitously in mechanical design. Suppose it is necessary to represent an arc of β radians taken from a circle of radius r. We follow the usual approach in [4] to create a quadratic NURBS template in the $x-y$ plane that can be affinely mapped to any position. The arc is represented initially as a quadratic rational B-spline with knot vector $\tau = [0,0,0,1,1,1]$. The representation will have one knot span. As shown in [4], the arc can be easily represented by the same Euclidean control points with many different functions w as long as $w_1 w_3 / w_2^2 = \sec^2(\beta/2)$. Commonly, $w_1 = w_3 = 1$ and $w_2 = \cos(\beta/2)$, so the arc is represented as

$$\boldsymbol{A}(t) = \sum_{i=1}^{3} \boldsymbol{P}_i R_{i,2}(t) w(t) = B_{1,2}(t) + \cos(\beta/2) B_{2,2}(t) + B_{3,2}(t),$$

where $\boldsymbol{P}_1 = r(1,0), \boldsymbol{P}_2 = r(1, \tan(\beta/2)), \boldsymbol{P}_3 = r(\cos\beta, \sin\beta)$ and the $R_{j,2}$ are defined as in Section 3.5. When $\beta = \pi/2$, and $r = 1, \boldsymbol{P}_1 = (1,0), \boldsymbol{P}_2 = (1,1), \boldsymbol{P}_3 = (0,1)$ and $w_2 = \sqrt{2}/2$. If $\beta = \pi$, $w_2 = 0$ which can lead to computational consistency problems. If $\beta > \pi$, then $w_2 < 0$, leading to yet other computational problems, such as potential zeros in the denominator. Thus a commonly used constraint is that $\beta < \pi$.

A full circle can be represented as three rotated instances of a $2\pi/3$ arc, giving rise to the NURBS representation

$$\boldsymbol{C}_3(t) = \sum_{i=1}^{7} \boldsymbol{P}_{3,i} R_{i,2,\tau_3}(t) \text{ defined with } w_3(t) = \sum_{i=1}^{7} w_{3,i} B_{i,2,\tau_3}(t),$$

$t \in [0,1]$,

where

$$\tau_3 = [0,0,0,1/3,1/3,2/3,2/3,1,1,1],$$

$$\mathscr{P}_3 = r[(-\sqrt{3},-1),(0,-1),(\sqrt{3},-1),(\sqrt{3}/2,1/2),(0,2),$$
$$(-\sqrt{3}/2,1/2),(-\sqrt{3},-1)],$$

and

$$\boldsymbol{w}_3 = [1, 1/2, 1, 1/2, 1, 1/2, 1].$$

The control points are successive vertices and midpoints, respectively, of the sides of an equilateral triangle that inscribes a circle of radius r having the origin as center.

Alternatively we can represent the circle as four rotated images of a quarter circle. This gives the NURBS curve

$$\boldsymbol{C}_4(t) = \sum_{i=1}^{9} P_{4,i} R_{i,2,\tau_4}(t), \text{ defined with } w_4(t) = \sum_{i=1}^{9} w_{4,i} B_{i,2,\tau_4}(t), \quad t \in [0,1],$$

where

$$\tau_4 = [0,0,0,1/4,1/4,1/2,1/2,3/4,3/4,1,1,1],$$

$$\mathscr{P}_4 = \frac{r}{2}[(-1,-1),(0,-1),(1,-1),(1,0),(1,1),(0,1),(-1,1),(-1,0),(-1,-1)],$$

$$\boldsymbol{w}_4 = [1, 1/\sqrt{2}, 1, 1/\sqrt{2}, 1, 1/\sqrt{2}, 1, 1/\sqrt{2}, 1].$$

For this configuration defining the circle in terms of an arc in each of the quadrants, the control points are the successively alternating vertices and midpoints of the sides of a square with that inscribed circle of radius r centered at the origin. The reader is referred to both [20,21], that discuss various ways to represent circles in fuller detail. These representations only guarantee C^0 across the boundaries of each of the arcs.

Both 3-arc and 4-arc representations are rational quadratic. A third rational representation of a circle is given by two semicircular arcs with cubic representations joined with C^1 smoothness. This representation, called $\boldsymbol{C}_2$ uses only six control points [20], is shown in Fig. 5(b), and has configuration

$$\text{knot vector}: \tau = [0,0,0,0,1/2,1/2,1,1,1,1], \tag{35}$$

$$\text{weights}: \boldsymbol{w} = [9,1,1/3,1/3,1,9] \tag{36}$$

$$\text{control coefficients}: \mathscr{P}_2 = \{(1,0),(1,2),(-1,2),(-1,-2),(1,-2),(1,0)\}. \tag{37}$$

In the figure, although the radii are drawn with uniformly spaced parameter values, notice the non-uniformity of the disc parameterization. That means that uniform h-refinement can well lead to non-uniformly sized analysis elements. That may be desirable for some tangent analysis problems. If it is not desirable then parameterizing the radii non-uniformly and creating curvilinear radii (to generate elements of more equal size) may be an appropriate completion when given $\boldsymbol{C}_2$ as a boundary representation.

Above we have used subscripts to reflect the number of distinct rational arc pieces used to represent a complete circle.

4.3.2. Solid discs from circular boundaries

The disc can be represented in many ways using NURBS. In this section we discuss three different representations that are all exact but differ in construction, degeneracies and smoothness. There are two widely used representations for completing the disc. The first, used in [22,23], is a polar type of parameterization of the disc induced by using $\boldsymbol{C}_4$ as one boundary. Select a point in the interior of the circle $\mathscr{O}$. Usually the center of the circle is selected for symmetry reasons, but another point could serve. Now select a representation for the unit interval. It could be the identity on any knot vector or any degree, it could be linear, quadratic, or it could be a cubic, like U_k given by (32). A tensor product representation is generated by selecting as the Euclidean part of the control points the scaled and rotated version of one of the representations of the line and one of the above representations of a circle. Translate, rotate and scale multiple instances of the line representation so there is one starting at $P_{4,j}$ and ending at $\mathscr{O}$ for each j. Name the rotated scaled instance of the ith coefficient of the line from $P_{4,j}$ to the origin be $D_{1,ij}$ with $w_{1,ij} = w_{4,j}$. Call this representation D_1.

This representation creates an orthogonal parameterization of the unit circle whose isoparametric lines are either circles or radii. See Fig. 5(a). The mesh is shown in the upper figure; isoparametric lines are drawn in the lower one. In this figure, the radial parameterization is linear. The lines drawn are showing parameterization,

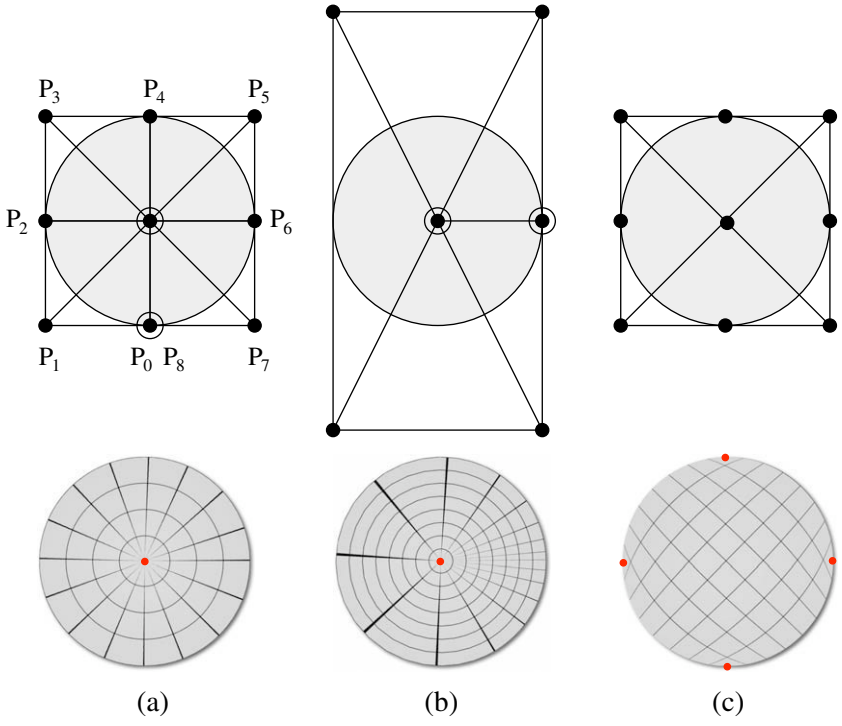

Fig. 5. Three ways to represent a disc of radius one. Column (a) is mapped with D_1, column (b) with D_2, and column (c) with D_3. Circles around black control points mean that more than one control point sits on the black control point location. Degeneracies are marked with red points. These representations are used to solve the drumhead problem. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

but not indicated analysis elements. An analogous solid representation can be generated from $\boldsymbol{C}_3$.

The resulting tensor product representation degenerates one whole boundary curve to a single point, so $J_{D_1}(\mathcal{O}) = 0$. The rate at which it goes to zero can be affected by modifying nearby control points so that the radii are not parameterized linearly, as in U_k. Thus the effect on $\boldsymbol{F} = D_1$ and its power to represent the solution space is adjustable without affecting the boundary geometry. It will affect the element shapes. The discussion in Section 5.3 concerns the impact on the induced function space and the ensuring impact on analysis results. Call D_1 the mapping that embodies $\boldsymbol{C}_4$ as one boundary in representing disc, places the origin as its opposite, and represents the radii linearly.

$\boldsymbol{C}_2$ is also a polar type. The initial NURBS representation for the disc has two rational cubic semicircles. Since there is one interior double knot, it is C^1 at the join. (Fig. 5(b)). The rest of the surface is generated using the same polar approach ad for D_1. However, the resulting is not as uniform an angular representation as for the first case. Call this mapping D_2.

Note, that for both mappings, an annulus could easily be modeled by choosing the circle representation for the smaller radius as the inner boundary and using a line representation that terminates at the control points of the smaller circle instead of the origin. The w's that define the rational space remain the same.

The third modeling choice is fundamentally different than the first two inasmuch as it uses opposite arcs as the matched boundaries of the tensor product representation.

$$\sigma_1(v) = \boldsymbol{A}(v);$$
$$\sigma_2(v) = R_\pi(\boldsymbol{A}(-v));$$
$$\sigma_3(u) = R_{\pi/2}(\boldsymbol{A}(u));$$
$$\sigma_4(u) = R_{3\pi/2}(\boldsymbol{A}(-u)),$$

where $\boldsymbol{A}$ is the $\pi/2$ radian circular arc, $R_\theta(\boldsymbol{P})$ means to rotate $\boldsymbol{P}$ through an angle θ. Nine control points determine the tensor product rational quadratic surface, eight of which are specified by the boundary control points. The remaining is chosen to be $\mathcal{O}$ and the associated w set to 1/2. Call this mapping D_3, as shown in Fig. 5(c). With respect to the number of basis functions, this is the most concise representation of a disc. In this case nine control points are needed as compared to 18 and 12 required for the first two representations, respectively. Furthermore, note that this mapping exhibits no interior degeneracy, but there are four locations on the boundaries at which the boundary curves meet at which the Jacobian vanishes. The rate at which the Jacobian goes to zero can be adjusted by modifying the initial NURBS representations of the boundary curves, and adjusting the rate at which the Jacobians go to 0 by modifying the locations of the control points on the interior, particularly those that result from h- or k-refinement. The boundary geometry is unperturbed by these modifications, but V_h changes, because $\boldsymbol{F}$ is different, even though they are all in a single $\mathbb{S}_2$. The various completions are not affine transformations of each other. Note that neither $\boldsymbol{C}_2$ nor $\boldsymbol{C}_3$ are suitable for use with this disc representation, but are quite reasonable for most other computational uses encountered in CAD. Again, see the discussion in Section 5.3 concerning the impact on the induced function space and the corresponding impact on analysis.

