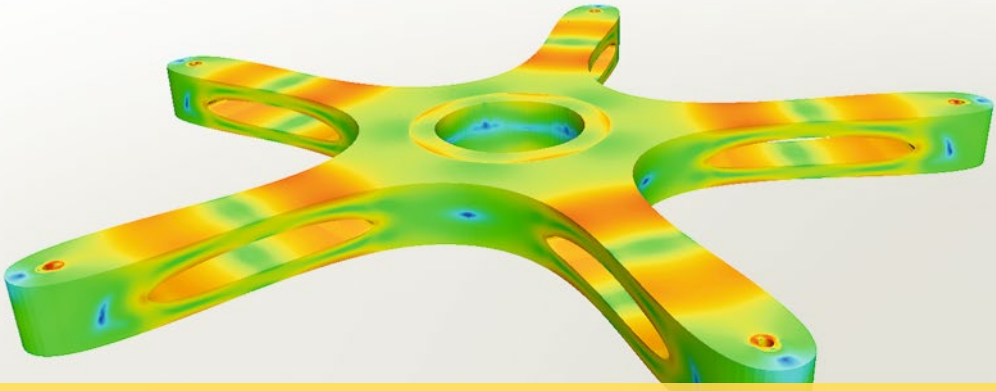


LECTURE NOTES IN COMPUTATIONAL SCIENCE AND ENGINEERING 107

Bert Jüttler · Bernd Simeon *Editors*

Isogeometric Analysis and Applications 2014

Springer

Lecture Notes in Computational Science and Engineering 107

Editors:

More information about this series at http://www.springer.com/series/3527

Bert Jüttler • Bernd Simeon
Editors

Isogeometric Analysis and Applications 2014

Springer

Editors
Bert Jüttler
Institut für Angewandte Geometrie
Johannes Kepler Universität Linz
Linz, Austria

Bernd Simeon
Fachbereich Mathematik
Technische Universität Kaiserslautern
Kaiserslautern, Germany

ISSN 1439-7358 ISSN 2197-7100 (electronic)
Lecture Notes in Computational Science and Engineering
ISBN 978-3-319-23314-7 ISBN 978-3-319-23315-4 (eBook)
DOI 10.1007/978-3-319-23315-4

Library of Congress Control Number: 2015959791

Springer Cham Heidelberg New York Dordrecht London

Cover figure: Isogeometric simulation using a trivariate multi-patch NURBS model (from Pauley, Nguyen, Mayer, Špeh, Weeger, Jüttler, pp. 1–21)

Printed on acid-free paper

Springer International Publishing AG Switzerland is part of Springer Science+Business Media (www.springer.com)

Foreword for the Proceedings Volume

The very comfortable conference venue in Annweiler am Trifels, Germany, facing the Trifels Castle, perched on the peak of the Sonnenberg, high above the Queich valley, provided a beautiful and inspirational setting for **IGAA 2014, Isogeometric Analysis and Applications**, April 7–10, 2014. Trifels Castle is where Richard the Lionheart, King of England, was imprisoned for three weeks in the year 1193. History was made then and it was made again at **IGAA 2014**, now one of several very important, focused conferences that have been held to assess the state-of-the-art in the emerging field of Isogeometric Analysis. This field began with a single paper published in the 2005 October 1 issue of the international journal *Computer Methods in Applied Mechanics and Engineering*, entitled "Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement." Almost immediately, Isogeometric Analysis attracted considerable attention in the research community, and at the present time it is enjoying explosive growth, as measured by the number of papers published on the topic and the citations to them in the literature. Currently, it represents perhaps the hottest topic within both Computational Mechanics and Computer-Aided Geometrical Modeling, two historically separate disciplines that have joined forces and embraced the vision of Isogeometric Analysis, that is, to reconstitute Computational Mechanics within the framework provided by Computational Geometry, thereby simplifying the analysis model development process and integrating engineering design and analysis into a unified framework. As research activity in Isogeometric Analysis has ramped up, industrial and commercial interest has followed, and we are now seeing considerable growth in these sectors as well. The promise of Isogeometric Analysis, that is, to improve the overall efficiency of the engineering product development process, is the driving force.

An interesting aspect of Isogeometric Analysis is the breadth of activities within it. One can pursue it from a very mathematical perspective, or a very practical engineering perspective, or anywhere in between. There seems to be something in it for everyone! It is a field that suits my broad scientific and engineering interests. From a mathematician's point of view, I may be an engineer. From an engineer's point of view, I may even be a mathematician. So be it. I would claim I am a

little bit of both, but I can also claim to be a Mechanical Design Engineer, at least I was 50 years ago! That was my first job at Grumman Aerospace Corporation (now Northrop Grumman) after obtaining my undergraduate degree in Mechanical Engineering. I enjoyed it and learned a lot about engineering in the real world, but it was not long before I gravitated toward engineering analysis. Design and analysis are the cornerstones of Isogeometric Analysis, and I feel I have practical experience with both. **IGAA 2014**, although including a little bit of everything, was somewhat weighted toward the mathematical side of the spectrum, reflecting the interests of the majority of the participants. Nevertheless, my invited talk at the conference was meant to be an overview, and I entitled it "Isogeometric Analysis: Where we are and where we are going." In it, I also tried to identify a few subtopics that I felt were enjoying increased attention, particularly ones that might have substantial impact on the field going forward. I think some of the thoughts presented therein are worth recalling in this Foreword to the book of selected papers emanating from the conference.

Perhaps a main theme that needs to be highlighted is the analysis suitability of design models. Certainly, T-splines and LR-splines are having significant impact due to their watertight and unstructured qualities. An additional theme is performing analysis directly on trimmed NURBS files, which heretofore has been considered unsuitable for analysis. This is important because trimmed NURBS are ubiquitous in contemporary engineering design and give no sign of going away, at least anytime soon. Hierarchically refined NURBS are now reaching maturity and are becoming a standard tool in the design-through-analysis arsenal.

Another theme that has begun to manifest itself is the surprising efficiency of Isogeometric Analysis in practical applications. This is facilitated by the significant gain in accuracy per degree of freedom compared with classical C^0-continuous finite element methods and decreased computational effort engendered by reduced integration and collocation techniques. Collocation techniques, which utilize the strong, rather than weak, form of the governing equations, are perhaps the ultimate reduced integration method. This topic has never had much success in the context of C^0-continuous finite element methods, but is undergoing a renaissance in Isogeometric Analysis due the smoothness of spline basis functions. Efficient solution algorithms that take advantage of the structure of Isogeometric Analysis matrix equations have also contributed to the gains in efficiency, as well as adaptive refinement strategies driven by a posteriori error estimates.

Other areas in which exciting developments are taking place include "thin" shell theory, which has also benefitted from smooth basis functions *without* derivative degrees of freedom. This has given new life to the classical Kirchhoff-Love shell theory, which requires C^1-continuous basis functions in the classical Galerkin weak formulation. These are readily available in Isogeometric Analysis, with the added benefit of the elimination of rotational degrees of freedom. Further developments have even generalized this approach to the Reissner-Mindlin "thick" shell theory and at the same time have eliminated shear locking entirely, independent of basis functions. These indeed are significant developments and are expected to have a profound effect on structural analysis in the coming years.

Smooth, higher-order accurate basis functions and precise geometrical modeling have also created a new beginning in boundary element analysis, whose potential has never before been attained due to inaccurate, faceted geometrical approximations and low-order, C^0-continuous basis functions. Several complete design-through-analysis procedures have been described using the Isogeometric Analysis version of boundary integral formulations. These have included shape optimization directly in the analysis-suitable CAD file, an enormous advantage in streamlining the design-through-analysis process. Shape and topology optimization are likewise general areas of significant Isogeometric Analysis activity.

The breadth of practical applications of Isogeometric Analysis is staggering, and some of applications to complex nonlinear problems and fluid-structure interaction are truly impressive. All this is being buttressed by many important mathematical studies involving complementary functional- and spectral-analysis approaches.

The papers in this volume only represent a small sampling of the many excellent presentations at **IGAA 2014**, but give one a glimpse of the significant recent progress in Isogeometric Analysis, which is rapidly becoming a mainstream analysis technology and a new paradigm for geometric design.

Austin, USA Thomas J.R. Hughes
June 23, 2015

Preface

This book contains a selection of articles that were presented at the second workshop on *Isogeometric Analysis and Applications – IGAA 2014*, held in Annweiler am Trifels, Germany, from April 7 to 10, 2014. The emerging field of Isogeometric Analysis lives from the interaction of geometric modeling and numerical analysis, and the workshop was a great success in stimulating the exchange of ideas between these two fields. We are glad to see that, over the last years, this interaction has resulted in a series of groundbreaking scientific results, some of which are covered by the articles in this volume. The articles can be structured in four groups that cover the most important branches of recent developments in Isogeometric Analysis.

The first group of four papers deals with various aspects of multi-patch methods in Isogeometric Analysis.

In their paper entitled "Multipatch Discontinuous Galerkin Isogeometric Analysis," U. Langer et al. present Galerkin-type methods that allow for discontinuities across the subdomain boundaries. The required interface conditions are weakly imposed by the terms associated with the boundary of the subdomains. The construction and the corresponding discretization error analysis of the resulting multi-patch schemes are given for heterogeneous diffusion model problems in 2D and 3D domains as well as on open and closed surfaces. The theoretical results are confirmed by extensive numerical experiments.

Next, E. Brivadis and her coauthors study the influence of quadrature errors on isogeometric mortar methods. After reviewing the recent mathematical theory, they investigate the effect of the errors caused by using numerical quadrature for the coupling integrals. Two approaches for designing quadrature rules and their impact on the convergence rate are analyzed.

M. Pauley et al. present a pipeline for the conversion of 3D models into a form suitable for Isogeometric Analysis. Given a boundary represented 3D model, they establish a chain of processes that result in a collection of volumetric NURBS patches. The suitability of the result for Isogeometric Analysis is also exemplified by performing stress simulations.

The fourth paper, entitled "Domain Decomposition Methods and Kirchhoff-Love Shell Multipatch Coupling in Isogeometric Analysis" and contributed by

A. Apostolatos et al., presents and compares several domain decomposition methods, including penalty and Lagrange multiplier methods. The methods, which are extended to account for geometrically nonlinear problems, provide highly accurate results, thus enabling the treatment of Kirchhoff-Love shells on multiple patches.

The second group of papers comprises three articles which are devoted to shell analysis and to T-splines. The last paper of the previous group also contributes to this topic.

C. Adam and coauthors propose a reduced shell element for Reissner-Mindlin geometric nonlinear analysis within the context of T-spline analysis. More precisely, the nonuniform Gauss-Legendre and patchwise reduced integrations for quadratic shape functions are both presented and compared to the standard full Gauss-Legendre scheme. The authors evaluate the effects of mesh distortion and local refinement, using both full and reduced numerical quadratures.

In their paper entitled "Multiresolution shape and topology optimisation with subdivision surfaces," F. Cirak and K. Bandara review the multiresolution shape optimisation technique based on subdivision surfaces and present its application to elastic solids, electrostatic field equations, and thin shells. The geometry of the domain is described with subdivision surfaces, and different resolutions of the same surface are used for optimization and analysis. In their paper, the authors discretize elastic solids with the immersed finite element method, electrostatic field equations with the boundary element method, and thin shells with the subdivision finite element technique. In all situations there is no need to generate and to maintain an analysis-suitable volume discretization.

In the next paper on "Atom Simplification and Quality T-mesh Generation for Multi-resolution Biomolecular Surfaces" by T. Liao et al., the structure of biomolecules is first simplified using an error-bounded atom elimination method. An extended cross field-based parameterization method is then developed to adapt the parametric line spacings to different surface resolutions. From the parameterization results, adaptive and anisotropic T-meshes for the further T-spline surface construction are extracted. A gradient flow-based method is finally used to improve the T-mesh quality, with the anisotropy preserved in the quadrilateral elements.

The two papers in the next group address an important topic that is rarely discussed in detail in the literature, namely, algorithmic and implementation aspects in Isogeometric Analysis.

D. Fußeder and B. Simeon concentrate on these aspects in the context of isogeometric shape optimization. They present a self-contained treatment of gradient-based shape optimization with Isogeometric Analysis, focusing on issues such as the computation of shape gradients in an isogeometric formulation and how to update B-spline and NURBS geometries.

The other paper in this group, entitled "Effective Integration of Sophisticated Operators in Isogeometric Analysis with igatools," has been contributed by N. Cavallini et al. It discusses aspects of igatools, which is a newly released library for operator assembly in Isogeometric Analysis. It is shown that the design of

the library makes it an effective tool in assembling isogeometric discretizations of sophisticated differential operators.

Finally, the two remaining papers address advanced topics from the theory of finite element methods and their application in Isogeometric Analysis.

S. Kleiss and S. Tomar present two-sided robust and sharp a posteriori error estimates in the isogeometric discretization of elliptic problems. These estimates, which are fully computable, provide robust, guaranteed, and sharp two-sided bounds of the exact error in the energy norm. The numerical realization and the quality of the computed error distribution are also covered.

The last paper of this volume, entitled "Multilevel Preconditioning for Variational Problems," which has been contributed by A. Kunoth, surveys the main ingredients for multilevel preconditioners in terms of higher-order B-splines for standard second- and fourth-order elliptic partial differential equations in variational form possessing smooth solutions. It addresses three types of multilevel preconditioners for which asymptotic optimality can be shown. Two of them, the so-called additive preconditioners, are specified for Isogeometric Analysis involving linear elliptic partial differential operators in terms of variants of the BPX preconditioner and wavelet preconditioners. The third class is formed by multiplicative preconditioners, specifically by multigrid methods. The performance of the different preconditioners is demonstrated by numerical examples.

Last but not least, the editors acknowledge the support from many people and organizations that has been crucial in organizing the workshop and in compiling this volume. In particular, we are indebted to the reviewers. Their comments have helped us greatly to identify the manuscripts suitable for publication in this book and moreover were useful for improving many of the papers substantially. We also wish to express our thanks to Springer Verlag for the constructive and supportive cooperation during the production of this book. Financial support from the *Fraunhofer Institute for Industrial Mathematics (ITWM)*, Kaiserslautern, from the *Center for Mathematical and Computational Modeling* $(\mathrm{CM})^2$ at Technische Universität Kaiserslautern, and from the program for equal opportunities at Technische Universität Kaiserslautern made it possible to attract participants from all over the world to the workshop. We are now looking forward to the next event in the workshop series on *Isogeometric Analysis and Applications*.

Linz, Austria — Bert Jüttler
Kaiserslautern, Germany — Bernd Simeon
July 2015

Contents

Multipatch Discontinuous Galerkin Isogeometric Analysis

Ulrich Langer, Angelos Mantzaflaris, Stephen E. Moore, and Ioannis Toulopoulos

Abstract Isogeometric Analysis (IgA) uses the same class of basis functions for both representing the geometry of the computational domain and approximating the solution of the boundary value problem under consideration. In practical applications, geometrical patches are used in order to get flexibility in the geometrical representation. This multi-patch representation corresponds to a decomposition of the computational domain into non-overlapping subdomains also called patches in the geometrical framework. We will present discontinuous Galerkin (dG) methods that allow for discontinuities across the subdomain (patch) boundaries. The required interface conditions are weakly imposed by the dG terms associated with the boundary of the subdomains. The construction and the corresponding discretization error analysis of such dG multi-patch IgA schemes is given for heterogeneous diffusion model problems in volumetric 2d and 3d domains as well as on open and closed surfaces. The theoretical results are confirmed by numerous numerical experiments which have been performed in G+SMO. The concept and the main features of the IgA library G+SMO are also described.

1 Introduction

Isogeometric Analysis (IgA), which was introduced by Hughes, Cottrell and Bazilevs in 2005, is a new discretization technology which uses the same class of basis functions for both representing the geometry of the computational domain and approximating the solution of problems modeled by Partial Differential Equations (PDEs), [23]. IgA uses the exact geometry in the class of Computer Aided Design (CAD) geometries, and thus geometrical errors introduced by approximation of the physical domain are eliminated. This feature is especially important in technical applications where the CAD geometry description is directly used in the production

U. Langer (✉) • A. Mantzaflaris • S.E. Moore • I. Toulopoulos
RICAM, Altenbergerstr. 69, A-4040 Linz, Austria
e-mail: ulrich.langer@ricam.oeaw.ac.at; ulanger@numa.uni-linz.ac.at;
angelos.mantzaflaris@ricam.oeaw.ac.at; stephen.moore@ricam.oeaw.ac.at;
ioannis.toulopoulos@ricam.oeaw.ac.at

B. Jüttler, B. Simeon (eds.), *Isogeometric Analysis and Applications 2014*,
Lecture Notes in Computational Science and Engineering 107,
DOI 10.1007/978-3-319-23315-4_1

process. Usually, IgA uses basis functions like B-Splines and Non-Uniform Rational B-Splines (NURBS), which are standard in CAD, and have several advantages which make them suitable for efficient and accurate simulation, see [8]. The mathematical analysis of approximation properties, stability and discretization error estimates of NURBS spaces have been well studied in [5]. Furthermore, approximation error estimates due to mesh, polynomial degree and smoothness refinement have been obtained in [6], and likewise a hybrid method that combines a globally C^1-continuous, piecewise polynomial finite element basis with rational NURBS-mappings have also been considered in [27].

In practical applications, the computational domain $\Omega \subset \mathbb{R}^d$ $(d = 2, 3)$ is usually represented by multiple patches leading to non-matching meshes and thus to patch-wise non-conforming approximation spaces. In order to handle non-matching meshes and polynomial degrees across the patch interfaces, the discontinuous Galerkin (dG) technique that is now well established in the Finite Element Analysis (FEA) of different field problems is employed. Indeed, dG methods have been developed and analyzed for many applications including elliptic, parabolic and hyperbolic PDEs. The standard dG finite element methods use approximations that are discontinuous across the boundaries of every finite element of the triangulation. To achieve consistent, stable and accurate schemes, some conditions are prescribed on the inter-element boundaries, see, e.g., Rivière's monograph [35].

In this paper, we present and analyze multipatch dG IgA methods for solving elliptic PDEs in volumetric 2d or 3d computational domains as well as on open and closed surfaces. As model problems, we first consider diffusion problems of the form

$$-\nabla \cdot (\alpha \nabla u) = f \text{ in } \Omega, \text{ and } u = u_D \text{ on } \partial\Omega, \tag{1}$$

where Ω is a bounded Lipschitz domain in $\mathbb{R}^d$, $d = 2, 3$, with the boundary $\partial\Omega$, f is a given source term, and u_D are given Dirichlet data. Neumann, Robin and mixed boundary conditions can easily be treated in the same dG IgA framework which we are going to analyze in this paper. The same is true for including a reaction term into the PDE (1). The diffusion coefficient α is assumed to be bounded from above and below by strictly positive constants. We allow α to be discontinuous across the inter-patch boundaries with possible large jumps across these interfaces. More precisely, the computational domain Ω is subdivided into a collection $\mathcal{T}_H(\Omega) := \{\Omega_i\}_{i=1}^N$ of non-overlapping subdomains (patches) $\Omega_1, \ldots, \Omega_N$, where the patches Ω_i are obtained by some NURBS mapping Φ_i from the parameter domain $\hat{\Omega} = (0, 1)^d \subset \mathbb{R}^d$. In every subdomain, the problem is discretized under the IgA methodology. The dG IgA method is considered in a general case without imposing any matching grid conditions and without any continuity requirements for the discrete solution across the interfaces $F_{ij} = \partial\Omega_i \cap \partial\Omega_j$. Thus, on the interfaces, the dG technique ensures the communication of the solution between the adjacent subdomains. For simplicity, we assume that the coefficients are piecewise constant with respect to the decomposition $\mathcal{T}_H(\Omega_i)_{i=1}^N$. It is well known that solutions of elliptic problems with discontinuous coefficients are generally not smooth [20, 26], Thus, we cannot expect that the dG IgA provides convergence rates like in the smooth case.

Here, the reduced regularity of the solution is of primary concern. Following the techniques developed by Di Pietro and Ern [12, 13] for dG FEA, we present an error analysis for diffusion problems in 2d and 3d computational domains with solutions belonging to $W^{l,p}(\Omega_i)$, $l \geq 2$, $p \in (\max\{1, 2d/(d+2(l-1))\}, 2]$, $i = 1, \ldots, N$. In correspondence with this regularity, we show optimal convergence rates of the discretization error in the classical "broken" dG – norm. Due to the peculiarities of the dG IgA, the proofs are quite technical and can be found in the paper [30] written by two of the co-authors. Solutions of elliptic problems with low regularity can also appear in other cases. For example, geometric singular boundary points (re-entrant corners), points with change of the boundary conditions, and singular source terms can cause low-regularity solutions as well, see, e.g., [20]. Especially, in the IgA, we want to use the potential of the approximation properties of NURBS based on high-order polynomials. In this connection, the error estimates obtained for the low-regularity case can be very useful. Indeed, via mesh grading, that has been used in the FEA for a long time [3, 33], we can recover the full convergence rate corresponding to the underlying polynomial degree. We mention that in the literature other techniques have been proposed for solving two dimensional problems with singularities. For example, in [34] and [24], the original B-Spline finite dimensional space has been enriched by generating singular functions which resemble the types of the singularities of the problem. Also in [7], by studying the anisotropic character of the singularities of the problem, the one dimensional approximation properties of the B-Splines are generalized in order to produce anisotropic refined meshes in the regions of the singular points. Our error analysis is accompanied by a series of numerical tests which fully confirm our convergence rate estimates for the multipatch dG IgA in both the full and the low regularity cases as well as the recovering of the full rate by means of mesh grading techniques. We mention that the detailed proofs of the main theoretical results can be found in [30] and [28], whereas all numerical results are new. It is very clear that our analysis can easily be generalized to more general classes of elliptic boundary value problems like plane strain or stress linear elasticity problems or linear elasticity problems in 3d volumetric computational domains.

We will also consider multipatch dG IgA of diffusion problems of the form (1) on sufficiently smooth, open and closed surfaces Ω in $\mathbb{R}^3$, where the gradient ∇ must now be replaced by the surface gradient ∇_Ω, see, e.g., [18], and the patches Ω_i, into which Ω is decomposed, are now images of the parameter domain $\hat{\Omega} = (0,1)^2$ by the mapping $\Phi_i : \hat{\Omega} \subset \mathbb{R}^2 \rightarrow \Omega_i \subset \mathbb{R}^3$. The case of matching meshes was considered and analyzed in [29] by two of the co-authors, see also [36] for a similar work. It is clear that the results for non-matching spaces and for mesh grading presented here for the volumetric (2d and 3d) case can be generalized to diffusion problems on open and closed surfaces. Until now, the most popular numerical method for solving PDEs on surfaces is the surface Finite Element Method (FEM). The surface FEM was first applied to compute approximate solutions of the Laplace-Beltrami problem, where the finite element solution is constructed from the variational formulation of the surface PDEs in the finite element space that is living on the triangulated surface approximating the real surface [15]. This method has been extended to the parabolic equations fixed

surfaces by Dziuk and Elliot [17]. To treat conservation laws on moving surfaces, Dziuk and Elliot proposed the evolving surface finite element methods, see [16]. Recently, Dziuk and Elliot have published the survey paper [18] which provides a comprehensive presentation of different finite element approaches to the solution of PDEs on surfaces with several applications. The first dG finite element scheme, which extends Dziuk's approach, was proposed by Dedner et al. [11] and has very recently been extended to adaptive dG surface FEM [10] and high-order dG finite element approximations on surfaces [2]. However, since all these approaches rely on the triangulated surface, they have an inherent geometrical error which becomes more complicated when approximating problems with complicated geometries. This drawback of the surface FEM can be overcome by the IgA technology, at least, in the class of CAD surfaces which can be represented exactly by splines or NURBS. Dede and Quateroni have introduced the surface IgA for fixed surfaces which can be represented by one patch [9]. They presented convincing numerical results for several PDE problems on open and closed surfaces. However, in many practical applications, it is not possible to represent the surface Ω with one patch. The surface multipatch dG IgA that allows us to use patch-wise different approximations spaces on non-matching meshes is then the natural choice.

The new paradigm of IgA brings challenges regarding the implementation. Even though the computational domain is partitioned into subdomains (patches) and elements (parts of the domain delimited by images of knot-lines or knot-planes), the information that is accessible by the data structure (parametric B-spline patches) is quite different than that of a classical finite element mesh (collection of triangles or simplices providing the triangulation of the domain). In this realm, existing finite element software libraries cannot easily be adapted to the isogeometric setting. Apart from the data structures used, another issue is the fact that FEA codes are focused on treating nodal shape function spaces, contrary to isogeometric function spaces. To provide a unified solution to the above (and many other) issues, we present the G+SMO[1] C++ library, which provides a unified, object-oriented development framework suitable to implement advanced isogeometric techniques, such as dG methods see [25].

The rest of the paper is organized as follows. In Sect. 2, we present and analyze the multipatch dG IgA for diffusion problems in volumetric 2d and 3d computational domains including the case of low-regularity solutions. We also study the mesh grading technology which allows us to recover the full convergence rates defined by the degree k of the underlying polynomials. The numerical analysis is accompanied by numerical experiments fully confirming the theoretical results. Section 3 is devoted to multipatch dG IgA of diffusion problems on open and closed surfaces. We present and discuss new numerical results including the case of jumping coefficients and non-matching meshes. The concept and the main features of the G+SMO library is presented in Sect. 4. All numerical experiments presented in this paper have been performed in G+SMO.

[1] *Geometry plus Simulation Modules*, http://www.gs.jku.at/gismo.

2 Multipatch dG IgA for PDEs in 3d Computational Domains

2.1 Multipatch dG IgA Discretization

The weak formulation of (1) reads as follows: find a function u from the Sobolev space $W^{1,2}(\Omega)$ such that $u = u_D$ on $\partial\Omega$ and satisfies the variational formulation

$$a(u, v) = l(v), \ \forall v \in W_0^{1,2}(\Omega) := \{v \in W^{1,2}(\Omega) : \ v = 0 \text{ on } \partial\Omega\}, \tag{2}$$

where the bilinear form $a(\cdot, \cdot)$ and the linear form $l(\cdot)$ are defined by the relations

$$a(u, v) = \int_\Omega \alpha \nabla u \cdot \nabla v \, dx \quad \text{and} \quad l(v) = \int_\Omega f v \, dx,$$

respectively. Beside Sobolev's Hilbert spaces $H^l(\Omega) = W^{l,2}(\Omega)$, we later also need Sobolev's Banach spaces $W^{l,p}(\Omega)$, $p \in [1, \infty)$, see, e.g., [1]. The existence and uniqueness of the solution u of problem (2) can be derived by Lax-Milgram's Lemma [19]. In order to apply the dG IgA methodology to problem (1), the domain Ω is subdivided into a union of subdomains $\mathscr{T}_H(\Omega) := \{\Omega_i\}_{i=1}^N$ such that

$$\bar{\Omega} = \bigcup_{i=1}^{N} \bar{\Omega}_i, \quad \text{with} \quad \Omega_i \cap \Omega_j = \emptyset, \ \text{if } j \neq i. \tag{3}$$

As we mentioned in the introduction, the subdivision of Ω is assumed to be compatible with the discontinuities of α, i.e. they are constant in the interior of Ω_i, that is $\alpha|_{\Omega_i} := \alpha^{(i)}$, and their discontinuities appear only across the interfaces $F_{ij} = \partial\Omega_i \cap \partial\Omega_j$, cf., e.g., [12–14]. Throughout the paper, we will use the notation $a \sim b$ meaning that there are positive constants c and C such that $ca \leq b \leq Ca$.

As it is common in IgA, we assume a parametric domain $\hat{\Omega}$ of unit length, e.g., $\hat{\Omega} = (0, 1)^d$. For any Ω_i, we associate d knot vectors $\Xi_n^{(i)}$, $n = 1, \ldots, d$, on $\hat{\Omega}$, which create a mesh $T_{h_i,\hat{\Omega}}^{(i)} = \{\hat{E}_m\}_{m=1}^{M_i}$, where $\hat{E}_m$ are the micro-elements, see details in [23]. We refer to $T_{h_i,\hat{\Omega}}^{(i)}$ as the *parametric mesh of* Ω_i. For every $\hat{E}_m \in T_{h_i,\hat{\Omega}}^{(i)}$, we denote by $h_{\hat{E}_m}$ *its diameter* and by $h_i = \max\{h_{\hat{E}_m}\}$ *the mesh size* of $T_{h_i,\hat{\Omega}}^{(i)}$. We assume the following properties for every $T_{h_i,\hat{\Omega}}^{(i)}$:

- quasi-uniformity: for every $\hat{E}_m \in T_{h_i,\hat{\Omega}}^{(i)}$ holds $h_i \sim h_{\hat{E}_m}$,
- for the micro-element edges $e_{\hat{E}_m} \subset \partial\hat{E}_m$ holds $h_{\hat{E}_m} \sim e_{\hat{E}_m}$.

On every $T^{(i)}_{h_i,\hat{\Omega}}$, we construct the finite dimensional space $\hat{\mathbb{B}}^{(i)}_{h_i}$ spanned by B-spline basis functions of degree k, see [23],

$$\hat{\mathbb{B}}^{(i)}_{h_i} = span\{\hat{B}^{(i)}_j(\hat{x})\}_{j=0}^{dim(\hat{\mathbb{B}}^{(i)}_{h_i})}, \tag{4}$$

where every basis function $\hat{B}^{(i)}_j(\hat{x})$ in (4) is derived by means of tensor products of one-dimensional B-spline basis functions, e.g.,

$$\hat{B}^{(i)}_j(\hat{x}) = \hat{B}^{(i)}_{j_1}(\hat{x}_1)\cdots\hat{B}^{(i)}_{j_d}(\hat{x}_d). \tag{5}$$

For simplicity, we assume that the basis functions of every $\hat{\mathbb{B}}^{(i)}_{h_i}$, $i = 1,\ldots,N$, are of the same degree k.

Every subdomain $\Omega_i \in \mathscr{T}_H(\Omega)$, $i = 1,\ldots,N$, is exactly represented through a parametrization (one-to-one mapping), cf. [23], having the form

$$\Phi_i : \hat{\Omega} \to \Omega_i, \quad \Phi_i(\hat{x}) = \sum_j C^{(i)}_j \hat{B}^{(i)}_j(\hat{x}) := x \in \Omega_i, \tag{6}$$

where $C^{(i)}_j$ are the control points and $\hat{x} = \Psi_i(x) := \Phi_i^{-1}(x)$. Using Φ_i, we construct a mesh $T^{(i)}_{h_i,\Omega_i} = \{E_m\}_{m=1}^{M_i}$ for every Ω_i, whose vertices are the images of the vertices of the corresponding mesh $T^{(i)}_{h_i,\hat{\Omega}}$ through Φ_i. If $h_{\Omega_i} = \max\{h_{E_m} : E_m \in T^{(i)}_{h_i,\Omega_i}\}$ is the subdomain Ω_i mesh size, then, based on definition (6) of Φ_i, we have the equivalence relation $h_i \sim h_{\Omega_i}$.

The mesh of Ω is considered to be $T_h(\Omega) = \bigcup_{i=1}^N T^{(i)}_{h_i,\Omega_i}$, where we note that there are no matching mesh requirements on the interior interfaces $F_{ij} = \partial\Omega_i \cap \partial\Omega_j$, $i \neq j$. For the sake of brevity in our notations, the interior faces of the boundary of the subdomains are denoted by $\mathscr{F}_I$ and the collection of the faces that belong to $\partial\Omega$ by $\mathscr{F}_B$, e.g. $F \in \mathscr{F}_B$ if there is a Ω_i such that $F = \partial\Omega_i \cap \partial\Omega$. We denote the set of all subdomain faces by $\mathscr{F}$.

Lastly, we define the B-spline space $\mathbb{B}_h(\mathscr{T}_H(\Omega)) = \mathbb{B}^{(1)}_{h_1} \times \ldots \times \mathbb{B}^{(N)}_{h_N}$ on Ω, where every $\mathbb{B}^{(i)}_{h_i}$ is defined on $T^{(i)}_{h_i,\Omega_i}$ as follows

$$\mathbb{B}^{(i)}_{h_i} := \{B^{(i)}_{h_i}|_{\Omega_i} : B^{(i)}_h(x) = \hat{B}^{(i)}_h \circ \Psi_i(x),\ \forall \hat{B}^{(i)}_h \in \hat{\mathbb{B}}^{(i)}_{h_i}\}. \tag{7}$$

We assume that the mappings Φ_i are regular in the sense that there exist positive constants c_m and c_M such that

$$c_m \leq |det(J_i(\hat{x}))| \leq c_M, \text{ for } i = 1,\ldots,N, \text{ for all } \hat{x} \in \hat{\Omega}, \tag{8}$$

where $J_i(\hat{x})$ denotes the Jacobian $\partial\Phi_i(\hat{x})/\partial(\hat{x})$ of the mapping Φ_i.

Now, for any $\hat{u} \in W^{m,p}(\hat{\Omega}), m \geq 0, p > 1$, we define the function

$$\mathscr{U}(x) = \hat{u}(\Psi_i(x)),\ x \in \Omega_i, \tag{9}$$

and the following relation holds true, see [30],

$$C_m \|\hat{u}\|_{W^{m,p}(\hat{\Omega})} \leq \|\mathscr{U}\|_{W^{m,p}(\Omega_i)} \leq C_M \|\hat{u}\|_{W^{m,p}(\hat{\Omega})}, \tag{10}$$

where the constants C_m and C_M depending on $C_m := C_m(\max_{m_0 \leq m}(\|D^{m_0}\Phi_i\|_\infty), \|det(\Psi_i^{'})\|_\infty)$ and $C_M := C_M(\max_{m_0 \leq m}(\|D^{m_0}\Psi_i\|_\infty), \|det(\Phi_i^{'})\|_\infty)$. The usefulness of inequalities (10) in the analysis is the following: every required relation can be proved in the parametric domain and then using (10) we can directly have the expression on the physical subdomain.

We use the B-spline spaces $\mathbb{B}_h^{(i)}$ defined in (7) for approximating the solution of (2) in every subdomain Ω_i. Continuity requirements for $\mathbb{B}_h(\mathscr{T}_H(\Omega))$ are not imposed on the interfaces F_{ij} of the subdomains, clearly $\mathbb{B}_h(\mathscr{T}_H(\Omega)) \subset L^2(\Omega)$ but $\mathbb{B}_h(\mathscr{T}_H(\Omega)) \not\subseteq W^{1,2}(\Omega)$. Thus, problem (2) is discretized by discontinuous Galerkin techniques on F_{ij}, see, e.g., [14]. Using the notation $v_h^{(i)} := v_h|_{\Omega_i}$, we define the average and the jump of v_h on $F_{ij} \in \mathscr{F}_I$, respectively, by

$$\{v_h\} := \frac{1}{2}(v_h^{(i)} + v_h^{(j)}), \quad \text{and} \quad [\![v_h]\!] := v_h^{(i)} - v_h^{(j)}, \tag{11}$$

and for $F_i \in \mathscr{F}_B$

$$\{v_h\} := v_h^{(i)}, \quad \text{and} \quad [\![v_h]\!] := v_h^{(i)}. \tag{12}$$

The dG-IgA scheme reads as follows: find $u_h \in \mathbb{B}_h(\mathscr{T}_H(\Omega))$ such that

$$a_h(u_h, v_h) = l(v_h) + p_D(u_D, v_h),\ \forall v_h \in \mathbb{B}_h(\mathscr{T}_H(\Omega)), \tag{13}$$

where

$$a_h(u_h, v_h) = \sum_{i=1}^{N} a_i(u_h, v_h) - \sum_{i=1}^{N} \Big(\frac{1}{2} s_i(u_h, v_h) - p_i(u_h, v_h)\Big), \tag{14}$$

with the bilinear forms

$$a_i(u_h, v_h) = \int_{\Omega_i} \alpha \nabla u_h \nabla v_h \, dx,$$

$$s_i(u_h, v_h) = \sum_{F_{ij} \subset \partial\Omega_i} \int_{F_{ij}} \{\alpha \nabla u_h\} \cdot \mathbf{n}_{F_{ij}} [\![v_h]\!] + \{\alpha \nabla v_h\} \cdot \mathbf{n}_{F_{ij}} [\![u_h]\!] \, ds,$$

$$p_i(u_h, v_h) = \begin{cases} \sum_{F_{ij} \subset \partial\Omega_i} \int_{F_{ij}} \mu \left(\frac{\alpha^{(j)}}{h_j} + \frac{\alpha^{(i)}}{h_i} \right) [\![u_h]\!][\![v_h]\!] \, ds, & \text{if } F_{ij} \in \mathscr{F}_I \\ \sum_{F_i \subset \partial\Omega_i} \int_{F_i} \mu \frac{\alpha^{(i)}}{h_i} [\![u_h]\!][\![v_h]\!] \, ds, & \text{if } F_i \in \mathscr{F}_B \end{cases}$$

$$p_D(u_D, v_h) = \sum_{F_i \subset \partial\Omega_i} \int_{F_i} \mu \frac{\alpha^{(i)}}{h_i} u_D v_h \, ds, \quad F_i \in \mathscr{F}_B$$

where $\alpha^{(i)} := \alpha|_{\Omega_i}$ and the unit normal vector $\mathbf{n}_{F_{ij}}$ is oriented from Ω_i towards the interior of Ω_j and the parameter $\mu > 0$ will be specified later in the error analysis, cf. [14].

For notation convenience in what follows, we will use the following expression

$$\int_{F_{ij}} \mu \Big(\frac{\alpha^{(j)}}{h_j} + \frac{\alpha^{(i)}}{h_i} \Big) [\![u_h]\!][\![v_h]\!] \, ds,$$

for both cases, $F_{ij} \in \mathscr{F}_I$ and $F_i \in \mathscr{F}_B$. In the latter case, we will assume that $\alpha^{(j)} = 0$.

2.2 Auxiliary Results

We will use the following auxiliary results which have been shown in [30].

Lemma 1 *Let $u \in W^{l,p}(\mathscr{T}_H(\Omega))$ with $l \geq 2$ and $p > 1$. Then there is a constant $C \geq 0$ determined according to the C_m and C_M of (10), such that for $F_{ij} \subset \partial\Omega_i$ holds*

$$\int_{F_{ij}} |u|^p \, ds \leq C \Big(h_i^{-1} \int_{\Omega_i} |u|^p \, dx + h_i^{p-1} \int_{\Omega_i} |\nabla u|^p \, dx \Big). \tag{15}$$

Lemma 2 *For all $v_h \in \mathbb{B}_{h_i}^{(i)}$ defined on $T_{h_i,\Omega_i}^{(i)}$, there is a positive constant C, depending on the mesh quasi-uniformity parameters and C_m and C_M of (10) but not on h_i, such that*

$$\|\nabla v_h\|_{L^p(\Omega_i)}^p \leq C h_i^{-p} \|v_h\|_{L^p(\Omega_i)}^p. \tag{16}$$

Lemma 3 *For all $v_h \in \mathbb{B}_{h_i}^{(i)}$ defined on $T_{h_i,\Omega_i}^{(i)}$ and for all $F_{ij} \subset \partial\Omega_i$, there is a positive constant C, which depends on the mesh quasi-uniformity parameters C_m and C_M of (10) but not on h_i, such that*

$$\|v_h\|_{L^p(F_{ij})}^p \leq C h_i^{-1} \|v_h\|_{L^p(\Omega_i)}^p. \tag{17}$$

Lemma 4 *Let $v_h \in \mathbb{B}_{h_i}^{(i)}$ such that $v_h \in W^{l,p}(E) \cap W^{m,q}(E)$, $E \in T_{h_i,\Omega_i}^{(i)}$, and $0 \leq m \leq l$, $1 \leq p,q \leq \infty$. Then there is a positive constant C, depended on the mesh quasi-uniformity parameters C_m and C_M of (10) but not on h_i, such that*

$$|v_h|_{W^{l,p}(E)} \leq C h_i^{m-l-\frac{d}{q}+\frac{d}{p}} |v_h|_{W^{m,q}(E)}. \tag{18}$$

2.3 Analysis of the dG IgA Discretization

Next, we study the convergence estimates of the method (13) under the following regularity assumption for the weak solution $u \in W^{1,2}(\Omega) \cap W^{l,p}(\mathscr{T}_H(\Omega))$ with $l \geq 2$ and $p \in (\max\{1, \frac{2d}{d+2(l-1)}\}, 2]$. For simplicity of the presentation, we assume that $l \leq k+1$. Nevertheless, for the case of highly smooth solutions, the estimates given below, see Lemma 5, can be expressed in terms of the underlying polynomial degree k. More precisely, the estimate $\delta(l,p,d) = l + (d/2 - d/p - 1)$, must be replaced by $\delta(l,p,d) = \min\{l + (d/2 - d/p - 1), k\}$. We use the enlarged space $W_h^{l,p} := W^{1,2}(\Omega) \cap W^{l,p}(\mathscr{T}_H(\Omega)) + \mathbb{B}_h(\mathscr{T}_H(\Omega))$, and will show that the dG IgA method converges in optimal rate with respect to $\|.\|_{dG}$ norm

$$\|u\|_{dG}^2 = \sum_{i=1}^{N} \Big(\alpha^{(i)} \|\nabla u^{(i)}\|_{L^2(\Omega_i)}^2 + p_i(u^{(i)}, u^{(i)}) \Big), \; u \in W_h^{l,2}. \tag{19}$$

For the error analysis, it is necessary to show the continuity and coercivity properties of the bilinear form $a_h(.,.)$ of (14) and interpolation estimates in $\|.\|_{dG}$ norm. We start by providing estimates on how well the quasi-interpolant $\Pi_h u$ approximates $u \in W^{l,p}(\Omega_i)$, see proof in [30].

Lemma 5 *Let $u \in W^{l,p}(\Omega_i)$ with $l \geq 2$ and $p \in (\max\{1, \frac{2d}{d+2(l-1)}\}, 2]$ and let $E = \Phi_i(E), E \in T_{h_i,\Omega}^{(i)}$. Then for $0 \leq m \leq l \leq k+1$, there exist constants $C_i := C_i\big(\max_{l_0 \leq l}(\|D^{l_0}\Phi_i\|_{L^\infty(\Omega_i)}), \|u\|_{W^{l,p}(\Omega_i)}\big)$, such that*

$$\sum_{E \in T_{h_i,\Omega_i}^{(i)}} |u - \Pi_h u|_{W^{m,p}(E)}^p \leq C_i h_i^{p(l-m)}. \tag{20}$$

Moreover, we have the following estimates for $F_{ij} = \partial\Omega_i \cap \partial\Omega_j$:

$$(i)\ h_i^{\beta} \|\nabla u^{(i)} - \nabla \Pi_h u^{(i)}\|^p_{L^p(F_{ij})} \leq C_i C_{d,p} h_i^{p(l-1)-1+\beta},$$

$$(ii)\ \Big(\frac{\alpha^{(j)}}{h_j} + \frac{\alpha^{(i)}}{h_i}\Big) \|[\![u - \Pi_h u]\!]\|^2_{L^2(F_{ij})} \leq$$

$$C_i \alpha^{(j)} \frac{h_i}{h_j} \Big(h_i^{\delta(l,p,d)} \|u\|^p_{W^{l,p}(\Omega_i)} \Big)^2 + C_j \alpha^{(i)} \frac{h_j}{h_i} \Big(h_j^{\delta(l,p,d)} \|u\|_{W^{l,p}(\Omega_j)} \Big)^2 +$$

$$C_j \Big(h_j^{\delta(l,p,d)} \|u\|_{W^{l,p}(\Omega_j)} \Big)^2 + C_i \Big(h_i^{\delta(l,p,d)} \|u\|_{W^{l,p}(\Omega_i)} \Big)^2,$$

$$(iii)\ \|u - \Pi_h u\|^2_{dG} \leq \sum_{i=1}^{N} C_i \Big(h_i^{\delta(l,p,d)} \|u\|_{W^{l,p}(\Omega_i)} \Big)^2 +$$

$$\sum_{i=1}^{N} \sum_{F_{ij} \subset \partial\Omega_i} C_i \alpha^{(j)} \frac{h_i}{h_j} \Big(h_i^{\delta(l,p,d)} \|u\|_{W^{l,p}(\Omega_i)} \Big)^2,$$

where $\delta(l,p,d) = l + (d/2 - d/p - 1)$.

We mention that the proof of estimate (iii) in Lemma 5 can be derived by using the estimates (i), (ii) and Lemma 1.

Lemma 6 *Suppose $u_h \in \mathbb{B}_h(\mathscr{S}(\Omega))$. There exist a positive constant C, independent of α and h_i, such that*

$$a_h(u_h, u_h) \geq C \|u_h\|^2_{dG}, \quad \forall\, u_h \in \mathbb{B}_h(\mathscr{S}(\Omega)). \tag{21}$$

Proof By (13), we have that

$$a_h(u_h, u_h) = \sum_{i=1}^{N} a_i(u_h, u_h) - \frac{1}{2} \sum_{i=1}^{N} s_i(u_h, u_h) + p_i(u_h, u_h) =$$

$$\sum_{i=1}^{N} \alpha_i \|\nabla u_h\|^2_{L^2(\Omega_i)} - 2 \sum_{F_{ij} \in \mathscr{F}} \int_{F_{ij}} \{\alpha \nabla u_h\} \cdot \mathbf{n}_{F_{ij}} [\![u_h]\!]\, ds$$

$$+ \sum_{F_{ij} \in \mathscr{F}} \mu \Big(\frac{\alpha^{(i)}}{h_i} + \frac{\alpha^{(j)}}{h_j} \Big) \|[\![u_h]\!]\|^2_{L^2(F_{ij})}. \tag{22}$$

For the second term on the right hand side, Lemma 2 and the trace inequality (17) expressed on $F_{ij} \in \mathscr{F}$ yield the bound

$$-\sum_{F_{ij}\in\mathscr{F}}\int_{F_{ij}}\{\alpha\nabla u_h\}\cdot\mathbf{n}_{F_{ij}}[\![u_h]\!]\,ds\geq$$

$$-C_{1,\varepsilon}\sum_{i=1}^{N}\alpha_i\|\nabla u_h\|^2_{L^2(\Omega_i)}-\sum_{F_{ij}\in\mathscr{F}}\frac{1}{C_{2,\varepsilon}}\Big(\frac{\alpha^{(i)}}{h_i}+\frac{\alpha^{(j)}}{h_j}\Big)\|[\![u_h]\!]\|^2_{L^2(F_{ij})}. \tag{23}$$

Inserting (23) into (22) and choosing $C_{1,\varepsilon}<1/2$ and $\mu>2/C_{2,\varepsilon}$, we obtain (21). □

Note that similar discrete coercivity result has been shown in [30] for numerical fluxes without symmetrized terms. Due to the assumed regularity of the solution, the normal interface fluxes $(\alpha\nabla u)|_{\Omega_i}\cdot\mathbf{n}_{F_{ij}}$ belongs (in general) to $L^p(F_{ij})$. The following bound for the interface fluxes in $\|.\|_{L^p}$ setting has been shown in [30].

Lemma 7 *There is a constant C such that the following inequality for $(u,v_h)\in W_h^{l,p}\times\mathbb{B}_h(\mathscr{S}(\Omega))$ holds true*

$$\sum_{i=1}^{N}\sum_{F_{ij}\subset\partial\Omega_i}\int_{F_{ij}}\{\alpha\nabla u\}\cdot\mathbf{n}_{F_{ij}}[\![v_h]\!]\,ds\leq \tag{24}$$

$$C\Big(\sum_{F_{ij}\in\mathscr{F}}\alpha^{(i)}h_i^{1+\gamma_{p,d}}\|\nabla u^{(i)}\|^p_{L^p(F_{ij})}+\alpha^{(j)}h_j^{1+\gamma_{p,d}}\|\nabla u^{(j)}\|^p_{L^p(F_{ij})}\Big)^{\frac{1}{p}}\|v_h\|_{dG},$$

where $\gamma_{p,d}=d(p-2)/2$.

Proof We use Hölder's inequality and then the results of Lemmas 3 and 4. □

Applying similar procedure as this in Lemma 7, we can show for the symmetrizing terms that there is a positive constant independent of grid size such that

$$\sum_{i=1}^{N}\sum_{F_{ij}\subset\partial\Omega_i}\int_{F_{ij}}\{\alpha\nabla v_h\}\cdot\mathbf{n}_{F_{ij}}[\![u]\!]\,ds\leq C_1\|v_h\|_{dG}\|u\|_{dG}. \tag{25}$$

Using the results (24) and (25), we can show the boundedness of the bilinear form, see details in [30].

Lemma 8 *There is a C independent of h_i such that for $(u,v_h)\in W_h^{l,p}\times\mathbb{B}_h(\mathscr{S}(\Omega))$*

$$a_h(u,v_h)\leq C(\|u\|^p_{dG}+\sum_{F_{ij}\in\mathscr{F}}h_i^{1+\gamma_{p,d}}\alpha^{(i)}\|\nabla u^{(i)}\|^p_{L^p(F_{ij})}+ \tag{26}$$

$$h_j^{1+\gamma_{p,d}}\alpha^{(j)}\|\nabla u^{(j)}\|^p_{L^p(F_{ij})}\Big)^{\frac{1}{p}}\|v_h\|_{dG},\ \text{where } \gamma_{p,d}=d(p-2)/2.$$

Next, we give the main error estimate for the dG IgA method.

Theorem 1 *Let $u \in W^{1,2}(\Omega) \cap W^{l,p}(\mathscr{T}_H(\Omega))$, with $l \geq 2$ and $p \in (p(d,l), 2]$ be the solution of (2). Let $u_h \in \mathbb{B}_h(\mathscr{S}(\Omega))$ be the dG IgA solution of (13). Then there are $C_i := C_i\Big(\max_{l_0 \leq l}\Big(\|D^{l_0}\Phi_i\|_{L^\infty(E)}\Big), \|u\|_{W^{l,p}(\Omega_i)}\Big)$, such that*

$$\|u - u_h\|_{dG} \leq \sum_{i=1}^{N}\Big(C_i\Big(h_i^{\delta(l,p,d)} + \sum_{F_{ij}\subset\partial\Omega_i}\alpha^{(j)}\frac{h_i}{h_j}h_i^{\delta(l,p,d)}\Big)\|u\|_{W^{l,p}(\Omega_i)}\Big), \tag{27}$$

where $\delta(l,p,d) = l + (d/2 - d/p - 1)$ and $p(d,l) = \max\{1, \frac{2d}{d+2(l-1)}\}$.

Proof First we need to prove the consistency of u, i.e. u satisfies (13). Then, we use a variation of Cea's Lemma (expressed in the dG framework), the results of Lemmas 6 and 8, as well the quasi-interpolation estimates of Lemma 5. A complete proof can be found in [30]. □

2.4 Numerical Examples

In this section, we present a series of numerical examples to validate the theoretical results, which have been presented.

2.4.1 Smooth and Low-Regularity Solutions

We restrict ourselves to a model problem in $\Omega = (-1,1)^3$, with $\Gamma_D = \partial\Omega$. The domain Ω is subdivided into four equal subdomains Ω_i, $i = 1,\ldots,4$, where for simplicity every Ω_i is initially partitioned into a mesh $T^{(i)}_{h_i,\Omega_i}$, with $h := h_i = h_j$, $i \neq j$, and $i,j = 1,\ldots,4$. Successive uniform refinements are performed on every $T^{(i)}_{h_i,\Omega_i}$ in order to compute numerically the convergence rates. We set the diffusion coefficient equal to one.

In the first test, the data u_D and f in (1) are determined such that the exact solution is $u(\mathbf{x}) = \sin(5\pi x_1)\sin(5\pi x_2)\sin(5\pi x_3)$ (smooth test case). The first three columns of Table 1 display the convergence rates. As it was expected, the convergence rates are optimal. In the second case, the exact solution is $u(x) = |x|^\lambda$. The parameter λ is chosen such that $u \in W^{l,p=1.67}(\Omega)$. Specifically, for $l = 2$, we get $\lambda = 0.2$ and, for $l = 3$, $\lambda = 1.2$. In the last columns of Table 1, we display the convergence rates for degree $k = 2, k = 3$ and $k = 4$ in case of having $l = 2$ and $l = 3$. We observe that, for each of the two different tests, the error in the dG-norm behaves according to the main error estimate given by (27).

Table 1 The numerical convergence rates of the dG IgA method

	Highly smooth			$k = 2$		$k = 3$		$k = 4$	
$\frac{h}{2^s}$	$k = 2$	$k = 3$	$k = 4$	$l = 2$	$l = 3$	$l = 2$	$l = 3$	$l = 2$	$l = 3$
Convergence rates									
$s = 0$	–	–	–	–	–	–	–	–	–
$s = 1$	0.445	2.916	0.083	0.510	1.425	0.406	1.124	0.369	0.926
$s = 2$	2.342	2.463	3.576	0.670	1.645	0.615	1.582	0.562	1.465
$s = 3$	2.100	3.141	4.089	0.699	1.669	0.694	1.692	0.683	1.674
$s = 4$	2.031	3.041	3.989	0.700	1.680	0.700	1.700	0.700	1.700

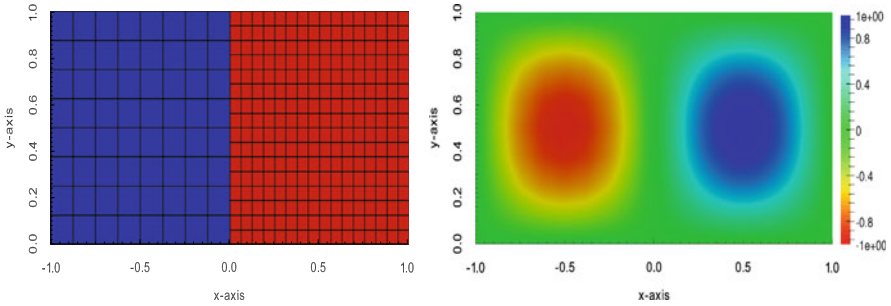


Fig. 1 Non-matching meshes: decomposition into 2 patches with underlying grid of ratio $R = 2$ (*left*), contours of u_h (*right*)

2.4.2 Non-matching Meshes

We consider the boundary value problem (1) with exact solution $u(x, y) = \sin(\pi x)$ $\sin(\pi y)$. The computational domain consist of two unit square patches, $\Omega_1 = (-1, 0) \times (0, 1)$ and $\Omega_2 = (0, 1) \times (0, 1)$, see Fig. 1 (left). The knot vectors representing the geometry are given by $\Xi_1^{(1,2)} = \Xi_2^{(1,2)} = \{0, 0, 1, 1\}$. We refine the mesh of the patches to a ratio R, i.e., the ratio of the grid sizes is $R = h_1/h_2$, see Fig. 1 (left). We solved the problem using equal B-spline degree k on all patches. We plot in Fig. 1 (right) the dG IgA solution u_h computed with $k = 2$. In Fig. 2, we present the decay of the L_2 and dG errors for $k = 1, 2, 3$ and ratios $R = 2^r$ with r from 1 up to 5. In Table 2, we display the convergence rates for large ratio $R = 40$. In both cases, we observe that the rates are the expected ones and are not affected by the different grid sizes of the meshes.