4.3.3. Volumetric models such as solid cylinders and solid tori

CAD systems generate multiple boundary models from the three circle representations above, cylinders (without the top and bottom surface), tori, and other types of extruded and swept surfaces. However, although CAD systems do not generate volumetric models, volumetric models of those shapes can be generated from the disc surface completions above. A variant of the disc of revolution has been used to generate geometry of vascular structures and to create the trial space for isogeometric analysis of blood through those structures [22]. It was used to generate geometry for optical lenses and carry the varying index of refraction volume attribute for computing the optical behavior of those lenses [24]. A sweep surface is defined as

$$\sigma(u, v) = A(u) + M_u(S(v))$$

where S is the cross section curve, A is a spine curve along which S is swept, and M_u is a transformation incorporating rotation and nonuniform scaling of $S(v)$ as a function of u. Unfortunately this representation has self-intersections wherever the radius of curvature of A is less than the first intersection of the curve normal of A with $\sigma(u, v)$. Generalized cylinders and tori also fall into this category. If the boundary has no self intersections, this can be made into a volumetric sweep by using the surface completion $S(v, w)$ of $S(v)$. If there are any self intersections, then this method is unsuitable.

Generalizations of this representation include allowing S to also be a function of u, and allowing S to be nonplanar. Both of those generalizations were combined in [25] to create a modeling technique for generalized cylinder-like objects with overhang regions. Such shapes cannot be modeled as a single NURBS patch if the cross section surfaces in the sweep are required to be planar. Because of the complexities of the boundary and some constraints on the isoparametric contours, there are no straightforward tech-

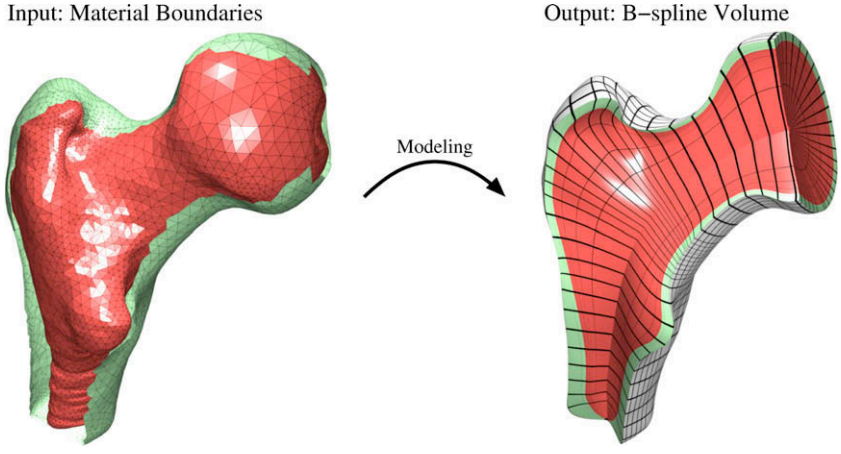

Fig. 6. Input: Two triangulated material boundary surfaces (left) and Output: a trivariate volumetric B-spline representation resulting from the reconstruction method in [25] (right). Note that the outer triangulated surface is partially cut-away in the left image so that the inner surface can be seen. Analogously, a wedge and the top are cut from the volumetric B-spline model on the right to visualize its interior.

niques for modeling some of them as images of multiple 3-cubes. For example, Fig. 6(left) cannot be modeled as a single sweep with planar cross sections. A partition of the data into multiple Θ domains will create mappings F that also split material properties (bone type), and Young's modulus (for linear elasticity) in unnatural, ways, not along isoparametric surfaces of the resulting mappings. Thus, it was deemed appropriate to allow some distortion in the parameterization, while still maintaining a quality model for analysis. An example of this approach is studied in Section 5.5.

5. Studies

In this section we examine studies that demonstrate how different modeling choices, in fact, can easily lead to different simulation results. The first mathematical model problem (Section 5.1) is the study of the eigenstructure of a system under different completion representations. We use a structural vibration analysis problem in both 1D and 2D. Then in Section 5.4, the Poisson equation is solved on different physical domains in 2D, where each domain is exactly represented with different choices of geometric model. In both sections h-refinement is applied and convergence rates are compared. Finally, in Section 5.5 we present a 3D study of the linear elastic deformation of a complex geometric model of a human femur.

5.1. Vibrations

The natural frequencies of a vibrating string are typically modeled by Eq. (11). Finding the natural frequencies can correspondingly be transformed to solving the system in Eq. (15). While the non-affine mapping F affects the eigenstructure, so does the choice of underlying B-spline space $\mathbb{S}$.

5.2. Longitudinal vibrations of a 1D elastic rod

Consider the eigenvalue problem (11) for $s = 1$ and $\Omega = [0, 1]$

$$-u''(t) = \lambda u(t), \quad t \in (0,1), \quad u(0) = u(1) = 0. \tag{38}$$

The exact eigenvalues and eigenfunctions for this problem are

$$\lambda_k = k^2\pi^2, \quad u_k(t) = \sin(k\pi t), \quad k = 1, 2, 3, \ldots. \tag{39}$$

The eigenfunctions u_k are orthogonal both with respect to the usual L_2 inner product and the energy inner product $a(u, v) = \int_0^1 u'(t)v'(t)dt$.

Cottrell et al. [17] solved this problem numerically using an isogeometric Rayleigh–Ritz method. It was demonstrated that with uniform knots one can get rid of outliers using a nonlinear mapping F. We demonstrate here that, with a non-uniform knot vector, a linear F also has no outliers. Also, we show that control mesh degeneracies in the interior of Ω have a negative effect on the eigenstructure.

We solve (38) numerically by the isogeometric Rayleigh–Ritz method (15) using four different spaces V_h generated by different bases $(\phi_i = \psi_i \circ F^{-1})_{i=1}^n$.

We use τ_1 and τ_2, as defined in Section 4.2 as knot vectors, providing uniform open and non-uniform open knots with larger reference space elements near the ends. The mappings to physical space are the identity, $\mathscr{I}_k$, and the uniformly spaced coefficients, $\mathscr{U}_k$, over each knot vector, $k = 1, 2$. Then $V_{\ell,\mathbb{S}_k}$ is the physical space of approximating functions space for the kth knot vector, where $\ell \in \{\mathscr{I}, U\}$. While $V_{\mathscr{I},\mathbb{S}_k} = \mathbb{S}_k$ is the spline space for the kth knot vector, $V_{U,\mathbb{S}_k} = span\{\phi = \psi(U_k^{-1}) : \psi \in \mathbb{S}_k\}$ is not a spline space.

As we have shown in Section 4.2, both U_1 and U_2 are increasing, concave on the first $d-1$ intervals, convex on the last $d-1$, and linear in between. Thus, U stretches the intervals near the boundaries and shrinks the interior ones with a constant scaling. The same behavior will be observed for other degrees, as long as n is sufficiently large, which occurs when we consider the discrete normalized spectrum. The choices of a and b are the stretch factor at the ends. If they are chosen too large, then the slope of the interior line segment becomes small. The consequence is that the slope of U_k^{-1} is large in that region. Since the values of J_F and $J_{F^{-1}}$ affect both the stiffness and mass matrices, they can adversely affect the eigenstructure. However, an optimal location will depend on the number of interior knots as well. This study was run with several different non-uniform knot vectors, although only one is shown below. Note that $\mathscr{I}_k' \equiv 1$, but U_k is more interesting in its behavior.

The *normalized discrete spectrum* is $\boldsymbol{\eta} = [\eta_0, \eta_1, \ldots, \eta_{N-1}]$ where η_k is the ratio between the eigenvalue $\sqrt{\lambda_{k,h}}$ and its corresponding exact solution, (39), i.e.,

$$\eta_k = \sqrt{\frac{\lambda_{k,h}}{\lambda_k}}. \tag{40}$$

Designed to have knots at prescribed distances from the endpoints of both sides, τ_2 has its remaining knots evenly-spaced across the rest of the interior interval. We demonstrate that the identity map with this basis creates $V_h = \mathbb{S}_{d,\tau_2}$ with no optical branches in the normalized discrete spectrum. It becomes clear that the mapping F, the space $\mathbb{S}_{d,\tau_2}$ and the particular DE being solved all interact to affect the appropriation properties of the resulting V_h and the

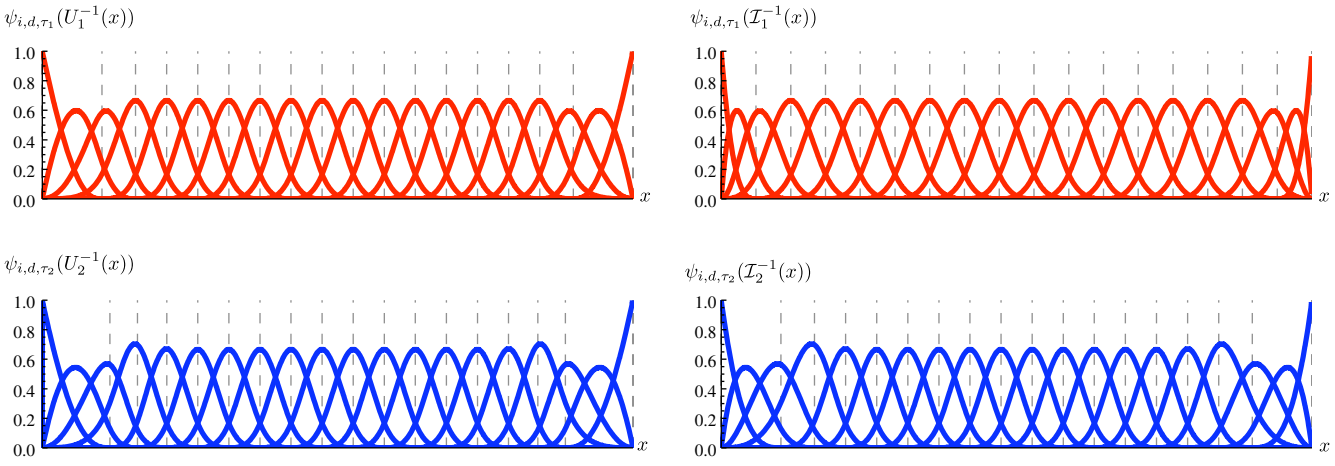

Fig. 7. Given the knot vectors τ_1 and τ_2, the first column shows $\psi_{i,d,\tau_k}(U_k^{-1}(x))$, and the second shows $\psi_{i,d,\tau_k}(\mathscr{I}_k^{-1}(x))(k = 1, 2)$. Note that, $\psi_{i,d,\tau_k}(\mathscr{I}_k^{-1}(x))$ are not stretched, but setting a and b appropriately causes the end functions to exhibit wider support and more resemble $\psi_{i,d,\tau_k}(U_k), k = 1, 2$.

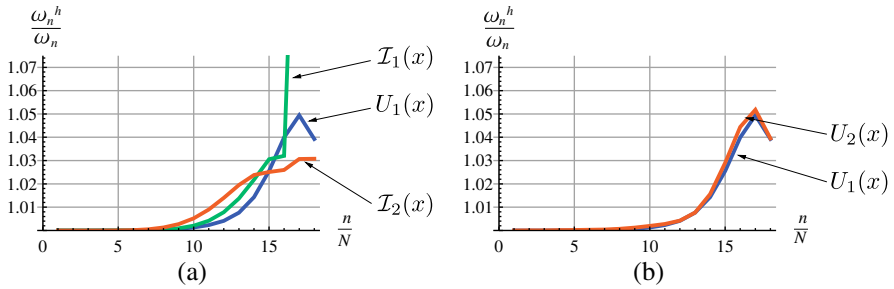

Fig. 8. Two cubic identity mappings constructed from different knot vectors. The particular choice of non-uniform knot vector yields a flatter normalized discrete spectra for the vibrating rod.

rates at which computed solutions converge towards the true solution.

Examine the ϕ-basis functions defined on Ω in Fig. 7 of the four mappings U_k and $\mathcal{I}_k$ where Fig. 4 shows their coefficients. By stretching the end elements, U_k stretches the basis functions at the boundaries so they extend farther into the interval and have a more rounded shape, compared to $\mathcal{I}_k$ that maintain the uniform knot spacing on Ω from Θ.

Given the uniform open knot vector τ_1, as observed in [17], for the identity on the uniform open knot vector $\mathcal{I}_1$, there are outliers of η towards the end of the spectrum that become worse with increasing degree. The U_1 spectrum does not produce outliers, shown in Fig. 8(a), and with growing degree η gets flatter. However, with the choice of τ_2, the linear map $\mathcal{I}_2$ performs better than U_1 in Fig. 8(a) where outliers are completely removed and η is flatter than using U_1. Note, that the η for U_2 is not as flat as with $\mathcal{I}_2$; the corresponding η are not as flat as with U_1. It is useful to also look at the eigenvectors. The eigenvectors allow us to generate the approximations to the eigenfunctions. For $k = 18$, using either U_1 or $\mathcal{I}_2$ approximates the true eigenfunction better than $\mathcal{I}_1$ does (see Fig. 9).

This example illustrates that the Jacobian of a mapping is not the sole factor governing numerical quality. Having the identity matrix as its Jacobian, the identity mapping on τ_1 produces equally-sized elements. A standard isoparametric mesh quality metric [26], would judge its quality as optimal. This study, however, demonstrates that non-uniform knot vectors perform better, producing, when knots are chosen as above, wider elements at the domain's boundary, and still maintain a constant Jacobian. In the same way that the FEA community has had to rethink (or expand their thinking) concerning mesh quality in the face of anisotropic mesh refinement usage [27–29], modeling for isogeometric analysis will need to carefully consider both the geometric and induced function space impact of uniformity and non-uniformity within the representation. In particular, element quality is not the only factor in deciding whether an analysis will succeed, it is necessary to understand the mapping $F : \theta \to \Omega$ as well.

Although it is not widely recognized in practice, we have demonstrated that knot vector choice is important. Although initial knot vectors may be decided during design and not up to the analyst, the methodology of mesh refinement is under the purview of the analyst. Uniform h-refinement may not be an optimal strategy. In higher dimensions the choice of knot vectors is more complicated. Given the tensor product nature of a NURBS, inserting a knot at a certain location might simultaneously produce a more favorable results in one region and a less favorable one in others. The use of T-splines [30] could be the solution to that problem.

5.2.1. *Influence of control mesh degeneracies on normalized discrete spectra*

In this section we examine how degeneracies in the control mesh affect normalized discrete spectra. During modeling, it can

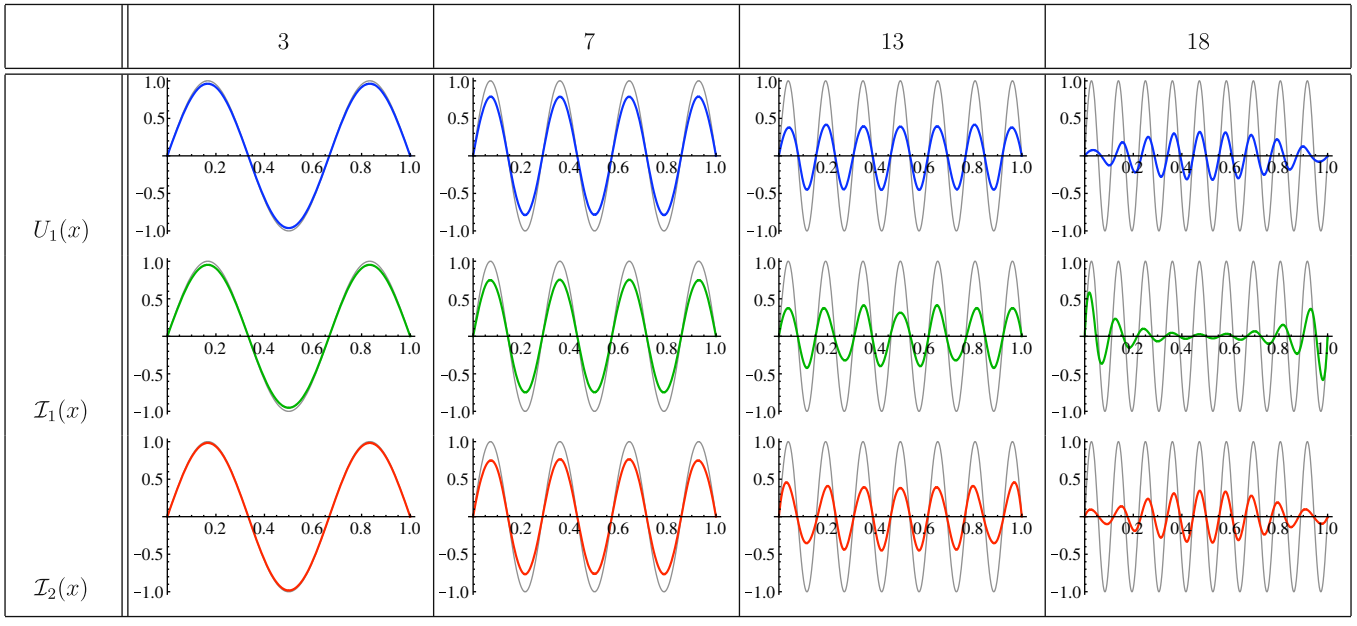

Fig. 9. Approximated eigenfunctions (exact solution in gray) for the modes $k = 3, 7, 13, 18$ using the non-uniform mapping U_1, and the identity mappings $\mathcal{I}_1$, $\mathcal{I}_2$.

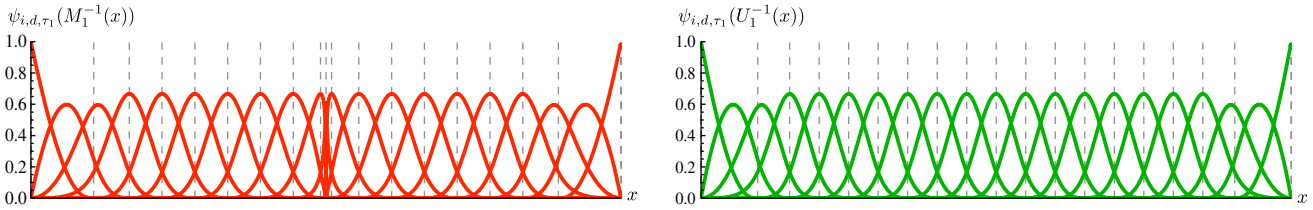

Fig. 10. The cubic-based basis functions defined on $\Omega = [0, 1]$ for $V_{M_1,\mathbb{S}_1}$ and $V_{U_1,\mathbb{S}_1}$, respectively. The control points for the M_1 are uniformly spaced, except the control point in the middle is duplicated. On the right, $\psi_{i,d,\tau_1}(U_1^{-1}(x))$ are shown for comparison.

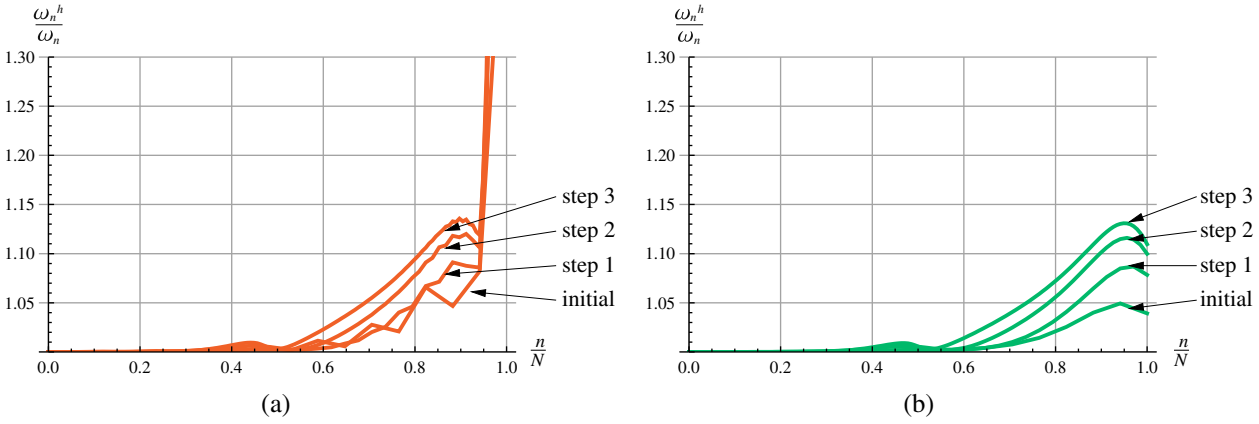

Fig. 11. *h*-Refinement: For both mappings (degenerate and non-degenerate), the normalized discrete spectra are slightly worse under uniform refinement. The uniformly refined space generated by M_1 always has outliers.

happen that the resulting control mesh of a model contains control points that coincide. In the following, we examine the mapping M_1 that is a cubic mapping with uniformly spaced control points except for a single instance of a single pair of identical repeated control points occurring about midway along the control point list, a configuration that creates a mesh degeneracy. The reader is referred to Fig. 10 which shows the basis functions $\psi_{i,d_1,\tau_1}(M_1^{-1}(x))$ and $\psi_{i,d_1,\tau_1}(U_1^{-1}(x)), i = 1, \ldots, n$ for $n = 20$ for comparison.

We apply *h*- and *k*-refinements to the spaces $V_{h,M_1,d_1} := V_h((\psi_{i,d_1,\tau_1}(M_1^{-1})))$ and $V_{h,U_1,d_1} := V_h((\psi_{i,d_1,\tau_1}(U_1^{-1})))$ resulting in spaces $V_{h/(2^m),M_1,d_1}$ and $V_{h/(2^m),U_1,d_1}$, respectively for m steps of *h*-refinement, and $V_{h/(2^{p-d_1}),M_1,p}$ and $V_{h/(2^{p-d_1}),U_1,p}$ when the *k*-refinement raises the degree to p, respectively. Note that the *h*-refining takes place after each step of the degree elevation, and halves all knot interval spacing. The corresponding normalized discrete spectra are computed. Fig. 11 shows the results for *h*-refinement and Fig. 12 shows the *k*-refined versions of the two models. The outlier problem is not ameliorated by refinement of either type. Although the normalized discrete spectrum becomes slightly worse under *h*-refinement, it becomes flatter under *k*-refinement – except for the outliers.

Investigations of M_1 and its inverse M_1^{-1} (Fig. 13) indicate possible reasons. The uniform spacing of the coefficients causes the mappings to stretch the boundary elements, creating a mapping that is concave for smaller values of t and convex for values of t close to 1. A multiple control point causes the parameterization to be nonlinear near the nodes corresponding to the multiple control point, pulling the density of the mapping towards that node.

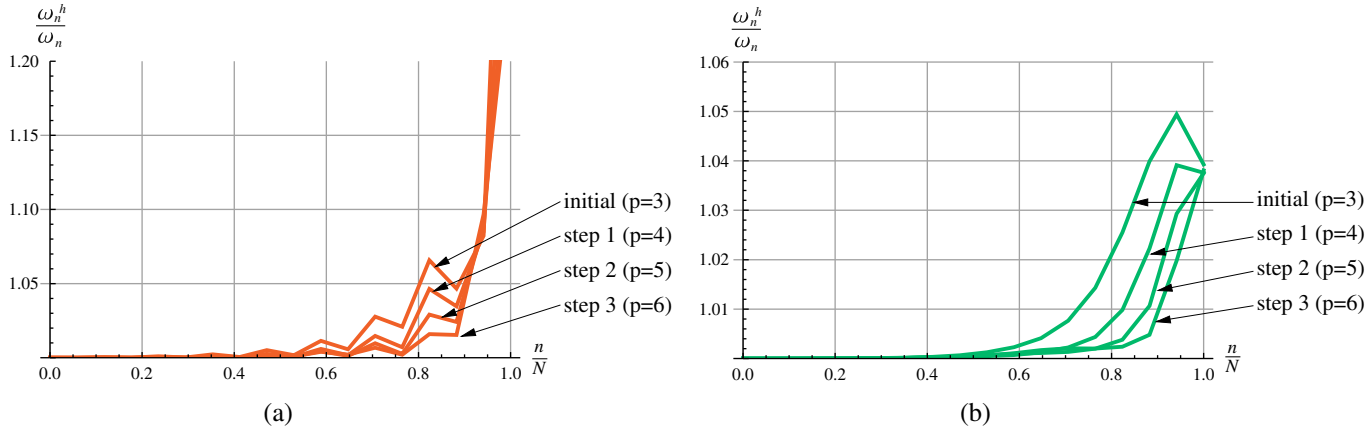

Fig. 12. *k*-Refinement: For both mappings (degenerate and non-degenerate), the normalized discrete spectra become flatter with elevated degree. However, the outlier exists throughout uniformly refining for the spaces based on M_1.

Hence the slope of M_1 becomes small near 1/2, and so the slope of M_1^{-1} gets large. So, in this small region the behavior of the Jacobian affects the conditioning of the stiffness matrix to cause larger maximum eigenvalues of the stiffness matrix and the outliers. An embedded 0 tangent direction and inflection point in M_2 (see Fig. 13(b)), cause M_2^{-1} to have a singularity at the corresponding value. The stiffness matrix conditioning is accordingly worse.