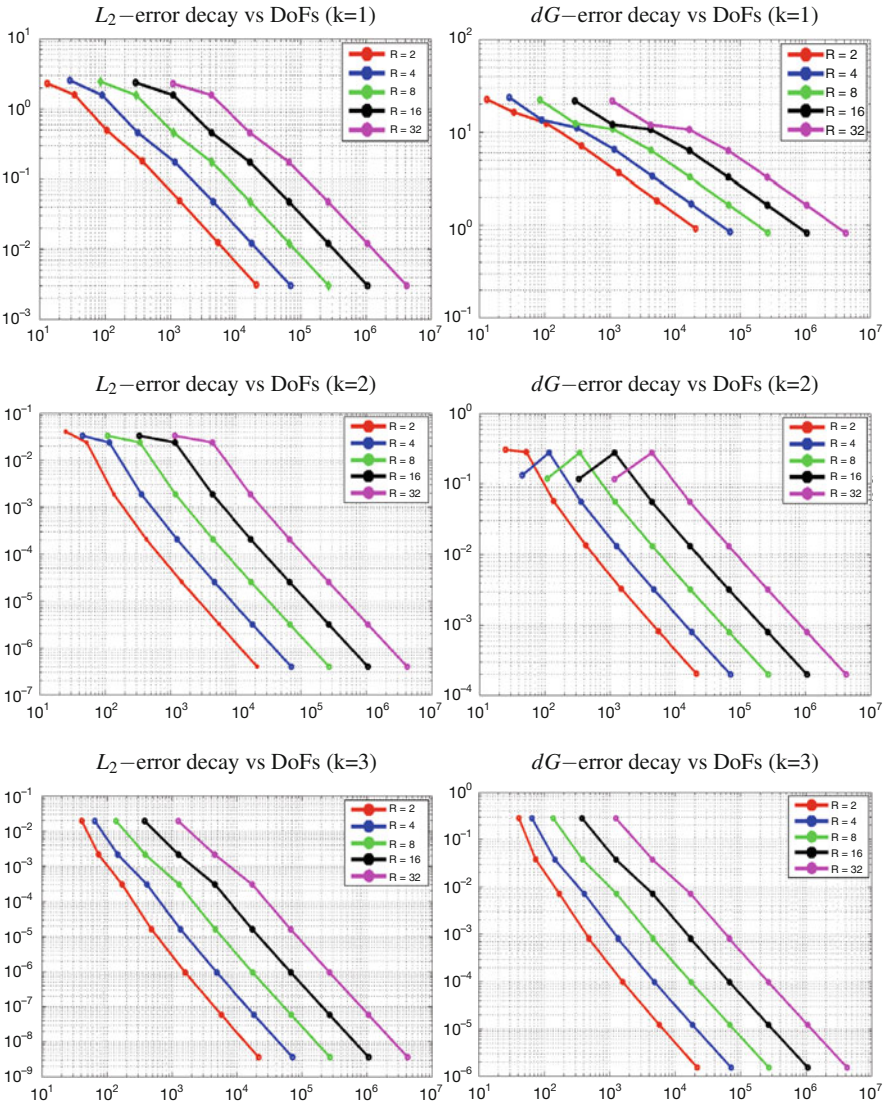


Fig. 2 Error decay in the L_2 (*left*) and dG (*right*) norms for polynomial degree $k = 1, 2, 3$ (*top* to *bottom*) and ratio $R = 2^r$ with r running from 1 to 5

Table 2 Non-matching meshes: error estimates for degrees $k = 1, 2, 3$ and $R = 40$

Dofs	L_2 error	Conv. rate	DG error	Conv. rate
Degree $k = 1$				
1685	0.31667	0	2.1596	0
6570	0.0867936	1.86732	0.97857	1.14202
25,946	0.0249492	1.79859	0.49992	0.968979
103,122	0.00638919	1.96529	0.251392	0.991758
411,170	0.00160452	1.99349	0.125887	0.997804
1.64205×10^{6}	0.000401493	1.9987	0.0629683	0.999435
6.56295×10^{6}	0.000100393	1.99972	0.0314873	0.999857
Degree $k = 2$				
1773	0.0330803	0	0.117511	0
6740	0.0237587	0.477516	0.278058	−1.2426
26,280	0.00186298	3.67278	0.0555191	2.32433
103,784	0.000205886	3.17769	0.0130802	2.0856
412,488	2.53126×10^{-5}	3.02392	0.00321704	2.02358
1.64468×10^{6}	3.17795×10^{-6}	2.99369	0.000800278	2.00716
6.5682×10^{6}	3.99389×10^{-7}	2.99223	0.000199727	2.00247
Degree $k = 3$				
1865	0.0196108	0	0.281761	0
6914	0.00216825	3.17704	0.037403	2.91324
26,618	0.00030589	2.82545	0.00706104	2.4052
104,450	1.60210×10^{-5}	4.25498	0.000803958	3.13469
413,810	9.49748×10^{-7}	4.07627	9.7687×10^{-5}	3.04088
1.64731×10^{6}	5.85542×10^{-8}	4.0197	1.21191×10^{-5}	3.01089
6.57346×10^{6}	3.64710×10^{-9}	4.00495	1.51195×10^{-6}	3.00280

2.5 Graded Mesh Partitions for the dG IgA Methods

We saw in the previous numerical tests that the presence of singular points reduces the convergence rates. In this section, we will study this subject in a more general form. We will focus on solving the model problem in domains with re-entrant corners on the boundary. Due to these singular corner points, the regularity of the solution (at least in a small vicinity) is reduced in comparison with the solutions in smooth domains [20]. As a result, the numerical methods applied on quasi uniform meshes for solving these problems do not yield the optimal convergence rate and thus a particular treatment must be applied. We will devise the popularly known graded mesh techniques which have widely been applied so far for finite element methods [3, 4].

2.5.1 Regularity Properties of the Solution Around the Boundary Singular Points

Let us assume a domain $\Omega \subset \mathbb{R}^2$ and let $P_s \in \partial\Omega$ be a boundary point with internal angle $\omega \in (\pi, 2\pi)$. We consider the local cylindrical coordinates (r, θ) with pole P_s, and define the cone, see Fig. 3 (left),

$$\mathscr{C} = \{(x, y) \in \Omega : x = r\cos(\theta), y = r\sin(\theta), 0 < r < R, 0 < \theta < \omega\}. \tag{28}$$

Then the solution in $\mathscr{C}$ can be written, [20],

$$u = u_r + u_s, \tag{29}$$

where $u_r \in W^{l\geq 2,2}(\Omega)$ and

$$u_s = \xi(r)\gamma r^{\lambda}\sin(\lambda\theta), \tag{30}$$

where γ is the *stress intensity factor* (is a real number depending only on f), and $\lambda = \pi/\omega \in (0, 1)$ is an exponent which determines the strength of the singularity. Since $\lambda < 1$, by an easy computation, we can show that the singular function u_s does not belong to $W^{2,2}(\Omega)$ but $u \in W^{2,p}(\Omega)$ with $p = 2/2 - \lambda$. The representation (30) of u_s helps us to reduce our study to the examination of the behavior of u in the vicinity of the singular point, since the regularity properties of u are determined by the regularity of u_s. The main idea is the following: based on the a priori knowledge of the analytical form of u_s in $\mathscr{C}$, we carefully construct a locally adapted mesh in $\mathscr{C}$ by introducing a grading control parameter $\mu := \mu(\lambda, k)$, such as allows us to prove that the approximation order of the method applied on this adapted mesh for u_s is similar with the order of the method applied on the rest of the mesh (maybe quasi uniform) for u_r.

2.5.2 The Graded Mesh for $\mathscr{T}_H(\Omega)$ and Global Approximation Estimates

The area $U_s := \{x \in \Omega : |P_s - x| \leq R, R \geq N_Z h, N_Z \geq 2\}$ is further subdivided into ring zones Z_ζ, $\zeta = 0, .., \zeta_M < N_Z$, with distance from P_s equal to $D_{(Z_\zeta, P_s)} := C(n_\zeta h)^{\frac{1}{\mu}}$, where $1 \leq n_\zeta < N_Z$ and $\frac{1}{2} \leq C \leq 1$. The radius of every zone is defined to be $R_{Z_\zeta} := D_{(Z_{\zeta+1}, P_s)} - D_{(Z_\zeta, P_s)} = C(n_{\zeta+1}h)^{\frac{1}{\mu}} - C(n_\zeta h)^{\frac{1}{\mu}}$.

For convenience, we assume that the initial subdivision $\mathscr{T}_H$ fulfill the following conditions, for an illustration see Fig. 3 (right) with $\zeta_M = 2$:

- The subdomains can be grouped into those which belong (entirely) to the area U_s and those that belong (entirely) to $\Omega \setminus U_s$. This means that there is no Ω_i, $i = 1, \ldots, N$ such that $U_s \cap \Omega_i \neq \emptyset$ and $(\Omega \setminus U_s) \cap \Omega_i \neq \emptyset$.

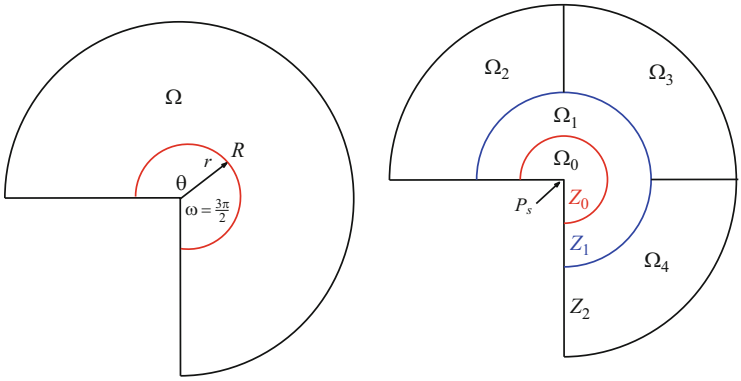


Fig. 3 *Left*: 2d domain with corner singularity. *Right*: zone partition and the subdomians Ω_i of $\mathscr{T}_H(\Omega)$

- Every ring zone Z_ζ is partitioned into "circular" subdomains Ω_{i_ζ} which have radius Ω_{i_ζ} equal to the radius of the zone, that is $R_{\Omega_{i_\zeta}} = R_{Z_\zeta}$. For computational efficiency reasons, we prefer -if its possible- every zone to be only represented by one subdomain.
- The zone Z_0 is represented by one subdomain, say Ω_0, and the mesh $T_{h_0}^{(0)}(\Omega_0)$ includes all $E \in T_h(\Omega)$ such that $\partial E \cap P_s \neq \emptyset$.

The graded meshes $T_{h_{i_\zeta}}^{(i_\zeta)}(\Omega_{i_\zeta})$ are mainly determined by the grading parameter $\mu(\lambda, k) \in (0, 1]$ and the mesh sizes h_{i_ζ} are chosen to satisfy the following properties: for Ω_{i_ζ} with distance $R_{\Omega_{i_\zeta}}$ from P_s, the mesh size h_{i_ζ} is defined to be $h_{i_\zeta} = \mathscr{O}(hR_{\Omega_{i_\zeta}}^{1-\mu})$ and for $T_{h_0}^{(i_0)}(\Omega_0)$ the mesh size is of order $h_{i_0} = \mathscr{O}(h^{\frac{1}{\mu}})$, more details are given in [28]. Based on previous properties of the $T_{h_{i_\zeta}}^{(i_\zeta)}(\Omega_{i_\zeta})$ meshes, we can conclude the relations

$$C_m h^{\frac{1}{\mu}} \leq h_{i_\zeta} \leq C_M h^{\frac{1}{\mu}}, \qquad \text{if } \overline{\Omega}_{i_\zeta} \cap P_s \neq \emptyset, \tag{31}$$

$$C_m h R_{\Omega_{i_\zeta}}^{1-\mu} \leq h_{i_\zeta} \leq C_M h D_{(Z_\zeta, P_s)}^{1-\mu}, \qquad \text{if } \overline{\Omega}_{i_\zeta} \cap P_s = \emptyset. \tag{32}$$

Using the local interpolation estimate of Lemma 5 in every $\Omega_{i_\zeta} \subset U_s$ and the characteristics of the meshes $T_{h_{i_\zeta}}^{(i_\zeta)}(\Omega_{i_\zeta})$, we can easily obtain the estimate

$$\|u_s - \Pi_h u_s\|_{dG(U_s)} \leq \sum_{i_\zeta} C_{i_\zeta} h_{i_\zeta}^{\lambda}, \tag{33}$$

since $u_s \in W^{l=1+\lambda,2}(\Omega)$ (and also $u \in W^{2,p=\frac{2}{2-\lambda}}(\Omega)$) and we do not consider non-matching grid interfaces. Using the mesh properties (31) and (32), estimate (33) and

Lemma 5, we can prove the following global error estimate of the proposed dG IgA method applied to problems with boundary singularities, see [28].

Theorem 2 *Let Z_ζ be a partition of $\Omega \subset \mathbb{R}^2$ into ring zones and let $\mathcal{T}_H$ to be a subdivision to Ω with the properties as listed in the previous paragraphs. Let $T_{h_i}^{(i)}(\Omega_i)$ be the meshes of subdomains as described above. Then for the solution u of (2) and the dG IgA solution u_h, we have the approximation result*

$$\|u - u_h\|_{dG} \leq Ch^r, \text{ with } r = \min(k, \lambda/\mu), \tag{34}$$

where the constant $C > 0$ depends on the characteristics of the mesh and on the mappings Φ_i (see (6)) but not on h_i.

2.5.3 Numerical Examples

In this section, we present a series of numerical examples in order to validate the theoretical analysis on the graded mesh in Sect. 2.5. The first example concerns a two-dimensional problem with a boundary point singularity (L-shape domain). The second example is the interior point singularity problem of the Sect. 2.4.

1. **Boundary Singular Point**
   One of the classical test cases is the singularity due to a re-entrant corner. The L-shape domain given by $(-1,1)^2 \setminus (-1,0)^2$. In Fig. 4 (left), the subdivision of Ω into two subdomains is presented. The exact solution is $u = r^{\frac{\pi}{\omega}} \sin(\theta\pi/\omega)$, where $\omega = 3\pi/2$. We set $\Gamma_D = \partial\Omega$ and the data f, u_D of (2) are specified by the given exact solution. The problem has been solved using B-splines of degree $k = 1$ and $k = 2$ and the grading parameter is $\mu = 0.6$ and $\mu = 0.3$ respectively. In Fig. 4 (middle), the graded mesh for $\mu = 0.6$ is presented and in Fig. 4 (right) the contours of the numerical solution are plotted. In Table 3, we present the convergence rates of the method without grading (left columns). As we can see, the convergence rates are determined by the regularity of the solution around the singular boundary point. In the right columns, we present the convergence rates corresponding to the graded meshes. We can see that the rates tend to be optimal with respect the B-spline degree.
2. **Interior Point Singularity**
   The domain is $\Omega = (-1,1)^2$. We consider a solution u of (1) with a point singularity at the origin given by $u(x) = |x|^\lambda$. We set $\lambda = 0.6$ and is easy to show that $u \in W^{l=1.6,2}(\Omega)$. We set $\alpha = 1$ in Ω. In the left columns of Table 3, we display the convergence rates for degrees $k = 1$ and $k = 2$ without mesh grading. The convergence rates are suboptimal and follow the approximation estimate (27). The problem has been solved again on graded meshes with $\mu = 0.6$ for $k = 1$ and $\mu = 0.3$ for $k = 2$, see Fig. 5. We display the convergence rates in the right columns of Table 3. The rates tend to be optimal as it was expected.

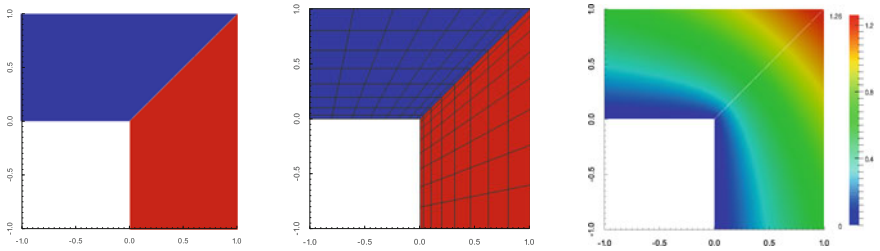


Fig. 4 L-shape test. *Left*: subdomains, *middle*: graded mesh with $\mu = 0.6$, *right*: contours of u_h

Table 3 The convergence rates for the L-shape test (*left*) and for the internal singularity test (*right*)

	No grading		With grading	
$\frac{h}{2^s}$	$k=1$	$k=2$	$k=1$ $\mu=0.6$	$k=2$ $\mu=0.3$
Convergence rates				
$s=0$	–	–	–	–
$s=1$	0.636	0.650	0.915	1.672
$s=2$	0.641	0.657	0.933	1.919
$s=3$	0.647	0.661	0.946	1.987
$s=4$	0.652	0.666	0.957	2.071

	No grading		With grading	
$\frac{h}{2^s}$	$k=1$	$k=2$	$k=1$ $\mu=0.6$	$k=2$ $\mu=0.3$
Convergence rates				
$s=0$	–	–	–	–
$s=1$	0.508	0.573	0.855	1.661
$s=2$	0.547	0.580	0.930	1.828
$s=3$	0.570	0.586	0.969	1.928
$s=4$	0.583	0.591	0.990	1.951

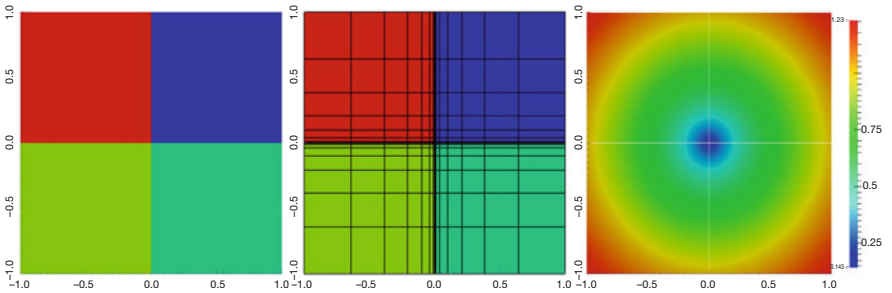


Fig. 5 Interior singularity test. *Left*: the subdomains Ω_i, $i = 1, \ldots, 4$, *middle*: the graded mesh with $\mu = 0.3$, *right*: the contours of u_h

3 Multipatch dG IgA for PDEs on Surfaces

3.1 Diffusion Problems on Open and Closed Surfaces

Let us now consider a diffusion problem of the form (1) on a sufficiently smooth, open surface Ω, the weak formulation of which can formally be written in the same form as (2) in Sect. 2: find a function $u \in W^{1,2}(\Omega)$ such that $u = u_D$ on the boundary

$\partial\Omega$ of the surface Ω and satisfies the variational formulation

$$a(u,v) = l(v), \ \forall v \in W_0^{1,2}(\Omega), \tag{35}$$

with the bilinear and linear forms $a(\cdot,\cdot)$ and $l(\cdot)$, but now defined by the relations

$$a(u,v) = \int_\Omega \alpha \, \nabla_\Omega u \cdot \nabla_\Omega v \, d\Omega \quad \text{and} \quad l(v) = \int_\Omega f v \, d\Omega,$$

respectively, where ∇_Ω denotes the surface gradient, see, e.g., Definition 2.3 in [18] for its precise description. For simplicity of the presentation, we here assume Dirichlet boundary condition. It is clear that other boundary conditions can be treated in the same framework as it was done in [29] for mixed boundary conditions. In the case of open surfaces with pure Neumann boundary condition and closed surfaces, we look for a solution $u \in W^{1,2}(\Omega)$ satisfying the uniqueness condition $\int_\Omega u(x)dx = 0$ and the variational equation (2) under the solvability condition $l(1) = 0$. In Sect. 3.4, we present and discuss the numerical results obtained for different diffusion problems on an open (Car) and on two closed (Sphere, Torus) surfaces.

3.2 Multipatch dG IgA Discretization

Let $\mathcal{T}_H(\Omega) = \{\Omega_i\}_{i=1}^N$ be again a partition of our physical computational domain Ω, that is now a surface, into non-overlapping patches (sub-domains) Ω_i such that (3) holds, and let each patch Ω_i be the image of the parameter domain $\hat{\Omega} = (0,1)^2 \subset \mathbb{R}^2$ by some NURBS mapping $\Phi_i : \hat{\Omega} \to \Omega_i \subset \mathbb{R}^3$, $\hat{x} = (\hat{x}_1, \hat{x}_2) \mapsto x = (x_1, x_2, x_3) = \Phi_i(\hat{x})$, which can be represented in the form

$$\Phi_i(\hat{x}_1, \hat{x}_2) = \sum_{k_1=1}^{n_1} \sum_{k_2=1}^{n_2} C^{(i)}_{(k_1,k_2)} \hat{B}^{(i)}_{(k_1,k_2)}(\hat{x}_1, \hat{x}_2) \tag{36}$$

where $\{\hat{B}^{(i)}_{(k_1,k_2)}\}$ are the bivariate NURBS basis functions, and $\{C^{(i)}_{(k_1,k_2)}\}$ are the control points, see [8] for a detailed description. We always assume that the mapping Φ_i is regular. Therefore, the inverse mapping $\hat{x} = \Psi_i(x) := \Phi_i^{-1}(x)$ is well defined for all patches Ω_i, $i = 1, \ldots, N$.

Now the dG IgA scheme for solving our surface diffusion problem (35) can formally be written in the form (13) as in Sect. 2: find $u_h \in \mathbb{B}_h(\mathcal{T}_H(\Omega))$ such that

$$a_h(u_h, v_h) = l(v_h) + p_D(u_D, v_h), \ \forall v_h \in \mathbb{B}_h(\mathcal{T}_H(\Omega)), \tag{37}$$

where $a_h(\cdot,\cdot)$ and $p_D(\cdot,\cdot)$ are defined by (14) provided that we replace the gradient ∇ by the surface gradient ∇_Ω. Since the dG bilinear form $a_h(\cdot,\cdot)$ is positive on

$\mathbb{B}_h(\mathscr{T}_H(\Omega)) \setminus \{0\}$ for sufficiently large μ, cf. Lemma 6, there exist a unique dG solution $u_h \in \mathbb{B}_h(\mathscr{T}_H(\Omega))$. The dG IgA scheme (37) is equivalent to a system of algebraic equations of the form

$$K_h \underline{u}_h = \underline{f}_h, \tag{38}$$

the solution $\underline{u}_h$ of which gives us the coefficients (control points) of u_h. In order to generate the entries of the system matrix K_h and the right-hand side $\underline{f}_h$, we map the patches Ω_i composing the physical domain, i.e., our surface Ω, into the parameter domain $\hat{\Omega} = (0,1)^2$. For instance, for the broken part $a_i(u_h, v_h)$ of the bilinear form $a_h(u_h, v_h)$, we obtain

$$\begin{aligned} a_i(u_h, v_h) &= \int_{\Omega_i} \alpha^{(i)}\, \nabla_\Omega u_h(x) \cdot \nabla_\Omega u_h(x)\, d\Omega \\ &= \int_{\hat{\Omega}_i} \alpha^{(i)}\, [J_i(\hat{x}) F_i^{-1}(\hat{x}) \hat{\nabla} \hat{u}_i(\hat{x})]^\top [J_i(\hat{x}) F_i^{-1}(\hat{x}) \hat{\nabla} \hat{v}_i(\hat{x})] g_i(\hat{x})\, d\hat{x} \\ &= \int_{\hat{\Omega}_i} \alpha^{(i)}\, (\hat{\nabla} \hat{u}_i(\hat{x}))^\top\, F_i^{-1}(\hat{x})\, \hat{\nabla} \hat{v}_i(\hat{x})\, g_i(\hat{x}) d\hat{x}\,, \end{aligned}$$

where $J_i(\hat{x}) = \partial \Phi_i(\hat{x}) / \partial \hat{x}$, $F_i(\hat{x}) = (J_i(\hat{x}))^\top (J_i(\hat{x}))$ and $g_i(\hat{x}) = (\det F_i(\hat{x}))^{1/2}$ denote the Jacobian, the first fundamental form and the square root of the associated determinant, respectively. These terms, coming from the parameterization of the domain, can be exploited for deriving efficient matrix assembly methods, cf. [31, 32]. Furthermore, we use the notations $\hat{u}_i(\hat{x}) = u_h(\Phi_i(\hat{x}))$ and $\hat{\nabla} = (\frac{\partial}{\partial \hat{x}_1}, \frac{\partial}{\partial \hat{x}_2})^\top$.

3.3 Discretization Error Estimates

In [29], we derived discretization error estimates of the form

$$\|u - u_h\|_{dG}^2 \leq C \sum_{i=1}^{N} \alpha^{(i)} h_i^{2t} \|u\|_{H^{1+t}(\Omega_i)}^2, \tag{39}$$

with $t := \min\{s, k\}$, provided that the solution u of our surface diffusion problem (35) belongs to $H^{1+s}(\mathscr{T}_H(\Omega)) = W^{1+s,2}(\mathscr{T}_H(\Omega))$ with some $s > 1/2$. In the case $t = k$, estimate (39) yields the convergence rate $\mathscr{O}(h^k)$ with respect to the dG norm, whereas the Aubin-Nitsche trick provides the faster rate $\mathscr{O}(h^{k+1})$ in the L_2 norm. Here k always denotes the underlying polynomial degree of the NURBS. This convergence behavior is nicely confirmed by our numerical experiments presented in [29] and in the next subsection.

In [29], we assumed matching meshes and some regularity of the solution of (35), namely $u \in H^{1+s}(\mathscr{T}_H(\Omega))$. It is clear that the results of Theorem 1, which

includes no-matching meshes and low-regularity solutions, can easily be carried over to diffusion problems on open and closed surfaces. The same is true for mesh grading techniques presented in Sect. 2.5.

3.4 Numerical Examples

3.4.1 Sphere

Let us start with a diffusion problem on a closed surface Ω that is given by the sphere $\Omega = \{(x, y, z) \in (-1, 1)^3 : x^2 + y^2 + z^2 = 1\}$ with unitary radius. The computational domain Ω is decomposed into 6 patches, see left-hand side of Fig. 6. The knot vectors representing the geometry of each patch are given as $\Xi_{1,2} = (0, 0, 0, 0, 0, 1, 1, 1, 1, 1)$ in both directions. Since the surface is closed, we impose the uniqueness constraint $\int_\Omega u \, d\Omega = 0$ on the solution. The right-hand side $f(r, \phi, \theta) = 12u(r, \phi, \theta)$, where the solution $u(r, \phi, \theta) = 12 \sin(3\phi) \sin^3(\theta)$ is an eigenfunction of the Laplace-Beltrami operator $(-\Delta_\Omega)$ satisfying the compatibility condition $\int_\Omega f \, d\Omega = 0$. The example can also be found in [21]. The diagrams displayed in Fig. 7 show the error decay with respect to the L_2 (left) and dG (right) norms for polynomial degrees $k = 1, 2, 3$, and 4. As expected by our theoretical results, we observe the full convergence rates $\mathcal{O}(h^{k+1})$ and $\mathcal{O}(h^k)$ for the L_2 norm and the dG norm, respectively. In Table 4, we compare the L_2 errors of dG IgA solutions with those produced by the corresponding continuous (standard) Galerkin (cG) IgA scheme for the polynomial degree $k = 5$. In the case of smooth solutions, the dG IgA is as good as the cG counterpart. The same is true for the errors with respect to the dG norm.

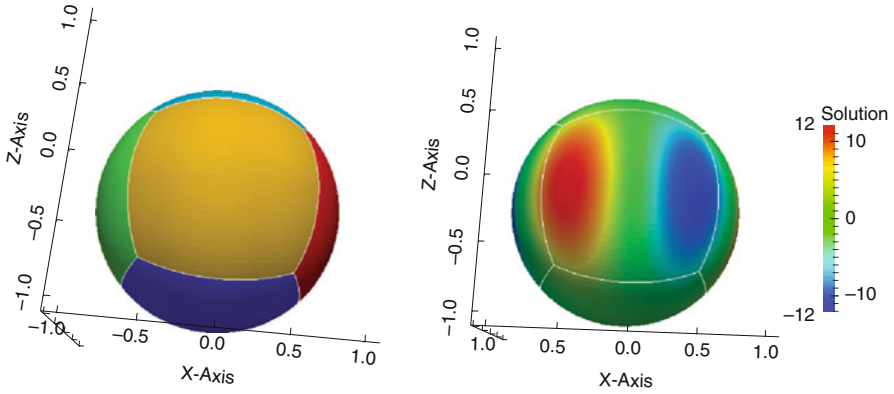


Fig. 6 Sphere: geometry and decomposition into 6 patches (*left*) and solution for the Laplace-Beltrami problem (*right*)

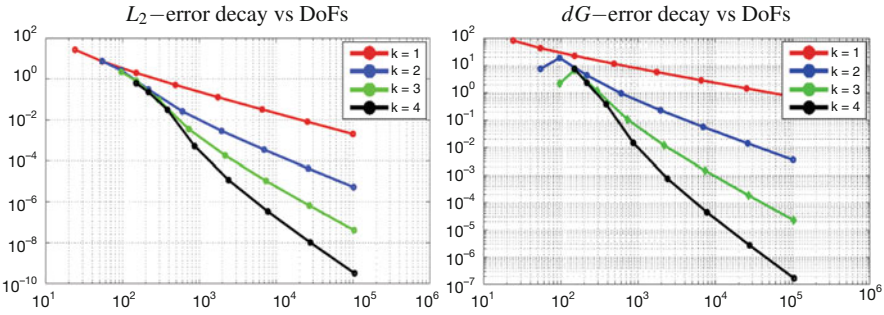


Fig. 7 Sphere: error decay in the L_2 (*left*) and dG (*right*) norms for polynomial degrees 1 to 4

Table 4 Sphere: comparison of the cG and dG IgA error decay in the L_2 norm for $k = 5$

$k = 5$	cG-IgA		dG-IgA	
Dofs	L_2 error	Conv. rate	L_2 error	Conv. rate
216	0.168803	0	0.166582	0
294	0.0602254	1.48689	0.0599854	1.16908
486	0.00900833	2.74104	0.00898498	2.55496
1014	8.90909×10^{-5}	6.65984	8.90774×10^{-5}	6.537
2646	8.64021×10^{-7}	6.68807	8.63906×10^{-7}	6.85905
8214	1.16592×10^{-8}	6.21152	1.16582×10^{-8}	6.27626
28,566	1.75119×10^{-10}	6.05699	1.75127×10^{-10}	6.06894

3.4.2 Torus

We now consider the closed surface

$$\Omega = \{(x, y) \in (-3, 3)^2, z \in (-1, 1) : \ r^2 = z^2 + (\sqrt{x^2 + y^2} - R^2)\}$$

that is nothing but a torus that is decomposed into 4 patches, see Fig. 8 left. The knot vectors $\Xi_1 = \{0, 0, 0, 0.25, 0.25, 0.50, 0.50, 0.75, 0.75, 1, 1, 1\}$ and $\Xi_2 = \{0, 0, 0, 1, 1, 1\}$ describe the NURBS used for the geometrical representation of the patches. We first consider the surface Poisson equation, also called Laplace-Beltrami equation, with the right-hand side

$$\begin{aligned} f(\phi, \theta) = r^{-2} &[9 \sin(3\phi) \cos(3\theta + \phi)] \\ &- \left[(R + r\cos(\theta))^{-2}(-10 \sin(3\phi) \cos(3\theta + \phi) - 6 \cos(3\phi) \sin(3\theta + \phi))\right] \\ &- \left[(r(R + r\cos(\theta))^{-1})(3 \sin(\theta) \sin(3\phi) \sin(3\theta + \phi))\right], \end{aligned}$$

where $\phi = \arctan(y/x)$, $\theta = \arctan(z/(\sqrt{x^2 + y^2} - R))$, $R = 2$, and $r = 1$. The exact solution is given by $u = \sin(3\phi) \cos(3\theta + \phi)$, cf. also [21]. We mention

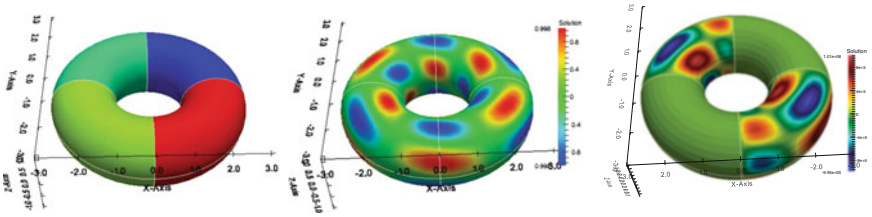


Fig. 8 Torus: geometry and decomposition into 4 patches (*left*), solution for the Laplace-Beltrami problem (*middle*), solution for jumping coefficients (*right*)

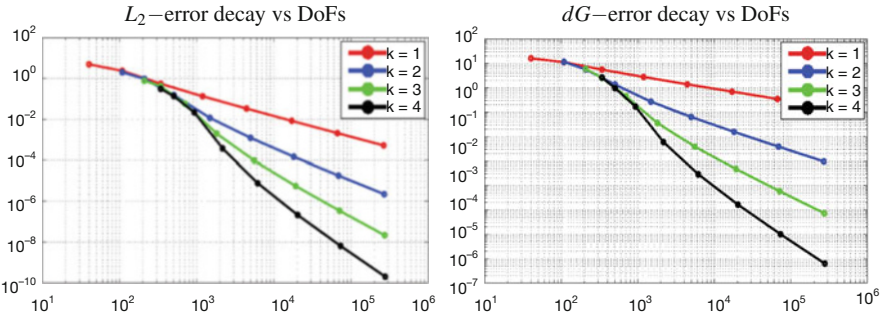


Fig. 9 Torus: error decay in the L_2 (*left*) and dG (*right*) norms for polynomial degrees 1 to 4

Table 5 Torus: comparison of the cG and dG IgA error decay in the L_2 norm for $k = 5$

$k = 5$	cG-IgA		dG-IgA	
Dofs	L_2 error	Conv. rate	L_2 error	Conv. rate
504	0.0974255	0	0.0973029	0
700	0.0433491	1.1683	0.043271	1.16908
1188	0.00736935	2.55639	0.00736339	2.55496
2548	7.9296×10^{-5}	6.53814	7.92948×10^{-5}	6.537
6804	6.83083×10^{-7}	6.85904	6.83068×10^{-7}	6.85905
21,460	8.8131×10^{-9}	6.27627	8.81294×10^{-9}	6.27626
75,348	1.3131×10^{-10}	6.0686	1.31276×10^{-10}	6.06894

that the functions u and f are chosen such that the zero mean compatibility condition holds. The IgA approximation to this solution is depicted in middle picture of Fig. 8. The diagrams displayed in Fig. 9 show the error decay with respect to the L_2 (left) and dG (right) norms for polynomial degrees $k = 1, 2, 3$, and 4. As expected by our theoretical results, we observe the full convergence rates $\mathcal{O}(h^{k+1})$ and $\mathcal{O}(h^k)$ for the L_2 norm and the dG norm, respectively. In Table 5, we compare the L_2 errors of dG IgA solutions with those produced by the corresponding cG IgA scheme for the polynomial degree $k = 5$. In the case of smooth solutions, the dG IgA is as good as the cG counterpart. The same is true for the errors with respect to the dG norm.

Table 6 Torus: L_2 and energy norm errors with degree $k = 2$

Dofs	L_2 error	Conv. rate	dG error	Conv. rate
108	2.04841×10^6	0	8.56088×10^6	0
208	1.03579×10^6	0.98377	4.89856×10^6	0.805403
504	128,215	3.0141	1.09244×10^6	2.1648
1480	15,431.9	3.05458	258,162	2.08121
4968	1808.17	3.09331	64,084.3	2.01023
18,088	213.252	3.0839	173,61.8	1.88406
268,840	Reference	Solution	Reference	Solution

Now we consider the case when the diffusion coefficients α have large jumps across the boundaries of the patches in which the torus was decomposed. More precisely, we assume that the diffusion coefficient $\alpha = \alpha^{(i)} > 0$ in the patch Ω_i, $i = 1, 2, 3, 4$, where $\alpha^{(2)} = \alpha^{(4)} = 1$ and $\alpha^{(1)} = \alpha^{(3)} = 10^{-6}$. The patches are arranged from blue Ω_1 to red Ω_4. The right-hand side f is the same as given above for the case of the Laplace-Beltrami problem, i.e. for $\alpha = 1$. Now the solution is not known, but we know that the solution is smooth in the patches $\overline{\Omega_i}$ and has steep gradients towards $\partial\Omega_2$ and $\partial\Omega_4$, see also of Fig. 8 right. Due to our theory, we can expect full convergence rates since we have an exact representation of the geometry. Indeed, we realize the full convergence rate by choosing a fine grid as the reference solution and comparing the solutions at each refinement step against this reference solution as shown in Table 6.

3.4.3 Car

We now consider the diffusion problem on an open, free-form surface. In order to demonstrate the fact that our results are general and not limited to academic examples, we apply our methods to a CAD model representing a car shell.

The surface is composed of eight quadratic B-spline patches, shown in Fig. 10 (left). The model exhibits several small bumps in the interior of the surface and sharp corners on the boundary. In addition, the patches have varying areas and meet along curved one-dimensional (quadratic) B-spline interfaces.

We choose a constant diffusion coefficient on the whole domain and we prescribe homogeneous Dirichlet conditions along the boundary. For the right-hand side, we used a (globally defined in $\mathbb{R}^3$) linear function which we restricted on the surface, in order to obtain a smooth solution to our problem.

As suggested by the isogeometric paradigm, we used quadratic B-spline basis functions, forwarded on the surface, as discretization basis. We started with a coarse grid, which was $h-$refined uniformly several times, in both parametric directions.

An analytic formula for the exact solution is not available. Therefore, we have chosen a fine grid of approximately one million degrees of freedom as the reference solution. Comparing against this solution (obtained by the corresponding cG IgA

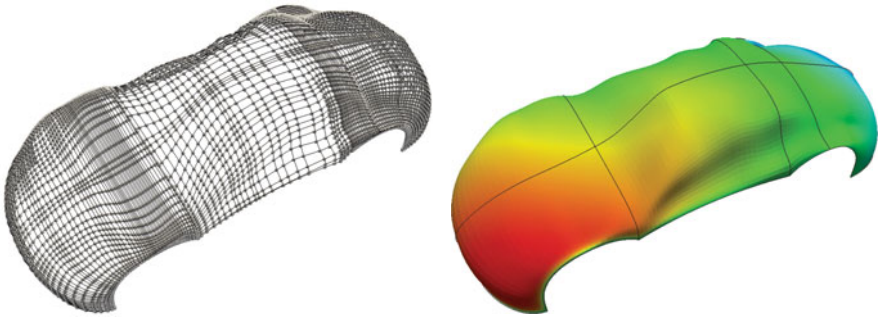


Fig. 10 Car: control points of the geometry (*left*), IgA solution of the Dirichlet surface diffusion problem (*right*). *Black lines* indicate patch interfaces

Table 7 Diffusion on a car-shell model. Numerical results for continuous Galerkin IgA with strong imposition of Dirichlet boundary (cG IgA), weak imposition (Nitsche-type) and finally patch-discontinuous Galerkin (dG IgA)

$k = 2$	cG IgA		Nitsche-type BCs		dG IgA	
DoFs	L_2 error	Rate	L_2 error	Rate	L_2 error	Rate
96	1.87598		1.77144		1.77029	
192	1.23006	0.608922	1.19201	0.571527	1.19188	0.570749
480	0.633648	0.956971	0.623767	0.934312	0.623764	0.934161
1440	0.241842	1.38962	0.239835	1.37897	0.239835	1.37896
4896	0.0659275	1.87511	0.0655902	1.87049	0.0655902	1.87049
17,952	0.0115491	2.5131	0.011502	2.5116	0.011502	2.5116
68,640	0.00155178	2.89578	0.00154625	2.89504	0.00154624	2.89504
∼268 K	0.00017567	3.14298	0.000175096	3.14256	0.000175094	3.14257
∼1 M	Reference	Reference	Reference	Reference	Reference	Reference

method), the expected convergence rates have been observed. Table 7 contains the numeric results for the L^2 norm, obtained using either the continuous (with strong or weak imposition of Dirichlet boundaries) or dG IgA method. Apart from the observed order of convergence, the magnitude of the error agrees in all cases as well.

4 The G+SMO C++ Library

Isogeometric analysis requires seamless integration of Finite Element Analysis (FEA) and Computer-aided design (CAD) software. The existing software libraries, however, cannot be adapted easily to the rising new challenges since they have been designed and developed for different purposes. In particular, FEA codes are traditionally implemented by means of functional programming, and are focused on treating nodal shape function spaces. In CAD packages, on the other hand, the

central objects are free-form curves and surfaces, defined by control points, which are realized in an object-oriented programming environment.

G+SMO is an object-oriented, template C++ library, that implements a generic concept for IGA, based on abstract classes for geometry map, discretization basis, assemblers, solvers and so on [25]. It makes use of object polymorphism and inheritance techniques in order to support a variety of different discretization bases, namely B-spline, Bernstein, NURBS bases, hierarchical and truncated hierarchical B-spline bases of arbitrary polynomial order, and so on.

The library is open-source and is licenced under the Mozilla Public License v2.0. The source code, together with a reference manual and wiki pages can be reached at http://www.gs.jku.at/gismo.

Our design allows the treatment of geometric entities such as surfaces or volumes through dimension independent code, realized by means of template meta-programming. Available features include simulations using continuous and discontinuous Galerkin approximation of PDEs, over conforming and non-conforming multi-patch computational domains. PDEs on surfaces as well as integral equations arising from elliptic boundary value problems. Boundary conditions may be imposed both strongly and weakly. In addition to advanced discretization and generation techniques, efficient solvers like multigrid iteration schemes are available. Methods for solving non-linear problems are under development. Finally, we aim to employ existing high-end libraries for large-scale parallelization.

In the following paragraphs we shall provide more details on the design and features of the library.

4.1 Description of the Main Modules

The library is partitioned into smaller entities, called modules. Currently, there are six (6) modules in G+SMO namely Core, Matrix, NURBS, Modeling, Input/Output and Solver modules.

The **Core module** is the backbone of the library. Here an abstract interface is defined for a *basis*, that is, a set of real-valued functions living on a parameter domain. At this level, we do not specify how these functions (or its derivatives) should be evaluated. However, a number of virtual member functions define an interface that should be implemented by derived classes of this type. Another abstract class is the *geometry* class. This object consists of a (still abstract) basis and a coefficient vector, and represents a patch. Note that parameter or physical dimension are not specified at this point. There are four classes directly derived from the geometry class; these are curve, surface, volume and bulk. These are parametric objects with known parameter space dimension 1, 2, 3 and 4 respectively.

Another abstract class is a *function* class. The interface for this class includes evaluation, derivations and other related operations. The *geometry* abstract class is actually deriving from the function class, demonstrating the fact that parametric geometries can be simply viewed as (vector) functions. Another interesting object

is the *multipatch* object. CAD models are composed of many patches. Therefore, a multipatch structure is of great importance. It contains two types of information; first, geometric information, essentially a list of geometry patches. Second, topological information between the patches, that is, the adjacency graph between patch boundaries, degenerate points, and so on. Let us also mention the *field* class, which is the object that typically represents the solution of a PDE. A *field* is a mathematical scalar or vector field which is defined on a parametric patch, or multipatch object. It may be evaluated either on the parameter or physical space, as the isogeometric paradigm suggests.

The **Matrix module** contains all the linear algebra related infrastructure. It is based on the third party library Eigen [22]. The main objects are dense and sparse matrices and vectors. Typical matrix decompositions such as LU, QR, SVD, and so on, are available. Furthermore, the user has also access to iterative solvers like conjugate gradient methods with different preconditioners. Finally, one can use popular high-end linear solver packages like SuperLU[2] through a common interface.

The **NURBS module** consists of B-Splines, NURBS, Bézier of arbitrary degrees and knot-vectors, tensor-product B-splines of arbitrary spatial dimension.

The **Modeling module** provides data structures and geometric operations that are needed in order to prepare CAD data for analysis. It contains trimmed surfaces, boundary represented (B-rep) solids and triangle meshes. Regarding modeling operations, B-Spline fitting, smoothing of point clouds, Coon's patches, and volume segmentation methods are available.

The **Input/Output module** is responsible for visualization as well as file reading and writing. For visualization, we employ Paraview or Axel (http://axel.inria.fr) and VTK at INRIA, France. An important issue is file formats. Even if NURBS is an industrial standard, a variety of different formats are used in the CAD industry to exchange NURBS data. In G+SMO, we have established I/O with popular CAD formats, which include the 3DM file format of Rhinoceros 3D modeler, the `X_T` format of Siemens' NX platform as well as an (exported) format used by the LS-DYNA general-purpose finite element program.

The **Assembler module** can already treat a number of PDE problems like convection-diffusion problems, linear elasticity, Stokes equations as well as diffusion problems on surfaces by means of continuous or discontinuous Galerkin methods, including divergence preserving discretizations for Stokes equations. Strong or weak imposition Dirichlet boundary conditions and Neumann-type conditions are provided. Boundary element IgA collocation techniques are also available.

Apart from the modules described above, there are several more which are under development. These include a hierarchical bases module, an optimization module and a triangular Bézier module, see Fig. 11.

[2] http://acts.nersc.gov/superlu

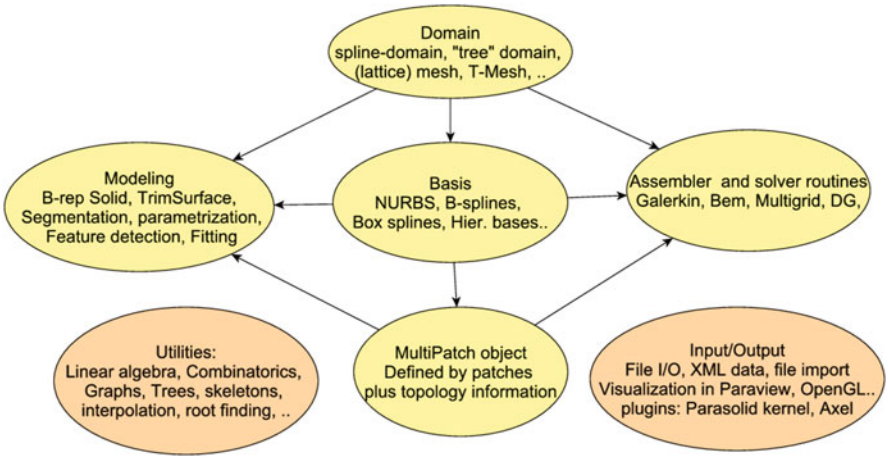


Fig. 11 A diagram of the different parts in the library and their interactions

4.2 Development Framework

The library uses a set of standard C++ tools in order to achieve cross-platform functionality. Some compilers that have been tested includes Microsoft's Visual C++, Mingw32, GCC, Clang and the Intel compiler.

The main development tools used are:

- CMake cross-platform build system for configuration of the code. A small set of options are available in order to enable optional parts of the library that require external packages. Furthermore, the user is able to enable debug mode build, that disables code optimization and facilitates development, by attaching debug information in the code.
- The code is available in an SVN repository. We have chosen the continuous integration as development policy. That is, all developers commit their contributions in a mainline repository.
- Trac software management system is used for bug reporting. Additionally, we use the integrated wiki for documentation and user guidance.
- A CDASH software testing server is employed for regular compilation and testing of the mainline code. Nightly builds are executed on different platforms, as well as continuous builds after user commits. This allows to easily correct errors and ensure the quality of the library. Coverage analysis and memory checks are also performed regularly.
- The in-source documentation system Doxygen is used heavily in the code.
- A mailing list is available for communication and user support.

Regarding tools that are employed in the library:

- The C++ Standard Library is employed. This is available by default on C++ modern installations.
- The Eigen C++ Linear Algebra library is used for linear algebra operations. This library features templated coefficient type, dense and sparse matrices and vectors, typical matrix decompositions (LU, QR, SVD,...) as well as iterative solvers and wrappers for high-end packages (SuperLU, PaStiX, SparseSuite,...).
- XML reader/writer for input/output of XML files, and other CAD formats (OFF, STL, OBJ, GeoPDEs, 3DM, Parasolid).
- Mathematical Expression Toolkit Library (ExprTk). This is an expression-tree evaluator for mathematical function expressions that allows input of functions similar to Matlab's interface.

4.3 Additional Features and Extensions

Apart from the basic set of tools provided in the library, the user has the possibility to enable features that require third-party software. The following connections to external tools is provided:

- OpenNurbs library used to support Rhino's 3DM CAD file format. This allows file exchange with standard CAD software.
- Connection to Parasolid geometric kernel. Enabling this feature allows input and output of the `x_t` file format. Also, the user can employ advanced modeling operations like intersection, trimming, boolean operations, and so on.
- Connection to LS-DYNA. The user can output simulation data that can be used by LS-DYNA's Generalized element module system, for performing for instance simulations on shells.
- Connection to IPOPT nonlinear optimization library. With this feature one can use powerful interior point constrained optimization algorithms in order to perform, for instance, isogeometric shape optimization.
- MPFR library for multi-precision floating point arithmetic. With this feature critical geometric operations can be performed with arbitrary precision, therefore guaranteeing a verified result.
- Graphical interface, interaction and display using Axel modeler.

4.4 Plugins for Third-Party Platforms

Finally, there are currently two plugins under development. The plugins allow third-party software to employ and exchange data with G+SMO.

- Axel Modeler is an open-source spline modeling package based on Qt and VTK. Our plugin allows to use G+SMO within this graphical interface, and provides user interaction (e.g. control point editing) and display.
- A Matlab interface is under development. This allows to use operations available in G+SMO within Matlab.

Acknowledgements This research was supported by the Research Network "Geometry + Simulation" (NFN S117) funded by the Austrian Science Fund (FWF) under the grand S117-03.

References

1. R.A. Adams, J.J.F. Fournier, *Sobolev Spaces* Pure and Applied Mathematics, vol. 140, 2nd edn. (Elsevier/Academic Press, Amsterdam, 2003)
2. P. Antonietti, A. Dedner, P. Madhavan, S. Stangalino, B. Stinner, M. Verani, High order discontinuous Galerkin methods on surfaces. SIAM J. Numer. Anal. **53**(2), 1145–1171 (2015)
3. T. Apel, F. Milde, Comparison of several mesh refinement strategies near edges. Commun. Numer. Methods Eng. **12**, 373–381 (1996)
4. T. Apel, A.-M. Sändig, J.R. Whiteman, Graded mesh refinement and error estimates for finite element solutions of elliptic boundary value problems in non-smooth domains. Math. Methods Appl. Sci. **19**(30), 63–85 (1996)
5. Y. Bazilevs, L. Beirão da Veiga, J. Cottrell, T. Hughes, G. Sangalli, Isogeometric analysis: approximation, stability and error estimates for *h*-refined meshes. Comput. Methods Appl. Mech. Eng. **194**, 4135–4195 (2006)
6. L. Beirão da Veiga, A. Buffa, J. Rivas, G. Sangalli, Some estimates for *hpk*-refinement in isogeometric analysis. Numerische Mathematik **118**(2), 271–305 (2011)
7. L. Beirão da Veiga, D. Cho, G. Sangalli, Anisotropic NURBS approximation in isogeometric analysis. Comput. Methods Appl. Mech. Eng. **209–212**(0):1–11 (2012)
8. J. Cottrell, T. Hughes, Y. Bazilevs, *Isogeometric Analysis: Toward Integration of CAD and FEA* (Wiley, Chichester, 2009)
9. L. Dede, A. Quarteroni, Isogeometric analysis for second order partial differential equations on surfaces. Comput. Methods Appl. Mech. Eng. **284**, 807–834 (2015)
10. A. Dedner, P. Madhavan, Adaptive discontinuous Galerkin methods on surfaces. Numerische Mathematik (2015). doi: 10.1007/s00211-015-0719-4
11. A. Dedner, P. Madhavan, B. Stinner, Analysis of the discontinuous Galerkin method for elliptic problems on surfaces. IMA J. Numer. Anal. **33**(3), 952–973 (2013)
12. D.A. Di Pietro, A. Ern, Analysis of a discontinuous Galerkin method for heterogeneous diffusion problems with low-regularity solutions. Numer. Methods Partial Differ. Equ. **28**(4), 1161–1177 (2012)
13. D.A. Di Pietro, A. Ern, *Mathematical Aspects of Discontinuous Galerkin Methods*. Mathématiques et Applications, vol. 69 (Springer, Berlin/Heidelberg, 2012)
14. M. Dryja, On discontinuous Galerkin methods for elliptic problems with discontinuous coefficients. Comput. Methods Appl. Math. **3**, 76–85 (2003)
15. G. Dziuk, Finite elements for the Beltrami operator on arbitrary surfaces, in *Partial Differential Equations and Calculus of Variations*, ed. by S. Hildebrandt, R. Leis. Lecture Notes in Mathematics, vol. 1357 (Springer, Berlin/Heidelberg, 1988), pp. 142–155
16. G. Dziuk, C. Elliott, Finite elements on evolving surfaces. IMA J. Num. Anal. **27**(2), 262–292 (2007)
17. G. Dziuk, C. Elliott, Surface finite elements for parabolic equations. J. Comput. Math. **25**(4), 385–407 (2007)

18. G. Dziuk, C. Elliott, Finite element methods for surface PDEs. Acta Numer. **22**, 289–396 (2013)
19. L.C. Evans, *Partial Differential Equestions*. Graduate Studies in Mathematics, vol. 19, 1st edn. (American Mathematical Society, Rhode Island, 1998)
20. P. Grisvard, *Elliptic Problems in Nonsmooth Domains*. Classics in Applied Mathematics, vol. 69 (SIAM, Philadelphia, 2011)
21. S. Gross, A. Reusken, *Numerical Methods for Two-phase Incompressible Flows*. Springer Series in Computational Mathematics, vol. 40 (Springer, Berlin/Heidelberg, 2011)
22. G. Guennebaud, B. Jacob et al., Eigen v3. http://eigen.tuxfamily.org (2010)
23. T. Hughes, J. Cottrell, Y. Bazilevs, Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. Comput. Methods Appl. Mech. Eng. **194**, 4135–4195 (2005)
24. J.W. Jeong, H.-S. Oh, S. Kang, H. Kim, Mapping techniques for isogeometric analysis of elliptic boundary value problems containing singularities. Comput. Methods Appl. Mech. Eng. **254**, 334–352 (2013)
25. B. Jüttler, U. Langer, A. Mantzaflaris, S.E. Moore, W. Zulehner, Geometry + Simulation Modules: Implementing Isogeometric Analysis. PAMM **14**(1), 961–962 (2014)
26. R.B. Kellogg, On the Poisson equation with intersecting interfaces. Appl. Anal. **4**, 101–129 (1975)
27. S.K. Kleiss, B. Jüttler, W. Zulehner, Enhancing isogeometric analysis by a finite element-based local refinement strategy. Comput. Methods Appl. Mech. Eng. **213–216**, 168–182 (2012)
28. U. Langer, A. Mantzaflaris, S.E. Moore, I. Toulopoulos, Mesh Grading in Isogeometric Analysis. Comput. Math. Appl. (2015). http://dx.doi.org/10.1016/j.camwa.2015.03.011
29. U. Langer, S. Moore, Discontinuous Galerkin Isogeometric Analysis of elliptic PDEs on surfaces. NFN Technical Report 12, Johannes Kepler University Linz, NFN Geometry and Simulation, Linz, 2014. http://arxiv.org/abs/1402.1185 and accepted for publication in the DD22 proceedings
30. U. Langer, I. Toulopoulos, Analysis of multipatch discontinuous Galerkin IgA approximations to elliptic boundary value problems. RICAM Reports 2014-08, Johann Radon Institute for Computational and Applied Mathematics, Austrian Academy of Sciences, Linz, 2014. http://arxiv.org/abs/1408.0182
31. A. Mantzaflaris, B. Jüttler, Exploring matrix generation strategies in isogeometric analysis, in *Mathematical Methods for Curves and Surfaces*, ed. by M. Floater, T. Lyche, M.-L. Mazure, K. Mørken, L. Schumaker. Lecture Notes in Computer Science, vol. 8177 (Springer, Berlin/Heidelberg, 2014), pp. 364–382
32. A. Mantzaflaris, B. Jüttler, Integration by interpolation and look-up for Galerkin-based isogeometric analysis. Comput. Methods Appl. Mech. Eng. **284**, 373–400 (2015)
33. L. Oganesjan, L. Ruchovetz, *Variational Difference Methods for the Solution of Elliptic Equations* (Isdatelstvo Akademi Nank Armjanskoj SSR, Erevan, 1979) (in Russian)
34. H.-S. Oh, H. Kim, J.W. Jeong, Enriched isogeometric analysis of elliptic boundary value problems in domains with cracks and/or corners. Int. J. Numer. Methods Eng **97**(3), 149–180 (2014)
35. B. Rivière, *Discontinuous Galerkin Methods for Solving Elliptic and Parabolic Equations: Theory and Implementation* (SIAM, Philadelphia, 2008)
36. F. Zhang, Y. Xu, F. Chen, Discontinuous Galerkin methods for isogeometric analysis for elliptic equations on surfaces. http://staff.ustc.edu.cn/~yxu/paper/IGA-DG.pdf (2014)

The Influence of Quadrature Errors on Isogeometric Mortar Methods

Ericka Brivadis, Annalisa Buffa, Barbara Wohlmuth, and Linus Wunderlich

Abstract Mortar methods have recently been shown to be well suited for isogeometric analysis. We review the recent mathematical analysis and then investigate the variational crime introduced by quadrature formulas for the coupling integrals. Motivated by finite element observations, we consider on the interface a quadrature rule purely based on the slave mesh as well as a method using on the interface quadrature rules based on the slave mesh and on the master mesh, resulting in a non-symmetric saddle point problem. While in the first case reduced convergence rates can be observed, in the second case the influence of the variational crime is less significant.

1 Introduction

Isogeometric analysis, introduced in 2005 by Hughes et al. in [17], is a family of methods that use B-splines and non-uniform rational B-splines (NURBS) as basis functions to construct numerical approximations of partial differential equations (PDEs). With isogeometric methods, the computational domain is generally split into patches. Within this framework, techniques to couple the numerical solution on

E. Brivadis (✉)
IUSS (Istituto Universitario di Studi Superiori di Pavia), Palazzo del Broletto, Piazza della Vittoria 15, 27100 Pavia, Italy

IMATI (Istituto di Matematica Applicata e Tecnologie Informatiche del CNR), Via Ferrata 1, 27100 Pavia, Italy
e-mail: ericka.brivadis@iusspavia.it

A. Buffa
IMATI (Istituto di Matematica Applicata e Tecnologie Informatiche del CNR), Via Ferrata 1, 27100 Pavia, Italy
e-mail: annalisa@imati.cnr.it

B. Wohlmuth • L. Wunderlich
M2 – Zentrum Mathematik, Technische Universität München, Boltzmannstraße 3, 85748 Garching, Germany
e-mail: wohlmuth@ma.tum.de; linus.wunderlich@ma.tum.de

B. Jüttler, B. Simeon (eds.), *Isogeometric Analysis and Applications 2014*,
Lecture Notes in Computational Science and Engineering 107,
DOI 10.1007/978-3-319-23315-4_2

different patches are required. To retain the flexibility of the meshes at the interfaces, mortar methods are very attractive.

Mortar methods are a popular tool for the coupling of non-matching meshes, originally introduced for spectral and finite element methods [4–6]. They were successfully applied in the context of isogeometric analysis [1, 13, 16]. A mathematical analysis enlightening the use of different dual spaces was recently presented in [8]. In this paper, starting from these latter results, we focus on one particular challenge in the realization of a mortar method, namely, the evaluation of the interface integrals which contain a product of functions, each of which being defined on a different mesh.

This article is structured as follows. In Sect. 2, we recall the basics of isogeometric mortar methods. In Sect. 3, we consider a review of numerical quadrature for mortar integrals such as additional aspects in the case of isogeometric analysis, illustrated by numerical results in Sect. 4.

2 Isogeometric Mortar Methods

In the following, we briefly present isogeometric mortar methods, for more details we refer to [8]. After stating the problem setting, we review isogeometric parametrizations, describe the domain decomposition into several NURBS patches and finally discuss suitable coupling spaces.

Let $\Omega \subset \mathbb{R}^d$, d the dimension being 2 or 3, be a bounded domain, $\alpha,\ \beta \in L^\infty(\Omega)$, $\alpha > \alpha_0 > 0$, and $\beta \geq 0$. We consider the following second order elliptic boundary value problem with homogeneous Dirichlet conditions

$$-\operatorname{div}(\alpha \nabla u) + \beta u = f \quad \text{in } \Omega, \tag{1a}$$

$$u = 0 \quad \text{on } \partial\Omega_D = \partial\Omega. \tag{1b}$$

We assume α and β to be piecewise sufficiently smooth.

2.1 Isogeometric Parametrization

Here, we present isogeometric concepts and notations used throughout the paper and refer to the classical literature [2, 10, 20, 23] for more details.

Let us denote by p the degree of the univariate B-splines and by Ξ an open univariate knot vector, where the first and last entries are repeated $(p+1)$-times, i.e.,

$$\Xi = \{0 = \xi_1 = \ldots = \xi_{p+1} < \xi_{p+2} \leq \ldots \leq \xi_n < \xi_{n+1} = \ldots = \xi_{n+p+1} = 1\}.$$

Let us define $Z = \{\zeta_1, \zeta_2, \ldots, \zeta_E\}$ as the knot vector without any repetition, also called breakpoint vector. For each breakpoint ζ_j of Z, we define its multiplicity m_j as its number of repetitions in Ξ. The Cox-de Boor algorithm, see [10], defines n univariate B-splines $\hat{B}_i^p(\zeta)$, $i = 1, \ldots, n$, based on the univariate knot vector Ξ and the degree p. We denote by $S^p(\Xi) = \text{span}\{\hat{B}_i^p(\zeta),\ i = 1, \ldots, n\}$ the corresponding spline space. The smoothness of B-splines is defined by the breakpoint multiplicity. More precisely, each basis function is C^{p-m_j} at each $\zeta_j \in Z$.

To define multivariate spline spaces, we introduce the multivariate knot vector $\boldsymbol{\Xi} = (\Xi_1 \times \Xi_2 \times \ldots \times \Xi_d)$, and for simplicity of notations assume in the following that the degree is the same in all parametric directions and denote it by p. Multivariate B-splines $\hat{B}_{\mathbf{i}}^p(\boldsymbol{\zeta})$ are defined by tensor product of univariate B-splines for each multi-index $\mathbf{i} \in \mathbf{I} = \{(i_1, \ldots, i_d) : 1 \leq i_\delta \leq n_\delta\}$. We denote by $S^p(\boldsymbol{\Xi})$ the corresponding spline space in the parametric domain.

Given a set of positive weights $\omega_{\mathbf{i}}$, we define NURBS functions $\hat{N}_{\mathbf{i}}^p(\boldsymbol{\zeta})$ as rational functions of B-splines and the weight function $\hat{W} = \sum_{\mathbf{i}\in\mathbf{I}} \omega_{\mathbf{i}} \hat{B}_{\mathbf{i}}^p(\boldsymbol{\zeta})$. We set $N^p(\boldsymbol{\Xi})$ as the multivariate NURBS space in the parametric domain.

For a set of control points $\mathbf{C}_{\mathbf{i}} \in \mathbb{R}^d$, $\mathbf{i} \in \mathbf{I}$, we define a parametrization of a NURBS surface ($d = 2$) or solid ($d = 3$) as a linear combination of NURBS and control points

$$\mathbf{F}(\boldsymbol{\zeta}) = \sum_{\mathbf{i}\in\mathbf{I}} \mathbf{C}_{\mathbf{i}} \hat{N}_{\mathbf{i}}^p(\boldsymbol{\zeta}),$$

and assume the regularity stated in [3, Assumption 3.1].

The knot vector $\boldsymbol{\Xi}$ forms a mesh in the parametric space $\hat{\Omega}$. We define the physical mesh $\mathcal{M}$ as the image of this parametric mesh through $\mathbf{F}$, and denote by $\mathbf{O}$ its elements. The h-refinement procedure, see [3, Section 2.1.3], yields a family of meshes denoted $\mathcal{M}_h$, each mesh being a refinement of the initial one. We assume quasi-uniformity for each mesh.

2.2 Description of the Computational Domain

Let the domain Ω be decomposed into K non-overlapping domains Ω_k, i.e.,

$$\overline{\Omega} = \bigcup_{k=1}^{K} \overline{\Omega}_k, \text{ and } \Omega_i \cap \Omega_j = \emptyset, i \neq j.$$

Each subdomain is a NURBS geometry, i.e., there exists a NURBS parametrization $\mathbf{F}_k$ based on a knot vector $\boldsymbol{\Xi}_k$ and a degree p_k, see Sect. 2.1, such that Ω_k is the image of the parametric space $\hat{\Omega} = (0, 1)^d$ by $\mathbf{F}_k$.

For $1 \leq k_1 < k_2 \leq K$, we define the interface as the interior of the intersection of the boundaries, i.e., $\overline{\gamma}_{k_1k_2} = \partial\Omega_{k_1} \cap \partial\Omega_{k_2}$, where $\gamma_{k_1k_2}$ is open. Let the non-empty

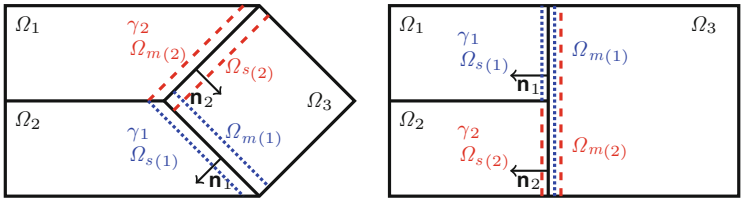


Fig. 1 Geometrical conforming case (*left*) and slave conforming case (*right*)

interfaces be enumerated by γ_l, $l = 1, \ldots, L$, and let us define the skeleton $\Gamma = \bigcup_{l=1}^{L} \gamma_l$ as the union of all interfaces. For each interface, one of the adjacent subdomains is chosen as the master side and one as the slave side. We denote the index of the former by $m(l)$, the index of the latter one by $s(l)$, and thus $\overline{\gamma}_l = \partial\Omega_{m(l)} \cap \partial\Omega_{s(l)}$. On the interface γ_l, we define the outward normal $\mathbf{n}_l$ of the master side $\partial\Omega_{m(l)}$.

We assume that for each interface the pull-back with respect to the slave domain is a whole face of the unit d-cube in the parametric space, which we call a *slave conforming* situation, see the right setting in Fig. 1. If we also assume that the pull-back with respect to the master domain is a whole face of the unit d-cube, we are in a fully geometrically conforming situation, see the left picture of Fig. 1. We note that in this latter case, the master-slave choice is arbitrary.

For each Ω_k, we introduce the space $H^1_*(\Omega_k) = \{v_k \in H^1(\Omega_k), v_{k|_{\partial\Omega \cap \partial\Omega_k}} = 0\}$, where we use standard Sobolev spaces, as defined in [15], endowed with their usual norms. In order to set a global functional framework on Ω, we consider the broken Sobolev space $V = \prod_{k=1}^{K} H^1_*(\Omega_k)$, endowed with the broken norm $\|v\|_V^2 = \sum_{k=1}^{K} \|v\|^2_{H^1(\Omega_k)}$. For any interface $\gamma_l \subset \partial\Omega_{s(l)}$, we define by $H^{-1/2}(\gamma_l)$ the dual space of $H^{1/2}_{00}(\gamma_l)$, which is the space of all functions that can be trivially extended on $\partial\Omega_{s(l)} \setminus \gamma_l$ by zero to an element of $H^{1/2}(\partial\Omega_{s(l)})$.

The mortar method is based on a weak coupling between different subdomains. Each subdomain is discretized independently and a weak coupling is performed on each interface.

2.3 Isogeometric Mortar Discretization

In the following, we set our non-conforming approximation framework. On each subdomain Ω_k, based on the NURBS parametrization, we introduce the approximation space $V_{k,h} = \{v_k = \hat{v}_k \circ \mathbf{F}_k^{-1}, \hat{v}_k \in N^{p_k}(\boldsymbol{\Xi}_k)\}$. We recall that under the assumptions on the mesh $\mathcal{M}_{k,h}$ and on the parametrization $\mathbf{F}_k$, this NURBS space has optimal approximation properties, see, e.g., [2]. On Ω, we define the discrete product space $V_h = \prod_{k=1}^{K} V_{k,h} \subset V$, which forms a $H^1(\Omega)$ non-conforming space

discontinuous over the interfaces. We denote in the following the maximal mesh size $h = \max_k h_k$ as the mesh parameter.

On the skeleton Γ, we define the discrete Lagrange multiplier product space M_h as $M_h = \prod_{l=1}^{L} M_{l,h}$, where $M_{l,h}$ has to be selected carefully.

From now on, we assume that jumps of α and β are solely located at the skeleton, and we define the bilinear and linear forms $a: V \times V \to \mathbb{R}$ and $f: V \to \mathbb{R}$, such that

$$a(u,v) = \sum_{k=1}^{K} \int_{\Omega_k} \alpha \nabla u \cdot \nabla v + \beta\, u\, v \,\mathrm{d}\mathbf{x}, \quad f(v) = \sum_{k=1}^{K} \int_{\Omega_k} f v \,\mathrm{d}\mathbf{x}$$

as well as

$$b(\lambda, v) = \sum_{l=1}^{L} \int_{\gamma_l} \lambda [v]_l \,\mathrm{d}\sigma,$$

where $[\cdot]_l$ denotes the jump from the master to the slave side over γ_l. The saddle point formulation of the isogeometric mortar method reads as follows: Find $(u_h, \lambda_h) \in V_h \times M_h$, such that

$$a(u_h, v_h) + b(\lambda_h, v_h) = f(v_h), \quad v_h \in V_h, \tag{2a}$$

$$b(\mu_h, u_h) = 0, \quad \mu_h \in M_h. \tag{2b}$$

We note that the Lagrange multiplier λ_h gives an approximation of the normal flux across the skeleton.

It is well known from the theory of mixed and mortar methods that two abstract requirements on each interface guarantee the method to be well-posed and of optimal order, see [4]. Namely, an appropriate approximation order of the dual space and a uniform inf-sup stability between the primal space and the dual space.

The first choice $M_{l,h}^0$ is a spline space of degree $p_{s(l)}$, defined on the interface γ_l based on the interface knot vector of the slave body $\Omega_{s(l)}$. Note that in the presence of any cross point, a suitable modification, e.g., a local degree reduction as presented in [8, Section 4.3], has to be applied. The alternative choice $M_{l,h}^2$ is an order $(p_{s(l)}-2)$ spline space defined on the interface γ_l based on the interface knot vector of the slave body $\Omega_{s(l)}$ for which the definition requires the trace space of $V_{s(l),h}$ to be a subset of $C^1(\gamma_l)$. We point out that although this combination is uniformly inf-sup stable with respect to the mesh-size, it results in an order reduction of $1/2$. The alternative candidate, a spline space of degree $(p_{s(l)} - 1)$ is not uniformly inf-sup stable, so we do not consider it any further.

Note that for simplicity of notations, we assume the same type of dual space to be used for all interfaces. The following theorem is shown in [8] and guarantees a-priori bounds.

Theorem 1 *Let $\theta = 0$ if $M_{l,h} = M_{l,h}^0$ and $\theta = 1/2$ if $M_{l,h} = M_{l,h}^2$. For $u \in H^{\sigma+1}(\Omega)$, $1/2 < \sigma \leq \min_k(p_k - \theta)$, the weak solution of (1) and (u_h, λ_h) the*

non-conforming approximation, see (2), it holds

$$\frac{1}{h^2}\|u-u_h\|^2_{L^2(\Omega)} + \|u-u_h\|^2_V \leq C\sum_{k=1}^{K} h_k^{2\sigma}\|u\|^2_{H^{\sigma+1}(\Omega_k)}$$

for the primal solution u and

$$\sum_{l=1}^{L}\| \alpha\nabla u\cdot\mathbf{n}_l|_{\Omega_{s(l)}} - \lambda_h\|^2_{H^{-1/2}(\gamma_l)} \leq C\sum_{k=1}^{K} h_k^{2\sigma}\|u\|^2_{H^{\sigma+1}(\Omega_k)}$$

for the dual solution $\lambda = \alpha\nabla u\cdot\mathbf{n}_l|_{\Omega_{s(l)}}$. *With* $0 < C < \infty$ *a generic constant that is independent of the mesh sizes but possibly depends on* p_k.

We highlight that while for both pairings the inf-sup stability is satisfied, the approximation order of the lower order dual space $M^2_{l,h}$ is sub-optimal. Indeed, in this case an order of one half is lost in the convergence order.

3 Mortar Integrals

To evaluate the bilinear form $b(\lambda, v)$ (and analogously $b(\mu, u)$), we need to evaluate for each interface γ_l the mortar integrals $\int_{\gamma_l}\lambda\ v^+\mathrm{d}\sigma$ and $\int_{\gamma_l}\lambda\ v^-\mathrm{d}\sigma$, where v^+ denotes the trace of v from the master domain $\Omega_{m(l)}$, and v^- the trace of v from the slave domain $\Omega_{s(l)}$. The Lagrange multiplier λ is based on the mesh of the slave domain $\Omega_{s(l)}$. To simplify the notation, let us restrict ourselves to the case of one single interface and drop the index l in the following.

One particular challenge in the realization of a mortar method is the evaluation of the first interface integral, referred as the master-slave mortar integral, due to the product $\lambda\ v^+$ of functions each of which being defined on a different mesh, see [7] for a method to bypass it in a finite element/wavelet context. Any quadrature rule based on the slave mesh does not respect the mesh lines of the master mesh and vice versa for a quadrature based on the master mesh.

It is obvious that the use of a suitable quadrature rule based on a merged mesh, i.e., a mesh which respects the reduced smoothness of the master and slave functions at their respective mesh lines, leads to an exact evaluation of the integral. However, the construction of this auxiliary mesh commonly named segmentation process is challenging, especially in the three dimensional case since the shape of the elements varies and is difficult to determine, see, e.g., [12, 16, 19, 21, 22]. Note that in an isogeometric context the merged mesh needs to be constructed in the physical space and then pulled back to the parametric space for each subdomain. The complexity of constructing such a mesh becomes even more severe in the case of non-linear and time-dependent problems, where the relative position of the meshes changes in every time or load step which implies to recompute the merged mesh at every step.

Due to this computational complexity, it has been seen very appealing to use a higher order quadrature rule either based on the slave mesh or on the master mesh to approximate the master-slave mortar integral, see [11, 14, 24] for some applications in the finite element and the isogeometric analysis context. However in the finite element case, early results in [9, 18] showed that this strategy does not necessarily yield optimal methods. More precisely, in the case that only the master mesh is chosen, the best approximation error is affected, while for the case only the slave mesh is chosen it is the consistency error. Numerical results confirmed the lack of optimality with the master integration approach, while with the slave integration approach reasonable results were obtained although not optimal in terms of the Lagrange multiplier norm.

Due to the global smoothness of splines, one could expect the sensitivity with respect to the quadrature rules for isogeometric methods to be less than for finite element methods. In the mortar context, according to the finite element results, it seems interesting to consider a slave integration rule. And, in case of maximal regularity, i.e., $V_{k,h} \subset C^{p_k-1}(\Omega_k)$ one also might expect the quadrature error on non-matching meshes to be significantly smaller than in the finite element case. These preliminary observations motivate us to study the different cases numerically.

Let us denote the quadrature rule based on the boundary mesh of the slave domain as $\sum_-$, i.e., $\int_\gamma \lambda\ v^+ \mathrm{d}\sigma \approx \sum_- \lambda\ v^+$. We precise that in the examples a Gaussian quadrature rule is used, and we vary the number of Gauss nodes. In all cases, we choose sufficiently many nodes, such that the integration on a merged mesh would have been exact. The mortar method with pure slave integration is obtained by evaluating all interface integrals in (2) using this quadrature rule, i.e., the discrete system reads as follows: Find $(\tilde{u}_h, \tilde{\lambda}_h) \in V_h \times M_h$, such that

$$\begin{aligned} a(\tilde{u}_h, v_h) + \sum\nolimits_-(v_h^+ - v_h^-)\ \tilde{\lambda}_h &= f(v_h), \quad v_h \in V_h, \\ \sum\nolimits_-(\tilde{u}_h^+ - \tilde{u}_h^-)\ \mu_h &= 0, \quad \mu_h \in M_h. \end{aligned}$$

The notation $\tilde{\cdot}$ is used to stress the difference to the discrete solution with exact integration.

In the next section, we present numerical examples which show severe deviations even in the isogeometric case. Hence, even though the global smoothness of the integrated function is increased compared to the finite element case, an integration approach based only on one mesh reduces the convergence order drastically.

Moreover, we consider an alternative approach which was proposed in [9, 18] using both integration rules. Additionally denoting $\sum_+$ a quadrature rule based on the boundary mesh of the master domain Ω_m, this approach, resulting in a non-symmetric saddle point problem, reads as follows: Find $(\tilde{u}_h, \tilde{\lambda}_h) \in V_h \times M_h$, such that

$$\begin{aligned} a(\tilde{u}_h, v_h) + \sum\nolimits_+ v_h^+\ \tilde{\lambda}_h - \sum\nolimits_- v_h^-\ \tilde{\lambda}_h &= f(v_h), \quad v_h \in V_h, \\ \sum\nolimits_-(\tilde{u}_h^+ - \tilde{u}_h^-)\ \mu_h &= 0, \quad \mu_h \in M_h. \end{aligned}$$

The non-symmetric saddle point problem, which corresponds to a Petrov–Galerkin approach in the primal formulation, was motivated by different requirements for the integration of the primal and dual test functions. Numerical examples showed error values very close to the case of exact integration, but we note that from the theoretical side even the well-posedness of the non-symmetric saddle point problem remains open. In the next section, we present numerical examples which show that also in an isogeometric context, the results are generally close to those from the exact integration case.

4 Numerical Results

In this section, we consider two-dimensional and three-dimensional settings in order to observe the effects of inexact quadrature rules, as presented in Sect. 3, on the optimality of the mortar method. We first set the problem settings and then give the results of several studies.

4.1 Two-Dimensional Example

As a first example, let us consider the Poisson problem $-\Delta u = f$ solved on the domain $\Omega = (0, 1) \times (-1, 1)$ which is decomposed into two patches by the interface $\gamma = \{(x, y) \in \Omega,\ y = 0\}$. The upper domain is set as the slave domain. The internal load and the boundary conditions are manufactured to have the analytical solution

$$u(x, y) = \cos(\pi x)\left(\cos\left(\frac{\pi}{2}y\right) + \sin(2\pi y)\right).$$

The normal derivative on the interface is given by $\partial u/\partial \mathbf{n}(x) = 2\pi \cos(\pi x)$, see Fig. 2. Neumann conditions are applied on the left and right boundary parts, such that no cross point modification is necessary.

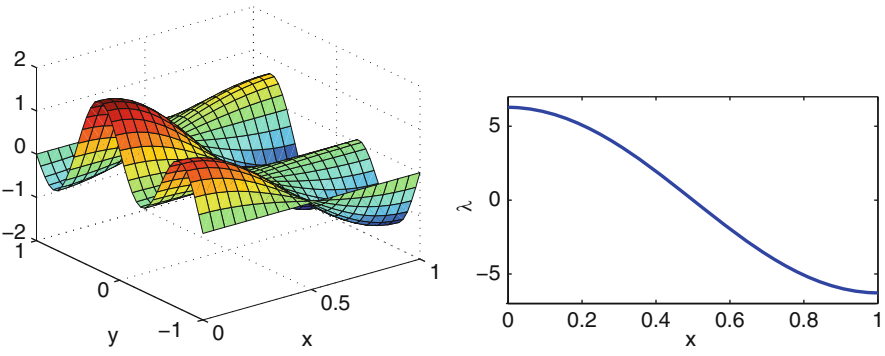


Fig. 2 *Left*: Primal solution on Ω. *Right*: Lagrange multiplier along the interface

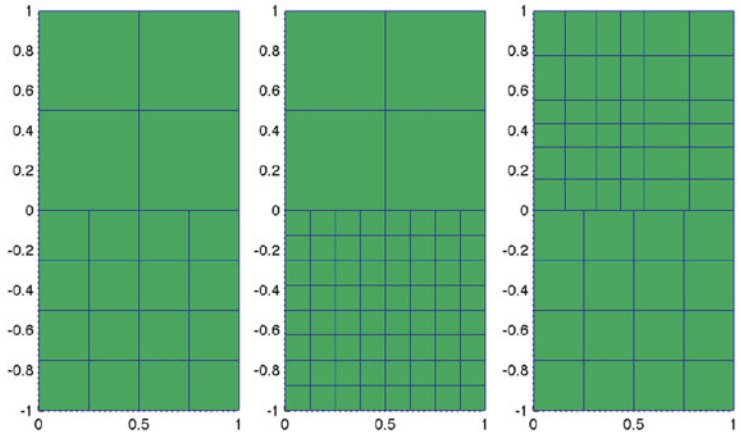


Fig. 3 Different meshes at mesh refinement level 1. From the left to the right: $M1$ to $M3$

Regarding the meshes, we consider three different cases, presented in Fig. 3. In the first two cases, the initial master mesh is a refinement of the initial slave mesh. The initial slave mesh consists of just one element. In the case $M1$, one uniform refinement step is applied to build the master mesh, in the case $M2$ two uniform refinement steps. Case $M3$ was chosen such that at no refinement level parts of the slave and master boundary meshes do coincide. The initial interior knots of the slave domain were chosen as $\{\pi/10, 1-\pi/7\}$ in both parametric directions, yielding nine elements. The initial master mesh consists of four uniform elements.

The cases $M1$ and $M2$ serve as a simple test to investigate the influence of the quadrature error. We note that inverting the role of the master and slave domains is not interesting in these cases as both integration approaches are able to exactly evaluate the master-slave mortar integral.

In the following, we provide different numerical error studies. Starting from the initial mesh, denoted by refinement level 0, we perform uniform refinements for the slave and the master domains. We note that the inter-element smoothness of the dual functions can influence the accuracy of the quadrature based on the master mesh, but not the one based on the slave mesh. Therefore for the slave integration approach, the equal order pairing with maximal smoothness is considered, i.e., $M_h = M_h^0 \subset C^{p-1}(\gamma)$, while for the non-symmetric approach we vary the dual degree. In all cases, the primal $L^2(\Omega)$ and the dual $L^2(\gamma)$ errors are computed by a comparison with the analytical solution stated above.

4.1.1 Slave Integration Approach

Firstly, we consider the case $M3$, see Fig. 3, to measure the impact of the integration error in a general situation. A numerical error study is provided in Fig. 4 for a

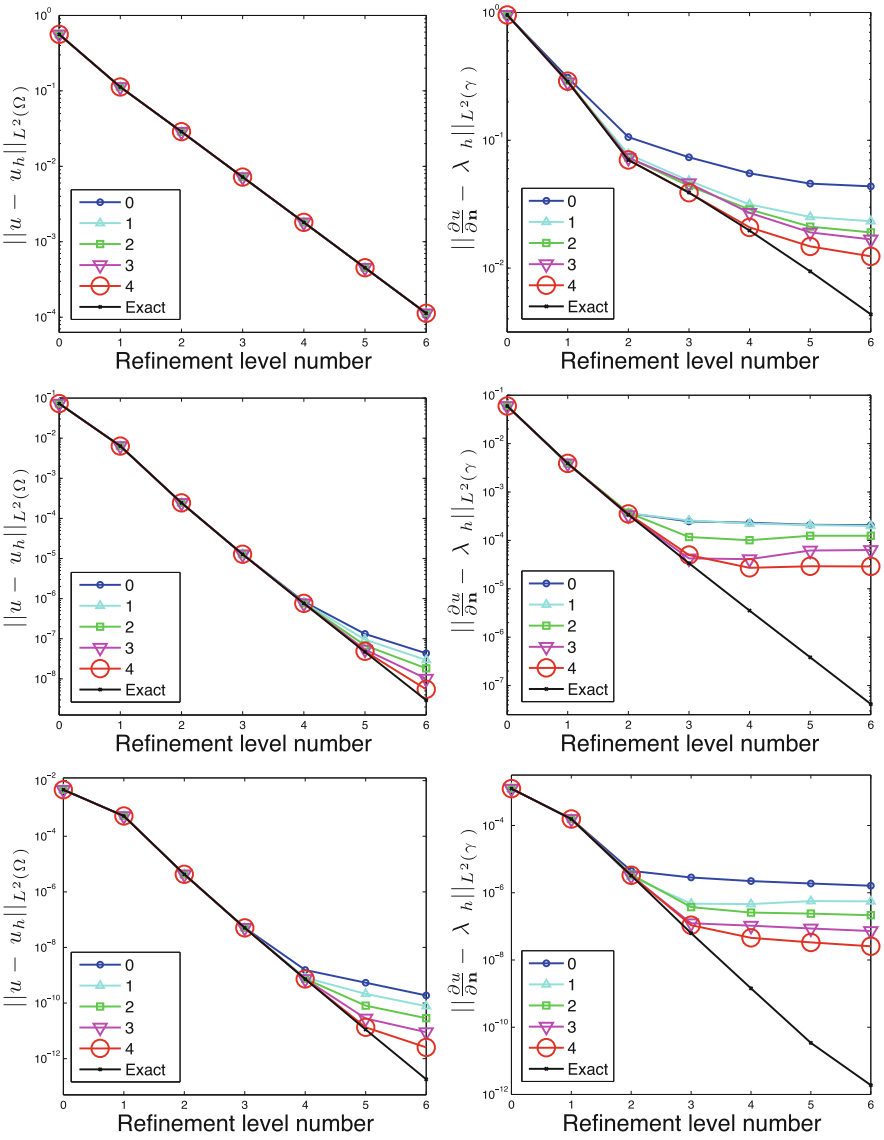


Fig. 4 2D results – L^2 primal (*left*) and dual (*right*) error curves for the case $M3$: equal order pairings with $p = 1, 3, 5$ (from top to bottom) for the slave integration approach and a different number of additional quadrature points

different number of additional Gauss points and different spline degrees. For a spline of degree p, we start with $p + 1$ Gauss points and investigate the effect of using a higher number of Gauss nodes. It can clearly be seen that the primal and dual solutions are both affected by the inexact quadrature, leading to non-optimal

Table 1 2D results – Last estimated order of convergence of the primal and dual L^2 errors for the cases $M1$ and $M2$: pairing $P5 - P5$ for the slave integration approach and a different number of additional quadrature points

Add.	Primal error		Dual error	
q.p.	Case $M1$	Case $M2$	Case $M1$	Case $M2$
0	1.63	1.74	0.50	0.50
1	1.63	1.54	0.50	0.50
2	1.63	1.55	0.50	0.50
3	1.63	1.58	0.50	0.50
4	1.63	1.56	0.50	0.50
5	1.63	1.50	0.50	0.50

methods. In all cases, the same characteristic behavior can be seen. Up to a certain refinement level, the results with inexact quadrature rules coincide with the ones with no quadrature error. Then, at a certain refinement level, the convergence order is reduced and the error is significantly larger than the exact integration one. The level, where this effect starts depends on the considered error norm, the order p and the number of additional Gauss nodes. Moreover, in this situation the higher order splines are more sensitive to the numerical quadrature approximation than the lower order splines.

In almost all cases of Fig. 4, we observe poor approximation results and a reduced convergence order which is numerically independent of the spline degree. Especially, the rate of the $L^2(\gamma)$ dual error is very low and in some cases no convergence can be observed anymore.

Secondly, we consider a simpler situation to show that even then the impact of the slave integration is noticeable. Let us focus on the cases $M1$ and $M2$, see Fig. 3, for which the master mesh is a refinement of the slave mesh. See Fig. 5 for a comparison of results between the cases $M1$ and $M2$ for a spline degree $p = 3$. We note that the low convergence orders of the primal and dual solutions, as remarked above, already appear in this simple context. Moreover, for a fixed number of slave elements, the error is increasing with the number of master elements. This is expected as there are more points of reduced smoothness which are not taken into account by the quadrature rule. Let us now consider the final numerical convergence rate in more details. In Table 1, estimated convergence orders for degree $p = 5$ are given. We notice that the dual $L^2(\gamma)$ rate breaks down to an order of $1/2$, while the $L^2(\Omega)$ primal rate lies about $3/2$.

Thirdly, we have additionally compared the case $M3$ with a similar situation in which the master and slave roles are reverted. The results also show that the integration error is increasing with the increase of the master element number. Thus, in accordance to the practical applications, in a slave integration context it seems worthwhile to choose the slave domain as the finest one.

Moreover, it can be observed that on coarse meshes using the slave integration method it is possible to recover the accuracy of the optimal mortar method simply by increasing the number of quadrature points, see Fig. 6. However, it has also been shown that the number of necessary quadrature points is drastically increasing with the refinement level. It can easily be seen that the number of Gauss points gets soon

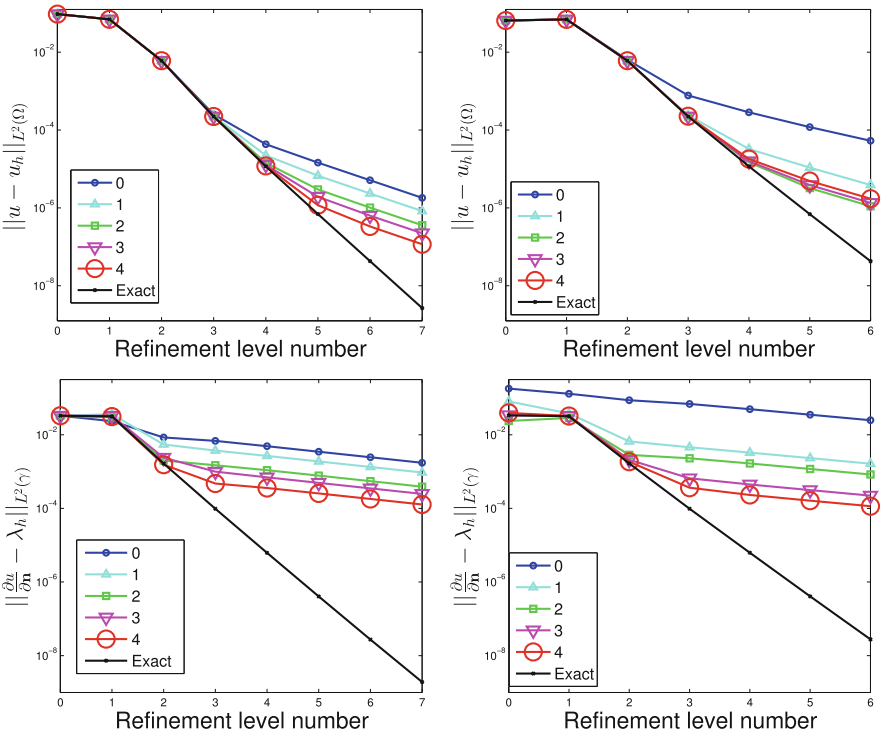


Fig. 5 2D results – L^2 primal (*top*) and dual (*bottom*) error curves for the cases $M1$ (*left*) and $M2$ (*right*): pairing $P3 - P3$ for the slave integration approach and a different number of additional quadrature points

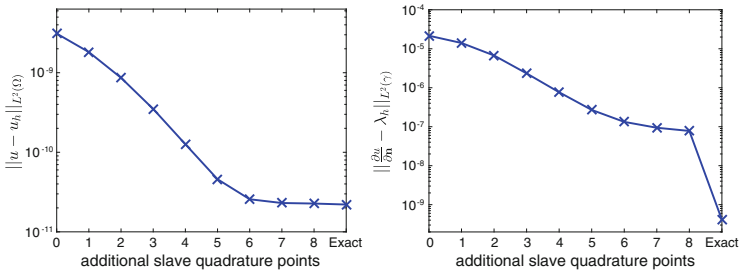


Fig. 6 2D results – L^2 primal (*left*) and dual (*right*) errors at refinement level number 6 as a function of the number of additional quadrature points for the case $M3$: pairing $P3 - P3$ for the slave integration approach

impracticably large, see the right picture of Fig. 6. Furthermore, in several cases, the deviation to the mortar method has been observed to be more severe for higher order functions.

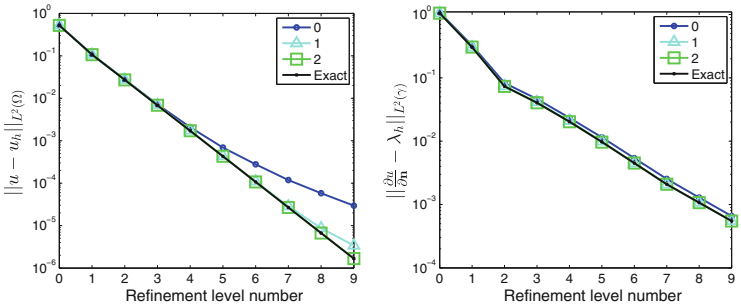


Fig. 7 2D results – L^2 primal (*left*) and dual (*right*) error curves for the case $M3$: equal order pairing $p = 1$ for the non-symmetric approach and a different number of additional quadrature points

4.1.2 Non-symmetric Approach

The non-symmetric saddle point problem based on the two different quadrature rules, see Sect. 3, was introduced to overcome the non-optimality of the pure slave integration approach in a finite element context. Due to the suboptimal results seen in the previous section, it is also interesting to consider it in an isogeometric context.

Firstly, we consider same degree pairings. In almost all tested cases, the results of the non-symmetric approach are comparable to the results of the exact integration case. However, we note that differences could still be seen in some cases. For example, for a degree $p = 1$ in the case $M3$, we obtained a non-optimal method, see in Fig. 7 the corresponding primal and dual error curves. Note that we do not show any curves in the cases where no disturbance is observed. For example for degree $p = 5$, we observed convergence almost up to machine precision without any remarkable difference compared to the exact integration case.

Secondly, we consider dual spaces with lower degrees than the primal ones. Note that in [8] stability for these pairings was only observed if the primal and the dual degrees have the same parity. Similar to the previously considered equal order case, the dual error did not show a significant deviation by the non-symmetric approach. In Fig. 8, primal error curves are shown for all stable different degree pairings up to a primal degree $p = 4$. We note that theoretically, we expect sub-optimal primal error rates even in the exact integration case, although often improved convergence rates were observed. For a dual degree $p - 2k,\ k \in \mathbb{N}$, we can expect a convergence of order $\mathcal{O}(h^{p-2k+5/2})$ in the $L^2(\Omega)$ norm, see the dashed lines in Fig. 8. For the $P4 - P2$ and $P3 - P1$ pairings, we observe small differences compared to the exact integration results, but note that the convergence rate is not significantly different than the theoretical expectation. The situation is different for the $P4 - P0$ and $P2 - P0$ pairings, for which the rate is more disturbed and even below the theoretical expectation. This can be explained by the discontinuity of the dual basis functions which introduces large errors in the integration approximation done with a rule based on the master mesh, which does not respect these discontinuities.

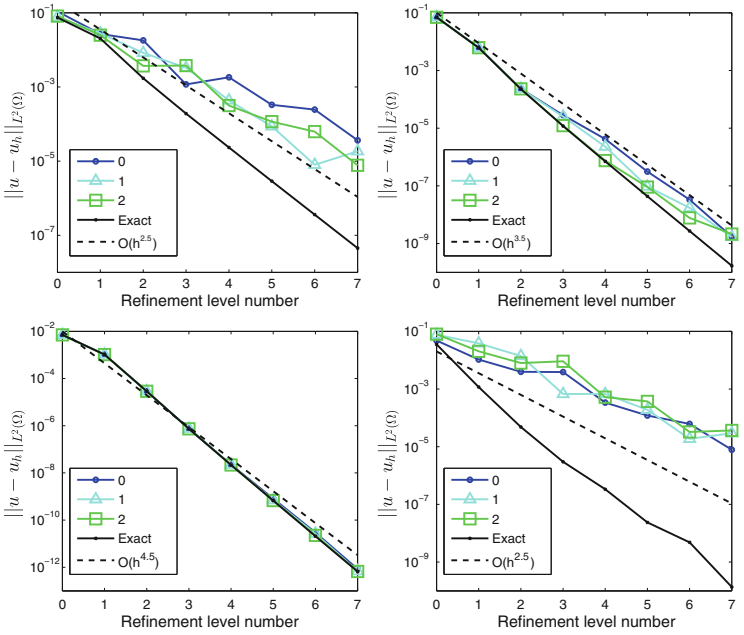


Fig. 8 2D results – L^2 primal error curves for the case $M3$: different order pairings for the non-symmetric approach and a different number of additional quadrature points. *Top left*: $P2-P0$. *Top right*: $P3-P1$. *Bottom left*: $P4-P2$. *Bottom right*: $P4-P0$

4.2 Three-Dimensional Example

As a second example, we consider a three-dimensional problem with a curved interface. Precisely, we consider the Poisson problem $-\Delta u = f$ on the domain $\Omega = (0,1)^3$, which is divided into two patches by the interface $\gamma = \{(x, y, \rho(x)), (x, y) \in (0,1)^2\}$, with $\rho(x, y) = 1/8\,(1 + x)(1 + y^2) + 1/5$, see Fig. 9. The bottom domain is set as the slave domain. The internal load and the boundary conditions are manufactured to have the analytical solution

$$u(x, y, z) = \cos(2\pi x)\cos(2\pi y)\sin(2\pi z).$$

Note that due to the curved interface, the normal derivative has a complex form, but is still explicitly computable. Neumann conditions are applied such that no cross point modification is necessary. The initial master mesh has 8 uniform elements, while the initial slave mesh has 8 elements given by the breakpoint vector $\{0, \pi/5, 1\}$ in each direction. In the following, we provide some numerical error studies, considering the slave integration approach as well as the non-symmetric approach.

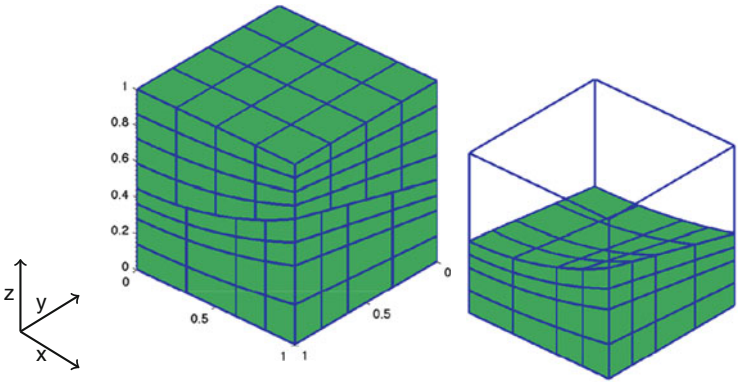


Fig. 9 Meshes at refinement level 1 (*left*) and the slave domain (*right*) illustrating the curved interface

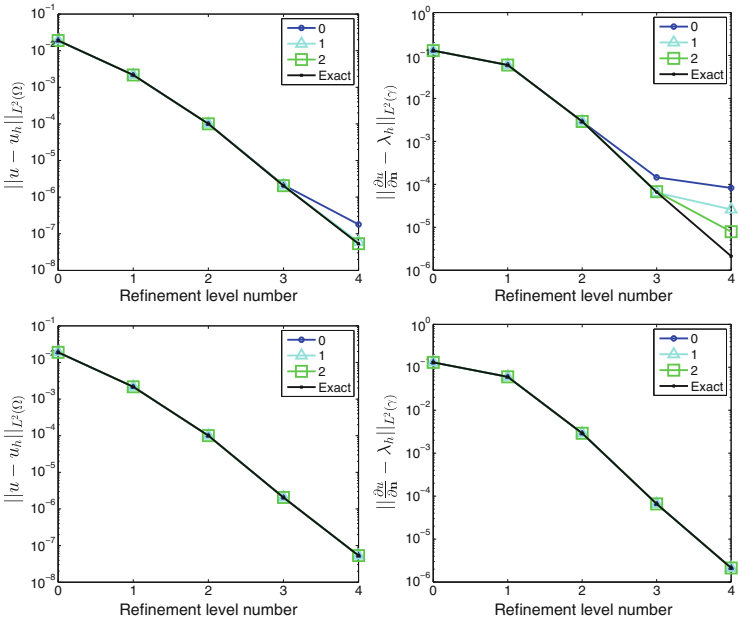


Fig. 10 3D results – L^2 primal (*left*) and dual (*right*) error curves for the pairing $P4 - P4$, for the slave integration approach (*top*) and the non-symmetric approach (*bottom*). Each of the curves being obtained with a different number of additional quadrature points

The obtained results are in accordance with the two-dimensional case results for both approaches. In Fig. 10, the deviation for the slave integration approach is shown for the $P4 - P4$ pairing. Although not shown here, we note that the results for the $P2 - P2$ and $P3 - P3$ pairing have a similar behavior. The non-symmetric approach does not lead to reduced rates considering equal order pairings, i.e., $M_h = M_h^0$, on

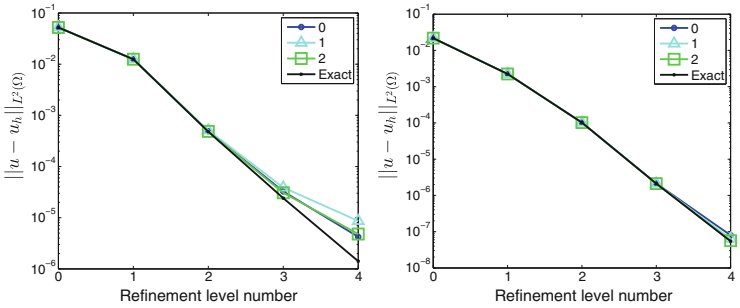


Fig. 11 3D results – L^2 primal error curves for the pairings $P3 - P1$ (*left*) and $P4 - P2$ (*right*), for the non-symmetric approach and a different number of additional quadrature points

the refinement levels we considered. As previously, using a lower order dual space, a difference to the exact integration case can be seen. See Fig. 11 for the disturbance in the primal variable of the $P3 - P1$ and $P4 - P2$ pairings.

5 Conclusion

After briefly reviewing optimal isogeometric mortar methods, a study on the possibility to approximate the mortar integrals by efficient numerical quadrature rules was performed.

The precise evaluation of mortar integrals is of a high computational complexity, since it requires the construction of a merged mesh, combining the non-matching meshes at the interface. While it would be desirable to use a quadrature rule based on the slave mesh, numerical examples show that this induces large deviations to the mortar method. Especially the convergence rate of the Lagrange multiplier is reduced to at most $1/2$, independently of the spline degree. While the method improves by increasing the number of quadrature points, the amount of points necessary to obtain nearly optimal results is not predictable.

To overcome these difficulties, we have considered a non-symmetric saddle point problem based on both master and slave integration rules, which was previously introduced in the finite element context. Numerical examples demonstrate that for most cases the reached accuracy is close to the optimal one.

Acknowledgements The first author has been supported by Michelin under the contract A10-4087. The third and the fourth authors have been supported by the International Research Training Group IGDK 1754, funded by the German Research Foundation (DFG) and the Austrian Science Fund (FWF), and by the German Research Foundation (DFG) in Project WO 671/15-1 and the Priority Programme "Reliable Simulation Techniques in Solid Mechanics. Development of Non-standard Discretisation Methods, Mechanical and Mathematical Analysis" (SPP 1748). The last author was supported by the Elite Network of Bavaria through its graduate program TopMath

and the TUM Graduate School through its Thematic Graduate Center TopMath. All supports are gratefully acknowledged.

References

1. A. Apostolatos, R. Schmidt, R. Wüchner, K.U. Bletzinger, A Nitsche-type formulation and comparison of the most common domain decomposition methods in isogeometric analysis. Int. J. Numer. Methods Eng. **97**, 473–504 (2014)
2. Y. Bazilevs, L. Beirão da Veiga, J.A. Cottrell, T.J.R. Hughes, G. Sangalli, Isogeometric analysis: approximation, stability and error estimates for h-refined meshes. Math. Models Methods Appl. Sci. **16**(7), 1031–1090 (2006)
3. L. Beirão Da Veiga, A. Buffa, G. Sangalli, R. Vásquez, Mathematical analysis of variational isogeometric methods. Acta Numer. **23**, 157–287 (2014)
4. F. Ben Belgacem, The mortar finite element method with Lagrange multipliers. Numer. Math. **84**, 173–197 (1999)
5. F. Ben Belgacem, Y. Maday, The mortar finite element method for three dimensional finite elements. Math. Model. Numer. Anal. **31**(2), 289–302 (1997)
6. C. Bernardi, Y. Maday, A.T. Patera, A new nonconforming approach to domain decomposition: the mortar element method, in *Nonlinear Partial Differential Equations and Their Applications*, vol. XI, ed. by H. Brezis, J.L. Lions (Collège de France, Paris, 1994), pp. 13–51
7. S. Bertoluzza, S. Falletta, An object-oriented implementation of the mortar method with approximate constraint. Technical report, Istituto di Matematica Applicata e Tecnologie Informatiche, Consiglio Nazionale delle Ricerche, Pavia (2004)
8. E. Brivadis, A. Buffa, B. Wohlmuth, L. Wunderlich, Isogeometric mortar methods. Comput. Methods Appl. Mech. Eng. **284**, 292–319 (2015)
9. L. Cazabeau, C. Lacour, Y. Maday, Numerical quadrature and mortar methods, in *Computational Science for the 21st Century* (Wiley, Chichester, 1997), pp. 119–128
10. J.A. Cottrell, T.J.R. Hughes, Y. Bazilevs, *Isogeometric Analysis. Towards Integration of CAD and FEA* (Wiley, Chichester, 2009)
11. L. De Lorenzis, I. Temizer, P. Wriggers, G. Zavarise, A large deformation frictional contact formulation using NURBS-based isogeometric analysis. Int. J. Numer. Methods Eng. **87**, 1278–1300 (2011)
12. M. Dittmann, M. Franke, I. Temizer, C. Hesch, Isogeometric analysis and thermomechanical mortar contact problems. Comput. Methods Appl. Mech. Eng. **274**, 192–212 (2014)
13. W. Dornisch, G. Vitucci, S. Klinkel, The weak substitution method – an application of the mortar method for patch coupling in NURBS-based isogeometric analysis. Int. J. Numer. Methods Eng. (2015). doi:10.1002/nme.4918
14. K. Fischer, P. Wriggers, Frictional 2D contact formulation for finite deformations based on the mortar method. Comput. Mech. **36**, 226–244 (2005)
15. P. Grisvard, *Elliptic Problems in Nonsmooth Domains* (SIAM, Philadelphia, 2011)
16. C. Hesch, P. Betsch, Isogeometric analysis and domain decomposition methods. Comput. Methods Appl. Mech. Eng. **213–216**, 104–112 (2012)
17. T.J.R. Hughes, J.A. Cottrell, Y. Bazilevs, Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. Comput. Methods. Appl. Mech. Eng. **194**, 4135–4195 (2005)
18. Y. Maday, F. Rapetti, B.I. Wohlmuth, The influence of quadrature formulas in 2D and 3D mortar element methods, in *Recent Developments in Domain Decomposition Methods. Some Papers of the Workshop on Domain Decomposition*, ETH Zürich, 7–8 June 2001 (Springer, 2002), pp. 203–221
19. T. McDevitt, T. Laursen, A mortar-finite element formulation for frictional contact problems. Int. J. Numer. Methods Eng. **48**, 1525–1547 (2000)

20. L. Piegl, W. Tiller, *The NURBS Book* (Springer, Berlin/New York, 1997)
21. M. Puso, A 3D mortar method for solid mechanics. Int. J. Numer. Methods Eng. **59**, 315–336 (2004)
22. M. Puso, T. Laursen, A mortar segment-to-segment contact method for large deformation solid mechanics. Comput. Methods Appl. Mech. Eng. **193**, 601–629 (2004)
23. L. Schumaker, *Spline Functions: Basic Theory*, 3rd edn. (Cambridge University Press, Cambridge, 2007)
24. M. Tur, F. Fuenmayor, P. Wriggers, A mortar-based frictional contact formulation for large deformations using Lagrange multipliers. Comput. Methods Appl. Mech. Eng. **198**, 2860–2873 (2009)

The Isogeometric Segmentation Pipeline

Michael Pauley, Dang-Manh Nguyen, David Mayer, Jaka Špeh, Oliver Weeger, and Bert Jüttler

Abstract We present a pipeline for the conversion of 3D models into a form suitable for isogeometric analysis (IGA). The input into our pipeline is a boundary represented 3D model, either as a triangulation or as a collection of trimmed non-uniform rational B-spline (NURBS) surfaces. The pipeline consists of three stages: computer aided design (CAD) model reconstruction from a triangulation (if necessary); segmentation of the boundary-represented solid into topological hexahedra; and volume parameterization. The result is a collection of volumetric NURBS patches. In this paper we discuss our methods for the three stages, and demonstrate the suitability of the result for IGA by performing stress simulations with examples of the output.