Since the maps M_i are unchanged in both h- and k-refinement, those modes isolate the effects to fewer, smaller elements, but alleviate this problem only to a limited extent, as we have seen in the above experiments. Generalized eigenvalue problems seem unsuitable candidates for use with models having control mesh degeneracies. Other analyses that are not sensitive in the same way may be suitable for use with models whose representations have multiple control points.

5.3. Drumhead problem

In this section we solve the 2D version of the generalized eigenvalue problem given in Eq. (11) on the disk centered at the origin with radius one and generate the normalized discrete spectrum. We use the three different disk representations developed as distinct surface completions to the circular boundary in Section 4.3.

The natural frequencies for the drumhead problem on the disk are the zeros of a Bessel function [31]. Using both quadratic and cubic degrees to investigate the effects of the single degree change, this study generates the spectra for D_1 and D_3. Thus, let $D_{i,j}$ indicate the ith representation in Fig. 5, and j indicates the degree.

Fig. 14(left) shows normalized spectra for the mappings $D_{i,j}$, all of which have undergone h-refinement using a uniform knot vector. It can be seen that $D_{3,j}$ produces a much flatter curve than the mappings $D_{1,j}$ and $D_{2,j}$. It also can be seen that an elevated degree has a negative impact on the result. The mapping $D_{2,3}$ performs the poorest. We speculate that this is related to non-uniform parameterization, because the elements that result under uniform h-refinement do not have uniform size and cannot represent the uniform spectral behavior as well. By referring to Fig. 14(right) the spectrum for the mapping $D_{3,2}$ is computed using a non-uniform h-refinement. The refinement process creates the elements for refined knot vector τ_2 from Section 4.2 in both parametric directions. The effect is to have larger elements near the circular boundary. The result is a flatter spectrum which has a maximum ratio of about 1.5, compared to approximately 2.2 obtained through uniform refinement.

5.4. Poisson equation on 2D domains

In this section we solve the Poisson Eq. (10) over four domains $\Omega_i \in \mathbb{R}^2, i = 1, 2, 3, 4$. Ω_1 is the unit square; Ω_2 is a 4×4 square; Ω_3 is the unit disc; and Ω_4 is the quarter annulus having inner radius

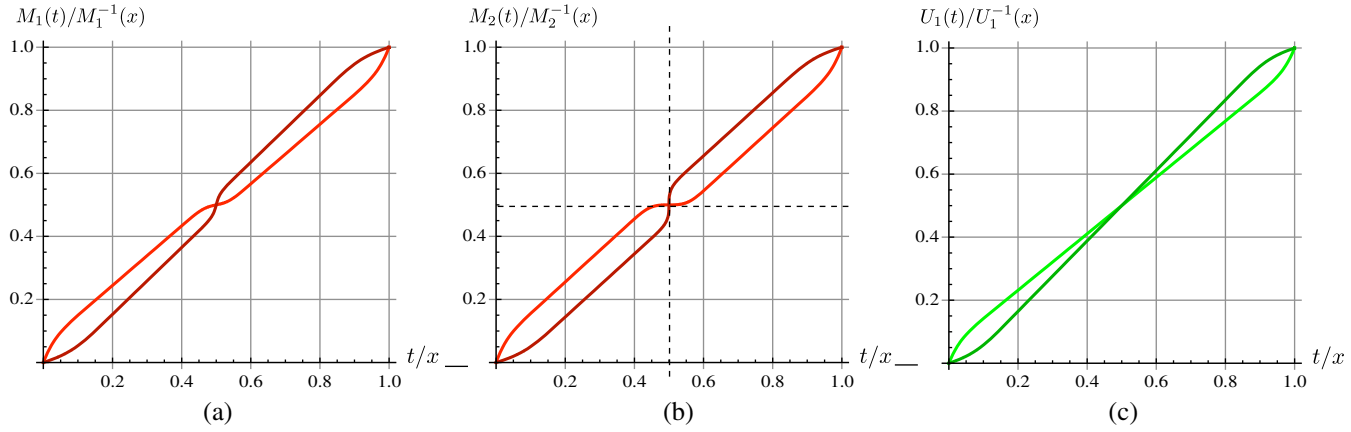

Fig. 13. (a) Degenerate mapping (red) with control point 1/2 duplicated and corresponding inverse mapping (darker red). The inverse has a high slope and therefore large first derivative at $t = 1/2$ causing a negative impact on the stiffness matrix conditioning (results in larger eigenvalues). (b) A triple control point embeds a horizontal tangent at $t = 1/2$ for M_2. Hence M_2^{-1} is singular at $t = 1/2$. However $t = 1/2$ is a knot so the singularity occurs only at an element boundary. (c) The comparison with U_1 (green) where slope of the inverse (dark green) is constant away from the boundary elements. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

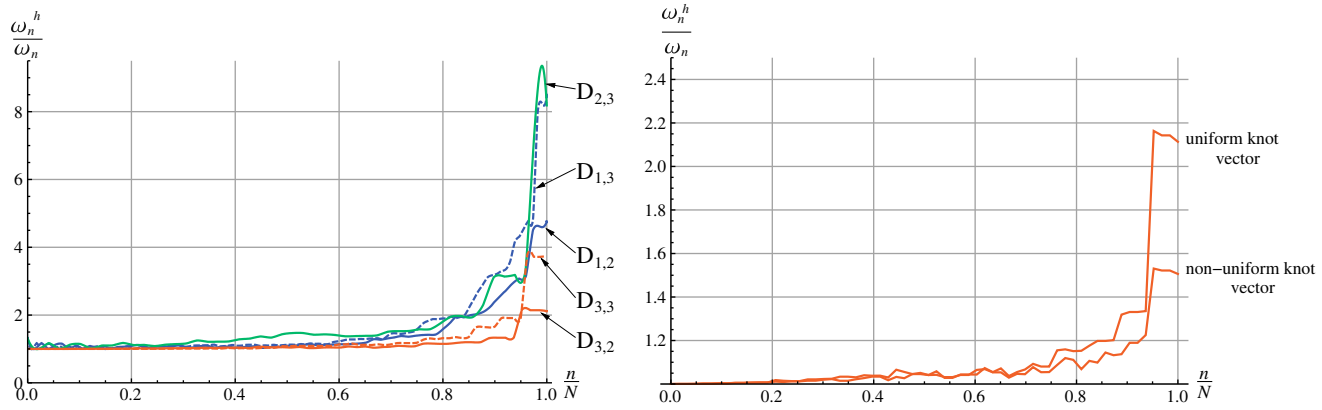

Fig. 14. (left): Normalized discrete spectra for different representations of a disc. (right): The disc model $D_{3,2}$ with a non-uniform knot vector, where refining knots are chosen as to create τ_2 (Section 4.2). It yields a flatter discrete spectrum.

one and outer radius of two. Although every domain is represented with an exact NURBS model, the completions from the boundary representations to the interiors differ. We consider relative aspects of mesh quality. We find a sequence of approximations u_h of the unknown solution u, such that $\lim_{h\to 0}\|u-u_h\|=0$. Thus, h controls the approximation quality and is related to the knot spacing, element size, and V_h.

We represent Ω_1 with two bicubic models $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$, each with degree $\boldsymbol{d}=(3,3)$ and knot vectors $\boldsymbol{T}=(\tau_1,\tau_1)$, where τ_1 the uniform open knot vector defined in Section 4.2. The coefficients for $\boldsymbol{S}_1$ are taken from the mapping U_1 and are uniformly spaced in each parametric direction. $\boldsymbol{S}_2$ uses coefficients from the identity map $\mathscr{I}$, and uses nodal representation to form the identity map on the unit square. Specifically, we define

$$\boldsymbol{S}_1(\boldsymbol{t})=\sum_{\boldsymbol{i}=1}^{\boldsymbol{n}}\left(\frac{i_1-1}{n_1-1},\frac{i_2-1}{n_2-1}\right)B_{\boldsymbol{i},\boldsymbol{d},\boldsymbol{T}}(\boldsymbol{t})$$
$$\boldsymbol{S}_2(\boldsymbol{t})=\sum_{\boldsymbol{i}=1}^{\boldsymbol{n}}\left(\tau^*_{1,i_1},\tau^*_{1,i_2}\right)B_{\boldsymbol{i},\boldsymbol{d},\boldsymbol{T}}(\boldsymbol{t}). \tag{41}$$

As in the 1D examples, $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$ differ in the choice of control points which results in the identity map or linear map $\boldsymbol{S}_2$, i.e., $\boldsymbol{S}_1(\boldsymbol{t})=(\boldsymbol{t})$ and in the nonlinear map $\boldsymbol{S}_2$. Fig. 15 shows the elements for $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$ for $n=m=10$.

Here the region Ω_2 is represented by the models $\boldsymbol{S}_3$ and $\boldsymbol{S}_4$, which are scaled and translated versions of $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$, respectively.

Then, Ω_3 is represented with the three different disc models D_1, D_2 and D_3, as in Section 4.3.

Finally, Ω_4 is represented using three different completions, $\boldsymbol{A}_1,\boldsymbol{A}_2$ and $\boldsymbol{A}_3$, shown in Fig. 16. All mappings are cubic in both parametric directions and use the same open knot vectors with interior knot multiplicity of two, i.e. all mappings are C^1. We use $\boldsymbol{A}_2$ and $\boldsymbol{A}_3$ to consider the differences between model quality criteria for meshes and how they apply to isogeometric models. They are generated from $\boldsymbol{A}_1$. The interior control points for $\boldsymbol{A}_2$ are slightly perturbed from those used in $\boldsymbol{A}_1$, creating analysis elements with wiggly boundaries. This situation could potentially arise from data-fitting algorithms or noisy data. Interior control points for $\boldsymbol{A}_3$ are chosen so that the knot-line curves are orthogonal where they cross. We still create perturbations so that element boundaries have some wiggles. Since one quality measure for isogeometric FEA meshes is that the element boundaries are orthogonal where they meet, this example sets up a similar scenario, but in the context of boundaries with wiggly sides.

In these studies we investigate Poisson's equations whose solutions are smooth analytical functions $u_i:\Omega_j\to\mathbb{R}$, where $i,j=1,2,3,4$ and defined as,

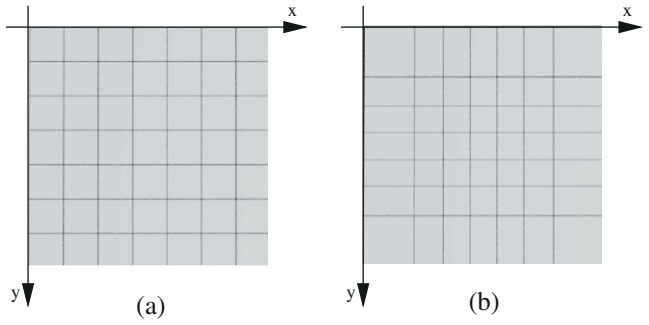

Fig. 15. Both NURBS models represent Ω exactly. $\boldsymbol{S}_2$ on the left is the identity map; the control points in $\boldsymbol{S}_1$ on the right are uniformly spaced giving a nonlinear parameterization.

$$u_1(x,y)=\sin(2\pi x)\sin(2\pi y), \tag{42}$$
$$u_2(x,y)=\frac{25\left(-\frac{1}{e^{100}}+e^{-12.5(x^2+y^2)}\right)}{2\pi}, \tag{43}$$
$$u_3(x,y)=J(4,J_{zero}(4,2)r(x,y))\sin(4\theta(x,y)), \tag{44}$$
$$u_4(x,y)=\sin(2\pi r(x,y))\sin(2\pi b(x,y)), \tag{45}$$

where

$$r(x,y)=\sqrt{x^2+y^2}-1 b(x,y)=\frac{2\arccos(x/\sqrt{x^2+y^2})}{\pi} \tag{46}$$

and, $\theta(x,y)$ defines the angle between the vector (x,y) and the Cartesian coordinate axes. In other words $r(x,y)$ and $\theta(x,y)$ convert the Cartesian coordinate (x,y) into the polar coordinate $(r(x,y),\theta(x,y))$. Furthermore, $J(n,z)$ is the nth Bessel function of the first kind at $z\in\mathbb{R}$, and $J_{zero}(n,m)$ is the mth zero of the nth Bessel function of the first kind. u_3, selected to be the $(4,2)$th mode of vibration of a drumhead [31], evaluates to zero on the boundary of Ω_3.