1 Introduction

In isogeometric analysis (IGA), the space of functions used to approximate the solution of a differential equation is the same space of functions used to represent the geometry. As such, IGA methods work directly with geometry represented by splines and have begun to close the gap between the geometry representations used for analysis and those used for design [1]. An additional benefit of IGA is that it can achieve the same accuracy as the finite element method (FEM) with fewer degrees of freedom (see [4]).

In order to realize the benefits of IGA when dealing with complex geometries, it is necessary to segment them into IGA-suitable pieces. A rich theory of FEM-suitable mesh generation exists (see e.g., [5, 8, 14, 15, 17, 21, 24]) but these methods

M. Pauley (✉) • D.-M. Nguyen • J. Špeh • B. Jüttler • D. Mayer
Institute of Applied Geometry, Johannes Kepler University, Altenberger Straße 69, 4040 Linz, Austria
e-mail: Michael.Pauley@jku.at; dmnguyen@cise.ufl.edu; Jaka.Speh@jku.at; Bert.Juettler@jku.at; david_mayer@aon.at

O. Weeger
Faculty of Mathematics, TU Kaiserslautern, P.O. Box 3049, 67653 Kaiserslautern, Germany
e-mail: oliver_weeger@sutd.edu.sg

B. Jüttler, B. Simeon (eds.), *Isogeometric Analysis and Applications 2014*,
Lecture Notes in Computational Science and Engineering 107,
DOI 10.1007/978-3-319-23315-4_3

aim at a segmentation of a solid into a large number of (approximately uniform) small tetrahedra and hexahedra. The goal of the *isogeometric segmentation pipeline* is to convert a boundary represented model (e.g. something designed in computer aided design (CAD) software) into an IGA-suitable volumetric model. For us this means segmenting the geometry into a small number of *topological hexahedra*, that is, deformed cubes.

Methods for parameterization of solids by a single B-spline or T-spline cube are developed in [11, 19, 25, 26]. A method for multiple patches is given in [22] which requires a predefined segmentation as input. Several papers [9, 10, 20, 23] provide methods for segmentation and parameterization of a solid by B-spline or T-spline cubes, allowing for singularities in the output.

The input into the pipeline can be either a triangulated boundary representation, or a trimmed non-uniform rational B-spline (NURBS) surface boundary representation. A trimmed surface consists of a *master spline* mapping a square into $\mathbb{R}^3$, together with a collection of *trimming curves* defining a domain of the surface as a subset of the square. The boundary representation for the pipeline consists of a collection of trimmed surfaces, together with adjacency information. The output of the pipeline is a representation of the solid as a collection of volumetric NURBS. The pipeline consists of the three following stages.

CAD model reconstruction. A triangulated model is converted into a trimmed NURBS surface boundary representation, for input into the remaining stages. This step is not necessary for models that are already represented with trimmed surfaces. It is necessary when only a triangulated solid is available, or when CAD data is available but uses features not supported by the remainder of the pipeline. In the latter case, the CAD model can be triangulated and then reconstructed to produce a solid in a format compatible with the pipeline.

Isogeometric segmentation. A trimmed NURBS surface boundary represented solid is segmented into a collection of boundary represented topological hexahedra. Our segmentation approach, which began development in [6, 12], is based on the edge graph of the solid – the graph defined by the vertices and edges between the trimmed surfaces.

In its present state, our isogeometric segmentation method requires manual interaction in two scenarios. Firstly, it is applicable to the class of contractible solids with 3-vertex-connected edge graphs. This means that a manual preprocessing step is required to ensure that the input satisfies these assumptions. In our ongoing work we are developing methods to automatically ensure these assumptions. Secondly, a post-processing step may be necessary if the segmentation is required to be free of T-joints. So far, we only have a way to eliminate them in special cases. Our belief is that T-joints will not be a barrier to IGA, as contemporary research is developing methods for handling non-matching interfaces [7].

Volume parameterization. A collection of boundary represented topological hexahedra is converted to a collection of volumetric splines.

Sections 2, 3,and 4 of the paper provide details and examples for the above stages of the pipeline. In Sect. 5, a segmentation produced by our pipeline is used to conduct mechanical simulations, showing the suitability of our results for

isogeometric analysis. Section 6 summarizes the state of the pipeline and outlines our ongoing and future work.

2 Segmentation of Triangulated Surfaces and CAD Model Reconstruction

Consider a complex model for which IGA-based simulation is required. If a collection of trimmed NURBS surfaces is provided describing its boundary, we can segment it into topological hexahedra using the methods in Sects. 3 and 4. However, often we only have access to a triangular mesh describing the boundary. This would happen, for example, if a triangulation of a model is available but the original model is not. Furthermore, if a CAD model is of low quality or uses features that are not supported by the segmentation software, it is useful to convert it into a triangle mesh from which a more suitable trimmed surface model can be created. Finally, a more complex CAD model is often obtained by combining several models produced by several users, possibly with different CAD systems. In this situation, using a triangulated CAD model is often the only feasible option.

Sophisticated methods exist for the reverse engineering of CAD data (see [18] and its references). The method we use can afford to be relatively straightforward because (i) we are primarily interested in engineering objects, which tend to have simple faces and sharp edges, compared to general, more free-form designs; (ii) we assume a high quality triangulation is provided as input, not noisy point cloud data; (iii) while more sophisticated methods can reproduce blend surfaces, we do not want them as the later stages of our pipeline are not prepared to deal with them.

- The triangulated boundary is segmented into regions using a *region growing algorithm*. Two triangles are part of the same region if there exists a path between them which passes only between triangles that meet at an angle close to π. Post-processing can be applied to further segment the large regions with a more strict angle tolerance, or to merge small regions with a less strict angle tolerance. An example is shown in Fig. 1.
- For each region, a triangulated domain is constructed whose triangles are in correspondence with the triangles of the region. The points in the interior of the

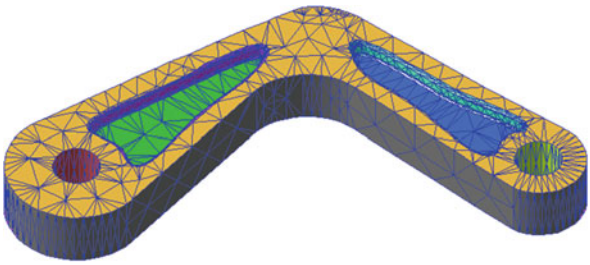


Fig. 1 The region growing algorithm segments the triangulation of the TERRIFIC demonstrator part into nine regions. Different regions are shaded different colors

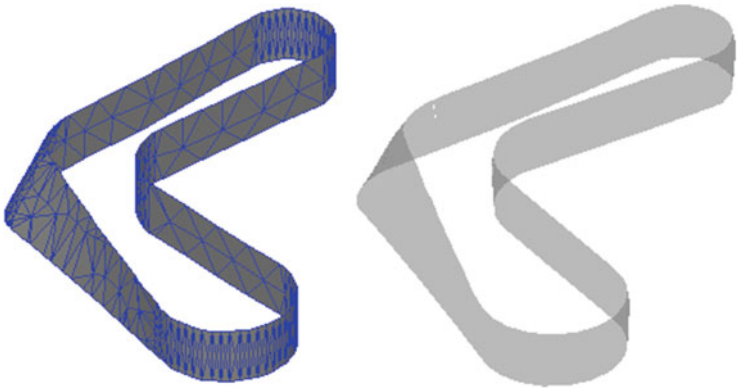


Fig. 2 Example of a NURBS trimmed surface fit to triangulated data

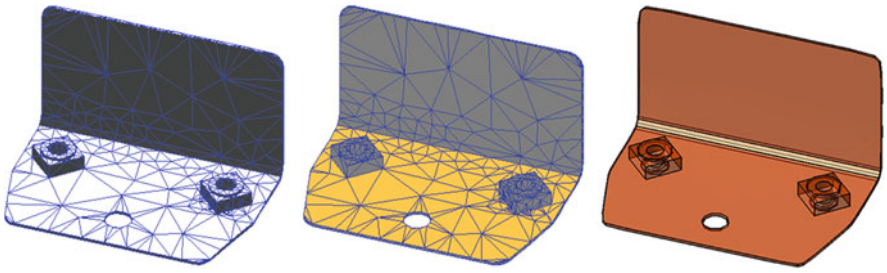


Fig. 3 (**a**) The triangulated car part. (**b**) The model is separated into regions. We highlight one here. (**c**) The CAD model reconstruction (translucent)

domain, and the positions of holes in the domain, are chosen by a method based on [2, 3].

- A tensor product NURBS master surface is constructed, by fitting the triangulation data using a least-squares approach. Finally, sharp corners in the domains are smoothed out where appropriate. An example of a surface fit to a detected region is shown in Fig. 2.

2.1 Examples

We demonstrate the procedure with the following examples.

Car part example. The procedure is applied to a car part consisting of 1,220 triangles. Angle tolerance is set to 40° and a secondary region growing phase is applied, with a tolerance of 20°, to those regions with area more than 20 % of the average. The result is shown in Fig. 3a–c.

Fig. 4 The TERRIFIC Demonstrator: reconstructed CAD model

TERRIFIC Demonstrator. The TERRIFIC demonstrator model was created as part of the TERRIFIC European Commission project for testing and demonstrating the project's research and development. The demonstrator in its original format is a boundary represented CAD model. However, feeding a triangulated version of the model through the CAD model reconstruction step allows us to clean the model up, eliminating the blends which the later steps of our pipeline do not support.

The model has 3,672 triangles. The region growing algorithm is used with a cutoff of 12° and regions are merged together if they have area less than 200 % of the average. The result is shown in Fig. 4.

3 Isogeometric Segmentation of Boundary Represented Solids

Here we describe our approach to the problem of segmenting a boundary-represented solid into topological hexahedra. The solid has sharp features, and its faces are described by NURBS-based trimmed surfaces.

The faces, edges and vertices of the solid form a structure called the *edge graph* of the solid (see Fig. 5). A graph is 3-vertex-connected if it is connected, it has at least four vertices and any two vertices can be deleted without disconnecting the graph. We make the following assumptions about the solid:

- the solid is contractible;
- the edge graph is simple and 3-vertex-connected.

Once these assumptions are satisfied, we have an automatic process to obtain a segmentation into topological hexahedra. The process is as follows.

1. Ensure that the solid satisfies the assumptions.
2. For each existing non-convex edge, segment the solid using a cutting loop (Sects. 3.1 and 3.2) or an extending loop (Sect. 3.4) containing the edge.
3. If the solid has no non-convex edges, segment it further using cutting loops until we arrive at pre-defined *base solids*.
4. Segment the base solids in a predefined way into topological hexahedra.

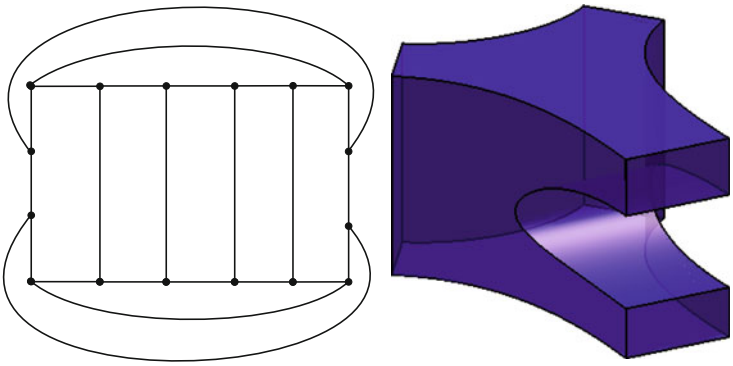


Fig. 5 An example of a contractible solid and a planar representation of its edge graph

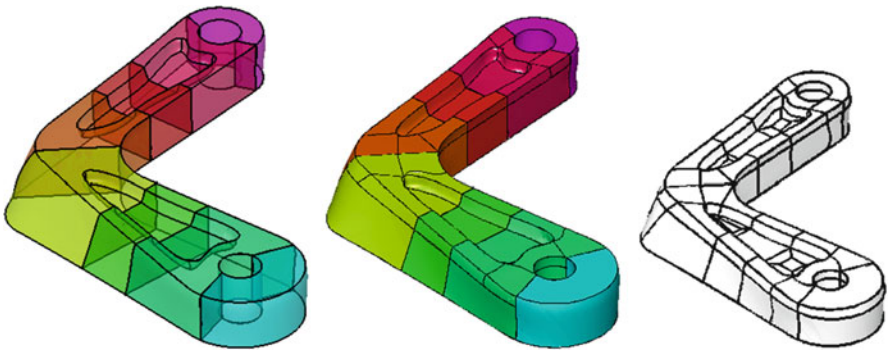


Fig. 6 (**a**) The TERRIFIC Demonstrator, preprocessed to cut it into contractible pieces with 3-vertex-connected edge graphs. (**b**) Segmentation into topological hexahedra. (**c**) The solid is further segmented to eliminate T-joints

Item 1 above is ongoing work: methods based on the concept of a Reeb graph from Morse theory, which has previously been used to good effect in *flow volume decomposition* [16], show promise for dealing with non-contractible solids. Items 2–4 are explained below in Sects. 3.1, 3.2, 3.3 and 3.4.

Figure 6 shows an example: this stage of the pipeline is applied to the TERRIFIC demonstrator. The demonstrator is first preprocessed to cut it into contractible pieces and artificial edges are added to make its graph 3-vertex-connected. Then these pieces are segmented into hexahedra using the methods of cutting loops and extending loops described in the following sections.

3.1 Theory for Contractible Solids with Only Convex Edges

An edge **e** of the solid is *convex* at a point **p** if the two incident faces at **e** meet at a convex interior angle at **p**. The edge **e** is called *convex* if it is convex at every point, and non-convex otherwise. Non-convex edges have a significant impact on the options for segmentation. In this section we restrict to solids with no non-convex edges; we extend our approach to the more general case in Sect. 3.2.

Our approach is based on the concept of a *cutting loop*; a cycle in the edge graph consisting of existing edges as well as *auxiliary edges* which can be created between any two vertices on the same face. A cutting loop is *valid* if it can be used as the boundary of a surface with the following properties:

- the surface is contained in the solid;
- the surface cuts the solid into exactly two pieces;
- both pieces satisfy our original assumptions, i.e., they are contractible and the edge graphs are simple and 3-vertex-connected.

For solids with only convex edges, validity is essentially a combinatorial condition, i.e., the structure of the graph matters but the geometry of the trimmed surfaces does not. This allows us to guarantee the existence of a valid cutting loop.

Theorem 1 ([6]) *Assume that all edges are convex. If the edge graph of the solid is not that of a tetrahedron, then a valid cutting loop exists.*

3.2 Theory for Contractible Solids with Non-convex Edges

Non-convex edges create additional complications. They must be eliminated during the segmentation process, because they prevent smooth parameterization (see Fig. 7). Furthermore, they create new validity criteria for cutting loops, which can sometimes require *auxiliary vertices* (new vertices created along the existing edges of the edge graph) to guarantee the existence (See [12, Section 3]) (Fig. 8).

Theorem 2 ([12]) *Given a non-convex edge of the solid, there exists a valid cutting loop containing that edge and possibly containing auxiliary vertices, such that cutting the solid with this loop eliminates the non-convex edge.*

This enables an extension of the algorithm outlined in Sect. 3.1. Cutting loops are used to cut the solid to remove all the non-convex edges. Once the solid has been segmented into pieces with only convex edges, the method of Sect. 3.1 can be applied.

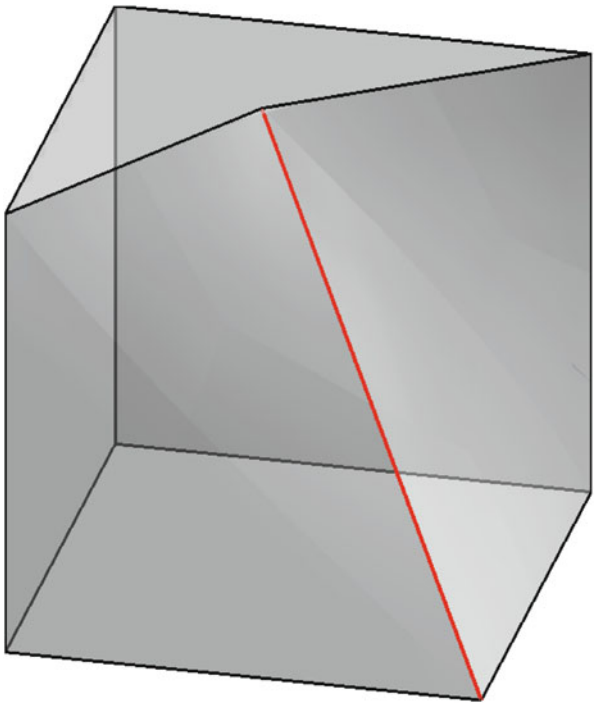


Fig. 7 A solid with a non-convex edge. The edge graph is isomorphic to that of a cube. However, the non-convex edge prevents C^1 volume parameterization by a cube. The segmentation algorithm eliminates non-convex edges as its first priority so that the resulting topological hexahedra have only convex edges

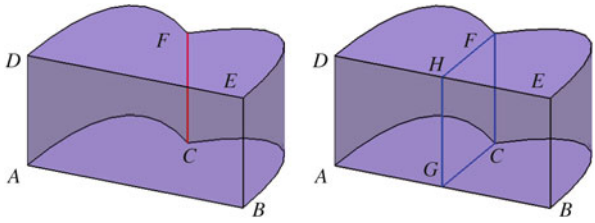


Fig. 8 *Left*: a solid with a non-convex edge CF. Without using auxiliary vertices, there is no valid cutting loop to cut CF, but it becomes possible with the addition of the vertices G and H

3.3 Description of the Algorithm

Theorems 1 and 2 mean that the following algorithm successfully segments the solid into topological hexahedra, given its edge graph structure:

- search for a valid cutting loop;
- use the cutting loop to construct a surface which cuts the solid into two pieces. The geometric aspects of this step are difficult. Firstly, any auxiliary edges of the loop must be turned into real edges by constructing a curve cutting the face. This problem is dealt with in the paper [13]. Secondly, a trimmed surface must be constructed having the cutting loop as its boundary;

- repeat the procedure for the two pieces until we arrive at *base solids*: a class of solids which have predefined segmentations into topological hexahedra. Our base solids include tetrahedra, hexahedra and prisms.

Each of the base solids is further split into topological hexahedra using pre-defined templates. More details of the algorithm are provided in [6, 12].

Typically, many valid cutting loops exist so a decision procedure is needed to pick one. We decide based on a cost which is a combination of the following components:

Combinatorial component. Aiming for a small number of hexes, our cost includes a component based on the graph theoretic structure:

- the number of edges of the cutting loop (e.g. preferring loops of length 4 since they more quickly lead to topological hexahedra).
- the number of edges of the newly created faces when a face is split by an auxiliary edge (e.g. preferring to split a 6-sided face into two 4-sided faces vs. a 5- and a 3-sided face).

Geometric component. Aiming for hexes with good shape, we include measurements of the planarity of the cutting loop.

A more detailed discussion, with examples, is provided in [12, Sections 5–6].

3.4 Surface Extension

The cutting loop approach can sometimes miss elegant segmentations that could be created by extending existing surfaces. For example, [12] provided an example where using the cutting loop method alone results in a segmentation with 31 topological hexahedra. Furthermore, they do not have ideal shape. Extending existing faces would allow a segmentation with only four hexes and higher quality shape (see Fig. 9). Surface extension can be done by modifying cutting loops into *extending loops* (see Fig. 10). In contrast to cutting loops, extending loops result in the construction of two different faces for the two new solids.

Extending loops can be defined as cutting loops with additional structure: each edge of the loop carries data of whether it extends one of the two incident faces, and if so, which one. The validity of an extending loop can be determined from the way faces meet at vertices and edges. Specifically,

- *it is only possible to extend a surface through a given edge* $\mathbf{e}$ *if* $\mathbf{e}$ *is everywhere concave*. This is because, if $\mathbf{e}$ is convex at any point, then the surface would be extended to outside the solid;
- at a vertex $\mathbf{v}$, consider the set of unit tangent vectors pointing into the solid. This set is a region of the unit sphere, whose boundary is a spherical polygon. See Fig. 11 for an example. Three faces $\mathbf{A}, \mathbf{B}, \mathbf{C}$ of a solid meet at a vertex $\mathbf{v}$; the resulting spherical polygon has three corresponding great arcs $\tilde{\mathbf{A}}, \tilde{\mathbf{B}}, \tilde{\mathbf{C}}$. *The geometry of this polygon determines the options available for extending a surface*

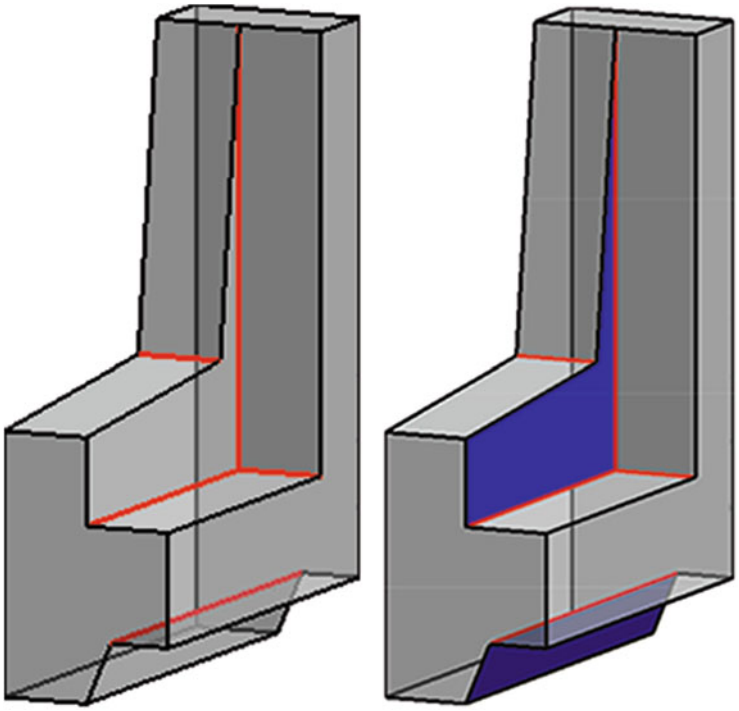


Fig. 9 Half-chair example. The solid in (**a**) was segmented into 31 topological hexahedra in [6]. However, a segmentation into just four topological hexahedra can be produced by extending the faces marked in (**b**) and merging them together as the first step. This motivates our ongoing work on automatic surface extension as a step in the segmentation procedure

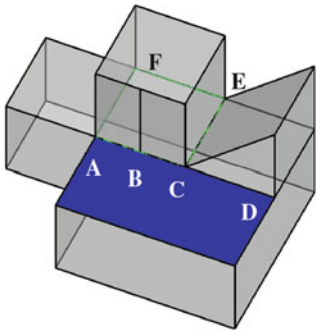


Fig. 10 Example of a solid with some surfaces that can be extended. Loop **A-B-C-E-F** is an extending loop which can be used to extend the highlighted face and cut the solid. An extending loop cannot contain the path **A-B-C-D** due to the geometry at vertex **C**

through **v**. In the example of Fig. 11, face **B** can be extended through **v**, resulting in face **A** being cut. This is because extending arc $\tilde{\mathbf{B}}$ into the interior of the

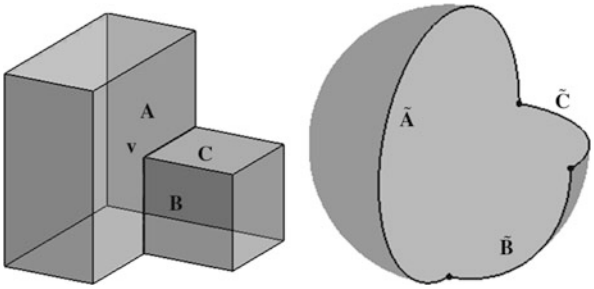


Fig. 11 In the solid (*left*), three faces $\mathbf{A}, \mathbf{B}, \mathbf{C}$ are incident to a vertex $\mathbf{v}$. The unit vectors at $\mathbf{v}$ pointing into the solid define a filled spherical polygon with arcs $\tilde{\mathbf{A}}, \tilde{\mathbf{B}}, \tilde{\mathbf{C}}$ containing the tangent vectors pointing along the faces $\mathbf{A}, \mathbf{B}, \mathbf{C}$. The geometry of the spherical polygon determines the available options for extending these faces through $\mathbf{v}$

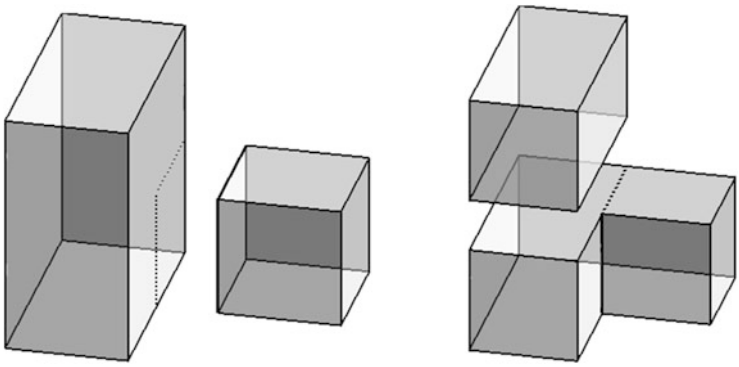


Fig. 12 Two possible results of cutting the solid of Fig. 11 (*left*) by extending surfaces

spherical polygon, it eventually intersects arc $\tilde{\mathbf{A}}$. A similar situation arises when extending face $\mathbf{C}$.

A slightly different situation occurs for extending face $\mathbf{A}$ through $\mathbf{v}$. If arc $\tilde{\mathbf{A}}$ is extended into the interior of the polygon, it does not intersect the interior of any of the other arcs, but instead closes up into a great circle. As a consequence, face $\mathbf{A}$ can be extended through $\mathbf{v}$ in such a way that the vertex $\mathbf{v}$ can be deleted from one of the two solids resulting from the cut. The question of whether two faces, meeting at a vertex, can be extended simultaneously into a single face, can also be answered using the geometry of the spherical polygon.

Compare these facts with [12, Proposition 2], where it is shown that the *spherical visibility problem* for the spherical polygon constructed above plays an important role in determining the validity of a cutting loop.

Two possible results of cutting the solid of Fig. 11 (left) are shown in Fig. 12.

The selection of an extending loop is similar to one for a cutting loop. A cost is assigned based on combinatorial and geometric properties. Minor complications arise from the fact that vertices can be deleted, and the new faces in the two resulting solids can have a different number of vertices.

There remains the question of the geometric realization of the extended surface, i.e., how do we construct the NURBS-based trimmed surface structure for a face which includes the existing face(s) and the newly constructed part? This question is still open; our current attitude is that it is better to only construct the new part, represent the extended face as a union of trimmed surfaces, and merge them together only at the time of volume fitting. This avoids creating an extra source of approximation.

Examples

We demonstrate the segmentation process for several models. For these models, a preprocessing step is applied to ensure that the model is contractible and the edge graph is simple and 3-vertex connected. This pre-processing step is ad-hoc as we have not developed a general method.

Chair stand piece. The chair stand piece (Fig. 13) was created to test various preprocessing strategies and demonstrate the algorithm for solids with convex edges. The result for one preprocessing strategy is shown in Fig. 14. A second

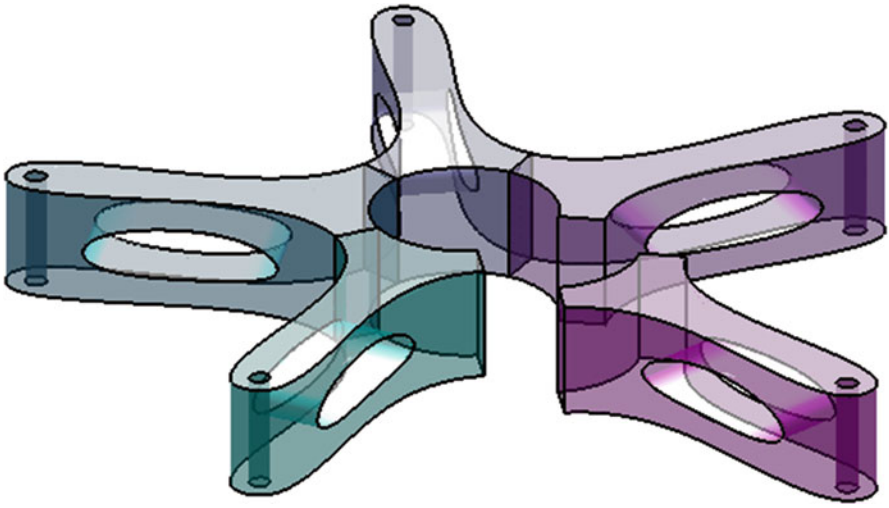


Fig. 13 The chair stand example. This solid has a 5-way symmetry, so one-fifth of it is extracted for isogeometric segmentation (see Fig. 14)

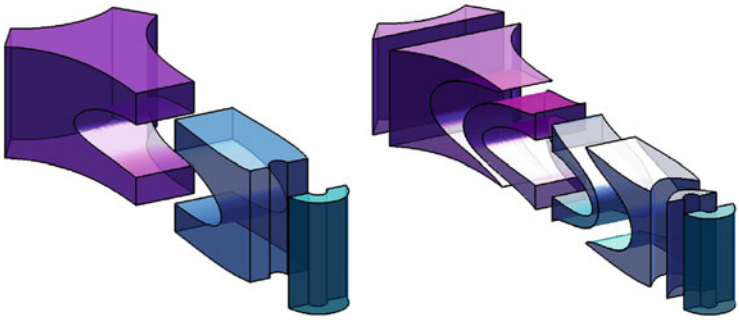


Fig. 14 *Left*: the chair stand piece, preprocessed to cut it into contractible pieces. *Right*: the result of the segmentation algorithm

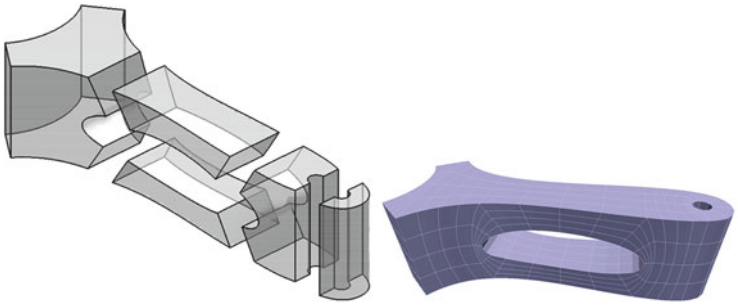


Fig. 15 (**a**) The chair stand piece, preprocessed to cut it into contractible pieces, using a different strategy from Fig. 14. (**b**) Representation of the chair stand as a watertight collection of tensor-product NURBS volumes

preprocessing strategy is shown in Fig. 15 where the resulting topological hexahedra are volume parameterized using the techniques in Sect. 4.

TERRIFIC Demonstrator The preprocessed TERRIFIC Demonstrator is shown in Fig. 6a. The solid is segmented into topological hexahedra using the surface extension and cutting loop methods described in the previous sections. The result is shown in Fig. 6b. The result of the removal of T-joints is shown in Fig. 6c.

Car part The segmentation procedure is applied to the CAD model reconstruction of the car part (See Sect. 2 and Fig. 3a–c). The preprocessed solid (Fig. 16, left) has a plane of symmetry. For one half, the resulting contractible pieces are segmented into topological hexahedra (Fig. 16, right). Data courtesy of Engineering Center Steyr.

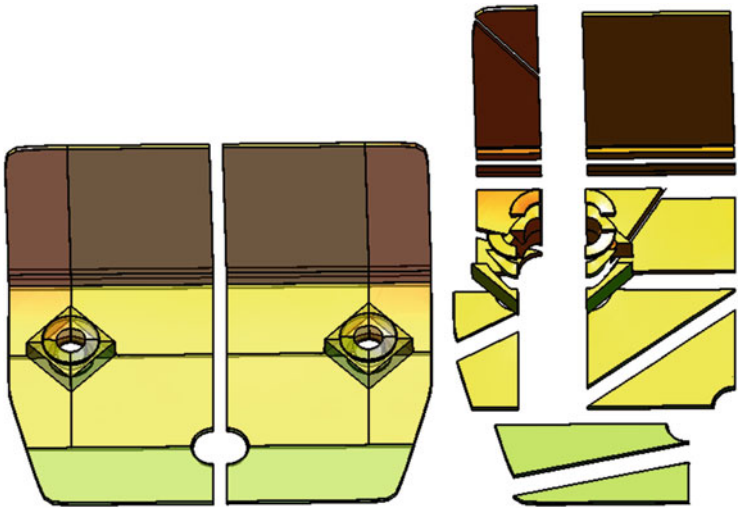


Fig. 16 *Left*: the car part, divided into two along its plane of symmetry and preprocessed to cut it into contractible pieces. *Right*: segmentation of one half of the Demonstrator into topological hexahedra

3.5 T-Joint Removal

For some simulation methods, it may be necessary to post-process the model to eliminate T-joints. We do not have a general method for this. A special-purpose method was developed which suffices for our segmentation of the TERRIFIC demonstrator. During the segmentation process, we keep track of adjacency information for the solids. When the segmentation of a solid results in the creation of a T-joint, new edges must be added to the adjacent solid(s) which must themselves be segmented. This segmentation may create new T-joints, resulting in a propagation of segmentations through the solid. The result for the TERRIFIC Demonstrator is shown in Fig. 6c. Ongoing research on advanced IGA techniques aim to handle meshes with T-joints directly (see Sect. 1).

4 Hexahedron Parameterization

The aim is to convert a boundary represented topological hexahedron into a volume parameterization. Besides the possibility of quite distorted hexes, difficulty arises from the fact that the boundary faces are trimmed surfaces, since the domain of such a surface is not a rectangle in general.

A single topological hexahedron is parameterized by the following procedure.

1. Sample each of the four edges of a boundary face at parameters such that the distance between consecutive points is approximately constant. This is necessary because the curves from adjacent boundaries do not generally have the same parameterization;
2. Apply 2-dimensional Coon's patches to the sampled points, to produce a grid of points in the domain;
3. Evaluate the grid points, to produce a grid on the 3D surface;
4. Applying the above steps for each of the six faces of the topological hexahedron, we now have a grid of sampled boundary points. Grid points agree along the edges because of the way the edges are sampled in Step 1. Apply 3D Coon's patches to the boundary points to produce a grid of interior points. (It may be possible to skip this step if the topological hexahedron is not too heavily distorted);
5. Fit a tensor product B-spline to the sampled points using a least squares method.

The fitting method could be easily modified to include more complex quality measures.

Applying the procedure to a collection of topological hexahedra produces a multi-patch domain (i.e., a domain composed of multiple volume-parameterized topological hexahedra) suitable for isogeometric analysis (see Sect. 5). If a segmentation of a solid into topological hexahedra has no T-joints, then we can also ensure watertightness of the collection of patches by the following procedure. (The collection is not automatically watertight because the construction of cutting faces during the segmentation, as well as the fitting of the volume patch, can cause errors in the corresponding control points of incident patches.)

(i) Find all the incident faces between patches (the easiest way to do this is to hold on to the adjacency information from the segmentation procedure);
(ii) Wherever two or more patches meet (more than two patches can meet at vertices or edges), the corresponding control points of all such patches are moved to the average.

This simple adjustment is made possible by the following fact: If two topological hexahedra meet face-to-face in a quadrilateral patch, then this patch is represented by identical trimmed NURBS surfaces for both hexahedra. This is guaranteed by our construction of the splitting surface, cf. Sect. 3.3. The steps 1–3 of our parameterization procedure therefore give identical parameterization of the matching boundary faces in both patches. Thus, the only possible deviations between the control points are due to the approximation of the inner points in step 5. They can safely be expected to be relatively minor and can be dealt with by the averaging step (ii).

5 Simulation

As applications of the segmentation presented in previous sections, we use the TERRIFIC Demonstrator and the chair stand in mechanical simulations, solving the linear elasticity partial differential equation using an isogeometric finite element discretization. In Sect. 5.1 we briefly present the linear elasticity problem. In Sect. 5.2 we describe the method for solution on a multi-patch domain. Results are presented in Sects. 5.3 and 5.4.

5.1 The Linear Elasticity Problem

The linear elasticity problem on a domain $\Omega \subset \mathbb{R}^3$ means finding a solution $\mathbf{u} : \Omega \to \mathbb{R}^3$ of the following boundary value problem:

$$\begin{aligned} -\operatorname{div} \boldsymbol{\sigma}(\mathbf{u}) &= \mathbf{b} \quad \text{in } \Omega, \\ \mathbf{u} &= \mathbf{g} \quad \text{on } \Gamma_u, \\ \boldsymbol{\sigma}(\mathbf{u}) \cdot \mathbf{n} &= \mathbf{t} \quad \text{on } \Gamma_n, \end{aligned} \tag{1}$$

where $\Gamma_u \subset \delta\Omega$ is the Dirichlet boundary, $\Gamma_n \subset \delta\Omega$ the Neumann boundary of the domain, the Cauchy stress $\boldsymbol{\sigma}$ is expressed by the linear St. Venant-Kirchhoff constitutive law,

$$\boldsymbol{\sigma}(\mathbf{u}) = 2\mu\, \boldsymbol{\varepsilon}(\mathbf{u}) + \lambda\ \operatorname{trace} \boldsymbol{\varepsilon}(\mathbf{u})\, \mathbf{I}$$

using Lamé constants $\mu, \lambda > 0$ and the linear strain tensor

$$\boldsymbol{\varepsilon}(\mathbf{u}) = \frac{1}{2}\left(\nabla \mathbf{u}^T + \nabla \mathbf{u}\right).$$

For a finite element discretization the weak form of (1) is required, i.e. seeking $\mathbf{u} \in H^1(\Omega), \mathbf{u}|_{\Gamma_u} = \mathbf{g}$:

$$\int_\Omega \boldsymbol{\varepsilon}(\mathbf{v}) : \boldsymbol{\sigma}(\mathbf{u})\, d\mathbf{x} = \int_\Omega \mathbf{v}^T \mathbf{b}\, d\mathbf{x} + \int_{\Gamma_n} \mathbf{v}^T\, \mathbf{t}\, d\Gamma \quad \forall \mathbf{v} \in H_0^1(\Omega).$$

Using the 4th-order elasticity tensor $\mathbb{C}$,

$$\mathbb{C}_{ijkl} = \lambda\, \delta_{ij}\delta_{kl} + \mu\left(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{kj}\right), \quad i,j,k,l = 1,2,3,$$

the equation can be rewritten as:

$$\int_{\Omega} \nabla \mathbf{v} : \mathbb{C} : \nabla \mathbf{u} \, d\mathbf{x} = \int_{\Omega} \mathbf{v}^T \mathbf{b} \, d\mathbf{x} + \int_{\Gamma_n} \mathbf{v}^T \, \mathbf{t} \, d\Gamma \quad \forall \mathbf{v} \in H_0^1(\Omega). \tag{2}$$

5.2 Isogeometric Treatment with Multi-patch Domain

The basis of an isogeometric method is a spline representation of the domain Ω and also the numerical solution field $\mathbf{u}^h$ [1]. This means that there exists a global mapping

$$\mathbf{g} : \Omega_0 \to \Omega, \quad \mathbf{x} = \mathbf{g}(\boldsymbol{\xi}) = \sum_{i=1}^{n} N_i^p(\boldsymbol{\xi}) \, \mathbf{c}_i, \tag{3}$$

which parameterizes the whole domain as a B-Spline or NURBS volume, using tensor product basis functions N_i^p of degree p, which are defined on the parameter domain Ω_0 using knot vectors $\Xi \in \mathbb{R}^{n+p+1}$, and control points $\mathbf{c}_i \in \mathbb{R}^3$. Then $\mathbf{u}^h$ is discretized as the push-forward of these spline functions onto the physical domain Ω and a set of displacement control points $\mathbf{d}_i \in \mathbb{R}^3$:

$$\mathbf{u}^h(\mathbf{x}) = \sum_{i=1}^{n} N_i^p \left(\mathbf{g}^{-1}(\mathbf{x})\right) \, \mathbf{d}_i. \tag{4}$$

Finally substitution of (4) into (2) and numerical evaluation of the integrals leads to the well-known matrix-vector form of the finite element discretization of the elasticity problem:

$$\mathbf{K}\mathbf{d} = \mathbf{f}, \tag{5}$$

where $N = 3n$ is the dimension of the system, $\mathbf{d}$ is the vector of unknown control point displacements, $\mathbf{K}$ the stiffness matrix and $\mathbf{f}$ the external force vector.

We have to deal with multi-patch partitionings of the initial domain Ω into b subdomains (patches):

$$\overline{\Omega} = \bigcup_{i=1}^{b} \overline{\Omega^i}, \quad \Omega^i \cap \Omega^j = \emptyset.$$

If two patches Ω_i and Ω_j are adjacent, i.e.

$$\overline{\Omega^i} \cap \overline{\Omega^j} = \Gamma^{ij} \neq \emptyset,$$

we call Γ^{ij} the interface between Ω^i and Ω^j. Thus we can discretize the solutions $\mathbf{u}^i := \mathbf{u}|_{\Omega^i}$ independently, but we have to enforce the conditions $\mathbf{u}^i = \mathbf{u}^j$ on Γ^{ij}. As we deal with conforming parameterizations, i.e. $\mathbf{g}_i|_{\Gamma_{ij}} \equiv \mathbf{g}_j|_{\Gamma_{ij}} \ \forall \ \Gamma_{ij}$, this continuity constraint simply means corresponding displacement control points have to be equal, i.e.

$$\mathbf{d}_k^i = \mathbf{d}_l^j \quad \forall\, \mathbf{d}_k^i \in \mathbf{u}^{h^i}|_{\Gamma_{ij}},\ \mathbf{d}_l^j \in \mathbf{u}^{h^j}|_{\Gamma_{ij}} \text{ with } \mathbf{c}_k^i = \mathbf{c}_l^j. \tag{6}$$

For the enforcement of these constraints basically two possibilities exist, either adding each constraint as additional equations in a Lagrangian multiplier approach, or eliminating degrees of freedom from the system.

5.3 TERRIFIC Demonstrator

The isogeometric volume parameterization consists of 70 patches of cubic B-Spline volumes, with a total of 24,010 control points and 72,030 degrees of freedom (DOFs). Coupling of displacement control points on 132 interfaces has to be enforced. When Lagrangian multipliers are used for coupling, the isogeometric finite element discretization of the model has 90,048 DOFs, otherwise when elimination is used, it has only 53,925 DOFs.

As material parameters of linear elasticity we chose

$$\lambda = 54.0 \text{ GPa}, \quad \mu = 27.8 \text{ GPa}.$$

As boundary conditions we take a clamping of the right bore hole (see Fig. 17) by a zero Dirichlet condition at Γ_u and a surface traction on the left bore hole as a Neumann boundary condition on Γ_n:

$$\mathbf{u} = \mathbf{0} \text{ m} \quad \text{on } \Gamma_u,$$
$$\mathbf{t} = (60.0,\ -42.0,\ 0.0)^T \text{ MPa} \quad \text{on } \Gamma_n.$$

Running the simulation, we get a maximal norm of displacement of 7.06 mm and internal energy of 70.1 J. The displaced object can be seen in Fig. 17. Comparison with a commercial finite element software shows a very good correspondence of results. Using ANSYS® with a mesh of 20,855 quadratic tetrahedral elements (type SOLID187) and 36,181 nodes resp. 108,543 DOFs, we have a maximum displacement of 7.00 mm and energy 70.2 J. It is also worth remarking that the stress distribution obtained from IGA in Fig. 17b looks very smooth.

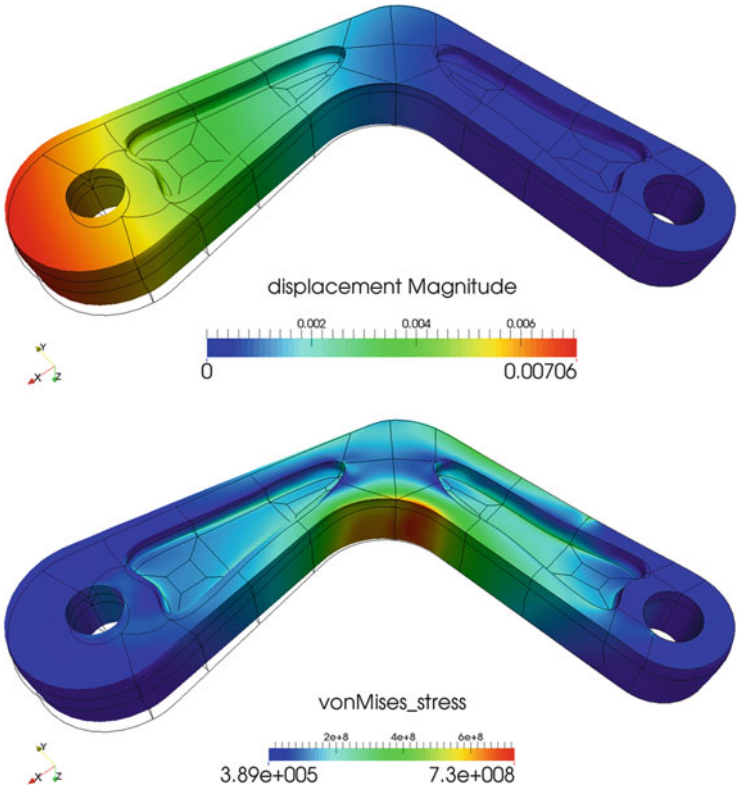


Fig. 17 Linear elasticity simulation of TERRIFIC Demonstrator. Displaced part, colored by (**a**) norm of displacement in [m] and (**b**) von Mises stress in [Pa]

5.4 Chair Stand

The chair stand model consists of 12 cubic B-Spline volumes with a total of 3,969 control points and 11,907 DOFs. Using coupling of patches with the Lagrangian multiplier approach we have a total of 14,231 DOFs.

For a realistic simulation of the chair stand we have added an inner cylinder connecting the stand with the seating and a small cylinder connecting to the wheels made from aluminum, and assumed the main parts of the stand are made from plastic (Polyamid P6). Thus the material parameters are:

$$\lambda_{Al} = 54.0\,\text{GPa}, \quad \mu_{Al} = 27.8\,\text{GPa},$$
$$\lambda_{P6} = 3.19\,\text{GPa}, \quad \mu_{P6} = 0.899\,\text{GPa}.$$

Assuming a person weighing 81.5 kg sitting on the chair, this would result in a vertical force of 0.10 MPa acting as Neumann boundary condition on the inner aluminum cylinder. Furthermore symmetry boundary conditions are applied on displacements, since we only model one fifth of the total chair stand. As the smaller cylinder connects to the wheel, which we assume as a frictionless link to the ground, no vertical displacement is possible there.

Running the simulation, we get a small vertical displacement of the inner cylinder of 0.82 mm and a maximal von Mises stress of 1.6 MPa. The total internal energy is 64.2 mJ. The deformed stand is visualized in Fig. 18. We also validate our results by a comparison to commercial finite element software ANSYS®. Automated mesh generation provides a discretization using a mix of 13,437 quadratic hexahedral and tetrahedral elements (types SOLID186 and SOLID187) with 22,518 nodes and 57,554 DOFs. The simulation results in an evaluated z-displacement of 0.81 mm, a von Mises stress of 1.6 MPa and a total energy of 62.8 mJ.

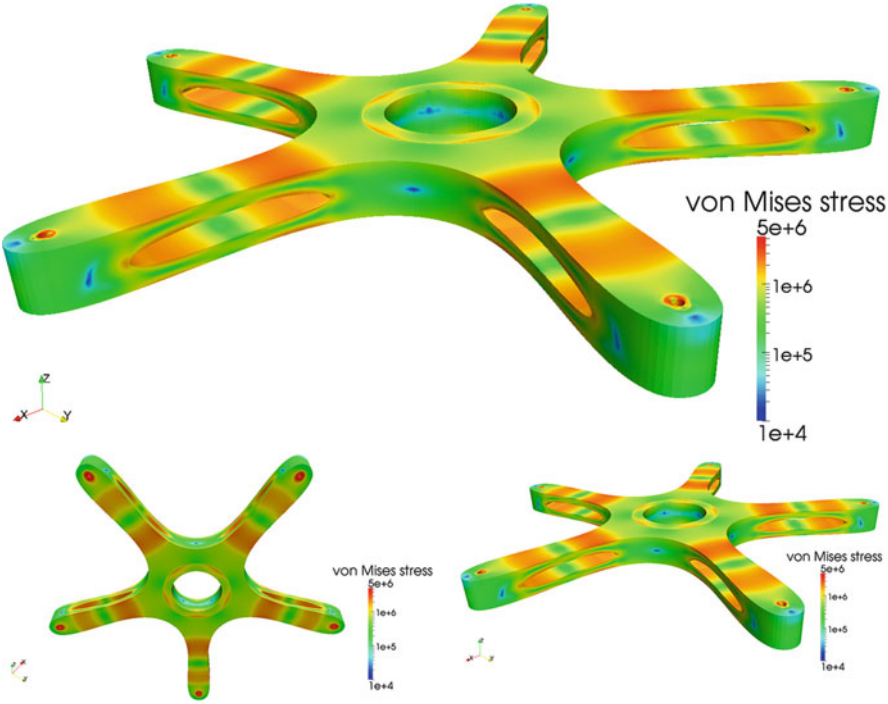


Fig. 18 Linear elasticity simulation of chair stand. Different views of displaced part, colored by von Mises stress in [Pa]

6 Conclusions and Further Work

We created a pipeline, including algorithms and software together with a rigorous mathematical treatment, for the the conversion of 3D models into an IGA-suitable form. The first step of the pipeline is the reconstruction, if necessary, of a CAD-like boundary representation from triangulated data. Then the boundary represented solid is segmented into a collection of topological hexahedra. Finally, volumetric NURBS patches are constructed for the hexahedra.

We used outputs of the pipeline to conduct simulations based on the linear elasticity problem. Results had good agreement with commercial finite element software, demonstrating the suitability of the segmentation and parameterization for simulation.

While the theory and software developed forms the core of the pipeline, there are several improvements already under development:

- improved geometric construction of the cutting curves and surfaces, with theoretical guarantees;
- addressing solids which are non-contractible or do not have 3-vertex-connected edge graphs without human interaction;
- improving the quality of isogeometric segmentation, in terms of the number and shape of the resulting topological hexahedra, using the surface extension methods described in Sect. 3.4.

Potential future developments include the use of splines allowing local refinement, such as LR- or THB-splines, and improving the reconstruction of cylindrical surfaces using periodic splines.

Acknowledgements This work has been funded by the European Commission projects TERRIFIC (Grant Agreement 284981), EXAMPLE (Grant Agreement 324340) and ITN INSIST (Grant Agreement 289361), and the Austrian Science Fund (FWF) project Geometry and Simulation (Project Number S 117). We thank Martin Schifko at Engineering Center Steyr and Stefan Boschert at Siemens Corporate Technology for some of the input data used in our work.