In experimenting we formulated five test cases involving solving the Poisson Eq. (12). They are shown in the table below.

Case	Physical space	Ideal solution	Mapping
1	Ω_1	u_1	$\boldsymbol{S}_1,\boldsymbol{S}_2$
2	Ω_2	u_2	$\boldsymbol{S}_3,\boldsymbol{S}_4$
3	Ω_3	u_1	D_1,D_2,D_3
4	Ω_3	u_3	D_1,D_2,D_3
5	Ω_4	u_4	A_1,A_2,A_3

Cases 1, 2, 4, and 5 use $u_i|_{\partial\Omega}\equiv 0$. Nonzero Dirichlet boundary conditions for Case 3 are chosen by a nodal interpolation of u_1.

We use uniform h-refinement, placing a new knot in the middle of every parametric knot span. The elements for building the approximate solution then are split in turn. The shape of the new elements is not an exact half split of the old because the mappings are not affine. Suppose there are m_ℓ control points after ℓ levels of h-refinement. The isogeometric solution after ℓh-refinements to u_i is denoted $\hat{u}^\ell_{j,i}$ for test case j result. The error is computed as

$$\epsilon_{\infty,j,i,\ell}=\sup_{\boldsymbol{x}\in\Omega}|\hat{u}^\ell_{j,i}(\boldsymbol{x})-u_i(\boldsymbol{x})|. \tag{47}$$

A piecewise linear convergence curve is defined by the points $(\sqrt{m_\ell},\epsilon_{\infty,j,i,\ell})$. In the following discussion, *good* or *poor* convergence means that the negative slope of the convergence curve is higher or lower, respectively, compared to the convergence of a different mapping. While the analysis solutions converge, as predicted, under h-refinement, these studies are aimed at determining how many levels of refinement are necessary before the asymptotic behaviors dominate. If the same quality result can be obtained on a simpler mesh, that situation is always desirable. We discuss each test in more detail.

We refer the reader to Figs. 17–19 that show log–log plots of the convergence curves for the five performed tests. For purposes of illustration, next to each log–log plot the exact solution u_i on the respective domain Ω_n is shown.

Test 1: Initially, both mappings have almost the same error. During the initial levels of refinement the linear map $\boldsymbol{S}_2$ has a better convergence rate than the nonlinear map $\boldsymbol{S}_1$. Then, both mappings converge with the same rate, i.e., the convergence curves become parallel. The error curve of $\boldsymbol{S}_2$ is below the curve of $\boldsymbol{S}_1$ by the respective offset, which means that the error of $\boldsymbol{S}_2$ is about an order of 10 smaller than the corresponding error of $\boldsymbol{S}_1$. So the convergence behavior of the Poisson problem is better under the identity map, differing from the better convergence behavior for $\boldsymbol{S}_1$ for the eigenvalue problem.

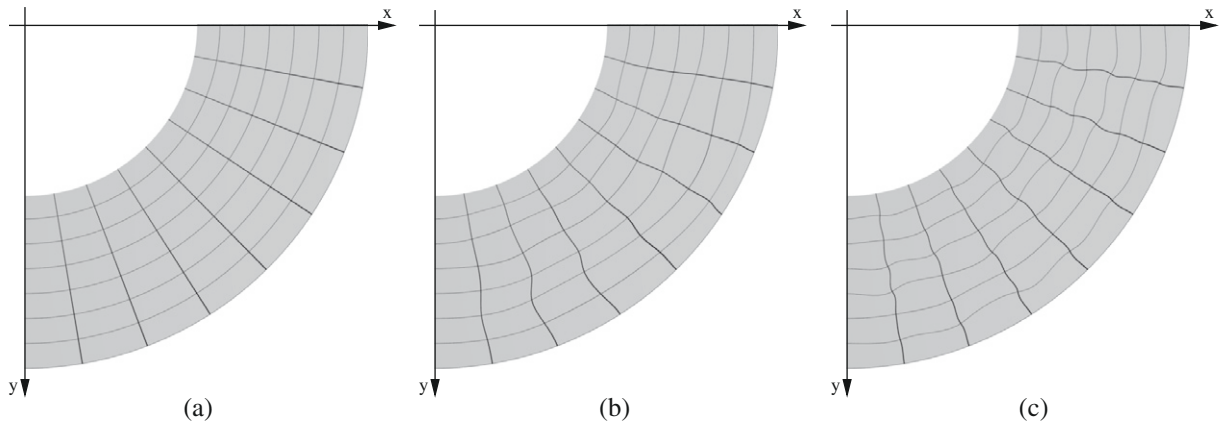

Fig. 16. Three exact representations of the boundary of a quarter annulus with an inner radius of one and an outer radius of two with different completions. In (a) a high quality representation, $\boldsymbol{A}_1$ is shown. In (b), control points for $\boldsymbol{A}_2$ in the interior are created by slightly perturbing those of $\boldsymbol{A}_1$. In (c), the control mesh points for $\boldsymbol{A}_3$ are chosen so element boundaries are orthogonal where they cross.

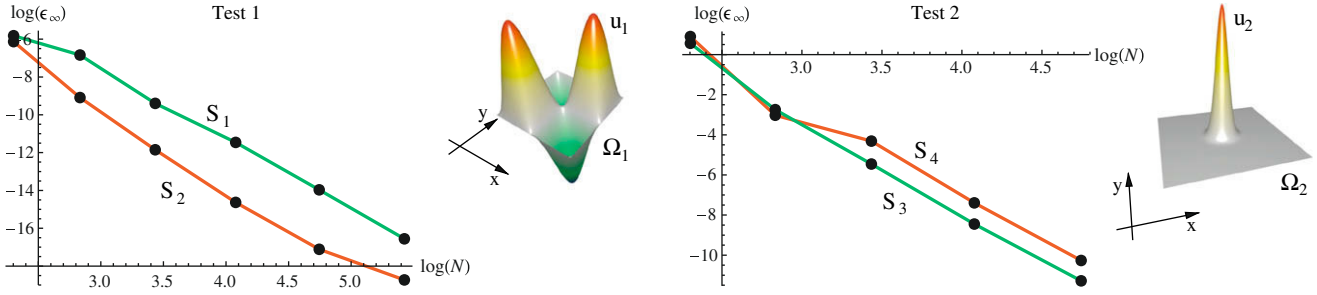

Fig. 17. The plots on the left show the error curves for different mappings, on which the exact solution on the respective domain is shown on the right.

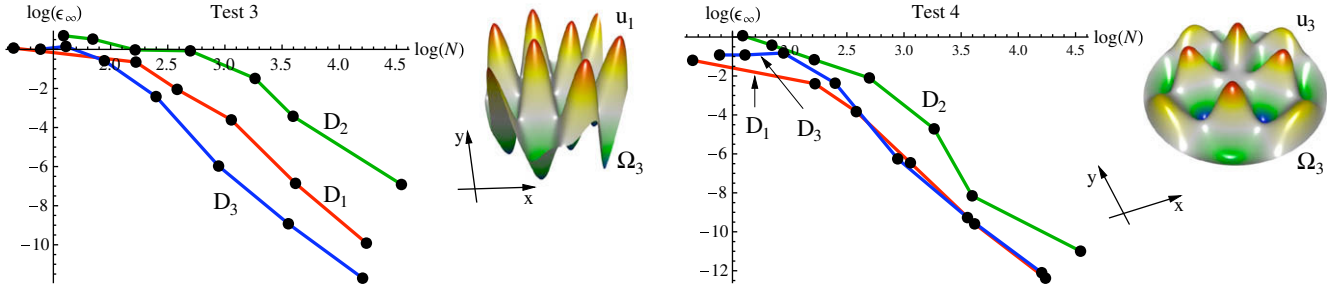

Fig. 18. Left: The error curves for different mappings; Right: The exact solution on the respective domain.

Test 2: A similar observation can be made for this test. Initially, in the second refinement step, both mappings have the same error. Then, in the subsequent refinement step, the error for the linear map $\boldsymbol{S}_4$ decreases slower. However, in further refinement steps, both error curves have the same slope. For this problem, contrary to Test 1, the nonlinear map $\boldsymbol{S}_3$ performs better. This is understandable inasmuch as the approximating space has greater basis function density near the center of the regions.

Test 3: During the first refinement steps, all three NURBS models converge rather slowly, although D_1 exhibits the least error. After about the fourth refinement step, the error for D_3 drops the most, and drops the least for D_2. Thereafter, all mappings converge at the same rate, where the error of D_1 is about an order of ten larger than D_3 and D_2 is about four orders of ten larger than D_3. This case has nonzero boundary conditions. Given the same number of functions to approximate u across all three mappings, D_3 will have a higher percentage that are nonzero on the boundary and so can be used to better approximate the true solution on the boundary. Each time both parameters are uniformly refined, D_1 only doubles the number of nonzero basis function on the boundary. But D_3 quadruples that number. So, for D_1 it would be better to h-refine more in the polar direction that in the radial direction to better approximate the boundary.

Test 4: The error curves look similar to those in Test 3. Initially, all mappings converge slowly; however, the curves of D_1 and D_3 start to overlap, while D_2 lags a bit to reach the same rate of convergence.

Test 5: In the final test, the initial errors for all mappings are different, with A_1 having the smallest error. Except for an initial slower convergence with A_2, all three mappings soon converge at the same rate, where the offsets between A_1 and A_2, and

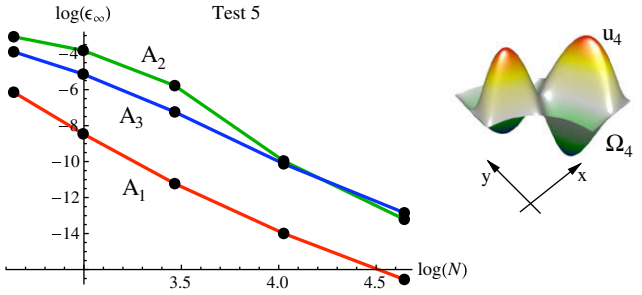

Fig. 19. Left: The plots show the error curves for different mappings. Right: The exact solution on the respective domain.

between A_1 and A_3 are considerable. In this case, the wiggles contributed to the problems for A_2 and A_3, the better behavior near the element boundary corners for A_3 alleviates some of the problems. So, although not necessary for convergence, the smoother sided elements of A_1 appear to help convergence.

Note that models $\boldsymbol{S}_1$ and $\boldsymbol{S}_3$ have the same stretching properties. The same holds for $\boldsymbol{S}_2$ and $\boldsymbol{S}_4$, where no stretching occurs due to the linear mapping properties. However, in Test 1, the linear map $\boldsymbol{S}_2$ performs better than the nonlinear mapping. In Test 2 the nonlinear map $\boldsymbol{S}_1$ performs better than $\boldsymbol{S}_2$. The reason for that lies in the different properties of the known solutions u_1 and u_2 as shown in Figs. 17 and 18. u_2 is everywhere zero but has a peak around the origin. $\boldsymbol{S}_3$ performs better because more elements are pulled into the center of the domain (see Fig. 15 (left)). Similar to the 1D case in Section 5.2, having the determinant of the Jacobian have value near 1 does not necessarily mean that a mapping performs better than a mapping which involves distortion, at least for issues not related to conditioning of the stiffness matrix. In the case of Test 1, however, the stretching leads to the neglect of other important regions, and therefore $\boldsymbol{S}_2$ performs better, because it more uniformly distributes the elements over the domain, a characteristic that treats the known function u_1 more favorably.

In the first three tests all mappings have roughly the same initial error, i.e., the convergence curves emerge from roughly the same starting position. This is not the case for Tests 4 and 5. In Test 4, D_1 has the smallest initial error, in Test 5, A_1 has the smallest error. Even though at a certain refinement stage, all mappings have the same asymptotic error estimation, the initial error shows that if a model is created with care, refinement steps can be saved, attaining better analysis with fewer elements. In case of Test 5, with the given true solution, wiggles or other perturbations have a negative impact on the initial error. Note that wiggles and other perturbations which were artificially introduced in our test models are common modeling artifacts and often unintentionally occur in practice during data-fitting.

5.5. Linear elastic deformation of a volumetric model

As a final example we examine the isotropic linear elastic deformation of a human femur. The femur is modeled with the methodology proposed in [25] and introduced in Section 4.3.3. The modeling input is an exterior and interior boundary triangle mesh, where the volume between the exterior and interior represents the cortical bone and the volume of the interior boundary represents the trabecular bone. From that, a single B-spline volume with proven approximation power is created. The model is C^2 but has a degeneracy along the cylindrical axis. The placement of the axis is user-guided, chosen depending on the model and simulation parameters. Such a model is difficult to decompose in multiple patches because of the following reasons. Patch boundaries may not be planar, and gluing them with certain continuity is difficult. Furthermore, since the object consists of different materials it is not clear how to represent it with multiple patches. Creating a single B-spline volume introduces more deformation in the geometry, but it avoids both problems, i.e. it does not involve any patch gluing, and respecting material attributes in the parameterization can be achieved without tremendous effort. Another innovative aspect of this model is that the angular parameterization is periodic. Unlike all previous modeling studies done for isogeometric analysis, the resulting representation takes advantage of the characteristics

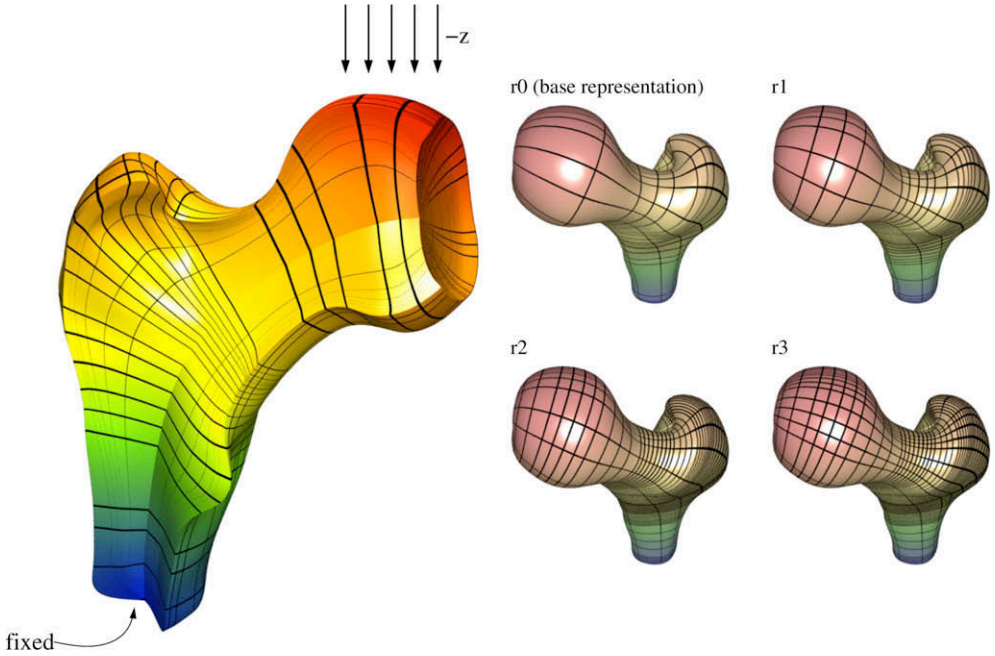

Fig. 20. Linear elastic deformation of a femur.

of periodic B-splines to ensure that the model is C^2 everywhere. Even more important, it means that V_h has only C^2 functions except at the natural bone boundary and the skeletal axis.

Both, cortical and trabecular part, use a Poisson ratio of 0.9. A Young's modulus of $17 \times 10^9 (\mathrm{N/m^2})$ is applied to the more solid cortical part; $100 \times 10^6 (\mathrm{N/m^2})$ is applied to the trabecular part. The boundary between the cortical and trabecular materials is an isoparametric surface of the trivariate B-spline representation.The bottom of the femur is held rigid in the x, y and z directions. Load is distributed over the apex of the femur in the direction of the base of the femur. Fig. 20 portrays the femur where magnitudes of the displacement are visualized.

Fig. 20 also shows h-refined versions r_i of the femur. r_0 is the initial representation which is a data-reduced version from the original representation. r_1 and r_0 are created by uniform h-refinement in u, where r_1 is created from r_0 and r_2 is created from $r_1 \cdot r_3$ and r_4 (not shown) are created by uniform h-refinement in v and w, respectively. Linear elasticity is applied to each of the representations with the same material parameters and boundary conditions resulting in the five solution representations $s_i \cdot s_i$ is compared to s_{i+1} by applying the respective refinement to s_i. The error $\boldsymbol{e}_i$ is the maximum L2-norm of the difference of the coefficients between s_i and s_{i+1} : $\boldsymbol{e} = \{1 \times 10^{-4}, 2 \times 10^{-5}, 3.8 \times 10^{-6}, 2 \times 10^{-7}\}$.

5.6. Result summary

In Section 5.2 we discussed different knot vector and mesh choices for the longitudinal vibrations of a 1D rod. It was shown that non-uniform knot vectors can improve the results and avoid outliers. Then, we examined the 2D version of this problem, the drumhead problem, completing the disk with different representations of the boundary. While none of the completions would have any effect on the design of the shape, the completion had strong effects on the normalized discrete spectrum.

Then, in Section 5.4 the Poisson equation was solved on different domains for different smooth known functions, where the domains were exactly represented with alternative choices of more or less common NURBS models. As shown in [32], sufficient h-refinement eventually converges to these cases where there is a starting NURBS volumetric model that meets certain criteria on the knot spacing and the Jacobian of the mapping. That is, no matter what regular mapping was initially chosen to represent a domain Ω, eventually the error decreases at the same rate. However, the refinement level at which a mapping reaches the asymptotic behavior and the resulting offset compared to other models, initially chosen by the modeler, allows us to make judgements about the quality of the model which could be observed in the above test cases. By a more careful model design, a design appropriate for analysis, computation time and efforts to refine a model can be saved.

We conclude that boundary modeling and boundary completion techniques can enhance the speed of convergence and quality of the results.

6. Analysis-aware modeling: considerations and issues

Now that we have provided a view of modeling from the modeler's perspective (Section 4) and have demonstrated the impact of modeling choices on numerical approximation of the solutions for several prototype problems (Section 5), we now try to infer a collection of considerations or "issues" of which it would behoove both the modeling and analysis communities to be aware.

Since we are discussing models for isogeometric analysis, we assume the geometry is represented by parametric functions that are either NURBS or a variant of NURBS called T-splines (or T-NURCCS – collections of T-splines into a mesh). Although many of the underlying behaviors of NURBS are applicable to T-NURCCS, we discuss only model characteristics for NURBS. Further most of the studies are performed on simple geometries since the exact solutions are known for those cases.

6.1. Issue: model completion

At this point, commercial CAD systems do not complete boundary representations of models in any general way to form full volume representations although there has been sporadic past research into this topic. There are multiple modeling operations and styles of modeling that can lead to the same boundary geometry. For example, a boundary model of a cylinder might be created by (1) representing a circle, (2) creating the sides of the cylinder by *extruding* the circle along a straight line path, (3) creating representations of two discs, (4) transforming the discs into the correct geometric location to form the cylinder top and bottom, and (5) communicating to the data structures that the appropriate edges of the side of the cylinder and the disc are the same. In this construction there is no parametric association between the parameterization of the ends of the cylinder and the sides, but the CAD system topology keeps the computations synchronized. This is lost when the representation is exported. Another representation can be created by representing a radius of the top of the cylinder, its associated side, and the associated bottom of the cylinder. Then that curve is rotated around the axis of the cylinder to form a boundary model whose parameterization is related to cylindrical coordinates. Neither of the above lends itself to completion to a full volume representation. Another approach discussed in Section 4.3.3 is suitable for extensions to full volume models. More complex geometries with shape irregularities are created employing a variety of design operators combined with boolean operations and hierarchical structuring leading to model representations for which there are no standard, generally applicable techniques to transfrom them to full volume models. Models derived from distinct design ontologies, like automotive modeling and aerospace aerodynamic modeling require different model completion operations. Analogously, suitable completions depend on the type of analysis being performed. Others have attempted to exploit mappings and mapping properties to improve solution quality or decrease undesirable numerical artifacts in non-isogeometric analysis methods, and in classical, higher order FEA [8,9,33], but on a case-by-case basis. Isogeometric analysis can incorporate such completions as a natural part of the modeling/analysis process.

Above all, demand will drive the development of new methodologies to deal with this. However, until embedded in CAD systems, the designer/analyst must make crucial decisions on making a boundary model into a volume model. For many complex boundary representations it is difficult to design a single model completion, although many are theoretically possible. Reconsidering the 2D boundary in Fig. 3(a) we observe that there is no natural approach for dividing it into mappings of multiple unit squares that meet smoothly, as a designer would want. Even if there existed such a solution, images on opposing sides of the unit square would not have shared parameterizations, or knot vectors. Some approaches for completing the interior could involve the medial axis, but many of these methods can have difficulties with parameterizations (defining the completion) near where medial axis branch points meet and around medial axis endpoints. This is a type of *paving* technique that can be made to work for the discrete case in 2D but is not generally suitable for 3D. In [15] a technique is presented for forming a NURBS representation of the interior for planar regions, one that automatically generates $\boldsymbol{F}$, appropriate mappings of collections of mappings from unit squares to the boundary and interior, to complete the interior. The resulting ϕ's

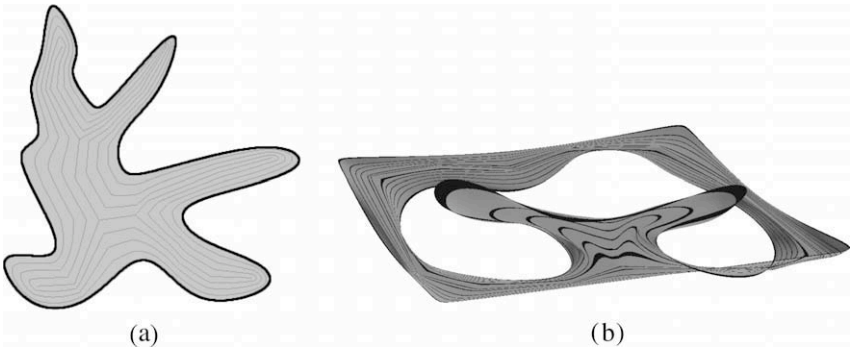

Fig. 21. (a) Filled boundary curves in 2D; (b) Filled boundary curves in 3D.

and V_h are C^0 across the boundaries of the images of the squares. But the curvilinear rectangular elements of the interior become degenerate square–triangles at the medial axis branch endpoints (see Fig. 21(a)). A related approach is aimed at creating smooth interiors for nonplanar boundary curves. Again the representation for the complex boundary curves does not lend itself to straightforward partitioning to create well-formed quadrilaterals. The approach creates a modified moving front approach to guarantee a well formed smooth surface completion to the boundary. The scheme builds on a form of simplified medial axis (see Fig. 3(b)) and its completion in Fig. 21(b). However, these approaches are not directly applicable to volume completions since the medial axis has surface sheets that would have different parameterizations, depending on which limiting boundary surface it considered.

6.2. Consideration: non-uniform knot vectors

Recall that geometric shape design is frequently a hierarchical process. That is, the overall shape is created and then finer shape features are designed into the model by adding degrees of freedom through knot insertion only in the region in which it is required. This is like a kind of localized, non-uniformly spaced *h*-refinement that keeps the geometric shape unchanged. Then the designer modifies coefficients locally to attain the needed shape. This is performed either by the CAD system automatically within shape modeling operators or by the designer during manual design operations. The process leads to highly non-uniform knot vectors. At the same time, the element size is not constrained by the knot interval size, although quite frequently the degrees of freedom are added in order to represent small, high frequency details. This affects both the characteristics of the isogeometric elements and the reference space that is generated, as well as its ability to represent a good approximation, since Δ_τ can be quite large.