References

1. J.A. Cottrell, T.J.R. Hughes, Y. Bazilevs, *Isogeometric Analysis: Toward Integration of CAD and FEA* (Wiley, Chichester/Hoboken, 2009)
2. M.S. Floater, Parametrization and smooth approximation of surface triangulations. Comput. Aided Geom. Des. **14**(3), 231–250 (1997)
3. M.S. Floater, Mean value coordinates. Comput. Aided Geom. Des. **20**(1), 19–27 (2003)
4. D. Großmann, B. Jüttler, H. Schlusnus, J. Barner, A.-V. Vuong, Isogeometric simulation of turbine blades for aircraft engines. Comput. Aided Geom. Des. **29**(7), 519–531 (2012). Geometric Modeling and Processing 2012
5. S. Han, J. Xia, Y. He, Constructing hexahedral shell meshes via volumetric polycube maps. Comput.-Aided Des. **43**(10), 1222–1233 (2011)

6. B. Jüttler, M. Kapl, D.-M. Nguyen, Q. Pan, M. Pauley, Isogeometric segmentation: the case of contractible solids without non-convex edges. Comput.-Aided Des. **57**, 74–90 (2014)
7. U. Langer, I. Toulopoulos, Analysis of multipatch discontinuous Galerkin IgA approximations to elliptic boundary value problems. Technical Report 1408.0182, arxiv.org, 2014
8. Y. Lu, R. Gadh, T.J. Tautges, Feature based hex meshing methodology: feature recognition and volume decomposition. Comput.-Aided Des. **33**(3), 221–232 (2001)
9. T. Martin, E. Cohen, Volumetric parameterization of complex objects by respecting multiple materials. Comput. Graph. **34**(3), 187–197 (2010). Shape Modelling International (SMI) Conference 2010
10. T. Martin, E. Cohen, R.M. Kirby, Volumetric parameterization and trivariate B-spline fitting using harmonic functions. Comput. Aided Geom. Des. **26**(6), 648–664 (2009)
11. T. Nguyen, B. Jüttler, Parameterization of contractible domains using sequences of harmonic maps, in *Curves and surfaces*. Lecture Notes in Computer Science, vol. 6920 (Springer, Heidelberg, 2012), pp. 501–514
12. D.-M. Nguyen, M. Pauley, B. Jüttler, Isogeometric segmentation. Part II: on the segmentability of contractible solids with non-convex edges. Graph. Models **76**(5), 426–439 (2014). Geometric Modeling and Processing 2014
13. D.-M. Nguyen, M. Pauley, B. Jüttler, Isogeometric segmentation: construction of auxiliary curves. Comput.-Aided Des. **70**, 89–99 (2016). Proceedings of GDSPM 2015, conditionally accepted
14. M. Nieser, U. Reitebuch, K. Polthier, CubeCover – parameterization of 3D volumes. Comput. Graph. Forum **30**(5), 1397–1406 (2011)
15. A. Sheffer, M. Etzion, A. Rappoport, M. Bercovier, Hexahedral mesh generation using the embedded Voronoi graph, in *Proceedings of the 7th International Meshing Roundtable*, Sandia National Laboratories (1999), pp. 347–364
16. B. Strodthoff, M. Schifko, B. Jüttler, Horizontal decomposition of triangulated solids for the simulation of dip-coating processes. Comput.-Aided Des. **43**(12), 1891–1901 (2011)
17. T.J. Tautges, T. Blacker, S.A. Mitchell, The whisker weaving algorithm: a connectivity-based method for constructing all-hexahedral finite element meshes. Int. J. Numer. Methods Eng. **39**(19), 3327–3349 (1996)
18. T. Varady, R. Martin, Reverse engineering, in *Handbook of Computer Aided Geometric Design*, ed. by G. Farin, J. Hoschek, M.-S. Kim (North-Holland, Amsterdam, 2002), pp. 651–681
19. X. Wang, X. Qian, An optimization approach for constructing trivariate *B*-spline solids. Comput.-Aided Des. **46**, 179–191 (2014)
20. W. Wang, Y. Zhang, L. Liu, T.J.R. Hughes, Trivariate solid T-spline construction from boundary triangulations with arbitrary genus topology. Comput.-Aided Des. **45**(2), 351–360 (2013)
21. D. White, L. Mingwu, S.E. Benzley, G.D. Sjaardema, Automated hexahedral mesh generation by virtual decomposition, in *Proceedings of the 4th International Meshing Roundtable*, Sandia National Laboratories (1995), pp. 165–176
22. G. Xu, B. Mourrain, R. Duvigneau, A. Galligo, Analysis-suitable volume parameterization of multi-block computational domain in isogeometric applications. Comput.-Aided Des. **45**(2), 395–404 (2013)
23. W. Yu, K. Zhang, S. Wan, X. Li, Optimizing polycube domain construction for hexahedral remeshing. Comput.-Aided Des. **46**, 58–68 (2014)
24. Y. Zhang, C. Bajaj, Adaptive and quality quadrilateral/hexahedral meshing from volumetric data. Comput. Methods Appl. Mech. Eng. **195**, 942–960 (2006)
25. Y. Zhang, W. Wang, T.J.R. Hughes, Solid T-spline construction from boundary representations for genus-zero geometry. Comput. Methods Appl. Mech. Eng. **249/252**, 185–197 (2012)
26. Y. Zhang, W. Wang, T.J.R. Hughes, Conformal solid T-spline construction from boundary T-spline representations. Comput. Mech. **51**(6), 1051–1059 (2013)

Domain Decomposition Methods and Kirchhoff-Love Shell Multipatch Coupling in Isogeometric Analysis

Andreas Apostolatos, Michael Breitenberger, Roland Wüchner, and Kai-Uwe Bletzinger

Abstract The necessity for solving the isogeometric Kirchhoff-Love shell problem into multiple domains has been exemplified especially in cases where the geometry comprises multipatches. In fact, geometries taken from *Computer Aided Geometric Design* involve in principle trimmed multipatches. Herein, the application and comparison of the most common *Domain Decomposition Methods* for the coupling of Kirchhoff-Love shell multipatches in isogeometric analysis is presented. The investigated methods comprise *Penalty* and *Lagrange Multipliers* methods. All methods are extended to account for geometrically nonlinear problems. The aforementioned methods provided highly accurate results, thus extending the Kirchhoff-Love shell analysis from a single to multiple patches which is a prerequisite for solving practical engineering problems using isogeometric analysis.

1 Introduction

In the past decades light-weight shell structures have been playing an ever-increasing role in various engineering applications as the cost can be significantly reduced by keeping vital mechanical properties such as the load carrying capacity. The Kirchhoff-Love or the Reissner-Mindlin shell models [1] were used among plenty of studies.

Isogeometric Analysis [2] (IGA) is a generalization of the well-established classical *Finite Element Analysis* (FEA) [3, 4] using a geometric basis, as used in *Computer-Aided Geometric Design* (CAGD), within the analysis process. The *Non-Uniform Rational B-Splines* (NURBS) basis functions [5] form the most common geometric basis and therefore are also used within this study, see [6]. The successful application of isogeometric analysis for Kirchhoff-Love shell problems was demonstrated in [7], which was initially confined into single-patch

A. Apostolatos (✉) • M. Breitenberger • R. Wüchner • K.-U. Bletzinger
Technische Universität München, Arcisstraße 21, D-80333 München, Germany
e-mail: andreas.apostolatos@tum.de; michael.breitenberger@tum.de; wuechner@tum.de; kub@tum.de

B. Jüttler, B. Simeon (eds.), *Isogeometric Analysis and Applications 2014*,
Lecture Notes in Computational Science and Engineering 107,
DOI 10.1007/978-3-319-23315-4_4

geometries. However, geometries taken from CAGD comprise typically trimmed multipatches [8]. Another issue is the tensor product structure that multidimensional NURBS basis has, making local refinement over the NURBS basis in its original form impossible. However, studies on local refinement and adaptivity over a modified NURBS basis can be found among other studies in [9, 10]. It is worth mentioning that since the variational index of the Kirchhoff-Love shell problem is equal to two, the basis functions must be at least one time continuously differentiable and therefore geometries involving kinks are ab initio excluded in standard NURBS-based IGA. Finally it is demonstrated in [11] that isogeometric analysis for shell structures can be extended to account for trimmed multipatch geometries which are not necessarily watertight, can join with an arbitrary angle along their common interfaces and which can have an independent from each other discretization level.

Within this study, three different *Domain Decomposition Methods* (DDMs) are introduced and compared for the multipatch coupling of the Kirchhoff-Love shell problem in both linear and nonlinear isogeometric analysis: a *Penalty*, a *Lagrange Multipliers* and an *augmented Lagrange Multipliers* method.

The Penalty method, firstly introduced in [12], aimed at enlarging the test and the solution spaces by not restricting the corresponding functions to comply with the Dirichlet boundary conditions. Since then it has been widely employed for the coupling of non-conforming finite element meshes, see for example [13–15]. Variationally consistent extensions of the Penalty method for boundary value problems have been proposed among other studies in [16, 17] and specifically for the coupling of linear multipatch thin and blended shell structures in [18] within IGA. The *Bending Strip Method* [19] (BSM) utilized a penalization over the bending stiffness of Kirchhoff-Love shell multipatches along their common interfaces, thus extending isogeometric shell analysis to multipatches. However, this method is restricted into matching interface parametrizations. In the present contribution, a Penalty method involving two penalty terms, for the interface displacement and rotations Dirichlet compatibility conditions, is presented, which releases this substantial restriction of matching parametrizations along the patch interfaces. As common for the Penalty methods, the underlying formulation is positive definite [12] but the accuracy of the numerical scheme depends on the choice of the penalty parameters. A rule for the choice of the penalty parameters involving scaling with the material matrices norms and the element size is herein proposed. The method is applicable for multipatches which join arbitrarily along their common interfaces.

The Lagrange Multipliers method is one of the most common methods for enforcing constraints within a finite element equation system, see for instance [14, 20, 21]. Having at hand two interface Dirichlet conditions, one for the displacement and one for the rotation field, two independent Lagrange Multipliers fields are chosen which act as the interface traction force and traction moment fields from a physical viewpoint, respectively. As being typical for the Lagrange Multipliers methods, the resulting linear equation system belongs to the category of saddle point problems, which necessitates that the discrete spaces satisfy

the *Ladyzenskaja-Babuska-Brezzi* [22] condition so that a unique solution to the problem can be guaranteed. The numerical investigations showed satisfactory results for the decomposed Kirchhoff-Love shell problem, provided that a suitable discretization for the Lagrange Multipliers fields is found.

At last, the augmented Lagrange Multipliers method for the Kirchhoff-Love shell problem is elaborated and discussed. This method is a combination of the Lagrange Multipliers and the Penalty method [15, 23]. This method belongs also to the class of saddle point problems and hence, the same restrictions as for the Lagrange Multipliers method apply in this case.

The application and comparison of the aforementioned methods is demonstrated using two linear and two nonlinear problems. Thus, isogeometric Kirchhoff-Love shell analysis is extended into multiple domains, towards the integration of CAGD and FEA into a large scale computational environment as in [11].

2 Theory

2.1 The Strong Form of the Problem

The model problem is an elastic body represented by a surface embedded in 3D space $\overline{\Omega} \subset \mathbb{R}^3$, where $\overline{\Omega}$ stands for the closure of Ω in the Euclidean space. The surface is described using a convective covariant space $\overline{\Omega_0} \subset \mathbb{R}^2$ through the map $\mathbf{S} : \overline{\Omega_0} \rightarrow \overline{\Omega}$. The mechanical behaviour of the elastic body is described by the *Kirchhoff-Love shell* theory. The process is assumed to be static, i.e. no inertia forces are to be considered. For each material particle $\mathbf{X} \in \overline{\Omega}$, the unknown displacement field $\mathbf{d} = \mathbf{d}(\mathbf{X})$ is defined in a Lagrangian description of the motion, that is, the current configuration is defined by the collection of material particles $\mathbf{x} = \mathbf{X} + \mathbf{d}$. In this way, the displacement field can have a representation in both the physical and the covariant space, that is [1],

$$\mathbf{d} = d_i^0 \mathbf{e}_i = \mathbf{d}_\parallel + d_3 \mathbf{A}_3 = d^\alpha \mathbf{A}_\alpha + d_3 \mathbf{A}_3 \quad \forall \mathbf{X} \in \overline{\Omega} \,, \tag{1}$$

where d_i^0, $\mathbf{d}_\parallel$ and d_3 stands for the components of the displacement field on the Cartesian space, the displacement field on the tangent space of the shell's surface and the displacement component normal to the shell's surface, respectively. According to standard notations, Greek and Latin indices range from 1 to 2 and from 1 to 3, respectively, if not stated otherwise. Moreover, $\mathbf{e}_i$ and $\{\mathbf{A}_\alpha, \mathbf{A}_3\}$ stand for the base vectors of the global Cartesian and the covariant space in the reference configuration, respectively. In particular, the covariant base vectors are given by [1],

$$\mathbf{A}_\alpha := \mathbf{X}_{,\alpha} \,, \tag{2a}$$

$$\mathbf{A}_3 := \bar{j}^{-1} \mathbf{A}_1 \times \mathbf{A}_2 \,, \tag{2b}$$

in $\overline{\Omega}$ with $\bar{j} = \|\mathbf{A}_1 \times \mathbf{A}_2\|_2$ and the commas standing for the partial derivatives with respect to the convective coordinates $\left(\theta^1, \theta^2\right) \in \overline{\Omega_0}$. The latter implies that $\mathbf{A}^3 = \mathbf{A}_3$ in what concerns the contravariant basis, meaning that the contravariant and the covariant components of any first order tensor defined over the convective space lateral to the shell's surface are equal. Hence notation d_3 is employed for both the covariant and the contravariant component of the displacement field in $\mathbf{A}_3$ direction, see [1]. In addition, the map $\mathbf{S}$ is assumed to be regular almost everywhere in Ω_0 meaning that $\bar{j}$ is non zero almost everywhere in Ω_0, see [24]. The boundary of the surface $\Gamma = \partial\Omega$ is assumed to be a piecewise C^1-continuous curve, so that the tangent, the normal and the binormal vectors can be uniquely defined almost everywhere in $\overline{\Omega}$. Let also $\mathfrak{S}^m$ be the space of all m-th order symmetric tensors. Then, the curvature tensor $\mathbf{B} \in \mathfrak{S}^2$ is given by [24],

$$\mathbf{B} = B_{\alpha\beta}\, \mathbf{A}^\alpha \otimes \mathbf{A}^\beta := -\mathbf{A}_\alpha \cdot \mathbf{A}_{3,\beta}\, \mathbf{A}^\alpha \otimes \mathbf{A}^\beta = \mathbf{A}_{\alpha,\beta} \cdot \mathbf{A}_3\, \mathbf{A}^\alpha \otimes \mathbf{A}^\beta \quad \forall \mathbf{X} \in \overline{\Omega}\ , \tag{3}$$

where $\{\mathbf{A}^\alpha, \mathbf{A}_3\}$ stands for the contravariant base vectors in the reference configuration. The small rotations assumption results in a linear relation between the rotations and the displacements. In this way, the total rotation field $\boldsymbol{\omega} \in \mathfrak{S}^1$ can be defined at each point $\mathbf{X} \in \overline{\Omega}$ by [1],

$$\boldsymbol{\omega}\left(\mathbf{d}\right) = -\boldsymbol{\epsilon} \cdot \left(\nabla d_3 + \mathbf{B} \cdot \mathbf{d}_\parallel\right) = \epsilon^{\alpha\beta} \left(d_{3,\beta} + B_{\beta\gamma}\, d^\gamma\right) \mathbf{A}_\alpha\ , \tag{4}$$

where $\boldsymbol{\epsilon} = \epsilon^{\alpha\beta}\, \mathbf{A}_\alpha \otimes \mathbf{A}_\beta$ stands for the permutation second order tensor with [1],

$$\epsilon^{\alpha\beta} = \frac{1}{\bar{j}} \begin{cases} 1, \text{ if } (\alpha, \beta) = (1, 2)\ , \\ -1, \text{ if } (\alpha, \beta) = (2, 1)\ , \\ 0, \text{ if } \alpha = \beta\ . \end{cases} \tag{5}$$

In addition, ∇ stands for the gradient operator over the convective space. The linear Hooke's law is employed for the description of the constitutive behaviour of the elastic material. Thus, the force and moment tensors corresponding to the second Piola-Kirchhoff stress measure $\mathbf{n}, \mathbf{m} \in \mathfrak{S}^2$ are linearly dependent on the membrane and bending strain tensors $\boldsymbol{\varepsilon}, \boldsymbol{\kappa} \in \mathfrak{S}^2$ through [25],

$$\mathbf{n}\left(\mathbf{d}\right) = \frac{Eh}{1-\nu^2} \mathbf{C} : \boldsymbol{\varepsilon}\left(\mathbf{d}\right) = \mathbf{C}_\mathrm{m} : \boldsymbol{\varepsilon}\left(\mathbf{d}\right)\ , \tag{6a}$$

$$\mathbf{m}\left(\mathbf{d}\right) = \frac{Eh^3}{12\left(1-\nu^2\right)} \mathbf{C} : \boldsymbol{\kappa}\left(\mathbf{d}\right) = \mathbf{C}_\mathrm{b} : \boldsymbol{\kappa}\left(\mathbf{d}\right)\ , \tag{6b}$$

in $\overline{\Omega}$, where $\mathbf{C} \in \mathfrak{S}^4$, E, ν and h is the elasticity tensor, the Young's modulus, the Poisson's ratio and the shell's thickness, respectively. Tensors $\mathbf{C}_\mathrm{m}, \mathbf{C}_\mathrm{b} \in \mathfrak{S}^4$ stand for the particular elasticity tensors for the membrane and the bending action of the Kirchhoff-Love shell, respectively. In what concerns the kinematics of the Kirchhoff-Love shell, the nonlinear membrane and bending strain tensors

corresponding to the Green-Lagrange strain measure are respectively defined by [7],

$$\boldsymbol{\varepsilon} = \varepsilon_{\alpha\beta}\, \mathbf{A}^{\alpha} \otimes \mathbf{A}^{\beta} = \frac{1}{2}\left(\mathbf{a}_{\alpha} \cdot \mathbf{a}_{\beta} - \mathbf{A}_{\alpha} \cdot \mathbf{A}_{\beta}\right)\, \mathbf{A}^{\alpha} \otimes \mathbf{A}^{\beta}\,, \tag{7a}$$

$$\boldsymbol{\kappa} = \kappa_{\alpha\beta}\, \mathbf{A}^{\alpha} \otimes \mathbf{A}^{\beta} = \left(B_{\alpha\beta} - b_{\alpha\beta}\right)\, \mathbf{A}^{\alpha} \otimes \mathbf{A}^{\beta}\,, \tag{7b}$$

where $\mathbf{a}_{\alpha}$ and $b_{\alpha\beta}$ stand for the base vectors of the covariant space and covariant components of the curvature tensor (3) over the current configuration. The linearised version of the strain components (7a) and (7b) writes [25],

$$\varepsilon_{\alpha\beta} := \frac{1}{2}\left(\mathbf{A}_{\beta} \cdot \mathbf{d}_{,\alpha} + \mathbf{A}_{\alpha} \cdot \mathbf{d}_{,\beta}\right)\,, \tag{8a}$$

$$\begin{aligned}\kappa_{\alpha\beta} := -\mathbf{A}_3 \cdot \mathbf{d}_{,\alpha\beta} + \mathbf{A}_{\alpha,\beta} \cdot \mathbf{A}_3 \frac{1}{j}\left((\mathbf{A}_2 \times \mathbf{A}_3) \cdot \mathbf{d}_{,1} - (\mathbf{A}_1 \times \mathbf{A}_3) \cdot \mathbf{d}_{,2}\right) \\ + \frac{1}{j}\left(\left(\mathbf{A}_{\alpha,\beta} \times \mathbf{A}_2\right) \cdot \mathbf{d}_{,1} - \left(\mathbf{A}_{\alpha,\beta} \times \mathbf{A}_1\right) \cdot \mathbf{d}_{,2}\right)\,.\end{aligned} \tag{8b}$$

Displacement and/or rotation Dirichlet boundary conditions are prescribed along a portion of the body's boundary $\Gamma_{\mathrm{d}} \subset \Gamma$, confining ourselves into the homogeneous Dirichlet boundary conditions without loss of generality. Forces $\mathbf{p}$ and/or moments $\mathbf{r}$ are applied over another portion of the shell's boundary $\Gamma_{\mathrm{n}} \subset \Gamma$ forming the so-called Neumann boundary conditions. Finally, it is also assumed that the elastic body is subject to body forces designated by $\mathbf{b} = b^{\alpha}\mathbf{A}_{\alpha} + b_3\mathbf{A}_3$. Subsequently, domain Ω is decomposed into a finite set of subdomains $\Omega^{(i)}$, $i = 1, \ldots, n \in \mathbb{N}$ for which the following relations hold,

$$\bigcup_{i=1}^{n} \overline{\Omega^{(i)}} = \overline{\Omega}\,, \tag{9a}$$

$$\overline{\Omega^{(i)}} \cap \overline{\Omega^{(j)}} = \overline{\Gamma_{\mathrm{c}}^{(i,j)}} \quad \forall\, (i,j) \in \mathcal{I}\,, \tag{9b}$$

$$\bigcup_{(i,j)\in\mathcal{I}} \overline{\Gamma_{\mathrm{c}}^{(i,j)}} = \overline{\Gamma_{\mathrm{c}}}\,, \tag{9c}$$

$\mathcal{I}$ being the set of all $n_{\mathrm{I}} \in \mathbb{N}$ intersections between the subdomains excluding pairs (i,i) and $\Gamma_{\mathrm{c}}^{(i,j)}$ being a set of piecewise C^1-continuous curves on the Euclidean space in the context of the non-overlapping DDMs. The Dirichlet and the Neumann boundaries, Γ_{d} and Γ_{n} respectively, are decomposed in the same fashion as Ω in (9a)–(9b). The aforementioned problem is depicted in Fig. 1. Subsequently, it is also assumed that the displacement, the rotation, the force and the moment fields are defined piecewisely in $\overline{\Omega}$ at each subdomain $\overline{\Omega^{(i)}}$. Whenever the displacement, the rotation, the force or the moment field is restricted into one subdomain, a superscript $i = 1, \ldots, n$ is used to indicate the domain of restriction, that is, $\mathbf{d}_{|\overline{\Omega^{(i)}}} = \mathbf{d}^{(i)}$,

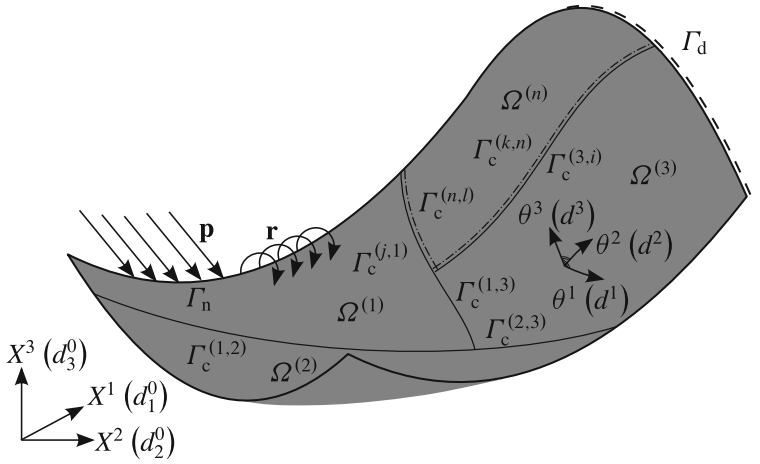


Fig. 1 Problem placement: multipatch Kirchhoff-Love shell structure

$\boldsymbol{\omega}\left(\mathbf{d}^{(i)}\right) = \boldsymbol{\omega}^{(i)}$, $\mathbf{n}\left(\mathbf{d}^{(i)}\right) = \mathbf{n}^{(i)}$ and $\mathbf{m}\left(\mathbf{d}^{(i)}\right) = \mathbf{m}^{(i)}$, respectively. Then, the Kirchhoff-Love of n-domain decomposed *Boundary Value Problem* (BVP) writes:

$$n^{\alpha\beta}\big|_{\alpha} - q^{\alpha} B_{\alpha}^{\ \beta} + b^{\beta} = 0\,, \quad \text{in } \Omega_0 \setminus \Gamma_{\mathrm{c}}\,, \tag{10a}$$

$$n^{\alpha\beta} B_{\alpha\beta} + q^{\alpha}\big|_{\alpha} + b_3 = 0\,, \quad \text{in } \Omega_0 \setminus \Gamma_{\mathrm{c}}\,, \tag{10b}$$

$$m^{\alpha\beta}\big|_{\alpha} - q^{\beta} = 0\,, \quad \text{in } \Omega_0 \setminus \Gamma_{\mathrm{c}}\,, \tag{10c}$$

$$\mathbf{d} = \mathbf{0}\,, \quad \text{on } \subset \Gamma_{\mathrm{d}}\,, \tag{10d}$$

$$\boldsymbol{\omega} = \mathbf{0}\,, \quad \text{on } \subset \Gamma_{\mathrm{d}}\,, \tag{10e}$$

$$\mathbf{n}^{\alpha} u_{\alpha} = \mathbf{p}\,, \quad \text{on } \subset \Gamma_{\mathrm{n}}\,, \tag{10f}$$

$$\mathbf{m}^{\alpha} u_{\alpha} = \mathbf{r}\,, \quad \text{on } \subset \Gamma_{\mathrm{n}}\,, \tag{10g}$$

$$\mathbf{d}^{(i)} - \mathbf{d}^{(j)} = \mathbf{0}\,, \quad \text{on } \Gamma_{\mathrm{c}}^{(i,j)}\ \forall i,j = 1,\ldots,n\,, \tag{10h}$$

$$\boldsymbol{\omega}^{(i)} + \boldsymbol{\omega}^{(j)} = \mathbf{0}\,, \quad \text{on } \Gamma_{\mathrm{c}}^{(i,j)}\ \forall i,j = 1,\ldots,n\,, \tag{10i}$$

where $\mathbf{u}$, $\mathbf{n}^{\alpha} = n^{\alpha\beta}\mathbf{A}_{\beta} + q^{\alpha}\mathbf{A}_3$ and $\mathbf{m}^{\alpha} = m^{\alpha\beta}\mathbf{A}_{\beta}$ stand for the normal to the Neumann boundary Γ_{n}, the traction force and the traction moment vector, respectively. The force and moment tensor components $n^{\alpha\beta}$ and $m^{\alpha\beta}$, respectively, that appear in the above mentioned BVP correspond to the first Piola-Kirchhoff stress measure for the geometrically nonlinear case. Those are connected to the second Piola-Kirchhoff stress measure through the deformation gradient tensor. However, in the geometrically linear case both stress measures coincide, for more details see [1]. Additionally, q^{α} stand for the contravariant components of the shear force vector. The compatibility of the displacement and the rotation field across

the interface Γ_c of the subdomains is ensured through the interface continuity conditions (10h)–(10i).

It is important to note that rotation continuity equation (10i) implies that the interface tangent to surfaces $\Omega^{(i)}$ and $\Omega^{(j)}$ vectors around which rotation vectors $\boldsymbol{\omega}^{(i)}$ and $\boldsymbol{\omega}^{(j)}$ are defined, have opposite direction. In case those interface vectors have the same direction, continuity condition (10i) should be reformulated such that the rotation vectors from each neighbouring patch pair $\Omega^{(i)}$ and $\Omega^{(j)}$ are equal and not opposite.

2.2 The Weak Form of the Decoupled Problem

The variational formulation of the BVP (10a), (10b), (10c), (10d), (10e), (10f) and (10g) without the consideration of the interface conditions writes: Find $\mathbf{d} \in \mathcal{V}$ such that:

$$a(\mathbf{h}, \mathbf{d}) = L(\mathbf{h}) \quad \forall \mathbf{h} \in \mathcal{V}, \tag{11}$$

where $\mathcal{V} \subset \left(H^2(\Omega \setminus \Gamma_c)\right)^3$ is defined by,

$$\mathcal{V} = \left\{ \mathbf{h} \in \left(H^2(\Omega \setminus \Gamma_c)\right)^3 \middle| \mathbf{h} = \mathbf{0} \text{ on } \subset \Gamma_d \text{ and } \boldsymbol{\omega}(\mathbf{h}) = \mathbf{0} \text{ on } \subset \Gamma_d \right\} . \tag{12}$$

On the other hand, form $a : \mathcal{V} \times \mathcal{V} \to \mathbb{R}$ and linear functional $L : \mathcal{V} \to \mathbb{R}$ in (11) are defined as follows,

$$a(\mathbf{h}, \mathbf{d}) := \int_{\Omega \setminus \Gamma_c} \boldsymbol{\varepsilon}(\mathbf{h}) : \mathbf{n}(\mathbf{d}) \, d\Omega + \int_{\Omega \setminus \Gamma_c} \boldsymbol{\kappa}(\mathbf{h}) : \mathbf{m}(\mathbf{d}) \, d\Omega , \tag{13a}$$

$$L(\mathbf{h}) := \int_{\Omega \setminus \Gamma_c} \mathbf{h} \cdot \mathbf{b} \, d\Omega + \int_{\Gamma_n} \mathbf{h} \cdot \mathbf{p} + \boldsymbol{\omega}(\mathbf{h}) \cdot \mathbf{r} \, d\Gamma . \tag{13b}$$

Space $\mathcal{V}$ is naturally equipped with the energy norm [26],

$$\|\mathbf{h}\|_{\mathcal{V}} = \sqrt{a(\mathbf{h}, \mathbf{h})} \qquad \forall \mathbf{h} \in \mathcal{V} , \tag{14}$$

In [26] it was shown that for the linear kinematics case, see Eqs. (8a) and (8b), the bilinear form (13a) is bounded and elliptic in $\mathcal{V}$ under the energy norm (14), meaning that variational problem (11) has a unique solution in $\mathcal{V}$. However, in the present case this solution is discontinuous across the subdomains, i.e. it does not represent the solution to boundary value problem (10a), (10b), (10c), (10d), (10e), (10f), (10g), (10h) and (10i), as it does not account for the interface continuity conditions (10h) and (10i).

2.3 The Weak Form of the Penalty Approach

Let us introduce space $\mathfrak{L} \subset \mathcal{V}$,

$$\mathfrak{L} := \left\{ \mathbf{h} \in \mathcal{V} \,\middle|\, \exists \text{ c.e. } \mathbf{h}^{(i)}, \mathbf{h}^{(j)} \text{ on } \Gamma_{\mathrm{c}}^{(i,j)} : \mathbf{h}^{(i)} - \mathbf{h}^{(j)} \in \left(L^2 \left(\Gamma_{\mathrm{c}}^{(i,j)} \right) \right)^3 \right. \\ \left. \text{and } \boldsymbol{\omega} \left(\mathbf{h}^{(i)} \right) + \boldsymbol{\omega} \left(\mathbf{h}^{(j)} \right) \in \left(L^2 \left(\Gamma_{\mathrm{c}}^{(i,j)} \right) \right)^3 \right\} , \tag{15}$$

which becomes an inner product space when equipped with the inner product $\langle \bullet, \bullet \rangle_{\mathfrak{L}} : \mathfrak{L} \times \mathfrak{L} \to \mathbb{R}$,

$$\langle \mathbf{h}, \mathbf{d} \rangle_{\mathfrak{L}} = a(\mathbf{h}, \mathbf{d}) + \langle \boldsymbol{\chi}_{\mathrm{d}}(\mathbf{h}), \boldsymbol{\chi}_{\mathrm{d}}(\mathbf{d}) \rangle_{0,\Gamma_{\mathrm{c}}} + \langle \boldsymbol{\chi}_{\omega}(\mathbf{h}), \boldsymbol{\chi}_{\omega}(\mathbf{d}) \rangle_{0,\Gamma_{\mathrm{c}}} , \tag{16}$$

for all $\mathbf{h}, \mathbf{d} \in \mathfrak{L}$, where $\boldsymbol{\chi}_{\mathrm{d}}$ and $\boldsymbol{\chi}_{\omega}$ stand for the jump operators for the displacement and the rotation fields across the interface Γ_{c}, respectively. Notation $\langle \bullet, \bullet \rangle_{0,\Gamma_{\mathrm{c}}}$ stands for the inner product in the $L^2(\Gamma_{\mathrm{c}})$ space, meaning that the second term in (16) writes,

$$\langle \boldsymbol{\chi}_{\mathrm{d}}(\mathbf{h}), \boldsymbol{\chi}_{\mathrm{d}}(\mathbf{d}) \rangle_{0,\Gamma_{\mathrm{c}}} = \int_{\Gamma_{\mathrm{c}}} \boldsymbol{\chi}_{\mathrm{d}}(\mathbf{h}) \cdot \boldsymbol{\chi}_{\mathrm{d}}(\mathbf{d}) \, \mathrm{d}\Gamma . \tag{17}$$

Moreover, space $\mathfrak{L}$ is equipped naturally with the norm $\|\bullet\|_{\mathfrak{L}} : \mathfrak{L} \to \mathbb{R}$ induced by the inner product (16),

$$\|\mathbf{h}\|_{\mathfrak{L}} = \sqrt{\langle \mathbf{h}, \mathbf{h} \rangle_{\mathfrak{L}}} \qquad \forall \mathbf{h} \in \mathfrak{L} . \tag{18}$$

Following [15] the variational formulation of the coupled problem using the Penalty approach writes: Find $\mathbf{d} \in \mathfrak{L}$ with $\mathbf{d}^{(i)} - \mathbf{d}^{(j)} = \boldsymbol{\chi}_{\mathrm{d}}^{(i,j)}(\mathbf{d}) = \boldsymbol{\chi}_{\mathrm{d}}(\mathbf{d})|_{\Gamma_{\mathrm{c}}^{(i,j)}}$ and $\boldsymbol{\omega}\left(\mathbf{d}^{(i)}\right) + \boldsymbol{\omega}\left(\mathbf{d}^{(j)}\right) = \boldsymbol{\chi}_{\omega}^{(i,j)}(\mathbf{d}) = \boldsymbol{\chi}_{\omega}(\mathbf{d})|_{\Gamma_{\mathrm{c}}^{(i,j)}}$ on $\Gamma_{\mathrm{c}}^{(i,j)}$, $i, j = 1, \ldots, n$ such that,

$$a(\mathbf{h}, \mathbf{d}) + \langle \boldsymbol{\chi}_{\mathrm{d}}(\mathbf{h}), \overline{\alpha}_{\mathrm{d}} \boldsymbol{\chi}_{\mathrm{d}}(\mathbf{d}) \rangle_{0,\Gamma_{\mathrm{c}}} + \langle \boldsymbol{\chi}_{\omega}(\mathbf{h}), \overline{\alpha}_{\omega} \boldsymbol{\chi}_{\omega}(\mathbf{d}) \rangle_{0,\Gamma_{\mathrm{c}}} = L(\mathbf{h}) , \tag{19}$$

for all $\mathbf{h} \in \mathfrak{L}$ with $\mathbf{h}^{(i)} - \mathbf{h}^{(j)} = \boldsymbol{\chi}_{\mathrm{d}}^{(i,j)}(\mathbf{h}) = \boldsymbol{\chi}_{\mathrm{d}}(\mathbf{h})|_{\Gamma_{\mathrm{c}}^{(i,j)}}$ and $\boldsymbol{\omega}\left(\mathbf{h}^{(i)}\right) + \boldsymbol{\omega}\left(\mathbf{h}^{(j)}\right) = \boldsymbol{\chi}_{\omega}^{(i,j)}(\mathbf{h}) = \boldsymbol{\chi}_{\omega}(\mathbf{h})|_{\Gamma_{\mathrm{c}}^{(i,j)}}$ on $\Gamma_{\mathrm{c}}^{(i,j)}$, $i, j = 1, \ldots, n$. Additionally, parameters $\overline{\alpha}_{\mathrm{d}}$ and $\overline{\alpha}_{\omega}$ which appear in Eq. (19) stand for the penalty factors for the displacement and the rotation field over the interface Γ_{c}, respectively, and they are strictly positive. Their distribution depends on the interface discretizations and a rule for their choice is proposed in Sect. 4. The penalty form $a_{\mathrm{p}} : \mathfrak{L} \times \mathfrak{L} \to \mathbb{R}$ governing variational formulation (19) is given by,

$$a_{\mathrm{p}}(\mathbf{h}, \mathbf{d}) := a(\mathbf{h}, \mathbf{d}) + \langle \boldsymbol{\chi}_{\mathrm{d}}(\mathbf{h}), \overline{\alpha}_{\mathrm{d}} \boldsymbol{\chi}_{\mathrm{d}}(\mathbf{d}) \rangle_{0,\Gamma_{\mathrm{c}}} + \langle \boldsymbol{\chi}_{\omega}(\mathbf{h}), \overline{\alpha}_{\omega} \boldsymbol{\chi}_{\omega}(\mathbf{d}) \rangle_{0,\Gamma_{\mathrm{c}}} , \tag{20}$$

for all functions $\mathbf{h}, \mathbf{d} \in \mathfrak{L}$. For the linear kinematics case, see Eqs. (8a) and (8b), form (20) is bilinear, by construction symmetric, bounded and coercive in $\mathfrak{L}$ which renders the formulation in general stable, meaning that one unique solution to problem (19) exists. However, this solution is not the same as for the strong form of the problem (10a), (10b), (10c), (10d), (10e), (10f), (10g), (10h) and (10i) for finite penalty parameters $\overline{\alpha}_{\mathrm{d}}$ and $\overline{\alpha}_{\omega}$, see [12]. This is the reason why the Penalty method is often called *variationally inconsistent*. The symmetry of the bilinear form (20) is obvious. Boundedness of a_{p} is easy to be shown using the *Cauchy-Schwarz* inequality in $L^2(\Gamma_{\mathrm{c}})$ space for the interface inner products. Coercivity of a_{p} follows from the positivity of the penalty parameters along the interface Γ_{c}. Numerical issues occurring due to the size of the penalty factors are discussed in Sect. 3.

2.4 The Weak Form of the Lagrange Multipliers Approach

The establishment of a weak formulation for the coupled system using the *Lagrange Multipliers* method demands two independent Lagrange Multipliers fields $\boldsymbol{\lambda}$ and $\boldsymbol{\zeta}$, i.e. one for each interface condition (10h) and (10i) respectively. In this way, the weak form of the coupled system using the Lagrange Multipliers writes [15]: Find $\mathbf{d} \in \mathfrak{L}$ with $\mathbf{d}^{(i)} - \mathbf{d}^{(j)} = \boldsymbol{\chi}_{\mathrm{d}}^{(i,j)}(\mathbf{d}) = \boldsymbol{\chi}_{\mathrm{d}}(\mathbf{d})|_{\Gamma_{\mathrm{c}}^{(i,j)}}$, $\boldsymbol{\omega}\left(\mathbf{d}^{(i)}\right) + \boldsymbol{\omega}\left(\mathbf{d}^{(j)}\right) = \boldsymbol{\chi}_{\omega}^{(i,j)}(\mathbf{d}) = \boldsymbol{\chi}_{\omega}(\mathbf{d})|_{\Gamma_{\mathrm{c}}^{(i,j)}}$ on $\Gamma_{\mathrm{c}}^{(i,j)}$, $i, j = 1, \ldots, n$ and $\boldsymbol{\lambda}, \boldsymbol{\zeta} \in \left(L^2(\Gamma_{\mathrm{c}})\right)^3$ such that,

$$a(\mathbf{h}, \mathbf{d}) + \langle \boldsymbol{\chi}_{\mathrm{d}}(\mathbf{h}), \boldsymbol{\lambda} \rangle_{0,\Gamma_{\mathrm{c}}} + \langle \boldsymbol{\chi}_{\omega}(\mathbf{h}), \boldsymbol{\zeta} \rangle_{0,\Gamma_{\mathrm{c}}} + \langle \boldsymbol{\mu}, \boldsymbol{\chi}_{\mathrm{d}}(\mathbf{d}) \rangle_{0,\Gamma_{\mathrm{c}}} + \langle \boldsymbol{\xi}, \boldsymbol{\chi}_{\omega}(\mathbf{d}) \rangle_{0,\Gamma_{\mathrm{c}}} = L(\mathbf{h}) \ , \tag{21}$$

for all $\mathbf{h} \in \mathfrak{L}$ with $\mathbf{h}^{(i)} - \mathbf{h}^{(j)} = \boldsymbol{\chi}_{\mathrm{d}}^{(i,j)}(\mathbf{h}) = \boldsymbol{\chi}_{\mathrm{d}}(\mathbf{h})|_{\Gamma_{\mathrm{c}}^{(i,j)}}$ and $\boldsymbol{\omega}\left(\mathbf{h}^{(i)}\right) + \boldsymbol{\omega}\left(\mathbf{h}^{(j)}\right) = \boldsymbol{\chi}_{\omega}^{(i,j)}(\mathbf{h}) = \boldsymbol{\chi}_{\omega}(\mathbf{h})|_{\Gamma_{\mathrm{c}}^{(i,j)}}$ on $\Gamma_{\mathrm{c}}^{(i,j)}$, $i, j = 1, \ldots, n$, for all $\boldsymbol{\mu}$ and for all $\boldsymbol{\xi}$ in $\left(L^2(\Gamma_{\mathrm{c}})\right)^3$, $\boldsymbol{\mu}$ and $\boldsymbol{\xi}$ being the Lagrange Multipliers test functions. From a physical viewpoint, the Lagrange Multipliers fields $\boldsymbol{\lambda}$ and $\boldsymbol{\zeta}$ represent the independent force traction and moment traction fields on the interface Γ_{c} which are necessary for the satisfaction of the interface force and moment equilibrium, respectively. Variational problem (21) is a saddle point problem since its weak solution is defined in $\mathcal{V}_{\mathrm{L}} = \mathcal{V} \times \left(L^2(\Gamma_{\mathrm{c}})\right)^3 \times \left(L^2(\Gamma_{\mathrm{c}})\right)^3$, thus having not to be necessarily existent or unique.

2.5 The Weak Form of the Augmented Lagrange Multipliers Approach

As already discussed in [15], the augmented *Lagrange Multipliers* method is a combination of the Lagrange Multipliers with the Penalty method, namely: Find

$\mathbf{d} \in \mathfrak{L}$ with $\mathbf{d}^{(i)} - \mathbf{d}^{(j)} = \boldsymbol{\chi}_{\mathrm{d}}^{(i,j)}(\mathbf{d}) = \boldsymbol{\chi}_{\mathrm{d}}(\mathbf{d})|_{\Gamma_{\mathrm{c}}^{(i,j)}}$ and $\boldsymbol{\omega}\left(\mathbf{d}^{(i)}\right) + \boldsymbol{\omega}\left(\mathbf{d}^{(j)}\right) = \boldsymbol{\chi}_{\omega}^{(i,j)}(\mathbf{d}) = \boldsymbol{\chi}_{\omega}(\mathbf{d})|_{\Gamma_{\mathrm{c}}^{(i,j)}}$ on $\Gamma_{\mathrm{c}}^{(i,j)}$, $i,j = 1,\ldots,n$ and $\boldsymbol{\lambda}, \boldsymbol{\zeta} \in \left(L^2(\Gamma_{\mathrm{c}})\right)^3$ such that,

$$\begin{aligned} a(\mathbf{h},\mathbf{d}) + \langle \boldsymbol{\chi}_{\mathrm{d}}(\mathbf{h}), \boldsymbol{\lambda} \rangle_{0,\Gamma_{\mathrm{c}}} + \langle \boldsymbol{\chi}_{\mathrm{d}}(\mathbf{h}), \overline{\alpha}_{\mathrm{d}} \boldsymbol{\chi}_{\mathrm{d}}(\mathbf{d}) \rangle_{0,\Gamma_{\mathrm{c}}} + \langle \boldsymbol{\chi}_{\omega}(\mathbf{h}), \boldsymbol{\zeta} \rangle_{0,\Gamma_{\mathrm{c}}} + \\ \langle \boldsymbol{\chi}_{\omega}(\mathbf{h}), \overline{\alpha}_{\omega} \boldsymbol{\chi}_{\omega}(\mathbf{d}) \rangle_{0,\Gamma_{\mathrm{c}}} + \langle \boldsymbol{\mu}, \boldsymbol{\chi}_{\mathrm{d}}(\mathbf{d}) \rangle_{0,\Gamma_{\mathrm{c}}} + \langle \boldsymbol{\xi}, \boldsymbol{\chi}_{\omega}(\mathbf{d}) \rangle_{0,\Gamma_{\mathrm{c}}} = L(\mathbf{h}) \ , \end{aligned} \tag{22}$$

for all $\mathbf{h} \in \mathfrak{L}$ with $\mathbf{h}^{(i)} - \mathbf{h}^{(j)} = \boldsymbol{\chi}_{\mathrm{d}}^{(i,j)}(\mathbf{h}) = \boldsymbol{\chi}_{\mathrm{d}}(\mathbf{h})|_{\Gamma_{\mathrm{c}}^{(i,j)}}$ and $\boldsymbol{\omega}\left(\mathbf{h}^{(i)}\right) + \boldsymbol{\omega}\left(\mathbf{h}^{(j)}\right) = \boldsymbol{\chi}_{\omega}^{(i,j)}(\mathbf{h}) = \boldsymbol{\chi}_{\omega}(\mathbf{h})|_{\Gamma_{\mathrm{c}}^{(i,j)}}$ on $\Gamma_{\mathrm{c}}^{(i,j)}$, $i,j = 1,\ldots,n$, for all $\boldsymbol{\mu}$ and for all $\boldsymbol{\xi}$ in $\left(L^2(\Gamma_{\mathrm{c}})\right)^3$. The weak solution of (22) is also defined in $\mathcal{V}_{\mathrm{L}}$, thus the augmented Lagrange Multipliers formulation belongs also to the family of saddle point problems. However, the penalty factors in this case do not have to be as large as in the Penalty method to achieve the same accuracy level in the interface conditions.

3 Discretization

3.1 Discrete Spaces

The numerical method used for the approximation of the given weak forms is IGA, as mentioned in Sect. 1. Within IGA the discrete spaces for the field approximation are taken from the geometry parametrization which is herein assumed to be the NURBS basis functions since they are dominant in CAGD. The convective coordinates $\left(\theta^1, \theta^2\right)$ introduced in Sect. 1 are thereafter designated by (ξ, η), complying with standard NURBS notations, see [2]. For each subdomain $\Omega^{(i)}$, $i = 1,\ldots,n$ and each interface $\Gamma_{\mathrm{c}}^{(k,l)}$, $(k,l) \in \mathcal{I}$, subdivisions $\mathcal{T}_{\mathrm{h}}^{(i)}$ and $\mathfrak{T}_{\mathrm{h}}^{(k,l)}$ are respectively chosen, which subdivide each $\Omega^{(i)}$ and $\Gamma_{\mathrm{c}}^{(k,l)}$ into isogeometric finite elements $\mathcal{T}_j^{(i)}$, $j = 1,\ldots,m^{(i)} \in \mathbb{N}$ and $\mathfrak{T}_j^{(k,l)}$, $j = 1,\ldots,m_{\lambda}^{(k,l)} \in \mathbb{N}$, respectively, for which the following relations hold,

$$\mathcal{T}_j^{(i)} \cap \mathcal{T}_k^{(i)} = \emptyset \quad \forall \mathcal{T}_j^{(i)}, \mathcal{T}_k^{(i)} \in \mathcal{T}_{\mathrm{h}}^{(i)} \text{ with } j \neq k \ , \tag{23a}$$

$$\mathfrak{T}_j^{(k,l)} \cap \mathfrak{T}_r^{(k,l)} = \emptyset \quad \forall \mathfrak{T}_j^{(k,l)}, \mathfrak{T}_r^{(k,l)} \in \mathfrak{T}_{\mathrm{h}}^{(k,l)} \text{ with } j \neq r \ , \tag{23b}$$

$$\bigcup_{j=1}^{m^{(i)}} \overline{\mathcal{T}_j^{(i)}} = \overline{\Omega^{(i)}} \ , \tag{23c}$$

$$\bigcup_{j=1}^{m_{\lambda}^{(k,l)}} \overline{\mathfrak{T}_j^{(k,l)}} = \overline{\Gamma_{\mathrm{c}}^{(k,l)}} \ , \tag{23d}$$

In the context of IGA the finite dimensional subspaces $\mathfrak{L}_\mathrm{h} \subset \mathfrak{L}$ and $\mathcal{L}_\mathrm{h} \subset \left(L^2\left(\Gamma_\mathrm{c}\right)\right)^3$ are chosen as,

$$\mathfrak{L}_\mathrm{h} = \left\{ \mathbf{h}_\mathrm{h} \in \mathfrak{L} \,\middle|\, \mathbf{h}_\mathrm{h} \in \left(\mathcal{R}^{p,q}\left(\mathcal{T}_j^{(i)}\right)\right)^3 \;\forall \mathcal{T}_j^{(i)} \in \mathcal{T}_\mathrm{h}^{(i)} \right\} , \tag{24a}$$

$$\mathcal{L}_\mathrm{h} = \left\{ \boldsymbol{\lambda}_\mathrm{h} \in \left(L^2\left(\Gamma_\mathrm{c}\right)\right)^3 \,\middle|\, \boldsymbol{\lambda}_\mathrm{h} \in \left(\mathcal{P}^0\left(\mathfrak{T}_j^{(k,l)}\right)\right)^3 \;\forall \mathfrak{T}_j^{(k,l)} \in \mathfrak{T}_\mathrm{h}^{(k,l)} \right\} , \tag{24b}$$

$\mathcal{R}^{p,q}\left(\mathcal{T}_j^{(i)}\right)$ being the space of all rational polynomials of polynomial orders p and q in the parametric image of $\mathcal{T}_j^{(i)}$. This parametric image is a rectangular domain $(\xi,\eta) \in \Xi^{(i)} \times H^{(i)} = \overline{\Omega_0^{(i)}}$, where $\overline{\Omega_0^{(i)}}$ is the image of $\overline{\Omega^{(i)}}$ under the inverse map $\mathbf{S}^{-1}$, within classical NURBS based isogeometric analysis, and $\Xi^{(i)}, H^{(i)}$ stand for the underlying knot vectors of patch $\Omega^{(i)}$, see [2] for more details. Additionally, the aforementioned rational polynomials are C^{p-m_k}-continuous across the elements within each patch $\Omega^{(i)}$, where p stands for the polynomial order of the basis crossing the element boundary and m_k the repetition of the knot at the element boundary, see [27] for more details. Equivalently, $\mathcal{P}^0\left(\mathfrak{T}_j^{(k,l)}\right)$ stands for the space of all constant functions in the parametric image of $\mathfrak{T}_j^{(k,l)}$. Assume that $\Phi^{(i)} = \left(\boldsymbol{\phi}_j^{(i)}\right)_j, j = 1,\ldots,\dim\left(\bigcup_{k=1}^{m^{(i)}} \mathcal{R}^{p,q}\left(\mathcal{T}_k^{(i)}\right)\right)^3$ and $\Psi^{(k,l)} = \left(\boldsymbol{\psi}_j^{(k,l)}\right)_j$, $j = 1,\ldots,\dim\left(\bigcup_{r=1}^{m_\lambda^{(k,l)}} \mathcal{P}^0\left(\mathfrak{T}_r^{(k,l)}\right)\right)^3 = 3m_\lambda^{(k,l)}$ are the set of basis functions at each subdomain $\Omega^{(i)}$ and each interface $\Gamma_\mathrm{c}^{(k,l)}$, respectively. Notation $\bigcup_{k=1}^{m^{(i)}} \mathcal{R}^{p,q}\left(\mathcal{T}_k^{(i)}\right)$ is herein abused for the space that the NURBS basis functions of arbitrary inter-element continuity span at each patch $\Omega^{(i)}$. In this way, a basis can be formed for the discrete spaces $\mathfrak{L}_\mathrm{h}$ and $\mathcal{L}_\mathrm{h}$ as $\Phi = \bigcup_{i=1}^{n} \Phi^{(i)} = \left(\boldsymbol{\phi}_j\right)_j, j = 1,\ldots,\dim\Phi$ and $\Psi = \bigcup_{(k,l)\in\mathcal{I}} \Psi^{(k,l)} = \left(\boldsymbol{\psi}_j\right)_j, j = 1,\ldots,\dim\Psi$, respectively. These bases are in turn discontinuous across each patch $\partial\Omega^{(i)}$ and interface $\partial\Gamma_\mathrm{c}^{(k,l)}$ boundary. Then, for any $\mathbf{h}_\mathrm{h} \in \mathfrak{L}_\mathrm{h}$ and any $\boldsymbol{\lambda}_\mathrm{h} \in \mathcal{L}_\mathrm{h}$ there exist reals $\hat{h}_i$ and $\hat{\lambda}_i$ such that,

$$\mathbf{h}_\mathrm{h} = \sum_{i=1}^{\dim\Phi} \hat{h}_i \boldsymbol{\phi}_i \quad \text{and} \quad \boldsymbol{\lambda}_\mathrm{h} = \sum_{i=1}^{\dim\Psi} \hat{\lambda}_i \boldsymbol{\psi}_i \,. \tag{25}$$

Important is to note that since $\mathfrak{L}_\mathrm{h}$ represents the discrete admissible space for the displacement field, the multiplicity of the knots within each patch $\Omega^{(i)}$ must be chosen smaller than the corresponding polynomial order of the basis such that the basis functions are at least one time continuously differentiable and hence the bending strain is well defined, see Eqs. (7b) and (8b). The same restriction does not apply for the discrete admissible space $\mathcal{L}_\mathrm{h}$ corresponding to the Lagrange Multipliers fields, and therefore discontinuous across the elements $\mathfrak{T}_j^{(k,l)}$ functions are chosen for the Lagrange Multipliers discretization. For all the forthcoming

numerical examples in Sect. 4, the multiplicity of the interior to the patches knots is chosen equal to one.

3.2 Discrete Equation System of the Decoupled Problem

Subsequently, the *Bubnov-Galerkin* approximation is employed, that is both the test and the trial spaces are projected onto the same finite dimensional subspace, as those are defined in (24a) and (24b). Then, projection of the weak form corresponding to the decoupled system (11) onto $\mathfrak{L}_{\mathrm{h}} \times \mathfrak{L}_{\mathrm{h}}$ yields the following discrete equation system for the geometrically linear setting,

$$\begin{bmatrix} \mathbf{K}^{(1)} & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{K}^{(n)} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{d}}^{(1)} \\ \vdots \\ \hat{\mathbf{d}}^{(n)} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{F}}^{(1)} \\ \vdots \\ \hat{\mathbf{F}}^{(n)} \end{bmatrix}, \tag{26}$$

where $\mathbf{K}^{(i)}$, $\hat{\mathbf{F}}^{(i)}$ and $\hat{\mathbf{d}}^{(i)}$ stand for the stiffness matrix with entries $K_{jk}^{(i)} = a\left(\boldsymbol{\phi}_j^{(i)}, \boldsymbol{\phi}_k^{(i)}\right)$, the load vector with entries $F_j^{(i)} = L\left(\boldsymbol{\phi}_j^{(i)}\right)$ and the discrete Control Point solution vector at each patch $\Omega^{(i)}$, respectively. For the geometrically nonlinear setting, the $\hat{i}$-th Newton-Rapshon iteration consists in the solution of the following linear system,

$$\begin{bmatrix} \mathbf{K}_{\mathrm{T}}^{(1)}\left(\hat{\mathbf{d}}_{\hat{i}}^{(1)}\right) & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{K}_{\mathrm{T}}^{(n)}\left(\hat{\mathbf{d}}_{\hat{i}}^{(n)}\right) \end{bmatrix} \begin{bmatrix} \delta\hat{\mathbf{d}}_{\hat{i}}^{(1)} \\ \vdots \\ \delta\hat{\mathbf{d}}_{\hat{i}}^{(n)} \end{bmatrix} = - \begin{bmatrix} \mathbf{R}^{(1)}\left(\hat{\mathbf{d}}_{\hat{i}}^{(1)}\right) \\ \vdots \\ \mathbf{R}^{(n)}\left(\hat{\mathbf{d}}_{\hat{i}}^{(i)}\right) \end{bmatrix}, \tag{27}$$

where $\mathbf{K}_{\mathrm{T}}^{(i)}$, $\delta\hat{\mathbf{d}}_{\hat{i}}^{(i)} = \hat{\mathbf{d}}_{\hat{i}+1}^{(i)} - \hat{\mathbf{d}}_{\hat{i}}^{(i)}$ and $\mathbf{R}^{(i)}\left(\hat{\mathbf{d}}_{\hat{i}}^{(i)}\right) = \mathbf{K}\left(\hat{\mathbf{d}}_{\hat{i}}^{(i)}\right)\hat{\mathbf{d}}_{\hat{i}}^{(i)} - \hat{\mathbf{F}}^{(i)}$ denote the tangent stiffness matrix, the Control Point displacement increment and the residual in patch $\Omega^{(i)}$, respectively.