Let $\{V_h\}_h$, denote the sequence of approximation spaces for a given NURBS geometry created from an initial approximation space by successive applications of *h*-refinement. It has been shown [32] that the approximations always converge to the true solution, when the knot vector is open uniform and *h*-refinement is applied globally each time. Each element is divided into 2^s elements in each pass. It is assumed that the Jacobian of $\boldsymbol{F}$ meets certain conditions as well. The constant related to convergence depends on both the parametric and physical shapes of all the elements. If one could remove the worst element shape behavior from affecting the convergence rate, one would be left considering the mesh widths. Results on univariate behavior [34] indicate that, excluding the behavior of $\boldsymbol{F}$ and $\boldsymbol{J_F}$ on convergence, convergence depends on the largest mesh widths. If *h*-refinement is applied to all elements, then smaller elements that already have high quality approximation power may get excessively refined in the pursuit of enhancing the approximation power of the whole mesh. One approach to this problem is to perform *h*-refinement adaptively. Since $\mathbb{S}$ is tensor product, this becomes more time consuming to do as the dimension *s* increases. Note that V_h is not tensor product since it depends in the inverse of $\boldsymbol{F}$.

6.3. Proposed guideline

Instead of performing *h*-refinement uniformly over the whole model, the guideline would indicate to insert possibly multiple knots in the largest knot intervals in order to reduce the size of $\overline{\Delta}_\tau = \max_i\{\tau_{i+1} - \tau_i\}$ in the refined analysis. Effectively this is an adaptive refinement based solely on parametric space convergence properties.

6.4. Issue: controlling mesh degeneracies

There are multiple types of control mesh degeneracies that can occur as part of the modeling process. Consider the case in which the knot vectors are uniform open. Locating the control points to affect certain geometries can lead to Jacobians with value zero, and even changes of sign of the Jacobian within a patch. Other types of degeneracies, such as multiple control points, may or may not cause a problem with the Jacobian. It is valuable to be aware of these problems in order to constrain them to existing only on the boundaries of the elements. Under the assumptions that the regions of zero Jacobian are a set of measure zero and that the induced function space is well-behaved (i.e. integrable) in physical space, integration such as done in collapsed-coordinates for spectral/*hp* elements should be applicable [9]. It is important to stress though that the flexibility that comes by allowing such degeneracies does not come without some price (in terms of things such as rate of convergence or computational complexity), and thus the analysis-aware modeler or modeling-aware analyst must be cognizant of how such choices impact analysis.

Consider the curve case. $\boldsymbol{F} : [0,1] \to [0,1]^q$, of degree d and a knot vector τ with simple interior knots, $\boldsymbol{F}(t) = \sum_{i=1}^{n} \gamma_i B_{i,d}(t), \gamma_i \in \mathbb{R}^q$.

Suppose $\gamma_{k-d} = \cdots = \gamma_{k-1}$ for some k. Let $p := k - d$. Then

$$\boldsymbol{F}(t) = \gamma_p(1 - B_{k,d}(t)) + \gamma_k B_{k,d}(t), \quad t \in \theta_k := [\tau_k, \tau_{k+1}).$$

Thus $\boldsymbol{F}$ reduces to a straight line on θ_k. Similarly,

$$\boldsymbol{F}(t) = \gamma_{p-1}(B_{p-1,d}(t)) + \gamma_p(1 - B_{p-1,d}(t)), \quad t \in \theta_{k-1} := [\tau_{k-1}, \tau_k),$$

is a straight line on $\theta_{k-1}, \boldsymbol{F}(\tau_k) = \gamma_p$, and the Jacobian will change sign at τ_k unless $\gamma_{p-1}, \gamma_p, \gamma_k$ lie on a straight line. If in addition $\gamma_k = \gamma_{k-1}$ then over all of θ_k, $\boldsymbol{F}$ evaluates to a single point. In Section 5.2.1 we present a study to see the effect that a degeneracy of this type has on the geometric elements in Ω, the basis for V_h, on the approximations, and on eigenstructures.

Even if the Jacobians are nonzero, control mesh degeneracies tend to induce distortions in the elements, some of which may be

desirable. It is important to note, however, that the resulting model representations may not have sufficient regularity to compute principal curvatures, principal directions, umbilics and crests. While not in general necessary for the analysis problem, these characteristics may be vital for computing assembly planning and manufacturing process planning on the volume boundary model.

As another example of a degeneracy, let $\boldsymbol{F}$ represent a univariate cubic ($d = 3$) geometry model with exactly one doubled control point $\boldsymbol{c}_k = \boldsymbol{c}_{k+1}$, defined over an open knot vector $\boldsymbol{\tau}$. The effects of the double control point on $\boldsymbol{F}'$ show up for $t \in [\tau_{k+1}, \tau_{k+4})$ It is in those three intervals that $\boldsymbol{F}'$ has only two nonzero terms. Thus, $\boldsymbol{F}'|_{[\tau_{k+1},\tau_{k+2})}$ lies on a straight line and never is 0. Uniform h-refinement is performed and applied to the definition of $\boldsymbol{F}$ to create a new sequence of children coefficients that in the interior generically looks like

$$\boldsymbol{c}_i^1 = \begin{cases} (\boldsymbol{c}_{j-2} + \boldsymbol{c}_{j-1}), & i = 2j-1, \\ (\boldsymbol{c}_{j-2} + 6\boldsymbol{c}_{j-1} + \boldsymbol{c}_j)/8), & i = 2j, \end{cases}$$

leading to the following children coefficients near the original double point.

$$\begin{aligned} &\vdots \\ &\boldsymbol{c}_{2k}^1 = (\boldsymbol{c}_{k-2} + 6\boldsymbol{c}_{k-1} + \boldsymbol{c}_k)/8, \\ &\boldsymbol{c}_{2(k+1)-1}^1 = (\boldsymbol{c}_{k-1} + \boldsymbol{c}_k)/2, \\ &\boldsymbol{c}_{2(k+1)}^1 = (\boldsymbol{c}_{k-1} + 7\boldsymbol{c}_k)/8, \\ &\boldsymbol{c}_{2(k+2)-1}^1 = \boldsymbol{c}_k, \\ &\boldsymbol{c}_{2(k+2)}^1 = (7\boldsymbol{c}_k + \boldsymbol{c}_{k+2})/8, \\ &\boldsymbol{c}_{2(k+3)-1}^1 = (\boldsymbol{c}_k + \boldsymbol{c}_{k+2})/2, \\ &\vdots \end{aligned}$$

Notice that h-refinement has split the control points so that there is no longer a double control point. However, observe that $\boldsymbol{c}_k$ is a control point again and there are two new control points on the line segment connecting $\boldsymbol{c}_{k-1}$ and $\boldsymbol{c}_k$ and two more new control points lie on the line segment between $\boldsymbol{c}_k = \boldsymbol{c}_{k+1}$ and $\boldsymbol{c}_{k+2}$. Each subsequent level of refinement will always have two sets of three control points positioned in this way, converging towards each other and to a point on the curve. The curve in that region looks like a quadratic shape that has nonlinear velocity. Each $\psi \in \mathbb{S}$ has uniform knots (except near the ends), but the ϕ do not behave that way. Rather they tend to cluster near the θ's that have double points.

In higher dimensions this effect is softened. This can be used on the bounding surface of a model to approximate the behavior of a fillet or chamfer, especially when used with NURBS, by allowing part or all of a row in the surface mesh to merge with part or all of the next row. Then the corresponding internal "rows" of the mesh can be more uniformly spread out.

Another way to use this property is to cluster more ϕ functions in regions that are expected to need a greater density of approximating functions. If this is done, other regions will have reduced approximating capabilities without more refinement.

6.5. Proposed guideline

Use of degeneracies in the mesh do not necessarily lead to poor *shape* behaviors or poor approximating spaces in physical space. The analysis is much more efficient if these degeneracies are avoided in the initial control structure of the volume. The modeler should be aware of these issues and their impact on analysis that might follow. However, as was be demonstrated in Section 5, the impact depends on the FEA being performed. However, the analyst who is making the model should visualize some of the most clustered ϕ's and check that they are in accordance with the expected density needs for approximating functions in solving the PDE, and the magnitude of the Jacobian in that localized part of space.

6.6. Issue one or many s-cubes for a reference domain

Sometimes there exists an option to use multiple s-cubes or a single s-cube for the reference domain. A difficult issue is decomposing the shape into images of s-cubes to create a compound mapping $\boldsymbol{F}$ that exhibits the best Jacobian behavior while honoring the shape and analysis requirements. It is frequently rather difficult to generate an $\boldsymbol{F}$ based on only one s-cube. Even when possible, it may lead to some distortions in the resulting parameterization causing Jacobians that are suboptimal. In the study in [17] involving a hyperboloid of revolution of constant cross section, two choices embodying each possibility of final representation are presented. It is only with knowledge of what should be expected from analysis that it is possible to decide which choice is superior. Some modeling operations will result in shapes better parameterized and completed to full volumetric representation for isogeometric analysis. Countervailing issues are the distortions in the parameterization affecting the quality of $\boldsymbol{F}^{-1}$ and its Jacobian and the difficulty of decomposing the shape into appropriate images of the s-cube (including the model completion issues). It is only with experience that such decisions can be made.

Proposed guideline: Develop "modeling for isogeometric analysis" lore and transform it into design rules that can be embedded into templates for families of parts, and templates and design rules for assembling them together, and insuring that V_h is C^0 across shared boundaries of s-cubes. This was done in [22,17] for modeling and analysis.

6.7. Issue: F may represent the geometry but have parameterization problems for analysis

Historically, shape representation has typically been concerned with its geometric form fidelity, instead of the quality of parameterization. This can easily arise when a surface comes about as the result of completing the surface from boundary curves. The problem becomes more complex in completing the volume from surfaces. One way of addressing this issue is to redesign the representation to approximate the original representation, but to have better properties for analysis. Using facets on the boundary and the interior, finite element analysis remeshing tends to do this. Creating a new representation is an intrinsic aspect reverse engineering approaches to model recovery. Hence adding analysis awareness to this semi-automatic process is auspicious for generating models better suited for both purposes, design and analysis.

6.8. Consideration: analysis-aware parameterization, knot selection, and representation

The most important decisions regarding whether a representation can be used successfully for both shape and analysis are usually made "uphill" in the design cycle, before consideration of analysis. Design feature curves are sometimes not the same as analysis feature curves. If a single representation is to be amenable to supporting both design and analysis, it must also be extensible to serve the needs of the other engineering activities such as fabrication because current CAD-models already provide limited but broad support for many diverse processes. Representations for each engineering methodology that have been developed in isolation are nearly invariably incompatible with each other. Part of the new isogeometric analysis cycle will be to create analysis aware representations whenever this can be done without representational conflicts with other engineering methodologies in design.

Some representation techniques do not lead to $\boldsymbol{F}$ becoming one-to-one. In models from CAD, such singularities occur only on the boundaries of elements, or when there are control mesh degeneracies. Control mesh degeneracies are discussed separately. Other types of boundary problems occur when an object like a disc, cylinder, or sphere is modeled in a way that transforms the Cartesian patch into a more polar/cylindrical/spherical coordinate system, as demonstrated in Section 4.3.

6.9. Consideration: generation of modeling quality metrics

In traditional FEA, it is well acknowledged that the choice of initial mesh greatly impacts the quality of the results. In the best case, the quality of the mesh impacts the constants that exist in the asymptotic error estimates and determines the level of refinement at which the asymptotic regime begins. In the worst case, the mesh quality under adaptive refinement leads to a successively worsening ill-conditioned system, one that ultimately renders the computations irrelevant for engineering purposes. In the case of finite element analysis, mesh quality is therefore an important issue when generating meshes that will be used in engineering analysis. Because of this reliance of analysis in practice on mesh quality, there is a large body of the literature addressing the subject. In particular, the reader is referred to [35] and references therein. Given a mesh, the question has to be raised, whether its quality is appropriate so that successful analysis can be applied to it. Application of h-refinement or knot insertion does not improve the initial mesh quality. In order to improve the quality of the initial mesh, tremendous effort has been devoted to generate high quality meshes used in finite elements. Generally, schemes are used which relocate the vertex (or node) positions without altering the connectivity of the mesh.

The important principle to distill from the mesh quality discussion is that the criteria that have been developed for assessing mesh quality are actually, in essence, guidelines aimed at optimizing the space of functions available for approximating the solutions to classes of analysis problems, albeit there exist specific analysis problems defined over specific domains for which meshes that are not compliant with those criteria can give better solutions (e.g. anisotropic meshes used in the study of hyperbolic problems [27–29]). The tacit goal is to optimize the function space of the approximations to the geometry and analysis at hand.