3.3 Discrete Equation System of the Penalty Approach

Projection of the weak form corresponding to the Penalty approach (19) onto $\mathfrak{L}_{\mathrm{h}} \times \mathfrak{L}_{\mathrm{h}}$ results into the following discrete equation system for the linear setting,

$$\begin{bmatrix} \mathbf{K}^{(1)} + \mathbf{K}_{\mathrm{p}}^{(1)} & \cdots & \mathbf{C}_{\mathrm{p}}^{(1,n)} \\ \vdots & \ddots & \vdots \\ \mathbf{C}_{\mathrm{p}}^{(n,1)} & \cdots & \mathbf{K}^{(n)} + \mathbf{K}_{\mathrm{p}}^{(n)} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{d}}^{(1)} \\ \vdots \\ \hat{\mathbf{d}}^{(n)} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{F}}^{(1)} \\ \vdots \\ \hat{\mathbf{F}}^{(n)} \end{bmatrix}, \tag{28}$$

where the entries of the additional stiffness and Penalty coupling matrices $\mathbf{K}_{\mathrm{p}}^{(i)}$ and $\mathbf{C}_{\mathrm{p}}^{(i,j)}$ are respectively defined as,

$$K_{\mathrm{p},k,l}^{(i)} := \left\langle \boldsymbol{\phi}_k^{(i)}, \overline{\alpha}_{\mathrm{d}} \boldsymbol{\phi}_l^{(i)} \right\rangle_{0,\Gamma_{\mathrm{c}}} + \left\langle \boldsymbol{\omega}\left(\boldsymbol{\phi}_k^{(i)}\right), \overline{\alpha}_{\omega} \boldsymbol{\omega}\left(\boldsymbol{\phi}_l^{(i)}\right) \right\rangle_{0,\Gamma_{\mathrm{c}}} , \tag{29a}$$

$$C_{\mathrm{p},k,l}^{(i,j)} := -\left\langle \boldsymbol{\phi}_k^{(i)}, \overline{\alpha}_{\mathrm{d}} \boldsymbol{\phi}_l^{(j)} \right\rangle_{0,\Gamma_{\mathrm{c}}} + \left\langle \boldsymbol{\omega}\left(\boldsymbol{\phi}_k^{(i)}\right), \overline{\alpha}_{\omega} \boldsymbol{\omega}\left(\boldsymbol{\phi}_l^{(j)}\right) \right\rangle_{0,\Gamma_{\mathrm{c}}} . \tag{29b}$$

For the geometrically nonlinear problem, the Newton-Raphson subproblem at the $\hat{i}$-th iteration, defined in (27), is written as,

$$\begin{aligned} &\begin{bmatrix} \mathbf{K}_{\mathrm{T}}^{(1)}\left(\hat{\mathbf{d}}_{\hat{i}}^{(1)}\right) + \mathbf{K}_{\mathrm{p}}^{(1)} & \cdots & \mathbf{C}_{\mathrm{p}}^{(1,n)} \\ \vdots & \ddots & \vdots \\ \mathbf{C}_{\mathrm{p}}^{(n,1)} & \cdots & \mathbf{K}_{\mathrm{T}}^{(n)}\left(\hat{\mathbf{d}}_{\hat{i}}^{(n)}\right) + \mathbf{K}_{\mathrm{p}}^{(n)} \end{bmatrix} \begin{bmatrix} \delta\hat{\mathbf{d}}_{\hat{i}}^{(1)} \\ \vdots \\ \delta\hat{\mathbf{d}}_{\hat{i}}^{(n)} \end{bmatrix} = \\ &\quad - \begin{bmatrix} \mathbf{R}^{(1)}\left(\hat{\mathbf{d}}_{\hat{i}}^{(1)}\right) \\ \vdots \\ \mathbf{R}^{(n)}\left(\hat{\mathbf{d}}_{\hat{i}}^{(n)}\right) \end{bmatrix} - \begin{bmatrix} \mathbf{K}_{\mathrm{p}}^{(1)} & \cdots & \mathbf{C}_{\mathrm{p}}^{(1,n)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{C}_{\mathrm{p}}^{(n,1)} & \cdots & \mathbf{K}_{\mathrm{p}}^{(n)} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{d}}_{\hat{i}}^{(1)} \\ \vdots \\ \hat{\mathbf{d}}_{\hat{i}}^{(n)} \end{bmatrix} . \end{aligned} \tag{30}$$

Note also that the additional interface terms which show up in the Penalty formulation (19) are linear with respect to the displacement field, thus making the extension from the geometrically linear to the geometrically nonlinear problem straightforward. It must be emphasized that the resulting system is positive definite given that the penalty parameters do not destroy its conditioning, and therefore must be carefully chosen. A rule for their choice is given in Sect. 4.1 where an assessment of this choice with respect to the domain and interface errors in the L^2-norm is in addition provided.

3.4 Discrete Equation System of the Lagrange Multipliers Approach

As discussed in Sect. 2.4 two independent Lagrange Multipliers fields are used for the corresponding variational form (21). Therefore, also two independent discrete Lagrange Multipliers spaces, one for the Lagrange Multipliers field representing the interface traction forces and one for the Lagrange Multipliers field representing the interface traction moments are chosen as $\mathcal{L}_{\mathrm{h}}^{\lambda}$ and $\mathcal{L}_{\mathrm{h}}^{\zeta}$, respectively, following (24b). Let also the set of basis functions for each space $\mathcal{L}_{\mathrm{h}}^{\lambda}$ and $\mathcal{L}_{\mathrm{h}}^{\zeta}$ be $\Psi^{\lambda} = \left(\boldsymbol{\psi}_j^{\lambda}\right)_j, j = 1, \ldots, \dim \Psi^{\lambda}$ and $\Psi^{\zeta} = \left(\boldsymbol{\psi}_j^{\zeta}\right)_j, j = 1, \ldots, \dim \Psi^{\zeta}$, respectively. Then, projection of the Lagrange Multipliers weak form (21) onto $\left(\mathfrak{L}_{\mathrm{h}} \times \mathcal{L}_{\mathrm{h}}^{\lambda} \times \mathcal{L}_{\mathrm{h}}^{\zeta}\right) \times \left(\mathfrak{L}_{\mathrm{h}} \times \mathcal{L}_{\mathrm{h}}^{\lambda} \times \mathcal{L}_{\mathrm{h}}^{\zeta}\right)$

provides us with the following equation system for the geometrically linear setting,

$$\begin{bmatrix} \mathbf{K}^{(1)} & \cdots & \mathbf{0} & \boldsymbol{\Lambda}^{(1)} & \boldsymbol{Z}^{(1)} \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ \mathbf{0} & \cdots & \mathbf{K}^{(n)} & \boldsymbol{\Lambda}^{(n)} & \boldsymbol{Z}^{(n)} \\ \left(\boldsymbol{\Lambda}^{(1)}\right)^{\mathrm{T}} & \cdots & \left(\boldsymbol{\Lambda}^{(n)}\right)^{\mathrm{T}} & \mathbf{0} & \mathbf{0} \\ \left(\boldsymbol{Z}^{(1)}\right)^{\mathrm{T}} & \cdots & \left(\boldsymbol{Z}^{(n)}\right)^{\mathrm{T}} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{d}}^{(1)} \\ \vdots \\ \hat{\mathbf{d}}^{(n)} \\ \hat{\boldsymbol{\lambda}} \\ \hat{\boldsymbol{\xi}} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{F}}^{(1)} \\ \vdots \\ \hat{\mathbf{F}}^{(n)} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} , \tag{31}$$

where the entries of matrices $\boldsymbol{\Lambda}^{(i)}$ and $\boldsymbol{Z}^{(i)}$ are given by,

$$\Lambda_{k,l}^{(i)} := \sum_{j=1}^{n} (-1)^{\alpha^{(i,j)}} \left\langle \boldsymbol{\phi}_k^{(i)}, \boldsymbol{\psi}_l^{\lambda} \right\rangle_{0,\Gamma_{\mathrm{c}}^{(i,j)}} , \tag{32a}$$

$$Z_{k,l}^{(i)} := \sum_{j=1}^{n} \left\langle \boldsymbol{\omega}\left(\boldsymbol{\phi}_k^{(i)}\right), \boldsymbol{\psi}_l^{\zeta} \right\rangle_{0,\Gamma_{\mathrm{c}}^{(i,j)}} , \tag{32b}$$

with $\alpha^{(i,j)} = 0$ if $i < j$ or $\alpha^{(i,j)} = 1$ if $j < i$. Additionally, $\hat{\boldsymbol{\lambda}}$ and $\hat{\boldsymbol{\xi}}$ denote the discrete solution vectors for both Lagrange Multiplier fields. Due to the linearity of the weak form on the Lagrange Multipliers fields, the extension to the nonlinear setting is also in this case straightforward. Thus, the $\hat{i}$-th Newton-Raphson iteration in this case writes,

$$\begin{aligned} &\begin{bmatrix} \mathbf{K}_{\mathrm{T}}^{(1)}\left(\hat{\mathbf{d}}_{\hat{i}}^{(1)}\right) & \cdots & \mathbf{0} & \boldsymbol{\Lambda}^{(1)} & \boldsymbol{Z}^{(1)} \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ \mathbf{0} & \cdots & \mathbf{K}_{\mathrm{T}}^{(n)}\left(\hat{\mathbf{d}}_{\hat{i}}^{(n)}\right) & \boldsymbol{\Lambda}^{(n)} & \boldsymbol{Z}^{(n)} \\ \left(\boldsymbol{\Lambda}^{(1)}\right)^{\mathrm{T}} & \cdots & \left(\boldsymbol{\Lambda}^{(n)}\right)^{\mathrm{T}} & \mathbf{0} & \mathbf{0} \\ \left(\boldsymbol{Z}^{(1)}\right)^{\mathrm{T}} & \cdots & \left(\boldsymbol{Z}^{(n)}\right)^{\mathrm{T}} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \delta\hat{\mathbf{d}}_{\hat{i}}^{(1)} \\ \vdots \\ \delta\hat{\mathbf{d}}_{\hat{i}}^{(n)} \\ \delta\hat{\boldsymbol{\lambda}}_{\hat{i}} \\ \delta\hat{\boldsymbol{\xi}}_{\hat{i}} \end{bmatrix} = \\ &- \begin{bmatrix} \mathbf{R}^{(1)}\left(\hat{\mathbf{d}}_{\hat{i}}^{(1)}\right) \\ \vdots \\ \mathbf{R}^{(n)}\left(\hat{\mathbf{d}}_{\hat{i}}^{(n)}\right) \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} \mathbf{0} & \cdots & \mathbf{0} & \boldsymbol{\Lambda}^{(1)} & \boldsymbol{Z}^{(1)} \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} & \boldsymbol{\Lambda}^{(n)} & \boldsymbol{Z}^{(n)} \\ \left(\boldsymbol{\Lambda}^{(1)}\right)^{\mathrm{T}} & \cdots & \left(\boldsymbol{\Lambda}^{(n)}\right)^{\mathrm{T}} & \mathbf{0} & \mathbf{0} \\ \left(\boldsymbol{Z}^{(1)}\right)^{\mathrm{T}} & \cdots & \left(\boldsymbol{Z}^{(n)}\right)^{\mathrm{T}} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{d}}_{\hat{i}}^{(1)} \\ \vdots \\ \hat{\mathbf{d}}_{\hat{i}}^{(n)} \\ \hat{\boldsymbol{\lambda}}_{\hat{i}} \\ \hat{\boldsymbol{\xi}}_{\hat{i}} \end{bmatrix} . \end{aligned} \tag{33}$$

It must be remarked that the discrete spaces chosen for the displacement and the Lagrange Multipliers fields must fulfil the so-called Ladyzenskaja-Babuska-Brezzi condition, alternatively known as *inf-sup* condition, so that one unique solution to the finite dimensional problem can be guaranteed. This is not a trivial problem and

an improper choice of the discrete spaces renders the discrete equation system ill-conditioned or even indefinite. The numerical investigations suggest the following way of choosing the discretization for the Lagrange Multipliers fields: As mentioned in Sect. 3.1 both Lagrange Multipliers fields are discretized with constant, discontinuous across the elements functions. In addition, field $\boldsymbol{\lambda}|_{\Gamma_c^{(k,l)}}$ is discretized using as many elements $n_\lambda^{(k,l)}$ as the coarsest of the subdomains $\Omega^{(k)}$ and $\Omega^{(l)}$ has and field $\boldsymbol{\zeta}|_{\Gamma_c^{(k,l)}}$ is discretized with $n_\zeta^{(k,l)} = \left\lceil \frac{n_\lambda^{(k,l)}}{2} \right\rceil$. These rules have only numerically proven to produce in most of the cases stable and not ill-conditioned discrete equation systems, thus further mathematical investigations are urged for this case.

3.5 Discrete Equation System of the Augmented Lagrange Multipliers Approach

As mentioned in Sect. 2.5, the augmented Lagrange Multipliers method is a combination of the Penalty and the Lagrange Multipliers method. Therefore, the discrete equation systems for both the geometrically linear and geometrically nonlinear problems write,

$$\begin{bmatrix} \mathbf{K}^{(1)} + \mathbf{K}_\mathrm{p}^{(1)} & \cdots & \mathbf{C}_\mathrm{p}^{(1,n)} & \boldsymbol{\Lambda}^{(1)} & \boldsymbol{Z}^{(1)} \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ \mathbf{C}_\mathrm{p}^{(n,1)} & \cdots & \mathbf{K}^{(n)} + \mathbf{K}_\mathrm{p}^{(n)} & \boldsymbol{\Lambda}^{(n)} & \boldsymbol{Z}^{(n)} \\ \left(\boldsymbol{\Lambda}^{(1)}\right)^\mathrm{T} & \cdots & \left(\boldsymbol{\Lambda}^{(n)}\right)^\mathrm{T} & \mathbf{0} & \mathbf{0} \\ \left(\boldsymbol{Z}^{(1)}\right)^\mathrm{T} & \cdots & \left(\boldsymbol{Z}^{(n)}\right)^\mathrm{T} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{d}}^{(1)} \\ \vdots \\ \hat{\mathbf{d}}^{(n)} \\ \hat{\boldsymbol{\lambda}} \\ \hat{\boldsymbol{\zeta}} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{F}}^{(1)} \\ \vdots \\ \hat{\mathbf{F}}^{(n)} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \tag{34}$$

$$\begin{aligned} &\begin{bmatrix} \mathbf{K}_\mathrm{T}^{(1)}\left(\hat{\mathbf{d}}_{\hat{i}}^{(1)}\right) + \mathbf{K}_\mathrm{p}^{(1)} & \cdots & \mathbf{C}_\mathrm{p}^{(1,n)} & \boldsymbol{\Lambda}^{(1)} & \boldsymbol{Z}^{(1)} \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ \mathbf{C}_\mathrm{p}^{(n,1)} & \cdots & \mathbf{K}_\mathrm{T}^{(n)}\left(\hat{\mathbf{d}}_{\hat{i}}^{(n)}\right) + \mathbf{K}_\mathrm{p}^{(n)} & \boldsymbol{\Lambda}^{(n)} & \boldsymbol{Z}^{(n)} \\ \left(\boldsymbol{\Lambda}^{(1)}\right)^\mathrm{T} & \cdots & \left(\boldsymbol{\Lambda}^{(n)}\right)^\mathrm{T} & \mathbf{0} & \mathbf{0} \\ \left(\boldsymbol{Z}^{(1)}\right)^\mathrm{T} & \cdots & \left(\boldsymbol{Z}^{(n)}\right)^\mathrm{T} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \delta\hat{\mathbf{d}}_{\hat{i}}^{(1)} \\ \vdots \\ \delta\hat{\mathbf{d}}_{\hat{i}}^{(n)} \\ \delta\hat{\boldsymbol{\lambda}}_{\hat{i}} \\ \delta\hat{\boldsymbol{\zeta}}_{\hat{i}} \end{bmatrix} = \\ &- \begin{bmatrix} \mathbf{R}^{(1)}\left(\hat{\mathbf{d}}_{\hat{i}}^{(1)}\right) \\ \vdots \\ \mathbf{R}^{(n)}\left(\hat{\mathbf{d}}_{\hat{i}}^{(n)}\right) \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} \mathbf{K}_\mathrm{p}^{(1)} & \cdots & \mathbf{C}_\mathrm{p}^{(1,n)} & \boldsymbol{\Lambda}^{(1)} & \boldsymbol{Z}^{(1)} \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ \mathbf{C}_\mathrm{p}^{(n,1)} & \cdots & \mathbf{K}_\mathrm{p}^{(n)} & \boldsymbol{\Lambda}^{(n)} & \boldsymbol{Z}^{(n)} \\ \left(\boldsymbol{\Lambda}^{(1)}\right)^\mathrm{T} & \cdots & \left(\boldsymbol{\Lambda}^{(n)}\right)^\mathrm{T} & \mathbf{0} & \mathbf{0} \\ \left(\boldsymbol{Z}^{(1)}\right)^\mathrm{T} & \cdots & \left(\boldsymbol{Z}^{(n)}\right)^\mathrm{T} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{d}}_{\hat{i}}^{(1)} \\ \vdots \\ \hat{\mathbf{d}}_{\hat{i}}^{(n)} \\ \hat{\boldsymbol{\lambda}}_{\hat{i}} \\ \hat{\boldsymbol{\zeta}}_{\hat{i}} \end{bmatrix}. \end{aligned} \tag{35}$$

As in the case of the Lagrange Multipliers method, the linear equation system in this case might also be indefinite, depending on whether the chosen discrete spaces fulfil the inf-sup condition or not. For the discretization of the Lagrange Multipliers fields, the same empirical rules as in Sect. 3.4 are employed.

4 Numerical Results

In this section, the DDMs are compared and systematically evaluated over two linear and two nonlinear problems. For the linear problems convergence studies over domain and interface L^2-norms are performed. The L^2-norm of a tensor $\boldsymbol{\tau} \in \mathfrak{S}^m$ over a set Ω is given by,

$$\|\boldsymbol{\tau}\|_{0,\Omega} := \left(\int_{\Omega} \sum_{i_1=1}^{m} \cdots \sum_{i_m=1}^{m} \tau_{i_1 \ldots i_m} \tau_{i_1 \ldots i_m} \, \mathrm{d}\Omega \right)^{\frac{1}{2}} . \tag{36}$$

On the other hand in what concerns the nonlinear problems, the corresponding load-displacement curves are given and compared against the corresponding results in [28]. The various quantities depicted in the following graphs are measured in *Systéme International d'Unités* (SI), if not otherwise stated.

4.1 Circular Plate Subject to Uniform Pressure Load

The first geometrically linear problem over which a convergence study for all the DDMs is performed is that of a circular plate with radius $\overline{R} = 50$ cm subject to uniform distributed load $\overline{p} = 0.01\,\mathrm{KN/m^2}$ at its surface. The problem is decomposed into five patches, see Fig. 2. An analytical solution in terms of

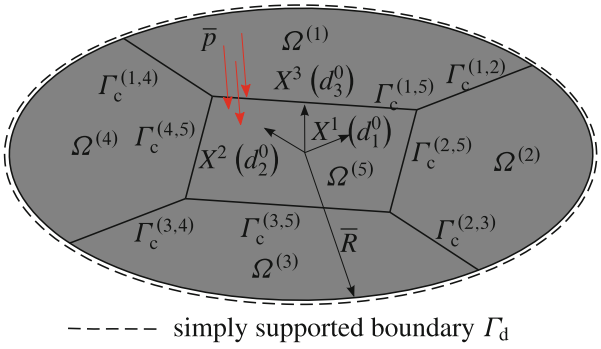


Fig. 2 Circular plate subject to uniform pressure load: problem placement

displacements for this problem exists [29],

$$\mathbf{d}\left(X^1, X^2\right) = -\frac{3\overline{p}\left(\overline{R}^2 - r^2\right)\left(1-\nu^2\right)}{16Eh^3}\left(\frac{5+\nu}{1+\nu}\overline{R}^2 - r^2\right)\mathbf{e}_3\,, \tag{37}$$

where $r^2 = \left(X^1\right)^2 + \left(X^2\right)^2 \leq \overline{R}^2$. The Young's modulus, Poisson's ratio and plate's thickness are chosen as $E = 1.8 \times 10^3\,\mathrm{KN/m^2}$, $\nu = 0.3$ and $h = 1\,\mathrm{cm}$, respectively. A convergence study is performed by refining patches $\Omega^{(1)}$, $\Omega^{(2)}$, $\Omega^{(3)}$, $\Omega^{(4)}$ and $\Omega^{(5)}$ with $\left\lceil \frac{5}{9}i \right\rceil^2$, $\left\lceil \frac{6}{9}i \right\rceil^2$, $\left\lceil \frac{4}{9}i \right\rceil^2$, $\left\lceil \frac{7}{9}i \right\rceil^2$ and i^2 biquadratic and bicubic elements, respectively, where $i = 4, \ldots, 40$ stands for the refinement level index. The isogeometric discretization of each patch is chosen equal in both surface parametric directions ξ and η. The resulting scaled deformation and the 2-norm of the displacement field $\|\mathbf{d}\|_2$ using the Penalty method, corresponding to refinement level $i = 8$ for the bicubic elements are shown in the set of Fig. 3. Within this study, the penalty parameters $\overline{\alpha}_\mathrm{d}$ and $\overline{\alpha}_\omega$ are chosen constant throughout the whole interface Γ_c, as a scaling of the matrix norms $\|\mathbf{D}_\mathrm{m}\|_\mathrm{F}$ and $\|\mathbf{D}_\mathrm{b}\|_\mathrm{F}$ corresponding to the material matrices for the membrane and the bending stiffness of the Kirchhoff-Love shell, respectively. In addition, this scaling is chosen as the inverse of the minimum element area size within the multipatch geometry, that is $h_{-1} = \left(\min_{\mathcal{T}_j \in \bigcup_{i=1}^5 \mathcal{T}_\mathrm{h}^{(i)}} \left|\mathcal{T}_j\right|\right)^{-1}$. For the case depicted in Fig. 3, $h_{-1} = 6.1169 \times 10^2\,\mathrm{m^{-2}}$ resulting in choosing the penalty parameters as $\overline{\alpha}_\mathrm{d} = 1.5729 \times 10^4\,\mathrm{KN/m^3}$ and $\overline{\alpha}_\omega = 0.1310\,\mathrm{KN/m}$. This choice is found to be optimal with respect to the domain error and interface displacement jump, for both the Penalty and the augmented Lagrange Multipliers methods, see Fig. 4. Then, in what concerns the refinement studies, the scaling related to the penalty parameters is chosen as 10^4 and 10^2 for the Penalty and the augmented Lagrange Multipliers method, respectively. The refinement studies consist in biquadratic as well as bicubic elements, see set of Fig. 5. The Penalty method is reaching its threshold in the order of 10^{-4} and 10^{-8} for the relative domain error and the interface jump in the L^2-norm, respectively.

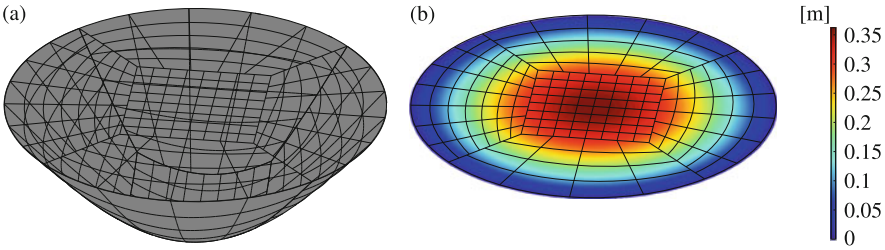


Fig. 3 Circular plate subject to uniform pressure load: postprocessing (Penalty). Setting corresponding to refinement level i=8. Patches $\Omega^{(1)}$, $\Omega^{(2)}$, $\Omega^{(3)}$, $\Omega^{(4)}$ and $\Omega^{(5)}$ modelled using 25, 36, 16, 49 and 64 bicubic elements, respectively. (**a**) Plate's deformation scaled by $\frac{3}{2}$. (**b**) The displacement field in 2-norm $\|\mathbf{d}\|_2$

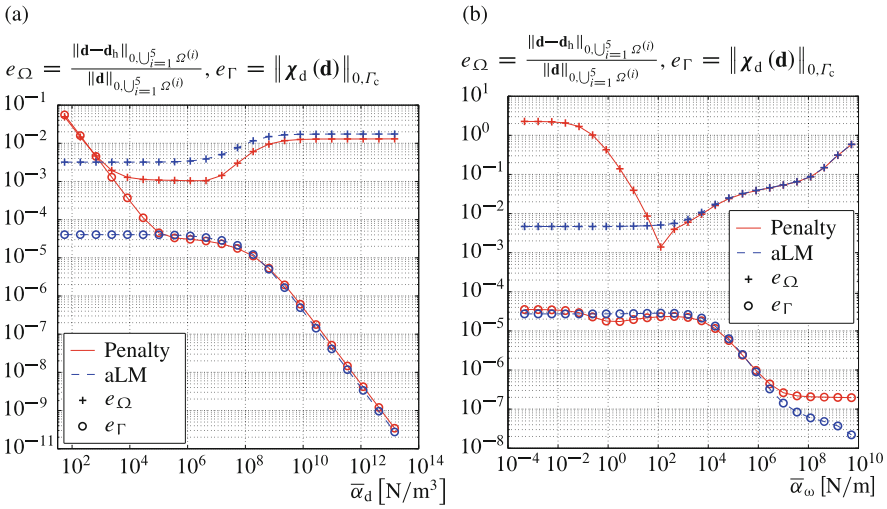


Fig. 4 Circular plate subject to uniform pressure load: penalty parameters. Relative domain error and interface displacement jump in the L^2-norm against the penalty parameters corresponding to refinement level i=8, see Fig. 3. (**a**) Error depending on $\overline{\alpha}_{\mathrm{d}}$ maintaining $\overline{\alpha}_{\omega} = 0.1310\,\mathrm{KN/m}$. (**b**) Error depending on $\overline{\alpha}_{\omega}$ maintaining $\overline{\alpha}_{\mathrm{d}} = 1.5729 \times 10^4\,\mathrm{KN/m^3}$

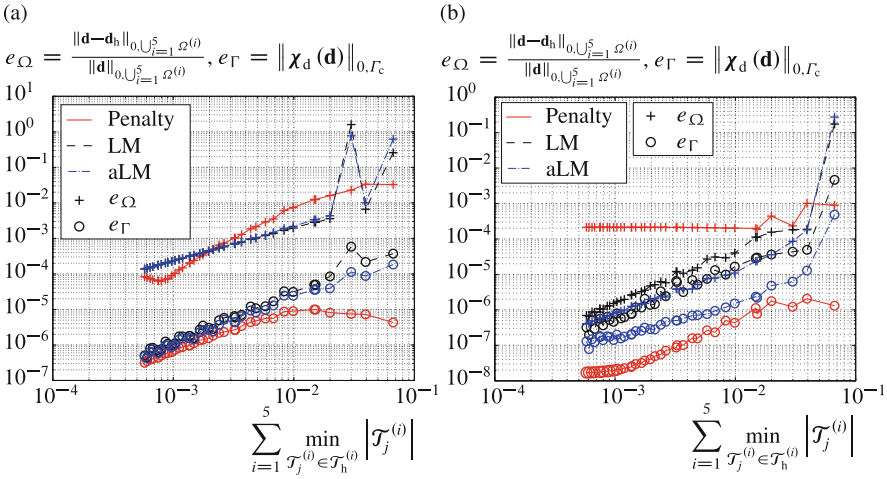


Fig. 5 Circular plate subject to uniform pressure load: convergence study. Relative domain error and interface displacement jump in the L^2-norm against the sum of the minimum element areas in the multipatch geometry. (**a**) Biquadratic basis. (**b**) Bicubic basis

Both Lagrange Multipliers methods are demonstrating uniform convergence with the augmented Lagrange Multipliers showing a slightly improved convergence compared to the Lagrange Multipliers method. That is due to the penalty factors whose magnitude is controlling the level of satisfaction to the interface constraints

(10h) and (10i). Important for the augmented Lagrange Multipliers method is not to use penalization parameters much larger than for the Penalty method, cause in this case the method would be dominated by the penalty terms destroying the uniform convergence expected from a Lagrange Multipliers method. The latter can be observed in Fig. 4 where although for small penalty parameters the error levels obtained by the Lagrange Multipliers are lower than the corresponding Penalty ones, when the penalization parameters are increased both methods yield essentially the same results, meaning that the Lagrange Multipliers play no more role.

4.2 Scordelis-Lo Roof

The second geometrically linear example which is employed for the comparison of the DDMs is the Scordelis-Lo roof shell [30]. Its Young's modulus, Poisson's ratio and thickness are chosen equal to $E = 4.32 \times 10^5\,\mathrm{KN/m^2}$, $\nu = 0$ and $h = 25\,\mathrm{cm}$, respectively. Its curvature in the X^1-direction is zero with length equal to $L = 50\,\mathrm{m}$, whereas in the X^2-X^3 plane it is representing a part of a cylinder with radius and arc length size equal to $\overline{R} = 25\,\mathrm{m}$ and $\overline{\phi}_0 = \frac{4\pi}{9}$, respectively. At $X^1 = -\frac{L}{2}$ and $X^2 = \frac{L}{2}$ the shell is supported with diaphragms. The latter means that along the Dirichlet boundary Γ_d the motion of the shell in X^1 and X^2 directions is confined. The problem is confronted in its original geometric setting without reducing it into one quarter by apply the necessary symmetry conditions. Therefore, the corner $\left(X^1, X^2, X^3\right) \equiv \left(-\frac{L}{2}, -\overline{R}\sin\frac{\overline{\phi}_0}{2}, \overline{R}\cos\frac{\overline{\phi}_0}{2}\right)$ has to be fixed so that the indeterminacy of the problem is eliminated. In addition, a uniform constant distributed load $\overline{p} = 9 \times 10^{-2}\,\mathrm{KN/m^2}$ is applied on the shell's surface in the X^3-direction. Then, the problem is decomposed symmetrically into four patches, see Fig. 6.

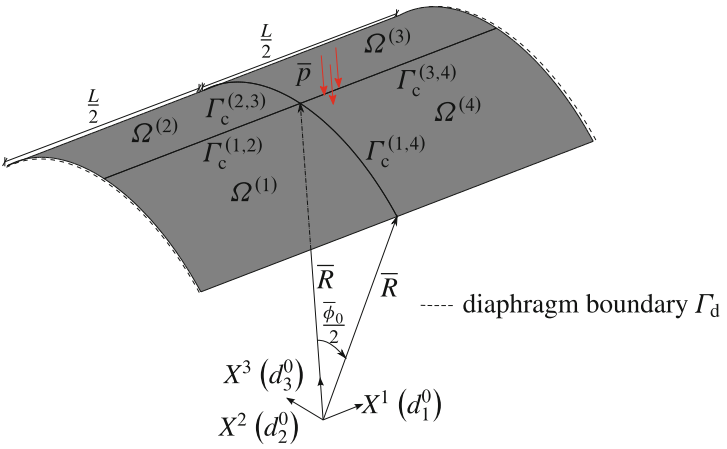


Fig. 6 Scordelis-Lo roof: problem placement

A convergence study is performed for low and high polynomial bases of the patches. As reference solution for the displacement vector $\mathbf{d}$, force $\mathbf{n}(\mathbf{d})$ and moment $\mathbf{m}(\mathbf{d})$ tensors fields, it is considered the solution of the same problem modelled with a single patch attaining a bidecic basis, sixty and forty elements in the X^1 and $X^2 \sin\overline{\phi} + X^3 \cos\overline{\phi}$ direction, respectively. The low polynomial order case consists in patch pairs $\Omega^{(1)}$, $\Omega^{(3)}$ and $\Omega^{(2)}$, $\Omega^{(4)}$ being modelled with bicubic and biquadratic basis, respectively. On the other hand, the high polynomial order case represents the patch pair $\Omega^{(1)}$, $\Omega^{(3)}$ and $\Omega^{(2)}$, $\Omega^{(4)}$ modelling with biquartic and bicubic basis respectively. The refinement study is performed by refining patches $\Omega^{(1)}$, $\Omega^{(2)}$, $\Omega^{(3)}$ and $\Omega^{(4)}$ with $\lceil \frac{4}{3}i \rceil \times \lceil \frac{4}{3}i \rceil$, $\left(\lceil \frac{7}{4}i \rceil + 2\right) \times \left(\lceil \frac{7}{4}i \rceil + 3\right)$, $\lceil \frac{6}{4}i \rceil \times \left(\lceil \frac{6}{4}i \rceil - 2\right)$ and $\left(\lceil \frac{8}{5}i \rceil + 4\right) \times \left(\lceil \frac{8}{5}i \rceil + 1\right)$ elements in ξ and η parametric directions, respectively, with $i = 1, \ldots, 26$. The parametric coordinates ξ and η, are chosen for all the patches to coincide with the longitudinal $X^1 \in \left[-\frac{L}{2}, \frac{L}{2}\right]$ and $\left(X^2 \sin\overline{\phi}, X^3 \cos\overline{\phi}\right)$ with $\overline{\phi} \in \left[-\frac{\overline{\phi}_0}{2}, \frac{\overline{\phi}_0}{2}\right]$ circumferential direction, respectively. Moreover, the refinement is chosen such that the resulting elements have a good aspect ratio as well as the interface parametrizations are kept highly non-conforming at each refinement level. The scaling of the penalty factors for the displacement and rotation penalization corresponding to the Penalty method are chosen as $10^3\,\mathrm{m}^{-2}$ and $0.25 \times 10^3\,\mathrm{m}^{-2}$, respectively. Accordingly, the scaling of the penalty factors for the displacement and rotation penalization corresponding to the augmented Lagrange Multipliers method are chosen as 1 and $0.25\,\mathrm{m}^{-2}$, respectively. Then, the seven times scaled deformation and the 2-norm of the rotation field obtained using the Lagrange Multipliers method and corresponding to refinement level $i = 5$ for the high polynomial order case can be seen in set of Fig. 7.

The convergence of the displacement field at each subdomain using the relative L^2-norm for the low and high polynomial order of the underlying bases can be seen in set of Fig. 8. More specifically, the Penalty method is clearly reaching its threshold at the relative displacement error order of 10^{-3} and 10^{-4} for patch pair $\Omega^{(1)}$, $\Omega^{(3)}$ and $\Omega^{(2)}$, $\Omega^{(4)}$, respectively.

In what concerns the Lagrange Multipliers methods, the resulting numerical displacement fields converge uniformly as the element area size goes to zero with

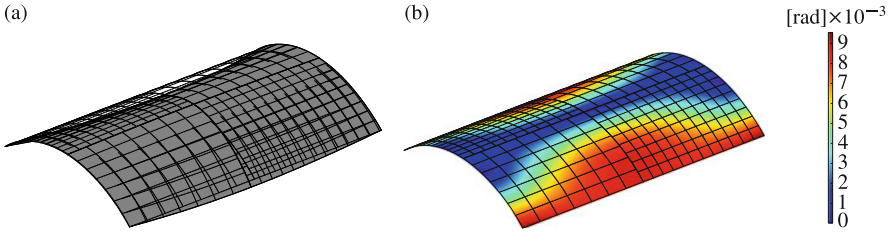


Fig. 7 Scordelis-Lo roof: postprocessing (Lagrange multipliers). Setting corresponding to refinement level i=5. Patches $\Omega^{(1)}$, $\Omega^{(2)}$, $\Omega^{(3)}$, $\Omega^{(4)}$ modelled using 49, 132, 48 and 108 elements, respectively, whereas pairs $\Omega^{(1)}$,$\Omega^{(3)}$ and $\Omega^{(2)}$,$\Omega^{(4)}$ have biquartic and bicubic bases, respectively. (**a**) Plate's deformation scaled by 7. (**b**) The rotation field in 2-norm $\|\boldsymbol{\omega}\|_2$

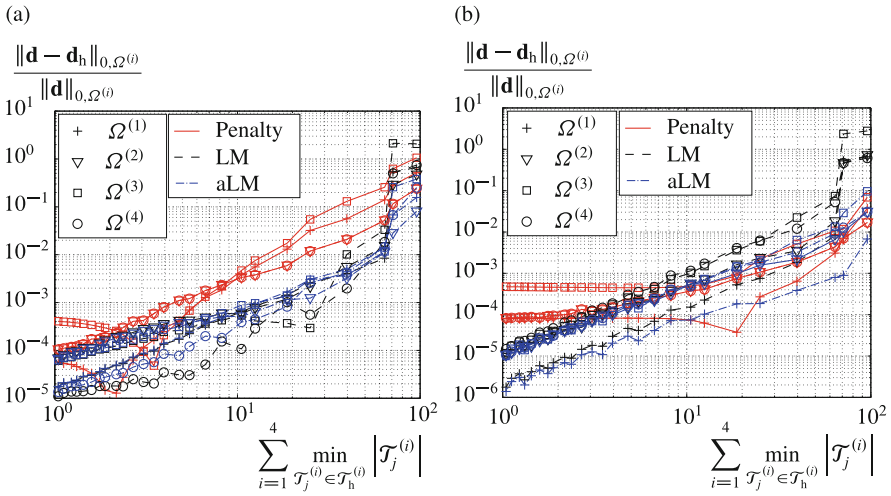


Fig. 8 Scordelis-Lo roof: convergence study. Relative displacement error in the L^2-norm at each subdomain against the sum of the minimum element areas for the multipatch geometry. (**a**) $\Omega^{(1)}$, $\Omega^{(3)}$ bicubic and $\Omega^{(2)}$, $\Omega^{(4)}$ biquadratic bases. (**b**) $\Omega^{(1)}$, $\Omega^{(3)}$ biquartic and $\Omega^{(2)}$, $\Omega^{(4)}$ bicubic bases

the augmented Lagrange Multipliers method demonstrating lower error levels in the coarse region of the isogeometric field discretization. On the other hand, both Lagrange Multipliers methods produce numerical fields whose relative errors are almost identical in the fine discretization region. This is explained by the fact that the penalty factors are kept constant with the refinement level while the Lagrange multipliers fields are successively refined and therefore becoming dominant as the element area size goes to zero.

As last, the convergence of the force and moment tensors, which are computed using Eqs. (6a) and (6b), is demonstrated for both polynomial order cases in the set of Figs. 9 and 10, respectively. Uniform convergence is observed for all but the Penalty method, which is reaching its threshold in the force tensor at the relative error of order 10^{-3} and 10^{-4} for the patch pairs $\Omega^{(2)}$, $\Omega^{(3)}$ and $\Omega^{(1)}$, $\Omega^{(4)}$, respectively, see Fig. 9b. Equivalently, the aforementioned method is reaching its threshold in the moment tensor at the relative error of order 10^{-2}, 10^{-3} and 10^{-4} for patches $\Omega^{(2)}$, $\Omega^{(4)}$ and pair $\Omega^{(1)}$, $\Omega^{(3)}$, respectively, see Fig. 10b. Important is to note that although the Penalty method is converging slower for the low order polynomial case in the displacement field before reaching its threshold, see Fig. 8a, the same does not necessarily hold for the resultant fields, that is the force and the moment tensors, see Fig. 9a and 10a, respectively. The Lagrange Multipliers methods in contrast to the Penalty method belong to the family of mixed formulations which result into saddle point problems. As mentioned in Sect. 2, a unique solution to the formulations resulting by using those methods is not guaranteed to exist or to be unique, a fact which is closely linked to the choice of the discrete Lagrange Multipliers fields. Moreover, it has been observed [14]

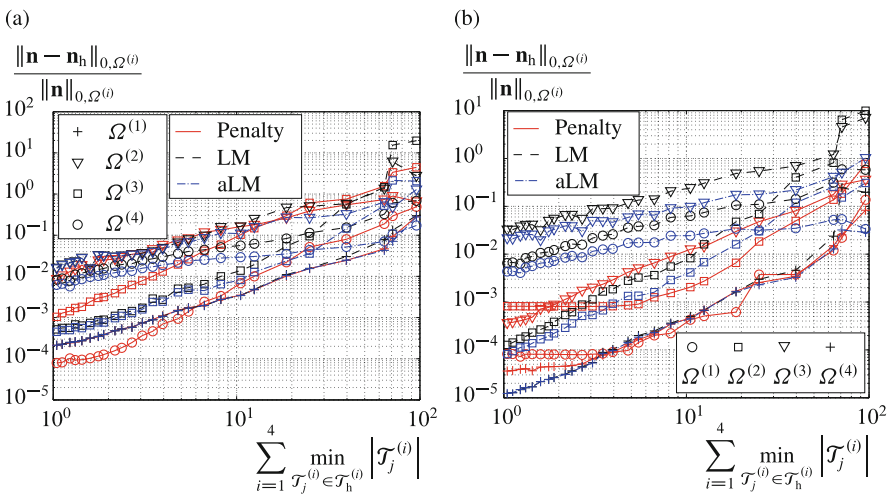


Fig. 9 Scordelis-Lo roof: convergence study. Relative force tensor error in the L^2-norm at each subdomain against the sum of the minimum element areas for the multipatch geometry. (**a**) $\Omega^{(1)}$, $\Omega^{(3)}$ bicubic and $\Omega^{(2)}$, $\Omega^{(4)}$ biquadratic bases. (**b**) $\Omega^{(1)}$, $\Omega^{(3)}$ biquartic and $\Omega^{(2)}$, $\Omega^{(4)}$ bicubic bases

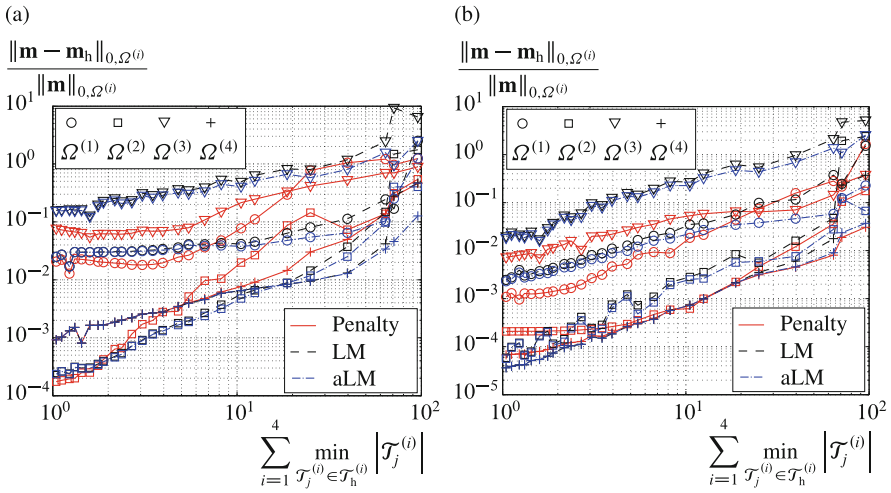


Fig. 10 Scordelis-Lo roof: convergence study. Relative moment tensor error in the L^2-norm at each subdomain against the sum of the minimum element areas for the multipatch geometry. (**a**) $\Omega^{(1)}$, $\Omega^{(3)}$ bicubic and $\Omega^{(2)}$, $\Omega^{(4)}$ biquadratic bases. (**b**) $\Omega^{(1)}$, $\Omega^{(3)}$ biquartic and $\Omega^{(2)}$, $\Omega^{(4)}$ bicubic bases

that mixed formulations may perform well for the approximation of the primary variables of a problem, in this case the displacement field, but not necessarily for the their derivatives, which in this case are the force and the moment traction fields.

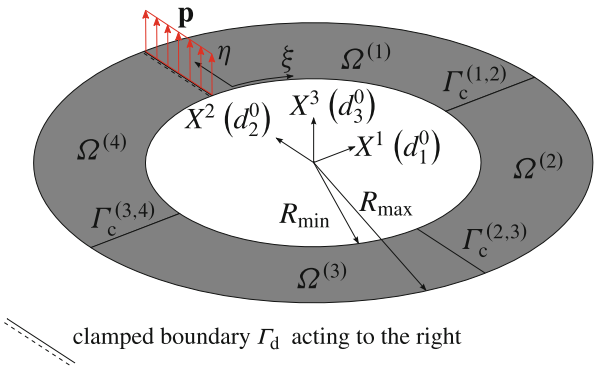


Fig. 11 Slit annular plate subject to tip lifting line force: problem placement

4.3 Slit Annular Plate Subject to Tip Lifting Line Force

The first nonlinear problem to be studied is that of a slit annular plate subject to tip lifting load. As reference, the results provided in [28] are used. The inner radius, the outer radius, the thickness of the plate and the load magnitude are $R_{\min} = 6\,\text{m}$, $R_{\max} = 10\,\text{m}$, $t = 3\,\text{cm}$ and $\mathbf{p} = p\,\mathbf{e}_3$ with $p = 8 \times 10^{-4}\,\text{KN/m}$, respectively. The Young's modulus and the Poisson's ratio are $E = 21 \times 10^3\ \text{KN/m}^2$ and $\nu = 0$, respectively. The problem is decomposed into four subdomains, see Fig. 11. Two settings for solving the aforementioned problem are used; the *coarse setting* for which patch $\Omega^{(1)}$, patch $\Omega^{(2)}$, patch $\Omega^{(3)}$ and patch $\Omega^{(4)}$ are discretized with thirty-six, eighteen, twenty-eight and ten elements, respectively, all having C^1-continuous biquadratic bases. On the other hand, the *fine setting* consists in discretizing patch $\Omega^{(1)}$, patch $\Omega^{(2)}$, patch $\Omega^{(3)}$ and patch $\Omega^{(4)}$ with two hundred nine, one hundred five, two hundred seventy-three and one hundred fifty-three elements, respectively, where patches $\Omega^{(1)}$, $\Omega^{(3)}$ have a C^1-continuous biquadratic basis and patches $\Omega^{(2)}$, $\Omega^{(4)}$ a C^2-continuous bicubic basis. The scaling of the penalty factors for the displacement and rotation penalization corresponding to the Penalty method are chosen as 10^2 and $10^3\,\text{m}^{-2}$, respectively. In what concerns the augmented Lagrange multipliers method, those factors are chosen as 10^{-1} and $10^0\,\text{m}^{-2}$, respectively.

For the solution of the nonlinear problem, sixty-one load steps are used with Newton-Raphson tolerance equal to 10^{-6}. In the set of Fig. 12, it is shown the ultimate deformation of the plate and the distribution of the physical moment component $\overline{m}^{22}$ along the plate for the fine setting corresponding to the solution provided by the Penalty method. The smoothness of the results across the patches which have strongly non-conforming bases exemplifies the applicability of the DDMs for the solution of the nonlinear Kirchhoff-Love shell problem in multiple domains. As last, the load-displacement curves corresponding to the d_3^0 component of the displacement field at points $\mathbf{X}_\text{A}$ and $\mathbf{X}_\text{B}$, with displaced coordinates $\mathbf{x}_\text{A} = \mathbf{X}_\text{A} + \mathbf{d}(\mathbf{X}_\text{A})$ and $\mathbf{x}_\text{B} = \mathbf{X}_\text{B} + \mathbf{d}(\mathbf{X}_\text{B})$, see Fig. 12a, for both the coarse and the fine settings and subsequently are given in Fig. 13 and compared to the results provided in [28]. Note that for both settings the DDMs deliver results very close

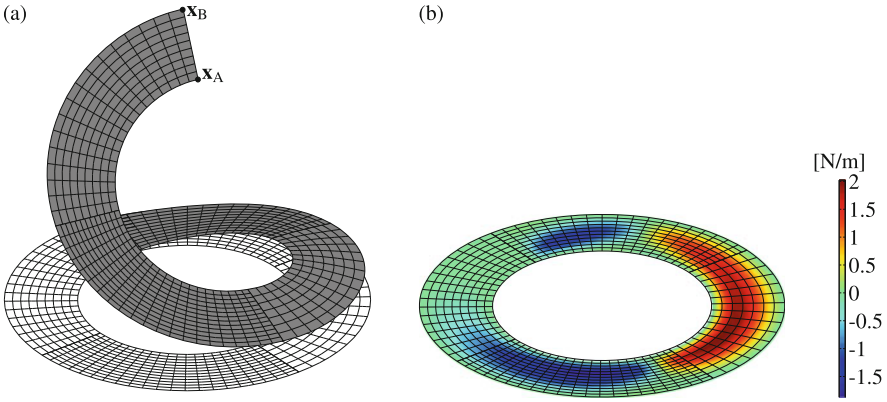


Fig. 12 Slit annular plate subject to tip lifting line force: postprocessing (Penalty). Results corresponding to the fine setting. (**a**) Ultimate deformation. (**b**) Ultimate moment field $\overline{m}^{22}$

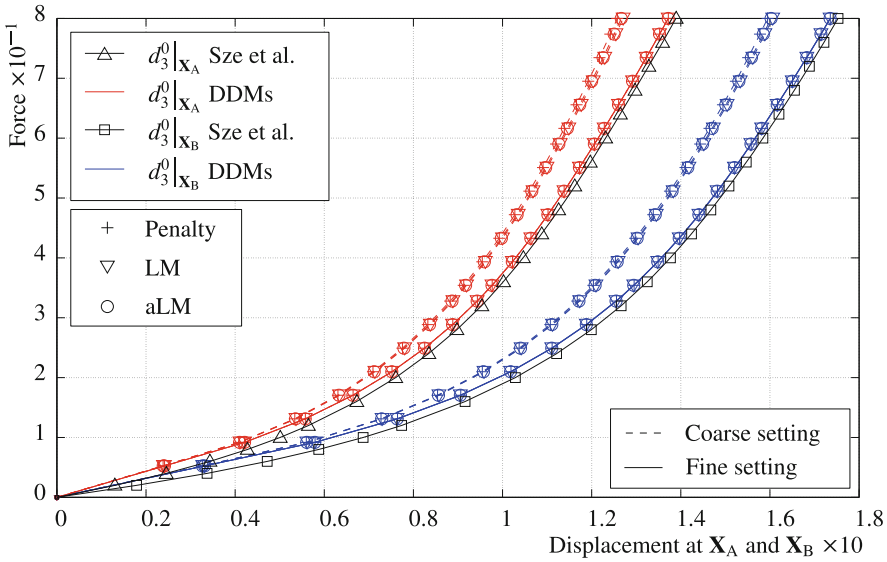


Fig. 13 Slit annular plate subject to tip lifting line force: load-displacement curves

to each other, and they are almost identical in the fine setting. The latter shows that all variational methods produce almost identical results as the mesh size goes to zero, given that the coupling parameters and Lagrange Multipliers discretizations are carefully chosen. It is worth mentioning that the reference results correspond to a Reissner-Mindlin shell formulation and therefore the reference load-displacement curves cannot be identical with the ones provided by the DDMs for a Kirchhoff-Love shell. However the problem considers a relatively thin structure and thus the results from both theories are very close to each other, as expected.

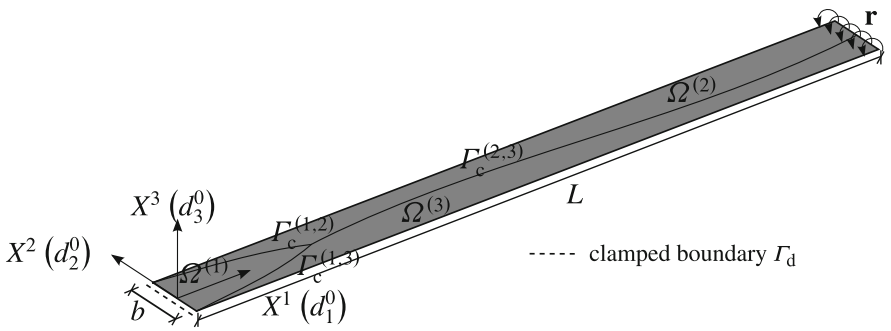


Fig. 14 Cantilever subjected to end moment: problem placement

4.4 Cantilever Subjected to End Moment

The second nonlinear problem to be studied is that of a cantilever subject to end moment. The material characteristics of the plate, that is, Young's modulus and Poisson's ratio are equal to $E = 1.2 \times 10^3\,\mathrm{KN/m^2}$ and $\nu = 0$, respectively. On the other hand, the geometrical characteristics of the plate, that is, its length, its width and its thickness are equal to $L = 12\,\mathrm{m}$, $b = 1\,\mathrm{m}$ and $h = 10\,\mathrm{cm}$, respectively. The cantilever is subject to end bending moment $\mathbf{r}$, with amplitude $\|\mathbf{r}\|_2 = 2\pi\frac{EI}{L} = 52.3333\,\mathrm{Nm}$, where $I = \frac{bh^3}{12} = 8.3333\,\mathrm{m}^4$ stands for the plate's moment of inertia. It is known that under this loading, the plate must bend into a circle of radius $\overline{R} = \frac{EI}{\|\mathbf{r}\|_2} = 1.9099\,\mathrm{m}$, see [28]. The computational domain Ω is decomposed into three subdomains, see Fig. 14. Those subdomains are neighbouring through three interface boundaries which are chosen curved. Interface boundary $\overline{\Gamma_c^{(1,2)}}$ is represented by a quadratic Bézier curve between the control points $\hat{\mathbf{X}}_1^{(1,2)} \equiv (0, -0.5, 0)$, $\hat{\mathbf{X}}_2^{(1,2)} \equiv (0.15, -0.3, 0)$ and $\hat{\mathbf{X}}_3^{(1,2)} \equiv (0.2, 0, 0)$. Similarly, interface boundary $\overline{\Gamma_c^{(1,3)}}$ is also represented by a quadratic Bézier curve between the control points $\hat{\mathbf{X}}_1^{(1,3)} \equiv (0, 0.5, 0)$, $\hat{\mathbf{X}}_2^{(1,3)} \equiv (0.15, 0.3, 0)$ and $\hat{\mathbf{X}}_3^{(1,3)} \equiv (0.2, 0, 0)$. Finally, interface boundary $\overline{\Gamma_c^{(2,3)}}$ is represented by a cubic Bézier curve between the control points $\hat{\mathbf{X}}_1^{(2,3)} \equiv (0.2, 0, 0)$, $\hat{\mathbf{X}}_2^{(2,3)} \equiv (0.4667, 0.5, 0)$, $\hat{\mathbf{X}}_3^{(2,3)} \equiv (0.7333, -0.5, 0)$ and $\hat{\mathbf{X}}_4^{(2,3)} \equiv (1, 0, 0)$. Note that within this decomposition, patch $\Omega^{(1)}$ attains a parametric singularity on its boundary point $\overline{\Gamma_c^{(1,2)}} \cap \overline{\Gamma_c^{(1,3)}}$ as its rectangular NURBS parametric space is degenerated into a triangle in the Cartesian space.