Suppose $\mathbb{H} = \{H_i\}_{i=1}^N$ is a collection of curvilinear hexahedra that intersect only along boundaries in a manner that the intersection is either null, a corner, an edge curve, or a complete shared curvilinear face, and $\bigcup_{i=1}^N H_i$ is an approximation to Ω.

In finite element isoparametric analysis, each H_i has straight edges and bilinear faces, and creates N bijective mappings $\boldsymbol{F}_i : [0,1]^s \to H_i$, that is, it uses N copies of a single reference template domain, say, $\theta_i = [0,1]^s, i = 1, \ldots, N$. Each $\boldsymbol{F}_i$ is smooth on $(0,1)^s$, the interior of the template. $\boldsymbol{F}$ is defined piecewise. Each collection $\{\psi_{kj}\}_j, k = 1, \ldots, N$ consists of polynomials defined over the reference element to a single H_k. Traditionally the ψ_{kj}'s are s-linear, i.e. linear in each of the s-dimensions separately. In high-order FEA, this is extended to becoming d-polynomial per dimension. $\boldsymbol{F}_i$ is usually a linear map in each variable separately, even for high-order FEA (e.g. solutions use piecewise high-order polynomials as bases that are C^0 across hexahedral boundaries), but the geometry is mapped only piecewise trilinearly (again C^0). So, the compositions $\psi_{kj}(\boldsymbol{F}^{-1})$ remain polynomials for tetrahedra and for planar-sided rectilinear hexahedra and the solution space is piecewise C^0 polynomials. Unfortunately creating a better approximation space for the PDE solution usually requires generating a new collection $\mathbb{H}$ that represents the boundary geometry and interior with higher fidelity.

Isogeometric analysis, while performed on more highly structured collections of elements, creates spaces for the approximate solution that can be tailored to both the geometry and the particular analysis while maintaining a desired degree of smoothness in the solution approximation spaces. Typically there are just a few parametric domains required to represent Ω and the bounding surface geometry can be represented exactly. In this case $H_i = F(\theta_i)$ can have curvilinear faces and edges. While the bases for V_h remain local, the trade-off for attaining higher smoothness is that the support of each basis function grows into adjoining elements. Thus the stiffness matrix is a bit less sparse and its computation is a bit more complex, and the resulting analysis elements are more structured.

The factors that enter into model quality are the spaces $\mathbb{S}$ chosen for each Θ, the mapping $\boldsymbol{F}$ chosen to represent the geometry, and the particular partial differential equation that must be solved. Since $\boldsymbol{F}$ is almost never s-linear, V_h is not a piecewise polynomial space, even though we can analyze its behavior in terms of $\mathbb{S}$ and $\boldsymbol{F}$. Because $\boldsymbol{F}$ represents the physical space exactly, and because $\mathbb{S}_{\boldsymbol{d},\boldsymbol{T}}$ defines $V_h, h-, p-$ and k-refinements are straightforward. It may be desirable to enrich the space for approximating the solution, but there is never a need to improve the quality of the geometry.

Issues that will confront the analyst in creating good models were raised in Section 2. Although it has been shown [32] that isogeometric analysis solution approximations converge under h-refinement, under certain conditions regarding $\boldsymbol{F}$ and $\mathbb{S}$, the constant in the convergence result and the number of refinements required to reach asymptotic behavior can vary widely. Furthermore, many geometric representations of models generated by CAD systems do not meet the conditions required by the convergence theorem on their boundaries, and almost never can be easily completed into a full volume model. We discuss the issues and propose guidelines in the context of unifying CAD (the shape designers) and CAE (the analysts) for creating a more (isogeometric) analysis-aware geometry representation.

6.10. Advanced issues

Boolean operations provide common modeling operations. Unfortunately, neither NURBS nor T-splines, nor subdivision surfaces are closed under such operations. Robustly computing these operations and suitably representing them is still a difficult vexing issue that has spawned a thriving *model repair* specialty activity (See Section 2). A variety of approaches has been fabricated for defining closed representations for surface boundaries with Booleans. All require a modified definition of the model boundary near the area of the boolean seam. One desirable solution would be an exact re-representation, as is done in the bracket example of [1,36,17]. However, that is only feasible for some models in which the Boolean operator has created intersection curves that can be represented, and the surface in which they occur can be represented exactly using that intersection curve as one of its 4 boundaries.

7. Summary and conclusions

Isogeometric analysis has demonstrated itself as a paradigm that may actually successfully bridge CAD modeling and FEA analysis. With analysis tools able to act natively upon the same mathematical building blocks employed in the modeling community, there exists a realistic chance that a seamless pipeline based on a shared representation might be in the future. It is important though to appreciate that the constraints under which these two communities currently work, and will continue to work, are some-

times complementary, often different, and occasionally competing. In this paper, we have attempted to provide insight into the modeler's perspective on the process of model design and construction, and to demonstrate and highlight explicitly that choices that arise within the modeling process may have consequences downstream the line when analysis is performed on an isogeometric model. We detail several of the outstanding issues and considerations within modeling and at the interface of modeling and analysis. These fundamental problems must be explored and addressed as the area of isogeometric analysis moves forward. We advocate a new area of research – *analysis-aware modeling* – by which modelers become cognizant of how their modeling choices impact the quality of analysis, and hence can incorporate this knowledge into the balancing act of design considerations and constraints that the modeler is already juggling. Isogeometric analysis is a superb vehicle to promote the marriage of CAD and FEA as it represents *modeling-aware analysis*. We hope to have demonstrated that there is a correspondingly important and symmetric need for the modeling community to reciprocate in developing analysis-aware modeling. We conclude by emphasizing that both can only be done through the continual interaction and dialog between the two communities.

Acknowledgments

This work was supported in part by NSF (CCF0541402), NSF Career Award CCF0347791, ARO W911NF0810517, and the Norwegian Research Council. The authors gratefully acknowledge the computational support and resources provided by the Scientific Computing and Imaging Institute at the University of Utah.

References

[1] T.J. Hughes, J.A. Cottrell, Y. Bazilevs, Isogeometric analysis: cad, finite elements, NURBS, exact geometry, and mesh refinement, Comput. Meth. Appl. Mech. Engrg. 194 (2005) 4135–4195.
[2] T. Dokken, V. Skytt, J. Haenisch, K. Bengtsson, Isogeometric representation and analysis - bridging the gap between CAD and analysis, in: 47th AIAA Aerospace Sciences Meeting Including The New Horizons Forum and Aerospace Exposition, January 5–8, 2009.
[3] T.W. Sederberg, G.T. Finnigan, X. Li, H. Lin, H. Ipson, ACM Trans. Graphics 27 (3) (2008) 79:1–79:8.
[4] Elaine Cohen, Richard F. Riesenfeld, Gershon Elber, Geometric Modeling with Splines: An Introduction, A.K. Peters, Ltd., Natick, MA, USA, 2001.
[5] Ch. Schwab, *p*- and *hp*-Finite Element Methods: Theory and Applications to Solid and Fluid Mechanics, Oxford University Press, USA, 1999.
[6] Philippe G. Ciarlet, The Finite Element Method for Elliptic Problems, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2002.
[7] Owe Axelsson, Iterative Solution Methods, Cambridge University Press, Cambridge, 1994.
[8] M.O. Deville, E.H. Mund, P.F. Fischer, High Order Methods for Incompressible Fluid Flow, Cambridge University Press, 2002.
[9] G.E. Karniadakis, S.J. Sherwin, Spectral/hp element methods for CFD – second ed., Oxford University Press, UK, 2005.
[10] Elaine Cohen, Tom Lyche, Richard F. Riesenfeld, Discrete B-splines and subdivision techniques in computer-aided geometric design and computer graphics, Comput. Graphics Image Process. 15 (2) (1980) 87–111.
[11] Elaine Cohen, Tom Lyche, L.L. Schumaker, Algorithms for degree-raising of splines, ACM Trans. Graphics 4 (3) (1986) 171–181.
[12] K. Morken, Products of splines as linear combinations of B-splines, Construct. Approx. 7 (1) (1991) 195–208.
[13] M.S. Casale, E.L. Stanton, An overview of analytic solid modeling, IEEE Comput. Graphics Appl. (1985) 45–56.
[14] Karen Lynn Paik, Trivariate B-splines. Master's Thesis, Department of Computer Science, University of Utah, June 1992.
[15] William Martin, Elaine Cohen, Surface completion of an irregular boundary curve using a concentric mapping, in: Proceedings of the Fifth Conference on Curves and Surfaces, Nashboro Press, 2003, pp. 293–302.
[16] Joel D. Daniels II, Elaine Cohen, Surface creation and curve deformations between two complex closed spatial spline curves, in: Springer-Verlag Lecture Notes in Computer Science 4077 (GMP 2006), 2006, pp. 221–234.
[17] J.A. Cottrell, T.J.R. Hughes, A. Reali, Studies of refinement and continuity in isogeometric structural analysis, Comput. Meth. Appl. Mech. Engrg. 196 (2007) 4160–4183.
[18] Steven A. Coons, Surfaces for computer-aided design of space forms. Technical Report MAC-TR-41, MIT, 1967.
[19] William J. Gordon, Spline-blended surface interpolation through curve networks, J. Math. Mech. 18 (10) (1969) 931–952.
[20] Leslie Piegl, Wayne Tiller, A menagerie of rational B-spline circles, IEEE Comput. Graphics Appl. 9 (5) (1989) 48–56.
[21] James Blinn, How many ways can you draw a circle?, IEEE Comput Graphics Appl. 7 (8) (1987) 39–44.
[22] Y. Zhang, Y. Bazilevs, S. Goswami, C.L. Bajaj, T.J.R. Hughes, Patient-specific vascular NURBS modeling for isogeometric analysis of blood flow, in: Proceedings of the 15th International Meshing Roundtable, Springer, Berlin, 2006, pp. 73–92.
[23] Y. Bazilevs, T.J.R. Hughes, Nurbs-based isogeometric analysis for the computation of flows about rotating components, Comput. Mech. 43 (2008) 143–150.
[24] William Martin, Elaine Cohen, Representation and extraction of volumetric attributes using trivariate splines, in: Symposium on Solid and Physical Modeling, 2001, pp. 234–240.
[25] Tobias Martin, Elaine Cohen, Mike Kirby, Volumetric parameterization and trivariate B-spline fitting using harmonic functions, in: SPM '08: Proceedings of the 2008 ACM Symposium on Solid and Physical Modeling, ACM, New York, NY, USA, 2008, pp. 269–280.
[26] Jonathan R. Shewchuk, What is a good linear element? interpolation, conditioning, and quality measures, in: 11th International Meshing Roundtable, 2002, pp. 115–126.
[27] J. Peraire, M. Vahdati, K. Morgan, O.C. Zienkiewicz, Adaptive remeshing for compressible flow computations, J. Comput. Phys. 72 (2) (1987) 449–466.
[28] M. Berzins, Mesh quality: a function of geometry, error estimates or both, in: Seventh International Meshing Roundtable, 1998, pp. 229–238.
[29] David A. Venditti, David L. Darmofal, Adjoint error estimation and grid adaptation for functional outputs: application to quasi-one-dimensional flow, J. Comput. Phys. 164 (1) (2000) 204–227.
[30] Thomas W. Sederberg, Jianmin Zheng, Almaz Bakenov, Ahmad Nasri, T-splines and T-NURCCS, ACM Trans. Graphics 22 (3) (2003) 477–484.
[31] Eugene Butkov, Mathematical Physics, Addison-Wesley Publishing Company, Reading, MA, 1968.
[32] L. Bazilevs, Y. amd Beirao da Veiga, J.A. Cottrell, T.J.R. Hughes, G. Sangalli, Isogeometric analysis: approximation, stability and error estimates for *h*-refined meshes, Math. Meth. Models Appl. Sci. 16 (2006) 1031–1090.
[33] B.A. Szabó, I. Babuška, Finite Element Analysis, John Wiley and Sons, New York, 1991.
[34] Larry L. Schumaker, Spline Functions: Basic Theory, second ed., Cambridge University Press, 2007.
[35] Anand Pardhanani, Graham F. Carey, Optimization of computational grids, Numer. Meth. Partial Diff. Equat. 4 (2) (1988) 95–117.
[36] J.A. Cottrell, A. Reali, Y. Bazilevs, T.J.R. Hughes, Isogeometric analysis of structural vibrations, Comput. Meth. Appl. Mech. Engrg. 195 (41–43) (2006) 5257–5296.