The tolerance for the Newton-Rapshon iterations is chosen 10^{-5} for twenty load steps in relation to the employed nonlinear scheme. In addition, the scaling of the both the displacement and the rotation penalty factors is chosen as 10^2 and 10^{-2} for the Penalty and the augmented Lagrange Multipliers method, respectively. Two settings, one *coarse* and one *fine*, are chosen for the current problem as in Sect. 4.3. The coarse setting consists in parametrizing patch $\Omega^{(1)}$, $\Omega^{(2)}$ and $\Omega^{(3)}$

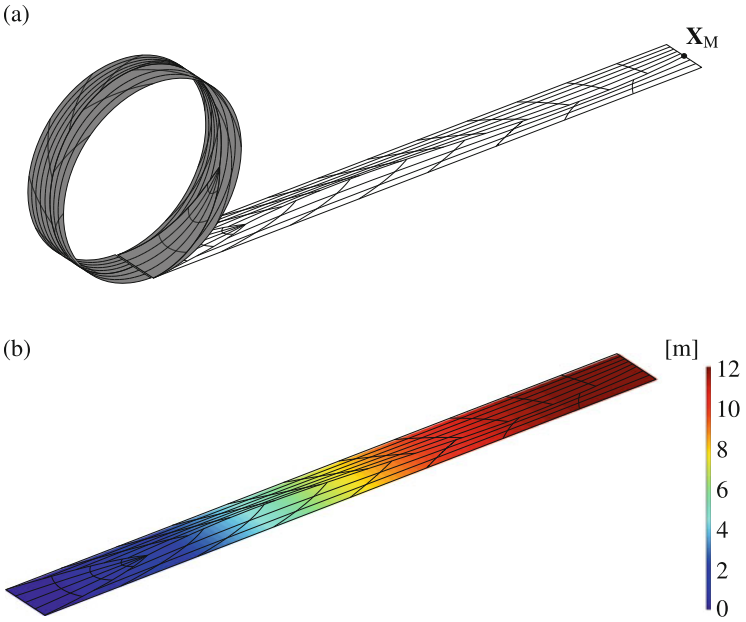


Fig. 15 Cantilever subjected to end moment: postprocessing (augmented Lagrange multipliers). Results corresponding to the fine setting. (**a**) Ultimate plate's deformation. (**b**) The ultimate displacement field in $\|\mathbf{d}\|_2$

with four, ten and twenty-one elements, respectively, all attaining C^2-continuous bicubic polynomial bases. In what concerns the fine setting, patches $\Omega^{(1)}$, $\Omega^{(2)}$ and $\Omega^{(3)}$ are parametrized with sixteen, twenty-four and fifty-five elements, respectively, all attaining C^3-continuous biquartic polynomial bases. The ultimate plate's deformation and the ultimate displacement field in the 2-norm using the augmented Lagrange Multipliers method and corresponding to the fine configuration is shown in set of Fig. 15. As expected, the plate is bent into a circle of the expected radius and the DDMs were proven to handle pointwise singularities even when those appear on the interface Γ_{c}. It must be however noted that the deformed configuration does not represent an exact circle if the control point weights of the NURBS parametrization are not chosen appropriately, see [7]. However, the finer the geometry space is chosen the better the approximation of the circle becomes.

Subsequently, load-displacement curves for the d_3^0 and d_1^0 components of the displacement field at point $\mathbf{X}_{\mathrm{M}}$ are given in Fig. 16, which is lying in the middle of the plate's tip, see Fig. 15a. The DDMs delivered highly accurate results in the fine setting and for the given choice of parameters as it can be verified by the almost overlapping curves with the analytical ones provided in [28]. On the other hand, for the coarse setting the results are as expected poor, as the discretization of the displacement field is not rich enough, with the Lagrange Multipliers methods

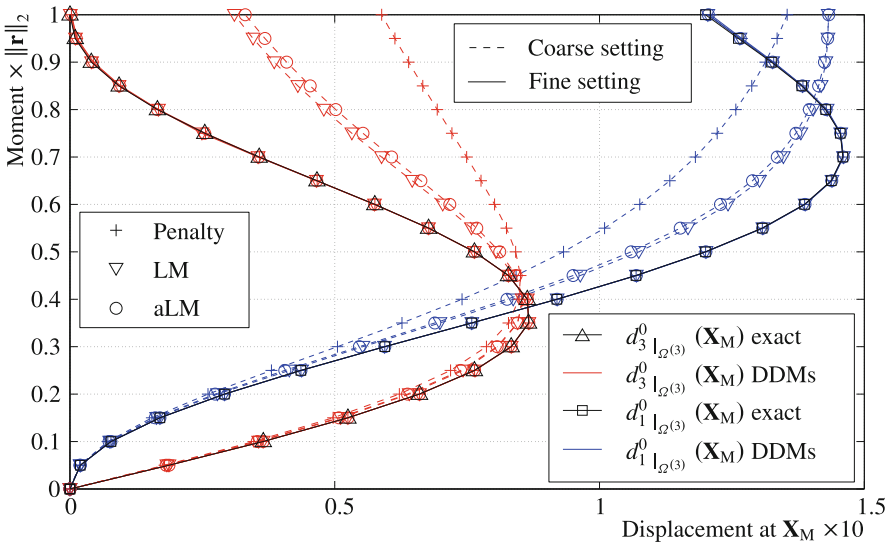


Fig. 16 Cantilever subjected to end moment: load-displacement curves

demonstrating better behaviour than the Penalty method. Additionally, it can be observed that the produced by the augmented Lagrange Multipliers method results are lying between those of the Penalty and the Lagrange Multipliers methods. The latter verifies that the augmented Lagrange Multipliers method is essentially a compensation between the other two methods, with the magnitude of the penalty parameters being decisive on its behaviour. The larger the penalty parameters are chosen, the better the interface continuity conditions (10h) and (10i) are satisfied but on the same time the overall error is increased and vice versa, see set of Fig. 4.

5 Conclusions and Outlook

In this study, three different DDMs are presented and compared for the decomposed Kirchhoff-Love shell problem in both linear and nonlinear isogeometric analysis. Investigated are a Penalty, the Lagrange Multipliers and the augmented Lagrange Multipliers method.

The Penalty method results in a positive definite discrete equation system given that the penalty parameters do not affect the conditioning of the equation system. The method performs robustly and its implementation is relatively simple. One drawback of the method is that its accuracy depends on the size of the penalty parameters which is typical for the Penalty methods. However, a discretization-dependent rule is proposed and numerically tested.

The Lagrange Multipliers method results in a saddle point discrete equation system. The latter means that the finite dimensional spaces must fulfil the Ladyzenskaja-Babuska-Brezzi condition so that one unique solution to the problem can be guaranteed. This is closely linked to the choice of the discretization of the Lagrange Multipliers fields, which is in general a difficult problem. Nevertheless, the method provided highly satisfactory results for the proposed choice of the Lagrange Multipliers fields discretization.

The augmented Lagrange Multipliers method inherits the properties from both the Lagrange Multipliers and the Penalty method. Therefore, the method results also in a saddle point problem for which the penalty parameters can be used to adjust the accuracy of the results without the necessity of being too large.

Extension of the given DDMs into a parameter-free variational formulation is by this study highly motivated, so that the parameters which are driving the coupled system are kept to minimum and thus minimizing the negative consequences of a possible improper choice.

Acknowledgements The support of the Deutsche Forschungsgemeinschaft (DFG) with grant numbers BL 306/26-1 and BL 306/23-2 is gratefully acknowledged.

References

1. Y. Basar, W. Krätzig, *Mechanik der Flächentragwerke* (Vieweg, Braunschweig, 1985)
2. T. Hughes, J. Cottrell, Y. Bazilevs, Isogeometric analysis: CAD, finite elements, NURBS, exact geometry, and mesh refinement. Comput. Methods Appl. Mech. Eng. **194**, 4135–4195 (2005)
3. K.-J. Bathe, *Finite Element Procedures* (Prentice Hall, Upper Saddle River, 2006)
4. O. Zienkiewicz, R. Taylor, J. Zhu, *The Finite Element Method, Its Basis and Fundamentals*, 6th edn. (Elsevier, Oxford, 2005)
5. L. Piegl, W. Tiller, *The NURBS Book*, 2nd edn. (Springer, New York, 1997)
6. R. Schmidt, J. Kiendl, K.-U. Bletzinger, R. Wüchner, Realization of an integrated structural design process: analysis-suitable geometric modelling and isogeometric analysis. Comput. Vis. Sci. **13**, 315–330 (2010)
7. J. Kiendl, K.-U. Bletzinger, J. Linhard, R. Wüchner, Isogeometric shell analysis with Kirchhoff-Love elements. Comput. Methods Appl. Mech. Eng. **198**, 3902–3914 (2009)
8. R. Schmidt, R. Wüchner, K.-U. Bletzinger, Isogeometric analysis of trimmed NURBS geometries. Comput. Methods Appl. Mech. Eng. **241–244**, 93–111 (2012)
9. M. Dörfel, B. Jüttler, B. Simeon, Adaptive isogeometric analysis based on t-splines. Comput. Methods Appl. Mech. Eng. **199**, 264–275 (2010)
10. T.-K. Uhm, Y. Kim, S.-K. Youn, A locally refinable t-spline finite element method for CAD/CAE integration. Struct. Eng. Mech. **80**, 225–245 (2008)
11. M. Breitenberger, A. Apostolatos, B. Philipp, R. Wüchner, K.-U. Bletzinger, Analysis in computer aided design: nonlinear isogeometric B-Rep analysis of shell structures. Comput. Methods Appl. Mech. Eng. **284**, 401–457 (2015)
12. I. Babuška, The finite element method with penalty. Math. Comput. **27**, 221–228 (1973)
13. I. Babuška, M. Zlámal, Nonconforming elements in the finite element method with penalty. SIAM J. Numer. Anal. **10**(5), 863–875 (1973)
14. J.D. Sanders, J.E. Dolbow, T.A. Laursen, On methods for stabilizing constraints over enriched interfaces in elasticity. Int. J. Numer. Methods Eng. **78**, 1009–1036 (2009)

15. A. Apostolatos, R. Schmidt, R. Wüchner, K.-U. Bletzinger, A Nitsche-type formulation and comparison of the most common domain decomposition methods in isogeometric analysis. Int. J. Numer. Methods Eng. **97**(7), 473–504 (2014)
16. D.N. Arnold, An interior penalty finite element method with discontinuous elements. SIAM J. Numer. Anal. **19**, 742–760 (1982)
17. M.F. Wheeler, An elliptic collocation-finite element method with interior penalties. SIAM J. Numer. Anal. **15**, 152–161 (1978)
18. Y. Guo, M. Ruess, Nitsche's method for a coupling of isogeometric thin shells and blended shell structures. Comput. Methods Appl. Mech. Eng. **284**, 881–905 (2015)
19. J. Kiendl, Y. Bazilevs, M.-C. Hsu, R. Wüchner, K.-U. Bletzinger, The bending strip method for isogeometric analysis of Kirchhoff–Love shell structures comprised of multiple patches. Comput. Methods Appl. Mech. Eng. **199**, 2403–2416 (2010)
20. C. Bernardi, Y. Maday, A. Patera, A new nonconforming approach to domain decomposition: the mortar element method, Technical report, Université Pierre at Marie Curie, Paris (1990)
21. E. Brivadis, A. Buffa, B. Wohlmuth, L. Wunderlich, Isogeometric mortar methods. Comput. Methods Appl. Mech. Eng. **284**, 292–319 (2015)
22. F. Brezzi, K.-J. Bathe, A discourse on the stability conditions for mixed finite element formulations. Comput. Methods Appl. Mech. Eng. **82**, 27–57 (1990)
23. A. Toselli, O. Widlund, *Domain Decomposition Methods-Algorithms and Theory* (Springer, Berlin/Heidelberg, 2005)
24. B.O. Neil, *Elementary Differential Geometry* (Academic, San Diego, 1997)
25. P. Kaufmann, S. Martin, M. Botsch, M. Gross, Implementation of discontinuous Galerkin Kirchhoff-Love shells, Technical report, ETH Zurich, Department of Computer Science (2009)
26. L. Noels, R. Radovitzky, A new discontinuous Galerkin method for Kirchhoff-Love shells. Comput. Methods Appl. Mech. Eng. **197**, 2901–2929 (2008)
27. J. Cottrell, T. Hughes, Y. Bazilevs, *Isogeometric Analysis: Toward Integration of CAD and FEA* (Wiley, Chichester, 2009)
28. K. Sze, X. Liu, S. Lo, Popular benchmark problems for geometric nonlinear analysis of shells. Finite Elem. Anal. Des. **40**, 1551–1569 (2004)
29. S. Timoshenko, S. Woinowsky-Krieger, K. (Firm), *Theory of Plates and Shells*, 2nd edn. (McGraw-Hill, New York, 1959), international edition 1959 – t.p. verso
30. A.C. Scordelis, K.S. Lo, Computer analysis of cylindrical shells. J. Am. Concr. Inst. **61**, 539–561 (1969)

A Reduced Integration for Reissner-Mindlin Non-linear Shell Analysis Using T-Splines

C. Adam, S. Bouabdallah, M. Zarroug, and H. Maitournam

Abstract We propose a reduced shell element for Reissner-Mindlin geometric non-linear analysis within the context of T-spline analysis. The shell formulation is based on the displacements and a first order kinematic in the thickness is used for the transverse shear strains. A total Lagrangian formulation is considered for the finite transformations. The update of the incremental rotations is made using the quaternion algebra. The standard two-dimensional reduced quadrature rules for structured B-spline and NURBS basis functions are extended to the more flexible T-meshes. The non-uniform Gauss-Legendre and patchwise reduced integrations for quadratic shape functions are both presented and compared to the standard full Gauss-Legendre scheme. The performance of the element is assessed with linear and geometric non-linear two-dimensional problems in structural analysis. The effects of mesh distortion and local refinement, using both full and reduced numerical quadratures, are evaluated.

1 Introduction

Isogeometric analysis (IGA) [1] is a recent concept introduced by T.J.R Hughes in 2004 that aims to bridge engineering design to analysis. B-spline and NURBS [2] shape functions, initially used as standard tools in industrial computer aided design (CAD) software, also provide promising properties for analysis. The geometric description in CAD systems needs to be translated in a representation suitable for

C. Adam (✉) • H. Maitournam
Laboratoire de Mécanique des Solides (CNRS UMR 7649), École Polytechnique, Route de Saclay, 91128 Palaiseau, France
e-mail: adam@lms.polytechnique.fr

S. Bouabdallah
Département Mécanique Numérique et Modélisation, École Supérieure d'Ingénieurs Léonard de Vinci, 8 avenue Léonard de Vinci, 92400 Courbevoie, France

M. Zarroug
Direction Scientifique et Technologies Futures, PSA Peugeot Citroën, Route de Gisy, 78140 Vélizy-Villacoublay, France

B. Jüttler, B. Simeon (eds.), *Isogeometric Analysis and Applications 2014*,
Lecture Notes in Computational Science and Engineering 107,
DOI 10.1007/978-3-319-23315-4_5

finite element analysis (FEA) or isogeometric analysis. This process is far from trivial when dealing with complex designs in industrial applications and represents a large amount of the overall process time.

Using different geometries results in several difficulties since both representations need to be updated in the same time, efficient shape optimizations of the CAD geometry require a tight integration between the solver and the optimizer with an automatic CAD geometry to mesh conversion and vice versa. These tasks are complex since meshes are disconnected from CAD geometries. Ideally, having a unique analysis suitable geometric model, which can be used directly from design to simulation, is essential to automatize the overall process.

Engineering designs present some drawbacks for analysis since there are less restrictions when manipulating the CAD geometry than when performing a numerical simulation. Some properties such NURBS trimming or patch merging represent bottlenecks to achieve an automatic process which would join both domains. The structured B-spline and NURBS shape functions are not enough flexible to represent general industrial designs in an analysis suitable fashion.

The notion of T-spline [3] extends the concepts of B-splines and NURBS. They benefit from the flexibility of the point-based splines [3] and the Cartesian structure of B-splines. Many complex geometries can be represented with a single T-spline. As hierarchical B-splines [4], they provide local refinement which is essential from an analysis point of view.

As a generalisation of B-splines, T-splines suffer from numerical locking pathologies and computational efficiency using a full Gauss-Legendre quadrature. Efficient quadrature rule is an essential ingredient to obtain efficient numerical simulations, regardless of the type of basis functions chosen. Full Gauss-Legendre quadrature is optimal for C^0 discretization methods such as standard Lagrange elements. However, it has been shown in [5] that optimal quadrature rules for spline-based analysis are patchwise rules with fewer quadrature points and different locations and weights.

Reduced integration is a controversial subject in computational mechanics which has significantly improved the efficiency of numerical simulations reducing the number of quadrature points and the locking phenomena occurring in thin structures modelled by thick shell elements. Some work on this topic has been initiated in e.g. [6–9] within the context of B-spline and NURBS analysis.

In this paper, we extend the reduced quadrature schemes presented in [10–13] to the more general T-spline analysis for two-dimensional problems. We limit our work to quadratic shape functions. The strengths of T-splines, reduced integration and geometric non-linear Reissner-Mindlin shell formulation are combined to obtain efficient isogeometric elements needed for complex and time consuming industrial simulations.

The paper is outlined as follows. In Sect. 2 we give a brief overview of the T-spline technology introducing the point-based splines and the concept of local knot vectors and refinement. In Sect. 3 we propose a Reissner-Mindlin shell element for geometric non-linear analysis of structures focusing on the update of the normals with arbitrary large rotations. The extension of non-uniform Gauss-Legendre and

patchwise reduced quadrature rules is presented in Sect. 4. Both integrations are performed in Bézier elements. The performance of the resulting reduced shell elements is assessed in Sect. 5 using standard linear and non-linear benchmark problems. We draw conclusions in Sect. 6.

2 T-Splines

T-splines generalize the concepts of B-splines and NURBS. Before describing T-splines, we introduce the notion of function subdivision which is a process that enlightens the concept of T-splines.

2.1 Blending Function Subdivision

Let us consider two quadratic B-spline basis $\mathbf{N}$ and $\tilde{\mathbf{N}}$ generated by the two open and global knot vectors $\Xi = \{0, 0, 0, 1, 2, 3, 4, 4, 4\}$ and $\tilde{\Xi} = \{0, 0, 0, 1, 1.5, 2, 3, 4, 4, 4\}$. We have the relation $\Xi \subset \tilde{\Xi}$ in the sense that the space of all curves that can be built from Ξ is included in the space of all curves that can be built from $\tilde{\Xi}$. In other words, each basis function provided by the knot vector Ξ can be expressed as a linear combination of the basis functions given by $\tilde{\Xi}$ [3]. These relations can be written in the following compact form

$$\mathbf{N} = \mathbf{P}\tilde{\mathbf{N}}, \tag{1}$$

where $\mathbf{P}$ is the operator of knot insertion. In order to describe the same curve $\mathcal{C}(\xi) = \mathbf{N}^{\mathrm{T}}\mathbf{B} = \tilde{\mathbf{N}}^{\mathrm{T}}\tilde{\mathbf{B}}$, a similar relation is obtained for both sets of control points $\mathbf{B}$ and $\tilde{\mathbf{B}}$

$$\tilde{\mathbf{B}} = \mathbf{P}^{\mathrm{T}}\mathbf{B}. \tag{2}$$

Figure 1 represents the two quadratic B-spline basis. On the left, blue curves are provided by the coarse knot vector Ξ while, on the right, red curves represent the new functions obtained from the refinement of the coarse shape functions. The three polynomials represented by dotted blue lines can be expressed as a linear combination of the four new functions denoted by red lines. Further details on the algorithm of the blending function refinement for cubic polynomials are presented in [3].

This example illustrates the notion of function subdivision and local refinement. In fact, knot insertion with B-splines or NURBS is a global process with multi-dimensional problems since it modifies the global knot vectors which generate the basis functions. However, instead of considering refinement of global knot vectors, we can consider the refinement of the basis functions, namely the refinement of the local knot vectors. The Cox-de-Boor formula [2] exhibits a local process to construct

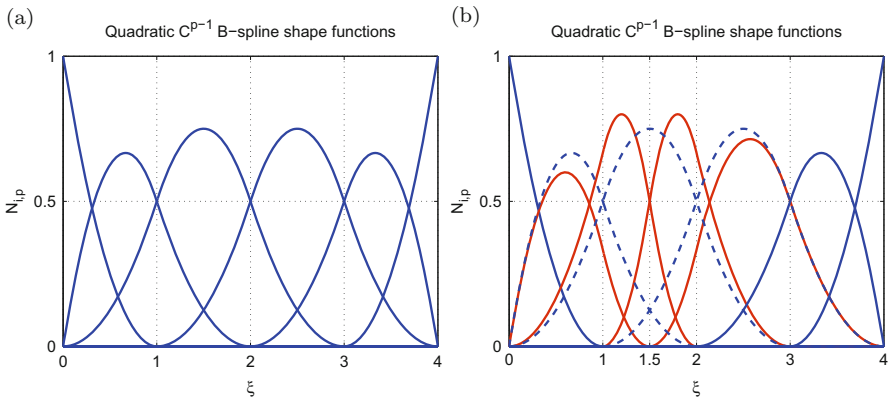


Fig. 1 Univariate B-spline basis functions of order two with both open knot vectors Ξ and $\tilde{\Xi}$. (**a**) Coarse knot vector $\Xi = \{0, 0, 0, 1, 2, 3, 4, 4, 4\}$. (**b**) Refined knot vector $\tilde{\Xi} = \{0, 0, 0, 1, 1.5, 2, 3, 4, 4, 4\}$

Table 1 Relations between the two basis $\mathbf{N}$ and $\tilde{\mathbf{N}}$

Coarse basis	Local knot Ξ_i^{loc}	Refined basis	Local knot $\tilde{\Xi}_i^{\text{loc}}$	Linear combination
N_1	[0 0 0 1]	$\tilde{N}_1$	[0 0 0 1]	$N_1 = \tilde{N}_1$
N_2	[0 0 1 2]	$\tilde{N}_2$	[0 0 1 1.5]	$N_2 = \tilde{N}_2 + \frac{1}{4}\tilde{N}_3$
N_3	[0 1 2 3]	$\tilde{N}_3$	[0 1 1.5 2]	$N_3 = \frac{3}{4}\tilde{N}_3 + \frac{3}{4}\tilde{N}_4$
N_4	[1 2 3 4]	$\tilde{N}_4$	[1 1.5 2 3]	$N_4 = \frac{1}{4}\tilde{N}_4 + \tilde{N}_5$
N_5	[2 3 4 4]	$\tilde{N}_5$	[1.5 2 3 4]	$N_5 = \tilde{N}_6$
N_6	[3 4 4 4]	$\tilde{N}_6$	[2 3 4 4]	$N_6 = \tilde{N}_7$
		$\tilde{N}_7$	[3 4 4 4]	

the spline polynomials of order p from the piecewise constant functions defined by the knot vectors. In that sense, to each basis function $N_{i,p}$ can be associated a local knot vector Ξ_i^{loc} of size $p + 2$, necessary and sufficient to construct the spline function.

Table 1 describes the local knot vectors Ξ_i^{loc} and $\tilde{\Xi}_i^{\text{loc}}$ as well as the existing relations between the two basis $\mathbf{N}$ and $\tilde{\mathbf{N}}$. The knot insertion $\xi = 1.5$ only impacts $p + 2 = 4$ basis functions.

Finally, knot insertion as subdivision function is a more general concept than the standard refinement for B-splines and NURBS. This notion is more general and exhibits interesting properties such as the local refinement which is a major drawback of B-splines and NURBS. T-splines are constructed from local knot vectors associated with the subdivision function process. Therefore, they permit meshes with the so called T-junctions and they contain the special cases of B-splines and NURBS. Another strength of T-splines is their capability to achieve a high continuity across patch boundaries without introducing new knots which propagate in each direction when merging two patches.

2.2 B-Splines, PB-Splines and T-Splines

The concept of point-based splines [3] has been introduced to benefit from the local construction and refinement of the splines. There is no notion of global knot vector, but only local and independent knot vectors. In that sense, the associated polynomials are called blending functions rather than basis functions since the properties of linear independence and partition of unity are no longer guaranteed. PB-splines are even more flexible than T-splines and have some nice properties for modelling geometries but the lack of structure in the knot vectors is a major deficiency which prevents them from being used in analysis.

T-splines represent a good compromise between the Cartesian structure of B-splines and the flexible PB-splines. They benefit from the strengths of both types of functions. Initially they were defined for bi-cubic surfaces but more recently, they were extended to three-dimensional problems and arbitrary degree p [14].

In this paper, we do not give an in-depth description of the T-spline theory which is very rich and complex. The interested reader can refer to e.g. [15] to have a more detailed study of this theory. We restrict our work to bi-quadratic basis functions which give the possibility of constructing a large part of complex geometries.

2.3 The T-Mesh

Figure 2 represents, on the left, a T-mesh constructed from two NURBS-patches denoted by blue and red lines respectively. It also includes a local refinement strategy represented by a green line. Several T-junctions are thus obtained. On the

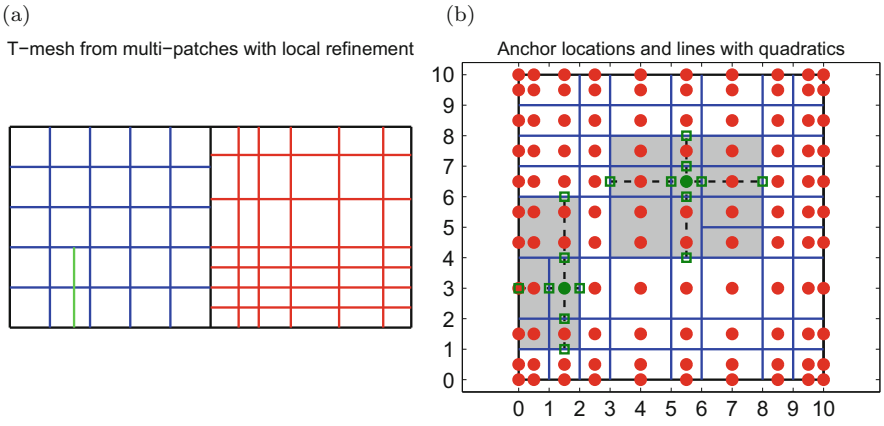


Fig. 2 T-mesh obtained from multi-patches merging (*blue* and *red lines*) with local refinement (*green line*). T-mesh with anchor locations (*red* and *green points*), local knot vectors (*green squares*) and supports of the blending functions (*grey areas*). (**a**) T-mesh with local refinement. (**b**) T-mesh with anchor locations and lines

right, Fig. 2 illustrates the construction of local knot vectors in each parametric direction. Let consider two bi-quadratic functions N_1 and N_2. They are represented in the T-mesh by two anchors (green dots) and their respective support is defined by the two darkened areas. Each anchor (red dots) is associated with one polynomial function and is located at the maximum of the blending function. They are equivalent to the Greville abscissae for B-splines and NURBS meshes. The construction of the anchor locations and lines, namely the local knot vectors defining the supports of the polynomials, is not detailed here but the interested reader can refer to [14]. The corresponding knots vectors are

$$\Xi_1^{\text{loc}} = [0\ 0\ 1\ 2] \text{ and } \mathcal{H}_1^{\text{loc}} = [1\ 2\ 4\ 6], \tag{3}$$

$$\Xi_2^{\text{loc}} = [3\ 5\ 6\ 8] \text{ and } \mathcal{H}_2^{\text{loc}} = [4\ 6\ 7\ 8]. \tag{4}$$

The T-mesh contains all the essential information for defining the local knot vectors and thus the shape functions. Moreover, the structure of the T-mesh defines the properties of linear independence and partition of unity. A T-mesh that guarantees these required properties for the analysis is called an analysis suitable T-mesh. Analysis suitable T-splines are a subset of general T-splines which satisfy several rules described in [16]. We introduce T-junction extensions which correspond to closed line segments that extend from the T-junctions in the opposite direction and cross $\lfloor (p+1)/2 \rfloor$ perpendicular edges or vertices. One may note that T-junction extensions are lines of reduced continuity. In practical terms, a T-mesh is well-characterized and suitable for analysis if no vertical T-junction extension intersects a horizontal T-junction extension. Both T-meshes represented in Fig. 2 are thus suitable for analysis.

The structure of an analysis suitable T-mesh is also flexible enough to describe complex geometries with locally different continuities and holes. Figure 3

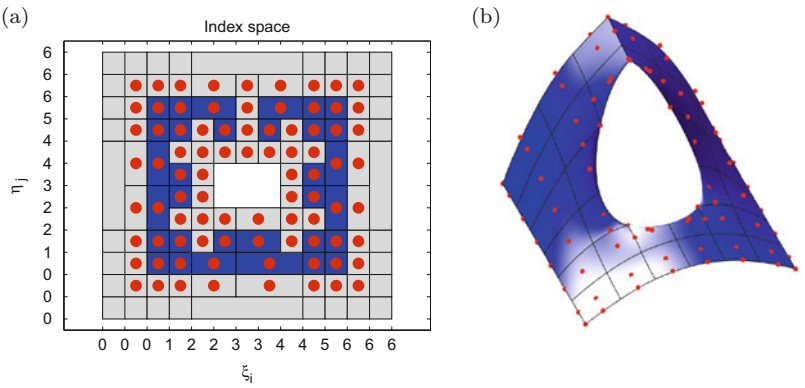


Fig. 3 T-mesh and associated geometry with a hole and local C^0 continuity. *Grey* elements have zero measure while *blue* elements have a positive measure. (**a**) Index space. (**b**) Physical space

represents, on the right, a complex bi-quadratic and three-dimensional surface containing a hole and a C^0 sharp edge which do not propagate within the domain. On the left, the T-mesh corresponding to the geometry is given in the index space so that multiple knots are visible. Grey elements have zero measure, there are thus non-physical, while blue elements have a positive measure. This T-mesh is suitable for analysis.

3 Large Deformations in Thick Shells

This section aims to develop the shell model presented in [10] for large deformations and moderate strains in Reissner-Mindlin shells. Both thick and thin shell structures are generally subjected to arbitrary large deformations and rotations but strains are moderate due to their particular shape ratio between the lengths of the in-plane directions and the thickness.

3.1 Green-Lagrange Strain Tensor

When dealing with moderate strains, the Green-Lagrange strain measure $\mathbf{E}$ [17] is a good candidate to generalize the standard infinitesimal strain tensor $\boldsymbol{\epsilon}$ used for small transformations. The Green-Lagrange tensor is well adapted to the Total Lagrangian Formulation (TLF), in which the discrete equations are formulated with respect to the reference configuration.

To compute $\mathbf{E}$, the current configuration $\mathbf{x}$ is expressed in the initial configuration $\mathbf{X}$ as follows

$$\mathbf{x} = \mathbf{X} + \mathbf{u}. \tag{5}$$

The Green-Lagrange strain measure is defined by the relation

$$\mathrm{d}\mathbf{x}^2 - \mathrm{d}\mathbf{X}^2 = 2\mathrm{d}\mathbf{X}^{\mathrm{T}}[\mathbf{E}]\mathrm{d}\mathbf{X}, \tag{6}$$

with

$$\mathrm{d}\mathbf{x} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}}\mathrm{d}\mathbf{X} = \frac{\partial(\mathbf{X}+\mathbf{u})}{\partial \mathbf{X}}\mathrm{d}\mathbf{X} = \mathbf{F}\mathrm{d}\mathbf{X}, \tag{7}$$

where the deformation gradient tensor $\mathbf{F}$ is related to both reference and current configurations as follows

$$\mathbf{F} = \mathbf{I_3} + \mathbf{D} \text{ and } \mathbf{D} = \frac{\partial \mathbf{u}}{\partial \mathbf{X}} = \mathbf{u}_{,\mathbf{X}}. \tag{8}$$

The Green-Lagrange tensor is thus

$$[\mathbf{E}] = \frac{1}{2}(\mathbf{F}^{\mathrm{T}}\mathbf{F} - \mathbf{I_3}) = \frac{1}{2}(\mathbf{D} + \mathbf{D}^{\mathrm{T}} + \mathbf{D}^{\mathrm{T}}\mathbf{D}). \tag{9}$$

Different measures exist when dealing with finite deformations. The Green-Lagrange strain tensor is a special case of more general strain measures, the Seth-Hill family of strain measures:

$$[\mathbf{E}_m] = \frac{1}{2m}(\mathbf{U}^{\mathbf{2m}} - \mathbf{I_3}) = \frac{1}{2m}((\mathbf{F}^{\mathrm{T}}\mathbf{F})^m - \mathbf{I_3}). \tag{10}$$

For several values of m we obtain the standard Green-Lagrange measure ($m = 1$), the Biot tensor ($m = 1/2$), the Euler-Almansi strain tensor ($m = -1$) or the logarithmic strains ($m = 0$). In the case of small strains, all the measures are similar. The logarithmic strain is a standard measure when dealing with arbitrary finite deformations. However, in structural mechanics, we recall that the displacements and rotations can be large but the strains generally remains moderate so that the Green-Lagrange measure is sufficient. The interested reader can find more details on the strain measures and the different formulations in [17].

3.2 Constitutive Law

We consider an elastic behaviour along with a plane stress hypothesis. The constitutive law relates the second Piola-Kirchhoff stresses $\mathbf{S}$ to the local Green-Lagrange strains $\mathbf{E}$ so that, in Voigt notation, we have

$$\mathbf{S} = \mathbf{C}\mathbf{E}, \tag{11}$$

where $\mathbf{C}$ is the standard constitutive matrix of the isotropic material for elastic shells under plane stress conditions in the case of Reissner-Mindlin's theory. In terms of iterative variation we thus obtain

$$\Delta\mathbf{S} = \mathbf{C}\Delta\mathbf{E}. \tag{12}$$

3.3 Virtual Work

The virtual work of internal forces can be written on the reference geometry V^0 as follows

$$\delta W^{\mathrm{int}} = \int_{V_0} \delta\mathbf{E}^{\mathrm{T}}\mathbf{S}\,\mathrm{d}V_0. \tag{13}$$

The iterative variation of the virtual work is

$$\Delta(\delta W^{\text{int}}) = \int_{V_0} \delta \mathbf{E}^{\mathrm{T}} \Delta \mathbf{S} + \Delta(\boldsymbol{\delta} \mathbf{E})^{\mathrm{T}} \mathbf{S} \, \mathrm{d}V_0. \tag{14}$$

The strain-displacement matrix can be rewritten in the following condensed form

$$\mathbf{E} = \left(\mathbf{H} + \frac{1}{2} \mathbf{A}\left(\mathbf{u}_{,\mathbf{X}}\right) \right) \mathbf{G}\mathbf{U} \tag{15}$$

where $\mathbf{U}$ is the vector containing the approximate displacements and rotations. The expression of the operators $\mathbf{H}$, $\mathbf{A}$ and $\mathbf{G}$ is detailed in [17].

Noting that $\delta \mathbf{A}\left(\mathbf{u}_{,\mathbf{X}}\right) \mathbf{G}\mathbf{U} = \mathbf{A}\left(\mathbf{u}_{,\mathbf{X}}\right) \mathbf{G}\delta\mathbf{U}$, we obtain the following virtual variation for the Green-Lagrange tensor

$$\delta \mathbf{E} = \left(\mathbf{H} + \mathbf{A}\left(\mathbf{u}_{,\mathbf{X}}\right)\right) \mathbf{G}\delta\mathbf{U} = \mathbf{B}_{\text{nl}}(\mathbf{u})\delta\mathbf{U}. \tag{16}$$

Finally, the virtual iterative variation of $\mathbf{E}$ is

$$\Delta(\delta \mathbf{E}) = \mathbf{A}\left(\Delta \mathbf{u}_{,\mathbf{X}}\right) \mathbf{G}\delta\mathbf{U} + \left(\mathbf{H} + \mathbf{A}\left(\mathbf{u}_{,\mathbf{X}}\right)\right) \mathbf{G}\Delta(\delta\mathbf{U}). \tag{17}$$

The first term $\Delta\mathbf{B}_{\text{nl}}(\mathbf{u})\delta\mathbf{U}$ is standard in solid element formulations while the second term only appears when large rotations are considered.

3.4 Tangent Stiffness Matrix

The tangent stiffness matrix $\mathbf{K_t}$ is defined using the virtual iterative variation of the work of internal forces such that

$$\Delta(\delta W^{\text{int}}) = \delta\mathbf{U}^{\mathrm{T}} \mathbf{K_t} \Delta\mathbf{U} \tag{18}$$

In this section, we only consider the first part of the virtual iterative variation of $\mathbf{E}$, neglecting the effects of large rotations.

Using Eqs. (16) and (18) and noting that $\mathbf{S}^{\mathrm{T}} \Delta\mathbf{B_{nl}}(\mathbf{u}) = \mathbf{S}^{\mathrm{T}} \mathbf{A}\left(\Delta \mathbf{u}_{,\mathbf{X}}\right) \mathbf{G} = \Delta\mathbf{U}^{\mathrm{T}} \mathbf{G}^{\mathrm{T}} \hat{\mathbf{S}} \mathbf{G}$ we obtain the following stiffness matrix

$$\mathbf{K_t} = \mathbf{K_{t1}} + \mathbf{K_{tf}} = \int_{V_0} \mathbf{B_{nl}}(\mathbf{u})^{\mathrm{T}} \mathbf{C_t}(\mathbf{S}) \mathbf{B_{nl}}(\mathbf{u}) \, \mathrm{d}V_0 + \int_{V_0} \mathbf{G}^{\mathrm{T}} \hat{\mathbf{S}} \mathbf{G} \, \mathrm{d}V_0 \tag{19}$$

where $\hat{\mathbf{S}}$ is defined by

$$\hat{\mathbf{S}} = \begin{bmatrix} [\mathbf{S}] & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & [\mathbf{S}] & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & [\mathbf{S}] \end{bmatrix} \tag{20}$$

The first term $\mathbf{K_{t1}}$ corresponds to the "material" part of the tangent stiffness matrix while the second term $\mathbf{K_{tff}}$ corresponds to the "initial stress" part.

3.5 Finite Rotations

Large displacements and rotations are common in structural analysis. In this section, we extend the displacement-based formulation given in [11] to arbitrary large rotations.

3.5.1 Displacement-Based Formulation

We briefly review the shell formulation given in small transformations. The interpolated displacements and the exact geometry of the shell are described by the same shape functions, that is

$$\mathbf{u}(\boldsymbol{\xi}) = \sum_{A=1}^{nm} R_A \left(\mathbf{U}_A + \frac{h}{2}\zeta(\mathbf{n}_A - \mathbf{n}_A^0) \right) = \sum_{A=1}^{nm} R_A \left(\mathbf{U}_A + \frac{h}{2}\zeta \boldsymbol{\theta}_A \times \mathbf{n}_A^0 \right), \tag{21}$$

where $\boldsymbol{\theta}_A$ is the rotation vector at the control point $\mathbf{X}_A$. It can be highlighted that these displacements are only valid for small rotations if we use a TLF. In this particular case, the simple relation

$$\mathbf{n}_A - \mathbf{n}_A^0 = (\mathbf{R}(\boldsymbol{\theta}_A) - \mathbf{I}_3)\mathbf{n}_A^0 \simeq \boldsymbol{\theta}_A \times \mathbf{n}_A^0 \tag{22}$$

remains true. However, in the general theory of finite transformations, we use the following relation

$$\mathbf{u}(\boldsymbol{\xi}) = \sum_{A=1}^{nm} R_A \left(\mathbf{U}_A + \frac{h}{2}\zeta(\mathbf{R}(\boldsymbol{\theta}_A) - \mathbf{I}_3)\mathbf{n}_A^0 \right). \tag{23}$$

Due to the Reissner-Mindlin's hypothesis, the current director vector $\mathbf{n}_A$ is not normal to the mid-surface of the shell and thus need to be correctly rotated to keep accurate results.

3.5.2 Parametrization of the Rotations

Finite rotations can be described by an orthogonal operator $\mathbf{R}$ or using a pseudo-vector of rotation. Different parametrizations are possible but we focus on the axis-angle representation $(\mathbf{e}, \phi)$ where $\mathbf{e}$ is a unit vector.

The orthogonal operator $\mathbf{R}$ has the three following eigenvalues $\{1, \exp(i\phi), \exp(-i\phi)\}$. The tensor can be represented using the compact form

$$\mathbf{R}(\mathbf{e}, \phi) = \cos(\phi)\mathbf{I} + (1 - \cos(\phi))\mathbf{e} \otimes \mathbf{e} + \sin(\phi)[\mathbf{e}\times], \tag{24}$$

with $||\mathbf{e}|| = 1$ and where the skew-symmetric tensor $[\mathbf{e}\times]$ is

$$\forall \mathbf{b} \in \mathbb{R}^3 : [\mathbf{e}\times]\mathbf{b} = \mathbf{e} \times \mathbf{b}. \tag{25}$$

Noting that, in the Euclidean space, $\mathbf{e} \otimes \mathbf{e} - [\mathbf{e}\times][\mathbf{e}\times] = \mathbf{I}$, we obtain

$$\mathbf{R}(\mathbf{e}, \phi) = \mathbf{I} + (1 - \cos(\phi))[\mathbf{e}\times][\mathbf{e}\times] + \sin(\phi)[\mathbf{e}\times]. \tag{26}$$

The pseudo-vector of rotation $\boldsymbol{\phi}$, of norm ϕ and direction $\mathbf{e}$, is introduced using the relation

$$\boldsymbol{\phi} = \phi\mathbf{e} = ||\boldsymbol{\phi}||\mathbf{e}. \tag{27}$$

In the Euclidean space, we have $\boldsymbol{\phi} = (\phi_1, \phi_2, \phi_3)^{\mathrm{T}}$ with $\phi = \sqrt{\phi_1^2 + \phi_2^2 + \phi_3^2}$. The pseudo-vector of rotation is not a vector since it does not satisfy the property of additivity. In fact, finite rotations of the three-dimensional space are not additive in general. The rotation operator can be rewritten as follows

$$\mathbf{R} = \cos(\phi)\mathbf{I} + \frac{1 - \cos(\phi)}{\phi^2}\boldsymbol{\phi} \otimes \boldsymbol{\phi} + \frac{\sin(\phi)}{\phi}[\boldsymbol{\phi}\times], \tag{28}$$

or recalling that, in the Euclidean space, $\boldsymbol{\phi} \otimes \boldsymbol{\phi} - [\boldsymbol{\phi}\times][\boldsymbol{\phi}\times] = \phi^2\mathbf{I}$, we obtain the standard Rodrigues' rotation formula

$$\mathbf{R} = \mathbf{I} + \frac{1 - \cos(\phi)}{\phi^2}[\boldsymbol{\phi}\times][\boldsymbol{\phi}\times] + \frac{\sin(\phi)}{\phi}[\boldsymbol{\phi}\times], \tag{29}$$

which has no singularity in $[0, 2\pi[$.

Using the Taylor series expansions of the functions sine and cosine, the Rodrigues' formula provides a simple algorithm to compute the exponential map

$$\mathbf{R} = \exp([\boldsymbol{\phi}\times]) = \sum_{k=0}^{\infty} \frac{[\boldsymbol{\phi}\times]^k}{k!} \tag{30}$$

so that the variation of the rotation operator is $\delta\mathbf{R} = [\delta\boldsymbol{\phi}\times]\mathbf{R}$. Moreover, the trigonometric relations

$$1 - \cos(\phi) = 2\sin^2\left(\frac{\phi}{2}\right) \text{ and } \sin(\phi) = 2\sin\left(\frac{\phi}{2}\right)\cos\left(\frac{\phi}{2}\right), \tag{31}$$

allows us to define new dependent parameters such that

$$r_0 = \cos\left(\frac{\phi}{2}\right) \text{ and } \mathbf{r} = \sin\left(\frac{\phi}{2}\right)\frac{\boldsymbol{\phi}}{||\boldsymbol{\phi}||}, \tag{32}$$

with the following relation $r_0^2 + \mathbf{r}\cdot\mathbf{r} = 1$.

The Euler parameters, or the unit quaternion $r = (r_0, \mathbf{r})$, give a simple representation of the orthogonal operator $\mathbf{R}$ as follows

$$\mathbf{R} = \exp([\boldsymbol{\phi}\times]) = (2r_0^2 - 1)\mathbf{I} + 2r_0[\mathbf{r}\times] + 2\mathbf{r}\otimes\mathbf{r}. \tag{33}$$

3.5.3 Variation of the Displacements

We recall that for all points Q in the shell we have

$$\mathbf{u}_Q = \mathbf{u}_P + z(\mathbf{R} - \mathbf{I}_3)\mathbf{n}^0 \tag{34}$$

where P is obtained by an orthogonal projection of Q onto the mid-surface of the shell and $\mathbf{n}^0$ is the initial normal. Noting that $\delta\mathbf{n}^0 = 0$, the virtual displacement $\delta\mathbf{u}_Q$ is expressed as

$$\delta\mathbf{u}_Q = \delta\mathbf{u}_P + z\delta\mathbf{R}\mathbf{n}^0 = \delta\mathbf{u}_P + z[\delta\boldsymbol{\phi}\times]\mathbf{R}\mathbf{n}^0 = \delta\mathbf{u}_P + z\delta\boldsymbol{\phi}\times\mathbf{n}. \tag{35}$$

A similar expression is obtained for the iterative displacement $\Delta\mathbf{u}_Q$, knowing that $\Delta\mathbf{n}^0 = 0$

$$\Delta\mathbf{u}_Q = \Delta\mathbf{u}_P + z\Delta\boldsymbol{\phi}\times\mathbf{n}. \tag{36}$$

Moreover, since we have $\Delta(\delta\mathbf{u}_P) = 0$, the variation for the iterative displacement $\Delta(\delta\mathbf{u}_Q)$ is

$$\Delta(\delta\mathbf{u}_Q) = z\delta\boldsymbol{\phi}\times(\Delta\boldsymbol{\phi}\times\mathbf{n}). \tag{37}$$

This term is not standard and does not exist in the classical theory of continuum shell elements. It implies a new contribution in the stiffness matrix which is non-symmetric. Further details about this additional term can be found in [17].

3.5.4 Update of the Rotations and Normals

The quaternion algebra gives simple formulas to update the pseudo-vector of rotation Φ and the current normal to the shell $\mathbf{n}$.

Given the two rotations $\mathbf{R}_1$ and $\mathbf{R}_2$, respectively represented by the quaternions r^1 and r^2, the resulting composition of the two rotations, namely $\mathbf{R} = \mathbf{R}_2\mathbf{R}_1$, is

parametrized by the quaternion r as follows

$$r = r^1 \circ r^2 = (r_0^1 r_0^2 - \mathbf{r}^1 \cdot \mathbf{r}^2, r_0^1 \mathbf{r}^2 + r_0^2 \mathbf{r}^1 + \mathbf{r}^1 \times \mathbf{r}^2), \tag{38}$$

where $r^1 \circ r^2$ denotes the product of the two quaternions r^1 and r^2. One may note that due to the presence of the tensor product $\mathbf{r}^1 \times \mathbf{r}^2$, the composition of two rotations in a three-dimensional space is not, in general, commutative.

Suppose it is known, at the increment i, the total rotation $\mathbf{R}_{i-1}$, that is its representation by the quaternion r^{i-1}, the field $(\mathbf{u}_{i-1}, \boldsymbol{\phi}_{i-1})$, its iterative variation $(\Delta \mathbf{u}_i, \Delta \boldsymbol{\phi}_i)$ and the normal $\mathbf{n}^{i-1}$. The displacement $\mathbf{u}$ is straightforward updated by the additive formula

$$\mathbf{u}_i = \mathbf{u}_{i-1} + \Delta \mathbf{u}_i. \tag{39}$$

However, a similar update can not be applied to the rotation $\boldsymbol{\phi}$ since the property of additivity is only valid for small rotations.

To update the rotation $\boldsymbol{\phi}$ using the computed iterative value $\Delta \boldsymbol{\phi}_i$, we first compute the quaternion Δr^i which parametrizes the iterative rotation

$$\Delta r^i = (\Delta r_0^i, \Delta \mathbf{r}^i) = \left(\cos\left(\frac{||\Delta \boldsymbol{\phi}_i||}{2} \right), \sin\left(\frac{||\Delta \boldsymbol{\phi}_i||}{2} \right) \frac{\Delta \boldsymbol{\phi}_i}{||\Delta \boldsymbol{\phi}_i||} \right). \tag{40}$$

The rotation field is thus updated using the product $r^i = \Delta r^i \circ r^{i-1}$

$$r^i = (r_0^i, \mathbf{r}^i) = (\Delta r_0^i r_0^{i-1} + \Delta \mathbf{r}^i \cdot \mathbf{r}^{i-1}, \Delta r_0^i \mathbf{r}^{i-1} + r_0^{i-1} \Delta \mathbf{r}^i + \Delta \mathbf{r}^i \times \mathbf{r}^{i-1}). \tag{41}$$

Recalling that $r = (\cos(\frac{\phi}{2}), \sin(\frac{\phi}{2})\mathbf{e})$, the pseudo-vector of rotation $\boldsymbol{\phi}$ can be defined in $[0, 2\pi[$ using the relations

$$\phi^i = 2 \tan^{-1}\left(\frac{||\mathbf{r}^i||}{r_0^i} \right) \text{ and } \boldsymbol{\phi}^i = \phi^i \frac{||\mathbf{r}^i||}{\mathbf{r}^i}. \tag{42}$$

The director vector $\mathbf{n}^i = \exp([\Delta \boldsymbol{\phi}^i \times])\mathbf{n}^{i-1}$ is easily updated as follows

$$\mathbf{n}^i = (2(\Delta r_0^i)^2 - 1)\mathbf{n}^{i-1} + 2\Delta r_0^i \Delta \mathbf{r}^i \times \mathbf{n}^{i-1} + 2(\Delta \mathbf{r}^i \cdot \mathbf{n}^{i-1})\Delta \mathbf{r}^i. \tag{43}$$

4 Reduced Integration

The reduced quadrature rules for NURBS-based analysis of structural mechanics given in [10, 11] are extended, in this section, to analysis suitable T-meshes.

We recall that due to the high order of the isogeometric method, a full Gaussian integration with $\lceil (p+1)/2 \rceil$ quadrature points in each direction is computationally expensive. Moreover, due to the high continuity of the spline functions, the full

integration provides to many quadrature points in the element [5]. The reduced integration presents the double advantage to significantly reduce the computational cost of the method and to alleviate the numerical locking occurring when modelling thin structures [11].

4.1 Bézier Elements and Full Gauss Quadrature

As for B-splines and NURBS, the continuity of T-splines in the physical space is determined by the local knot vectors, defining the blending functions in the parameter space. For polynomials of order p, a multiplicity m of a knot value in the local knot vector implies a regularity C^{p-m} of the blending functions across this knot value. Unlike Cartesian structure of B-splines and NURBS, in which the lines of reduced continuity are global and unambiguous, the lines of reduced continuity in a T-mesh are local to the basis functions and do not generally propagate throughout the domain. Therefore, T-splines may have locally several degrees of continuity within the domain as previously illustrated in Fig. 3.

We have introduced the concept of T-junction extension which, in practical, is a simple and efficient tool to verify whether the T-mesh is suitable for analysis or not. We recall that these T-junction extensions are lines of reduced continuity. Therefore, an extended T-mesh can be constructed adding all continuity reduction lines to the initial T-mesh. The union off all edges and T-junction extensions represents the lines across which the continuity of the blending functions is less than C^{∞}. The extended T-mesh splits the parameter and physical spaces into several regions which define the Bézier elements. This notion of element is essential to perform a correct numerical quadrature.

The numerical integration is thus performed in the regions delimited by the edges of the extended T-mesh which correspond to the Bézier elements. Figure 4 illustrates

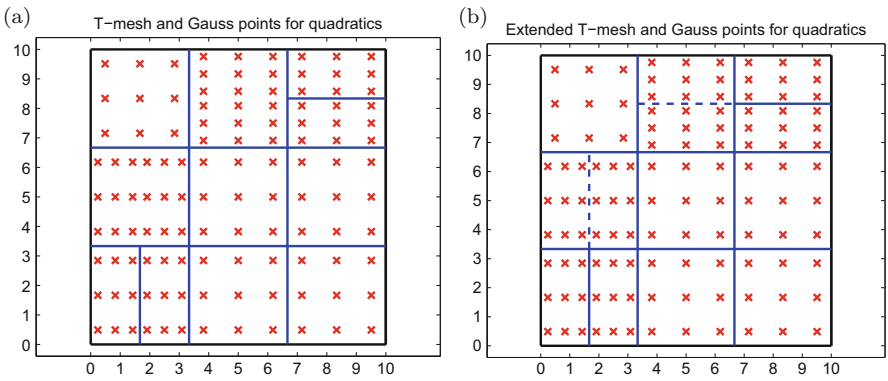


Fig. 4 T-mesh (**a**) and Bézier elements (**b**) with full Gauss quadrature for bi-quadratic blending functions

the full Gauss quadrature for analysis suitable T-splines. The numerical quadrature is performed in each Bézier elements, represented on the right, which are defined adding the lines or reduced continuity (dotted blue lines) to the edges of the T-mesh (solid blue lines).

4.2 Reduced Gauss Quadrature

The non-uniform reduced Gauss quadrature presented in [10] in the context of B-spline meshes, is extended in this work to unstructured analysis suitable T-meshes. The reduced number of quadrature points used in B-spline/NURBS analysis has shown to alleviate numerical locking as well as to improve the computational efficiency, especially as the degree of continuity is high. As with a full Gauss quadrature, the reduced rule is applied in each Bézier element in which the continuity of the blending function is C^{∞}. In practical terms, the local continuity of each Bézier element edge is given by the local knot vectors of the blending functions that are positive on this element.

Figure 5 represents the same analysis suitable T-mesh as previously and the non-uniform reduced Gauss quadrature rule which is constructed locally, namely in a standard element-by-element fashion. Using the concepts introduced in [12] for quadratic polynomials, the target space is $\mathcal{S}_{q-1}^{2p} = \mathcal{S}_0^4$, achieved with a full Gauss or patchwise quadrature, and the approximation space corresponding to this given scheme is $\mathcal{S}_s^r = \mathcal{S}_{-1}^1$.

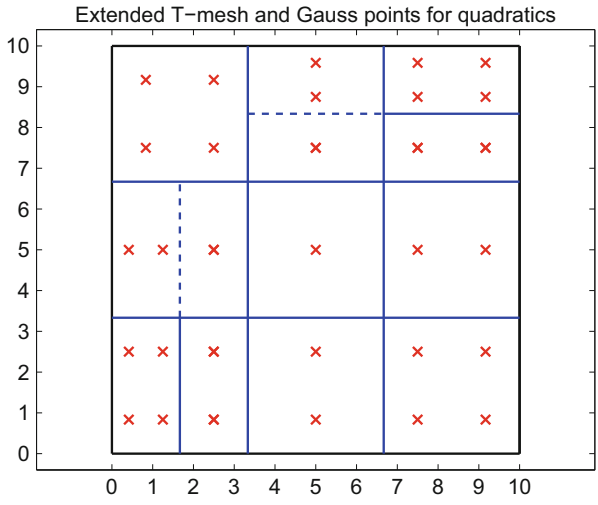


Fig. 5 Extended T-mesh and reduced Gauss quadrature rule

4.3 Reduced Patchwise Quadrature

The reduced patchwise rules given in [12] are applied in the context of T-spline analysis. We recall that, in this work, it is shown that the reduced Gauss rule is not optimal in term of accuracy since it exactly integrates the approximation space $\mathcal{S}^1_{-1}$ which is not included in the target space $\mathcal{S}^4_0$. For C^1 quadratic shape functions, the over-integrated approximation space $\mathcal{S}^2_0$ appears to be optimal when performing a reduced integration in the sense that it significantly reduces the numerical locking and computational effort without introducing Hourglass modes. Moreover, it is the smallest space that guarantees these properties for C^1 quadratics.

The reduced patchwise rules are constructed by tensor products of several one-dimensional rules. We look through all the Bézier elements of the extended T-mesh in both parametric directions since the patchwise rules are not constructed locally, that is from the continuity given by the local knot vectors associated with the positive functions in the element. The complexity of the T-mesh, namely the number of holes or T-junctions, the different degrees of continuity, defines the number of one-dimensional rules to compute in order to construct the two-dimensional quadrature scheme. However, each construction of a one-dimensional rule is independent, that is it can be straightforward performed in parallel. A patchwise one-dimensional rule is defined in a given parametric direction and between two local knots of C^{-1} continuity, namely between two borders of the domain.

Figure 6 represents the two one-dimensional rules that have to be constructed in order to compute the two-dimensional scheme. Again, we use the same T-mesh as with the other presented integrations. We observe that the same number of quadrature points is used as with the non-uniform reduced Gauss quadrature. However, the locations of these points are not standard since the integration rule is not defined locally, it depends on the element in the patch.

Due to the flexible structure of the T-splines, in particular the possibility to perform a local refinement, blending functions of generally distinct supports lie in

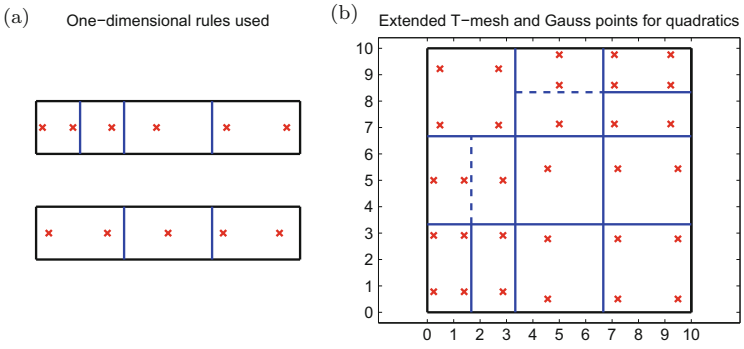


Fig. 6 Extended T-mesh and reduced patchwise quadrature rule. (**a**) One-dimensional patchwise reduced rules used. (**b**) Reduced patchwise quadrature scheme

the element. Therefore, the patchwise rule can not, in principle, exactly integrate all functions within the element. However, we have seen that the refined blending functions can be expressed as a linear combination of the coarse functions when performing a blending function subdivision. Considering the Bézier elements for the construction of the quadrature rules guarantees that all the blending functions in the element, even with a coarse support, are exactly integrated. We note that some shape functions are thus over-integrated near a T-junction, which may imply a slight numerical locking since there are locally more quadrature points than control points. This observation can be made for both quadrature rules presented in this work. For instance, the one-dimensional rule with six quadrature points, presented in Fig. 6, also exactly integrates the blending functions which are exactly integrated by the other one-dimensional rule with only five quadrature points.

5 Numerical Results

In this section, we perform some classical benchmarks for structural analysis. We present numerical solutions for linear elastic shells and geometric non-linear structures. All the calculations are done for the thick shell formulation presented in the previous section. A standard Galerkin formulation is employed. Full Gauss, non-uniform reduced Gauss and reduced patchwise quadratures are used for the numerical integration and compared in several examples. The first example is a bending plate which exhibits strong numerical locking under mesh distortion with full Gauss-Legendre quadrature. The two following examples come from the so-called shell obstacle course: the pinched cylinder and the pinched hemisphere. The relevance of these problems to the assessment of shell analysis have been discussed extensively in the literature [18]. The pinched hemisphere is also considered within the context of geometric non-linear analysis to assess the performance of the shell formulation and the numerical quadrature with large transformations. A set of popular benchmark problems for geometric non-linear analysis of shells is given in [19].

5.1 Bending Plate

The first example, represented in Fig. 7, is a simply supported thin square plate, of length L, with a vertical concentrated loading applied at its center. The case of a uniform pressure loading is not presented in this paper but similar results are obtained. Due to the symmetry of the problem, only a quarter of the plate (blue area in Fig. 7) is simulated. The geometric property $L/h = 10^3$ results in a thin shell structure which is expected to suffer from locking pathologies, in particular transverse shear locking. In this example, no T-junction is considered. We first assess

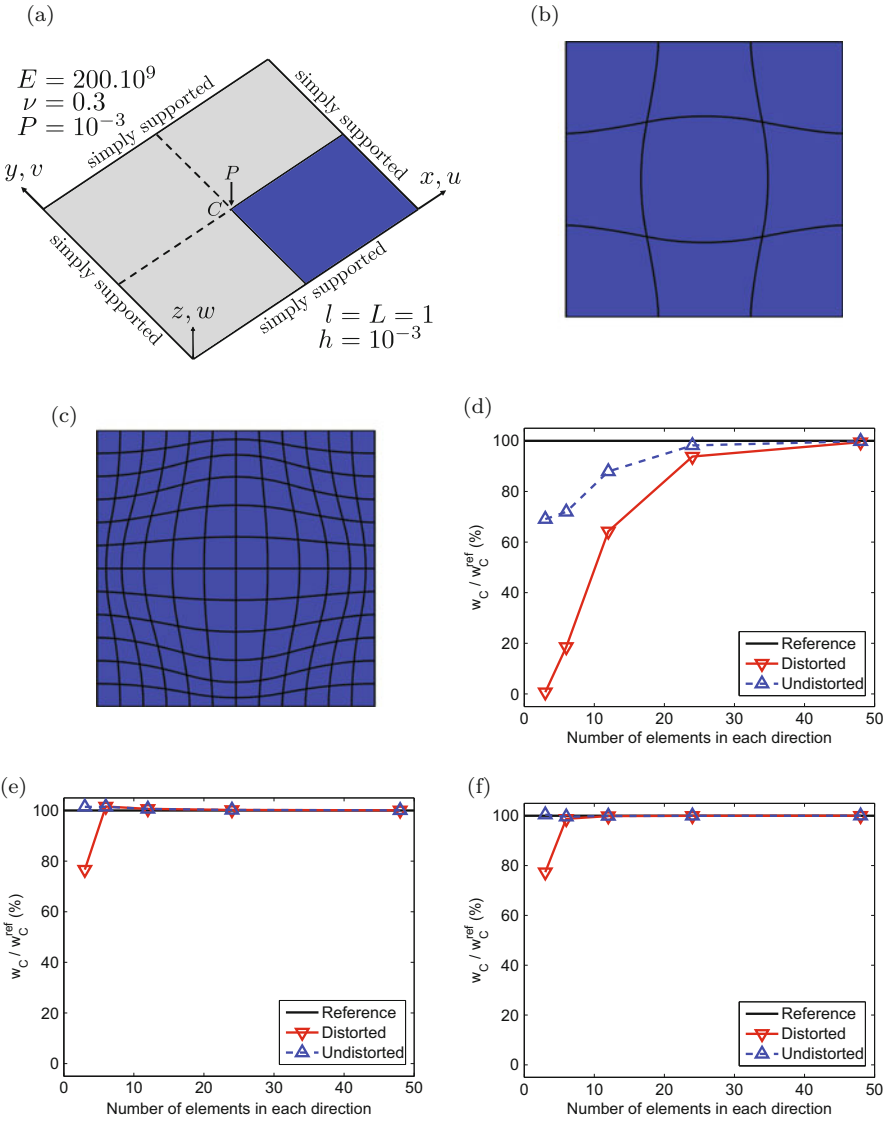


Fig. 7 Central deflection of the bending plate for different numerical quadratures. (**a**) Problem description and geometric data. (**b**) Distorted mesh 3×3. (**c**) Distorted mesh 12×12. (**d**) Full Gauss-Legendre quadrature. (**e**) Reduced Gauss-Legendre quadrature. (**f**) Reduced patchwise quadrature

the performance of both presented reduced rules with respect to mesh quality and numerical locking.

For each integration, namely full, reduced Gauss-Legendre and reduced patch-wise, two meshes are compared: a regular mesh and a distorted mesh. The numerical

results are compared with the reference central deflection $w_C^{\text{ref}} = 6.3341 \cdot 10^{-6}\,\text{m}$ [11]. An initial mesh distortion is applied on the coarsest mesh with three elements in each direction. A strategy of k-refinement is then applied to generate the refined meshes without modifying the initial mesh distortion.

Figure 7 illustrates that full quadrature is sensitive to mesh distortion in the sense that it strengthens the locking pathologies and decreases the accuracy of the numerical solution. However, in this example, both reduced quadratures are not sensitive to numerical locking neither to mesh distortion. The accuracy of the solution is significantly improved with respect to the full quadrature. Moreover, we have the following relation for the ratio of the number of quadrature points

$$\lim_{n_{\text{elem}} \to +\infty} n_{\text{GP}}^{\text{full}} / n_{\text{GP}}^{\text{reduced}} = 4, \tag{44}$$

which means that the computational effort is also significantly reduced.

5.2 *Pinched Cylinder*

The second example is the so called pinched cylinder subjected to equal and opposite radial forces at the top and bottom mid-spans. The two ends are supported by rigid diaphragms (Fig. 8). The geometry of the cylindrical shell exhibits the shape ratio as in the previous problem. This is a severe test case evaluate the capacity of the shell element to describe the membrane and bending deformations, especially near the application points of the concentrated forces. This problem is characterized by a bending dominant behaviour and results have revealed a tendency towards membrane and shear locking. A severe membrane locking is expected as bending is inextensible. Again, due to the symmetry of the problem, only a quarter of the shell (blue area in Fig. 8a) is modelled. To evaluate the convergence, the vertical displacement w_C^h at the loading points is compared with the reference solution $w_C^{\text{ref}} = 1.8248 \cdot 10^{-7}\,\text{m}$ [11].

Several NURBS meshes are generated using the standard k-refinement procedure. The T-meshes are represented in Fig. 8. We assess the performance of both full and reduced Gauss-Legendre quadratures for NURBS and T-spline meshes. We observe that combining the reduced integration with the locally refined T-meshes significantly improves the accuracy of the numerical solution using less degrees of freedom and integration points. It thus appears as a promising technique to perform complex simulations, involving localized deformations, with a reasonable computational effort.

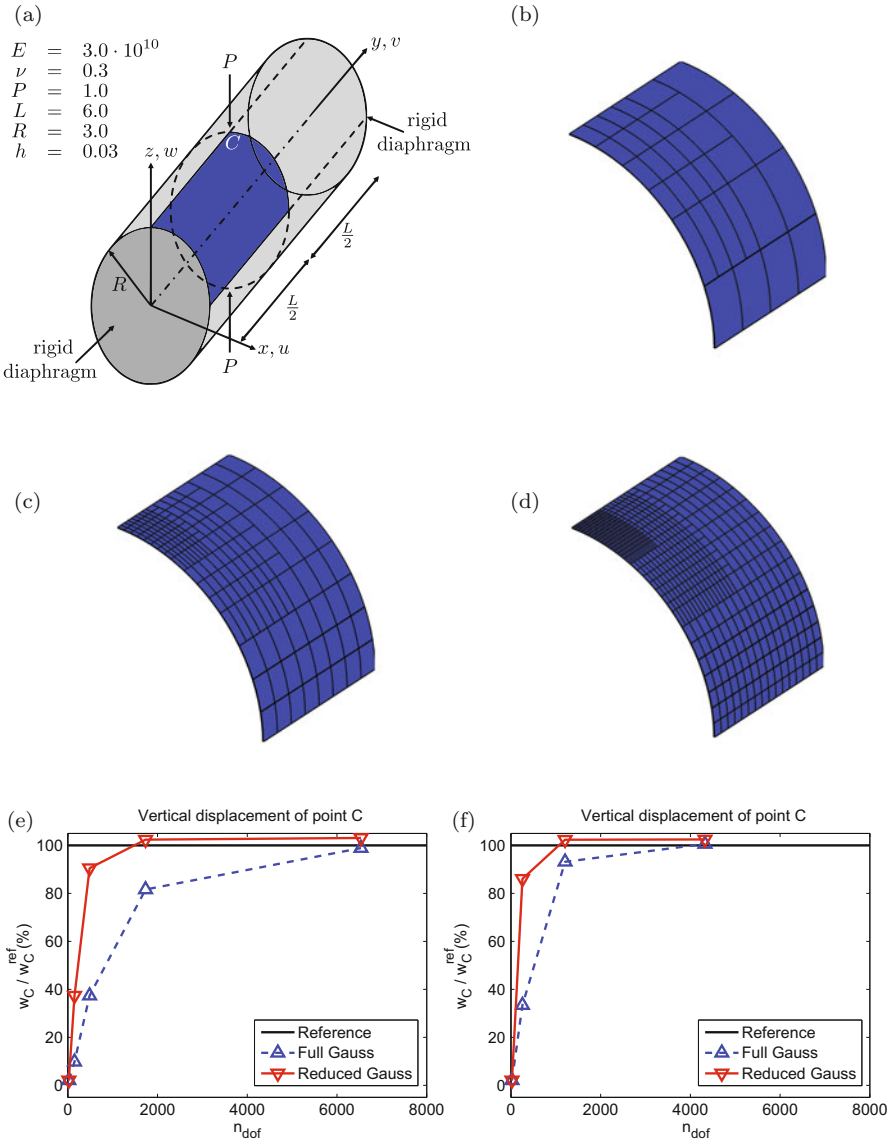


Fig. 8 Vertical displacement at the center of the pinched cylinder for different numerical quadratures and types of meshes. (**a**) Problem description and geometric data. (**b**) $n_{\text{dof}} = 258$. (**c**) $n_{\text{dof}} = 1206$. (**d**) $n_{\text{dof}} = 4326$. (**e**) NURBS meshes. (**f**) T-spline meshes

5.3 *Pinched Hemisphere*

The last example considered in this work is the pinched hemisphere, with an 18° hole at its pole, which comes from a set of popular benchmark problems for geometric non-linear analysis of shells [19]. The pinched hemispherical shell, represented in Fig. 9, is subjected to equal and opposite concentrated forces $P_{\max}$ applied at the four cardinal points of its equator. The two ends are stress free. The rigid body motion is eliminated fixing the vertical displacement at an application point of the forces. The slenderness ratio is $R/h = 250$. This doubly curved shell problem is characterized by inextensible bending modes and large rigid body rotations. Membrane energy is almost zero. This test ensures that the element does not suffer from membrane and shear locking. Due to symmetry, only one quadrant is meshed (blue area in Fig. 9). To assess the accuracy of the shell formulation, the horizontal displacement u_A^h at the loading points is compared with the reference solution using ABAQUS S4R, a four-node reduced shell element, given in [19].

In the non-linear solution procedure, the full Newton-Raphson method is used in a TLF. The convergence criterion employed is the 0.1 % force tolerance. The maximum load $P_{\max}$ is subdivided into several load increments which are not necessarily uniform. At the end of each load increment, a converged intermediate solution is obtained. This procedure reduces the degree of non-linearity from an intermediate solution state to another and enhances the chance of obtaining the desired solution corresponding to the maximum load. If the increments are correctly chosen or computed, each step converges in a few iterations.

A standard NURBS mesh with twelve C^1 quadratic elements in each direction is used. Figure 9 illustrates that, as with linear analysis, the reduced Gauss scheme provide a better accuracy than the full Gauss-Legendre quadrature within the context of non-linear analysis. The shell formulation with finite transformations also exhibits a good accuracy. The resulting reduced isogeometric shell element shows

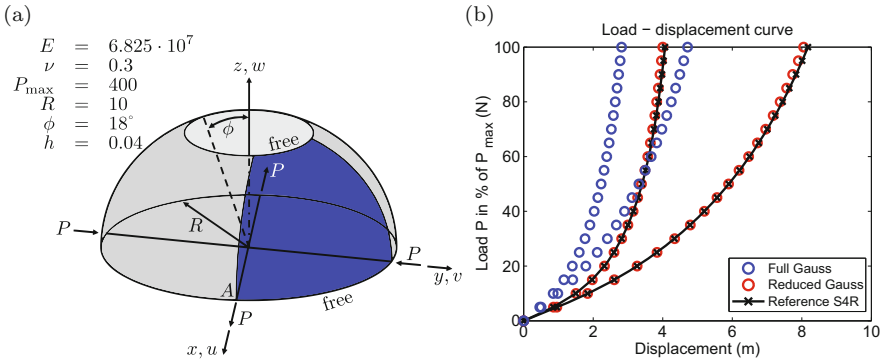


Fig. 9 Horizontal displacement at the center of the pinched cylinder for different numerical quadratures and types of meshes. (**a**) Problem description and geometric data. (**b**) Numerical solution for a 12 × 12 mesh

satisfying performance. If a local refinement, using T-junctions, is added, accuracy and computational efficiency are even more improved, as shown in the previous examples.

6 Conclusions

The concept of T-splines extends the NURBS technology and permits local refinement, holes, which is a solution to the trimmed surface problem, merging of patches without introducing new knots which propagate. These properties are essential in a design context and T-splines represent a promising integrated approach to engineering design and analysis. This technology is attractive for both design and analysis and has recently become available in industrial design software.

In this paper, we explore various integration schemes and adapt the B-spline/NURBS tensor-product rules to T-splines. The uni-dimensional reduced quadrature rules for quadratic shape functions, with non-uniform Gauss-Legendre or patchwise integration, have been extended to bi-dimensional shell problems within the context of unstructured T-splines. A geometric non-linear Reissner-Mindlin shell formulation, based on the displacements, has been proposed. The update of the finite rotations in a TLF is handled using the quaternion algebra.

The non-uniform reduced Gauss-Legendre integration can be straightforward extended to more complex T-meshes. The local continuity of each Bézier element edge, required to construct the reduced element rule, is given by the local knot vectors of the blending functions that are positive on this element. However, we look through all the Bézier elements of the extended T-mesh in both parametric directions to construct the one-dimensional patchwise rules essential for the final two-dimensional tensor-product rule. The latter require a higher computational effort than Gauss rules, e.g. one calculation during the pre-processing phase, but they also offer a better accuracy.

The performance of the resulting reduced shell element is assessed on several linear and geometric non-linear problems. We have observed that, C^1 quadratic blending functions are sensitive to mesh distortion and numerical locking when performing a full integration with a thin structure. They both decrease the accuracy of the numerical solution. However, the reduced integrations alleviate these pathologies and exhibit good performance in terms of accuracy and computational efficiency.

Acknowledgements This work was supported by the "OpenLab Computational Mechanics" created collectively by PSA Peugeot Citroën and the Laboratoire de Mécanique des Solides, École Polytechnique. This support is gratefully acknowledged.

References

1. T.J.R Hughes, J.A. Cottrell, Y. Bazilevs, Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. Comput. Methods Appl. Mech. Eng. **194**, 4135–4195 (2005)
2. L. Piegl, W. Tiller, *The NURBS Book*, 2nd edn. (Springer, New York, 1997)
3. T.W. Sederberg, J. Zheng, A. Bakenov, A. Nasri, T-splines and T-NURCCSs. ACM Trans. Graph. **22**, 477–484 (2003)
4. A.-V. Vuong, C. Giannelli, B. Jüttler, B. Simeon, A hierarchical approach to adaptive local refinement in isogeometric analysis. Comput. Methods Appl. Mech. Eng. **200**, 3554–3567 (2011)
5. T.J.R. Hughes, A. Reali, G. Sangalli, Efficient quadrature for NURBS-based isogeometric analysis. Comput. Methods Appl. Mech. Eng. **199**, 301–313 (2010)
6. R. Echter, M. Bischoff, Numerical efficiency, locking and unlocking of NURBS finite elements. Comput. Methods Appl. Mech. Eng. **199**, 374–382 (2010)
7. R. Echter, B. Oesterle, M. Bischoff, A hierarchic family of isogeometric shell finite elements. Comput. Methods Appl. Mech. Eng. **254**, 170–180 (2013)
8. D. Schillinger, S.J. Hossain, T.J.R. Hughes, Reduced Bézier element quadrature rules For quadratic and cubic splines in isogeometric analysis. Comput. Methods Appl. Mech. Eng. **277**, 1–45 (2014)
9. J.C. Simo, D. Fox, M.S. Rifai, On the stress resultant geometrically exact shell model. Part III: computational aspects of the nonlinear theory. Comput. Meth. Appl. Mech. Eng. **79**, 21–70 (1990)
10. C. Adam, S. Bouabdallah, M. Zarroug, H. Maitournam, Improved numerical integration for locking treatment in isogeometric structural elements, Part I: beams. Comput. Methods Appl. Mech. Eng. **278**, 1–28 (2014)
11. C. Adam, S. Bouabdallah, M. Zarroug, H. Maitournam, Improved numerical integration for locking treatment in isogeometric structural elements, Part II: plates and shells. Comput. Methods Appl. Mech. Eng. **284**, 106–137 (2015)
12. C. Adam, T.J.R. Hughes, S. Bouabdallah, M. Zarroug, H. Maitournam, Selective and reduced numerical integration for NURBS-based isogeometric analysis. Comput. Methods Appl. Mech. Eng. **284**, 732–761 (2015)
13. C. Adam, A contribution to static and dynamic isogeometric analysis of structures. Ph.D. thesis, École Polytechnique (2015)
14. Y. Bazilevs, V.M. Calo, J.A. Cottrell, J.A. Evans, T.J.R. Hughes, S. Lipton, M.A. Scott, T.W. Sederberg, Isogeometric analysis using T-splines. Comput. Methods Appl. Mech. Eng. **199**, 229–263 (2010)
15. T.W. Sederberg, D.L. Cardon, G.T. Finnigan, N.S. North, J. Zheng, T. Lyche, T-spline simplification and local refinement. ACM Trans. Graph. **23**, 276–283 (2004)
16. M.A. Scott, X. Li, T.W. Sederberg, T.J.R. Hughes, Local refinement of analysis-suitable T-splines. Comput. Methods Appl. Mech. Eng. **213**, 206–222 (2012)
17. M.A. Crisfield, *Non-linear Finite Element Analysis of Solids and Structures* (Wiley, West Sussex, 1996)
18. R. MacNeal, R. Harder, A proposed standard set of problems to test finite element accuracy. Finite Elem. Anal. Des. **1**, 3–20 (1985)
19. K.Y. Sze, X.H. Liu, S.H. Lo, Popular benchmark problems for geometric nonlinear analysis of shells. Finite Elem. Anal. Des. **40**, 1551–1569 (2004)

Multiresolution Shape Optimisation with Subdivision Surfaces

Fehmi Cirak and Kosala Bandara

Abstract We review our recent work on multiresolution shape optimisation and present its application to elastic solids, electrostatic field equations and thin-shells. In the spirit of isogeometric analysis the geometry of the domain is described with subdivision surfaces and different resolutions of the same surface are used for optimisation and analysis. The analysis is performed using a sufficiently fine control mesh with a fixed resolution. During shape optimisation the geometry is updated starting with the coarsest control mesh and then moving on to increasingly finer control meshes. The transfer of data between the geometry and analysis representations is accomplished with subdivision refinement and coarsening operators. Moreover, we discretise elastic solids with the immersed finite element method, electrostatic field equations with the boundary element method and thin-shells with the subdivision finite element technique. In all three discretisation techniques there is no need to generate and maintain an analysis-suitable volume discretisation.

1 Introduction

As widely discussed in isogeometric analysis literature, the geometry representations used in today's computer aided design (CAD) and computational analysis software are inherently incompatible [19]. This is particularly limiting in shape optimisation in which a geometry model is iteratively updated based on the results of a computational analysis [6, 8, 17]. It is tedious and often impossible to repeatedly map the results from an analysis mesh back to the CAD model. To remedy this, the shape optimisation of shells and solids by directly optimising the CAD geometry model using isogeometric analysis has been recently explored [12, 16, 21, 37].

In the present paper we review the multiresolution shape optimisation technique and present its application to linear elastic solids, electrostatic field equations and thin-shell structures [2–4]. We discretise elastostatic boundary value problems using immersed, or embedded, finite elements, see, e.g., [28, 29, 31, 32], which have clear

F. Cirak (✉) • K. Bandara
Department of Engineering, University of Cambridge, Cambridge, CB2 1PZ, UK
e-mail: fc286@cam.ac.uk; kkmb2@cam.ac.uk

B. Jüttler, B. Simeon (eds.), *Isogeometric Analysis and Applications 2014*,
Lecture Notes in Computational Science and Engineering 107,
DOI 10.1007/978-3-319-23315-4_6

advantages when applied to structural shape optimisation [1, 18]. In order to cope with unbounded domains we discretise electrostatic problems with the boundary element method. Crucially, in immersed finite elements and boundary elements the geometry of the domain boundary can be updated without needing to generate or maintain domain meshes. Shell structures are modelled as a surface and are discretised with subdivision finite elements, which use the subdivision surfaces as basis functions [11, 12].

The domain boundaries and the shell mid-surface are represented with subdivision surfaces. Although historically subdivision and related techniques have originated in computer graphics, they recently became available in several CAD software packages, including Autodesk Fusion 360, PTC Creo and CATIA. As will be demonstrated in this paper, subdivision surfaces provide an elegant isogeometric analysis-suitable, bidirectional mapping between the geometry and analysis models. In subdivision a geometry is described using a control mesh and a limiting process of repeated refinement [27, 40]. The refinement rules are usually adapted from knot refinement rules for splines [9, 14, 22]. We consider the Catmull-Clark [9] scheme based on quadrilateral meshes and the Loop [25] scheme based on triangular meshes. The Catmull-Clark scheme is the generalisation of cubic tensor-product b-splines to unstructured meshes and the Loop scheme is the generalisation of quartic box-splines. Both subdivision schemes lead to smooth surfaces even in case of unstructured meshes with extraordinary vertices. The hierarchy of control meshes underlying subdivision surfaces lends itself naturally to multiresolution decomposition of geometries [26, 41]. The size of the geometric region influenced by each vertex depends on the resolution of the control mesh, editing coarser levels leads to large-scale changes while editing finer levels lead to small-scale changes.

The introduced multiresolution optimisation approach relies on subdivision curves/surfaces for the description of boundaries. The multiresolution paradigm allows us to describe the same geometry with control meshes of different resolution for analysis and optimisation purposes. For finite element and boundary element analysis a relatively fine control mesh is used in order to minimise the discretisation errors. In contrast, the degrees of freedom in optimisation (i.e., design variables) are chosen as the vertex coordinates of a coarser control mesh. Perhaps counterintuitively, the use of the same fine control mesh for optimisation and analysis leads to suboptimal optimisation results with possible non-physical geometry oscillations. This behaviour is also known from earlier work on structural optimisation in which the finite element nodes were used as optimisation design variables [8, 17]. This paper will demonstrate that the best optimisation results are achieved when starting with a coarse control mesh increasingly finer control meshes are optimised. During the optimisation iterations the refinement level of the control mesh is increased each time a minimum is reached.

2 Multiresolution Subdivision Surfaces

2.1 Subdivision Refinement of Spline Curves

The refinability property of cubic b-splines can be utilised to derive a corresponding subdivision scheme. To illustrate this, we consider the coarse knot sequence $\xi_i^0 = 0, 1, 2, 3, \ldots$ and the fine knot sequence $\xi_i^1 = 0, 0.5, 1, 1.5, 2, 2.5, 3, \ldots$. We denote the b-splines on the coarse knot sequence with $B_i^0(\xi)$ and the ones on the fine knot sequence with $B_i^1(\xi)$, see Fig. 1. According to the b-spline refinability equation, see, e.g., [7, 40], it is possible to represent the coarse b-splines (in the interior) as a linear combination of the fine b-splines

$$B_i^0(\xi) = \sum_{j=0}^{4} S_{ij} B_{2i+j-2}^1(\xi) \quad \text{with} \quad S_{ij} = \frac{1}{8}\binom{4}{i}, \tag{1}$$

where S_{ij} is the subdivision matrix with its entries given in terms of the usual binomial coefficients. In order to keep the derivations simple we exclude from our discussion the b-splines close to the boundaries. For subsequent derivations (1) is best expressed in matrix notation

$$\boldsymbol{B}^0 = \boldsymbol{S}\boldsymbol{B}^1 . \tag{2}$$

Next, we consider a spline curve defined in terms of the coarse b-splines and the corresponding control vertices, i.e.,

$$\boldsymbol{x}_h(\xi) = \boldsymbol{B}^0 \cdot \boldsymbol{x}^0 \tag{3}$$

with matrix $\boldsymbol{x}^0$ containing the coordinates of control vertices. The number of columns of $\boldsymbol{x}^0$ is equal to space dimension and the number of rows is equal to the number of all control vertices.

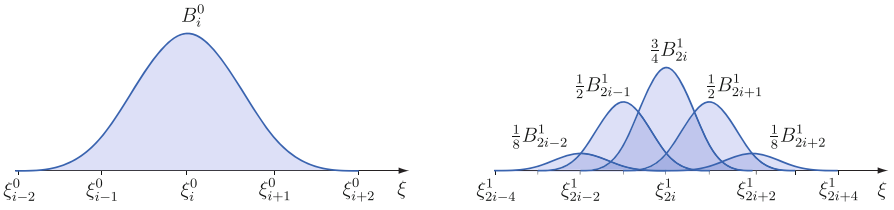


Fig. 1 Refinement relation for cubic b-splines

Introducing the refinement relation (2) into (3) the spline curve can be expressed with

$$\boldsymbol{x}_h(\xi) = \left(\boldsymbol{S}\boldsymbol{B}^1\right) \cdot \boldsymbol{x}^0 = \boldsymbol{B}^1 \cdot \left(\boldsymbol{S}^\mathsf{T}\boldsymbol{x}^0\right) . \tag{4}$$

This implies that the control vertex coordinates on the finer level can be computed with the subdivision relation

$$\boldsymbol{x}^1 = \boldsymbol{S}^\mathsf{T}\boldsymbol{x}^0 . \tag{5}$$

In subdivision schemes the described refinement approach is applied recursively, i.e.,

$$\boldsymbol{x}^{\ell+1} = \boldsymbol{S}^\mathsf{T}\boldsymbol{x}^\ell , \tag{6}$$

where $\boldsymbol{x}^{\ell+1}$ and $\boldsymbol{x}^\ell$ are two matrices containing the coordinates of all the vertices at levels $\ell+1$ and ℓ with $\ell \geq 0$. Although the dimensions of $\boldsymbol{S}$ increase with ℓ we denote all subdivision matrices with $\boldsymbol{S}$ since each row has the same non-zero components only shifted relative to adjacent rows [3]. The successive refinement of a given control polygon using (6) is illustrated in Fig. 2.

For computer implementation and generalisation to bivariate splines, it is instructive to think that the refinement of a control polygon according to (6) consists of a refinement and an averaging step. In the refinement step each segment of the polygon is subdivided into two segments, see Fig. 3a. Subsequently, the vertex

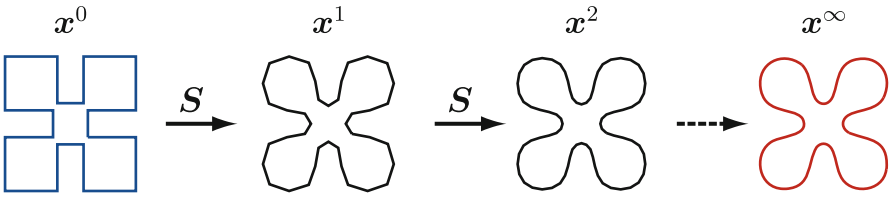


Fig. 2 Subdivision refinement of a given control polygon (shown *left*)

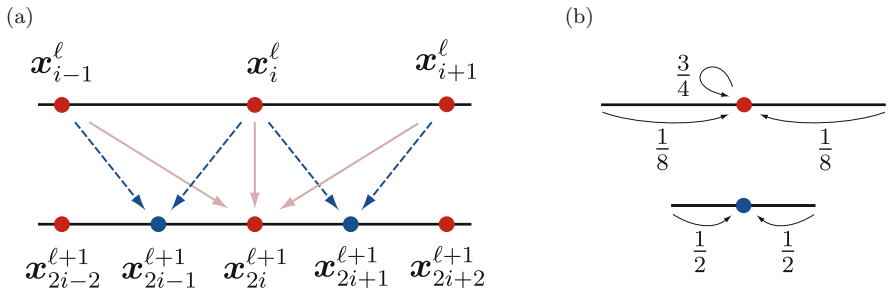


Fig. 3 Subdivision refinement for cubic spline curves. (**a**) Refinement by bisectioning and vertex renumbering. (**b**) Stencils for even (*top*) and odd (*bottom*) vertices

coordinates of the refined polygon are determined by averaging the coarse vertex coordinates with the two stencils shown in Fig. 3b. The even vertex stencil applies to vertices that are already present in the coarse polygon and the odd vertex stencil applies to vertices that are only present in the refined polygon. The naming odd and even is motivated by the consecutive numbering of vertices where newly inserted vertices receive odd numbers. According to Fig. 3b, for a given coarse polygon of level ℓ with vertex coordinates $\boldsymbol{x}_i^\ell$ a refined polygon of level $\ell + 1$ with vertex coordinates $\boldsymbol{x}_i^{\ell+1}$ is computed with

$$\boldsymbol{x}_{2i}^{\ell+1} = \frac{1}{8}\boldsymbol{x}_{i-1}^{\ell} + \frac{3}{4}\boldsymbol{x}_i^{\ell} + \frac{1}{8}\boldsymbol{x}_{i+1}^{\ell}\,, \qquad \boldsymbol{x}_{2i+1}^{\ell+1} = \frac{1}{2}\boldsymbol{x}_i^{\ell} + \frac{1}{2}\boldsymbol{x}_{i+1}^{\ell}\,. \tag{7}$$

2.2 *Subdivision Refinement of Bivariate Surfaces*

Similar to tensor-product b-splines the tensor products of one-dimensional subdivision stencils yield the corresponding subdivision stencils for surfaces. The tensor product construction works only for quadrilateral meshes and in the refinement step each quadrilateral is subdivided into four quadrilaterals, see Fig. 4a. It is evident that the tensor-product stencils only apply to meshes in which each vertex within the domain is connected to four faces. The number of faces connected to a vertex is referred to as the valence of that vertex. The domain vertices with a valence other than four are known as extraordinary vertices or star-vertices.

For subdivision surfaces using quadrilateral meshes, the original stencils proposed by Catmull and Clark [9], shown in Fig. 5, are used. Note, the vertex stencil in Fig. 5c depends on the valence v of the vertex and reduces to a standard tensor product stencil for structured meshes with valence $v = 4$. There is mathematical theory which shows that the resulting surface is C^2 continuous almost everywhere except at the extraordinary vertices where it is only C^1 continuous [27]. In addition, there are also extended subdivision stencils for vertices on edges, creases and corners, see, e.g., [5, 10, 39]. In this context, a crease is a line on the surface across which the surface is only C^0 continuous. As an illustrative example, Fig. 6

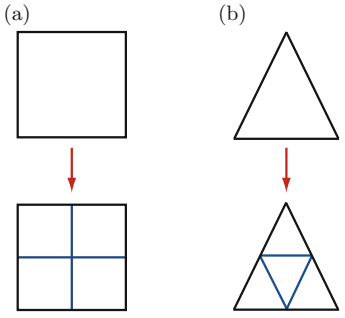


Fig. 4 Refinement by quadrisectioning. (**a**) Quadrilaterals. (**b**) Triangles

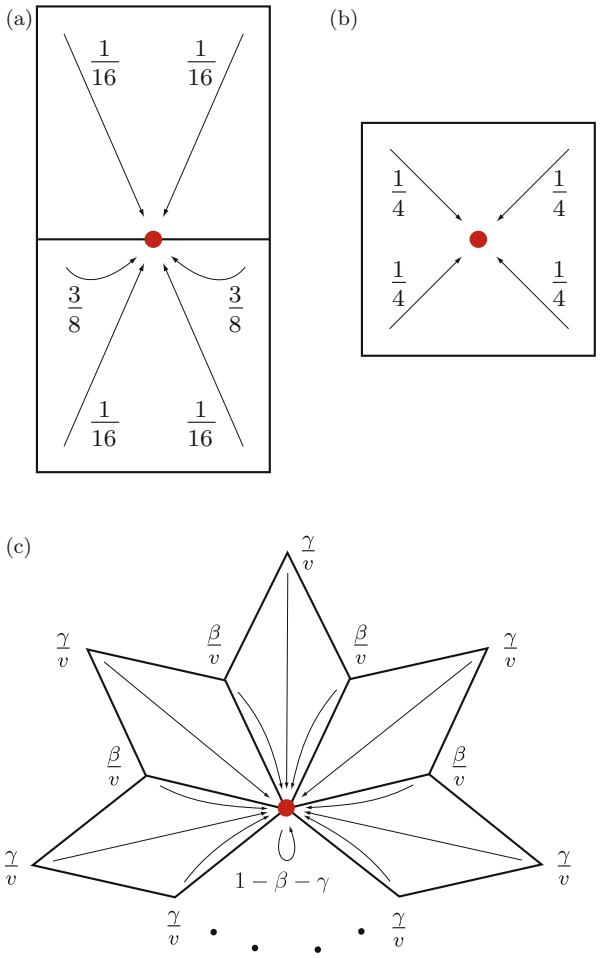


Fig. 5 Subdivision stencils for the Catmull-Clark scheme. Each of the stencils are used for computing the coordinates of vertices of the type indicated by the *red dot*. The weights for the vertex stencil are $\beta = \frac{3}{2v}$ and $\gamma = \frac{1}{4v}$, where v is the valence of the vertex. (**a**) Edge stencil. (**b**) Face stencil. (**c**) Vertex stencil

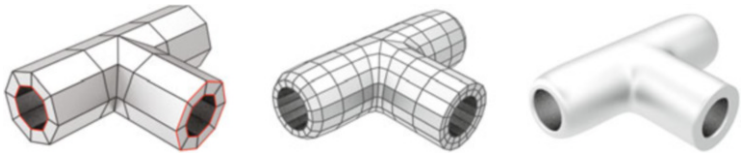


Fig. 6 Refinement of a given quadrilateral control mesh (shown *left*) with Catmull-Clark subdivision. In the vicinity of the edges in *red* modified subdivision stencils are applied in order preserve the sharp edge. The once subdivided mesh is shown in the *middle* and the limit surface on the *right*

shows the subdivision refinement of a control mesh for a T-junction geometry with extraordinary vertices and prescribed crease edges.

In case of triangular control meshes the subdivision scheme introduced by Loop [25] can be used. On three-directional triangular meshes (with each vertex having valence six) the Loop scheme yields quartic box-splines. In this context a vertex is regular when it is inside the domain and incident to six edges, or is on the boundary of the domain and incident to four edges. In the refinement step of the Loop scheme, each triangle of the control mesh is subdivided into four triangles by introducing new vertices at the edge midpoints, as shown in Fig. 4b. Subsequently, the vertex coordinates of the refined mesh are computed with the subdivision stencils given in Fig. 7 [38]. Figure 8 shows a mechanical connector geometry containing extraordinary vertices and sharp features described with the extended subdivision surfaces.

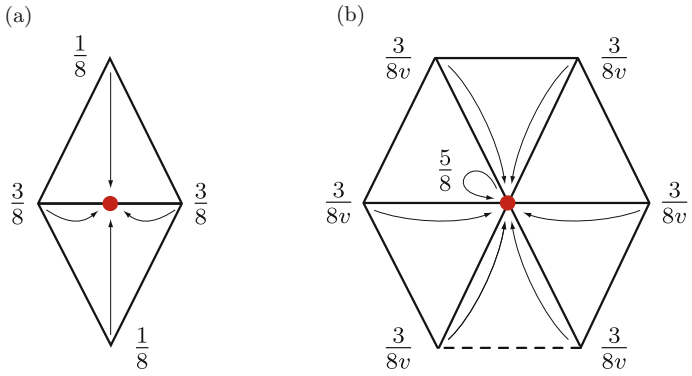


Fig. 7 Subdivision stencils for the Loop scheme. Each of the stencils are used for computing the coordinates of vertices of the type indicated by the *red dot*. (**a**) Edge stencil. (**b**) Face stencil

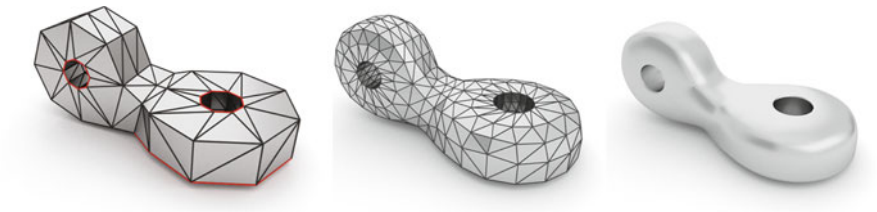


Fig. 8 Subdivision refinement of a given triangular control mesh (shown *left*) with Loop subdivision. In the vicinity of the edges in *red* modified subdivision stencils are applied in order preserve the sharp edge. The once subdivided mesh is shown in the *middle* and the limit surface on the *right*

2.3 Multiresolution Editing

Subdivision surfaces represent a limit surface with a nested hierarchy of control meshes of increasing resolution. As known in computer graphics, this property lends itself to efficient multiresolution editing of surfaces [26, 41]. The basic idea in multiresolution editing is to modify the coarse mesh vertex coordinates to apply large-scale changes (to the limit surface) and to modify the fine mesh vertex coordinates to add localised changes. By way of example, this is illustrated in Fig. 9 for the connector geometry previously introduced in Fig. 8. First the control mesh vertex coordinates $\boldsymbol{x}^0$ at level $\ell = 0$ are modified with $\boldsymbol{x}^0 + \boldsymbol{d}^0$, where $\boldsymbol{d}^0$ can be thought of as a user given perturbation vector. In the considered example, $\boldsymbol{d}^0$ applies displacements only to the vertices placed on one of the hole edges. Subsequent computation of the limit surface (by repeated subdivision) leads to a geometry with rather large scale changes. Alternatively, the edge of the hole can be displaced on level $\ell = 1$, i.e., $\boldsymbol{x}^1 + \boldsymbol{d}^1 = \boldsymbol{S}\boldsymbol{x}^0 + \boldsymbol{d}^1$. This results in a more localised change on the limit surface. It can be shown that the area of influence for each vertex extends over two rings of adjacent triangles.

The multiresolution editing algorithms available in computer graphics allow us to simultaneously edit coarse and fine resolutions, see, e.g., [3, 41]. This is achieved by a wavelet-like decomposition of the geometry into a low resolution part and a collection of wavelet coefficients expressing perturbations from the low resolution part [26, 41]. To compute such a decomposition it is necessary to define, in addition to the subdivision refinement, a coarsening operation

$$\boldsymbol{x}^\ell = \boldsymbol{R}\boldsymbol{x}^{\ell+1} . \tag{8}$$

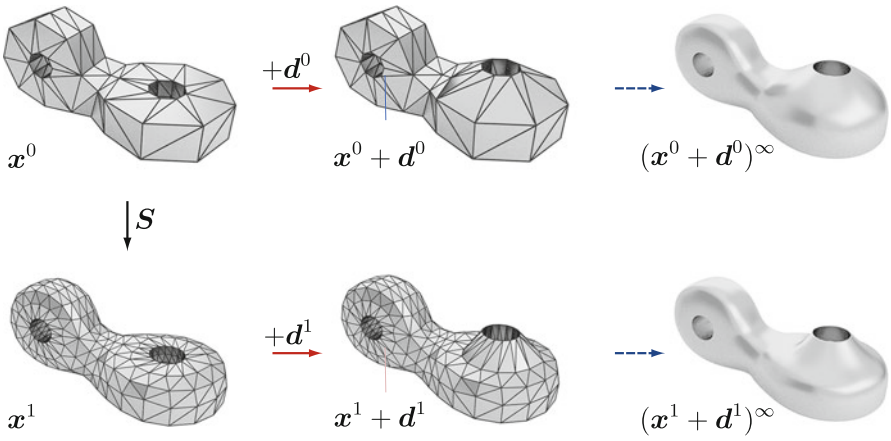


Fig. 9 Multiresolution editing of the connector geometry introduced in Fig. 8. The geometry is modified by moving the edge of one of the holes in the vertical direction. On the first row the modification is performed on level $\ell = 0$ and on the second row it is performed on level $\ell = 1$. Notice the effect of the modification level on the limit surface (*last column*)

The coarsening matrix $\boldsymbol{R}$ enables the computation of the coarse control mesh $\boldsymbol{x}^\ell$ corresponding to a given edited fine control mesh $\boldsymbol{x}^{\ell+1}$. In contrast, recall that the subdivision matrix $\boldsymbol{S}$ enables the computation of a refined mesh from a given coarse mesh, cf. (6). Different choices for the matrix $\boldsymbol{R}$ are possible. For instance, it can be determined with discrete least squares fitting

$$\boldsymbol{x}^\ell = \underset{\boldsymbol{y}^\ell}{\operatorname{argmin}} \|\boldsymbol{x}^{\ell+1} - \boldsymbol{S}\boldsymbol{y}^\ell\|^2 , \tag{9}$$

which leads to

$$\boldsymbol{S}^\mathsf{T}\boldsymbol{S}\boldsymbol{x}^\ell = \boldsymbol{S}^\mathsf{T}\boldsymbol{x}^{\ell+1} . \tag{10}$$

By comparison with (8) we observe that the coarsening matrix has to be $\boldsymbol{R} = (\boldsymbol{S}^\mathsf{T}\boldsymbol{S})^{-1}\boldsymbol{S}^\mathsf{T}$. For more details we refer to [3, 4].

Instead of using least squares fitting the coarsening matrix $\boldsymbol{R}$ can also be defined based on quasi-interpolation [23] or smoothing [41]. On the other hand coarsening by simply subsampling of the fine control mesh usually leads to artefacts in form of oscillations in the coarse control mesh. The proposed least squares fit approach is not very common in computer graphics because of the need for interactivity and fast processing times. Although the least squares matrix in (10) is sparse its solution cannot be found at interactive rates.

Subdivision surfaces do not by themselves provide the possibility of simultaneously editing coarse and fine control meshes. For instance, after a fine control mesh is edited it is not possible to further edit a coarser level in order to apply larger scale changes to the geometry. Simultaneous editing of different levels can be achieved with a wavelet-like multiresolution decomposition of the control meshes, as discussed in [3, 41].

3 Governing Equations of Shape Optimisation

In this section, we revisit the governing equations for shape optimisation of linear elastic solids, electrostatic field equations and thin-shells. The considered cost functions are the structural compliance for elastic solids and thin-shells and the normal flux over the domain boundaries for electrostatic problems. For computing the derivatives of the cost function with respect to the domain boundary perturbations we consider both the continuous and the discrete adjoint formulations. Specifically, for elastic and electrostatic field equations we use the continuous formulation and for thin-shells we use the discrete formulation. In the continuous formulation the differentiation is first performed analytically and subsequently the resulting boundary value problems are discretised with a method of choice. In contrast, in the discrete formulation first the governing equations are discretised and subsequently the resulting algebraic equations differentiated.

3.1 Continuous Shape Sensitivity Formulation

In continuous shape sensitivity analysis it is necessary to differentiate functionals that are defined on varying domains. To this end, borrowing terminology from continuum mechanics, we consider a reference configuration with domain Ω and boundary Γ and a perturbed configuration with domain Ω_t and boundary Γ_t, see Fig. 10. The corresponding deformation $\boldsymbol{\varphi}_t(\boldsymbol{x})$ which maps material points $\boldsymbol{x} \in \Omega$ onto $\boldsymbol{x}_t \in \Omega_t$ is assumed to be of the form

$$\boldsymbol{x}_t = \boldsymbol{\varphi}_t(\boldsymbol{x}) = \boldsymbol{x} + t\boldsymbol{V}\,, \tag{11}$$

where $\boldsymbol{V}$ is a prescribed constant vector field and t is a scalar parameter. It is convenient to interpret $\boldsymbol{V}$ as a velocity field and t as the time parameter.

In the sequel the derivatives of domain and boundary integrals with respect to boundary perturbations are needed. First, we consider the generic domain integral

$$I_1(\Omega_t) = \int_{\Omega_t} \psi(\boldsymbol{x}_t)\ \mathrm{d}\,\Omega_t \tag{12}$$

with a scalar integrand $\psi(\boldsymbol{x}_t)$. At the reference configuration $\boldsymbol{x}$, the derivative of the domain integral in the direction of $\boldsymbol{V}$ is defined with

$$\frac{\partial I_1}{\partial \Omega}\boldsymbol{V} = \frac{d}{dt} I_1(\boldsymbol{x} + t\boldsymbol{V})\Big|_{t=0} = \frac{d}{dt}\int_{\Omega_t} \psi(\boldsymbol{x}_t)\ \mathrm{d}\,\Omega_t\Big|_{t=0}\,. \tag{13}$$

By applying standard techniques from continuum mechanics and shape calculus, see [3, 13, 33], we obtain

$$\frac{\partial I_1}{\partial \Omega}\boldsymbol{V} = \int_{\Omega} (\nabla\psi(\boldsymbol{x})\cdot\boldsymbol{V} + \psi(\boldsymbol{x})\nabla\cdot\boldsymbol{V})\ \mathrm{d}\,\Omega = \int_{\Gamma} \psi(\boldsymbol{x})(\boldsymbol{V}\cdot\boldsymbol{n})\ \mathrm{d}\,\Gamma\,, \tag{14}$$

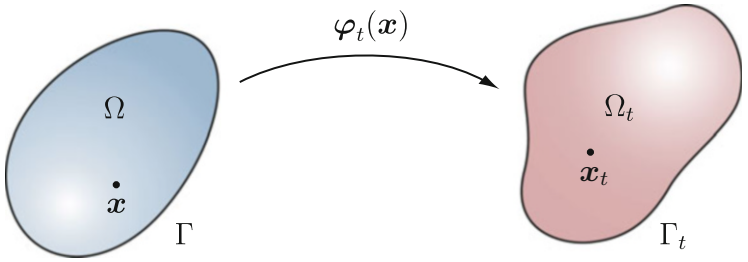


Fig. 10 Reference and the perturbed configurations (*left* and *right*, respectively)

where $\boldsymbol{n}$ is the unit normal to the boundary. As to be expected, this integral is zero when the perturbation direction $\boldsymbol{V}$ is orthogonal to the boundary normal $\boldsymbol{n}$. Perturbations tangential to the boundary do not lead to a change in I_1.

Next, we consider the generic boundary integral

$$I_2(\Gamma_t) = \int_{\Gamma_t} \psi(\boldsymbol{x}_t)\ \mathrm{d}\,\Gamma_t \tag{15}$$

with a scalar integrand $\psi(\boldsymbol{x}_t)$. The differentiation of this integral at the reference configuration $\boldsymbol{x}$ in the direction of $\boldsymbol{V}$ is more involved and can be found in standard texts on shape calculus, see [13, 33],

$$\frac{\partial I_2}{\partial \Gamma}\boldsymbol{V} = \int_{\Gamma} \left(\nabla\psi(\boldsymbol{x}) \cdot \boldsymbol{n} + H(\boldsymbol{x})\psi(\boldsymbol{x})\right)(\boldsymbol{V}\cdot\boldsymbol{n})\ \mathrm{d}\,\Gamma\,, \tag{16}$$

where $H(\boldsymbol{x})$ is the mean curvature on Γ.

3.1.1 Compliance Optimisation in Elasticity

As the first specific optimisation problem we consider linear elasticity. The equilibrium equation for a solid body with the domain Ω is given by

$$\nabla\cdot\boldsymbol{\sigma}(\boldsymbol{u}) + \boldsymbol{f} = \boldsymbol{0} \qquad \text{in } \Omega\,, \tag{17a}$$

$$\boldsymbol{u} = \boldsymbol{0} \qquad \text{on } \Gamma_D\,, \tag{17b}$$

$$\boldsymbol{\sigma}(\boldsymbol{u})\boldsymbol{n} = \bar{\boldsymbol{t}} \qquad \text{on } \Gamma_N\,, \tag{17c}$$

where $\boldsymbol{\sigma}$ is the stress tensor, $\boldsymbol{u}$ is the displacement vector, $\boldsymbol{f}$ is the external load vector and $\bar{\boldsymbol{t}}$ is the prescribed traction on the Neumann boundary Γ_N with the outward normal $\boldsymbol{n}$. On Γ_D, for simplicity, only homogenous Dirichlet boundary conditions are assumed.

We assume a homogenous linear elastic material model

$$\boldsymbol{\sigma}(\boldsymbol{u}) = \boldsymbol{C} : \boldsymbol{\epsilon}(\boldsymbol{u}) \tag{18}$$

with the linear elastic strain tensor

$$\boldsymbol{\epsilon}(\boldsymbol{u}) = \frac{1}{2}(\nabla\boldsymbol{u} + \nabla^{\mathsf{T}}\boldsymbol{u})\,. \tag{19}$$

The aim of shape optimisation is to find a shape Ω that minimises a cost functional $J(\Omega, \boldsymbol{u})$ such that

$$\min_{\Omega} J(\Omega, \boldsymbol{u}) \tag{20}$$

where the displacement vector $\boldsymbol{u}$ has to satisfy (17). In practice there are usually additional constraints, e.g., pertaining to the volume of the domain Ω, which are neglected here for brevity. In case of structural compliance, the cost functional $J(\Omega,\boldsymbol{u})$ reads

$$J(\Omega,\boldsymbol{u}) = \int_{\Omega} \boldsymbol{\sigma}(\boldsymbol{u}) : \boldsymbol{\epsilon}(\boldsymbol{u}) \,\mathrm{d}\,\Omega = \int_{\Omega} \boldsymbol{f}\cdot\boldsymbol{u} \,\mathrm{d}\,\Omega + \int_{\Gamma_N} \bar{\boldsymbol{t}}\cdot\boldsymbol{u} \,\mathrm{d}\,\Gamma\,. \tag{21}$$

In adjoint shape sensitivity analysis the minimisation problem (20) with the equilibrium equation (17) as a constraint is expressed with the functional

$$\begin{aligned} L(\Omega,\boldsymbol{u},\boldsymbol{\lambda}) = J(\Omega,\boldsymbol{u}) &+ \int_{\Omega} \nabla\boldsymbol{\lambda} : \boldsymbol{\sigma}(\boldsymbol{u}) \,\mathrm{d}\,\Omega - \int_{\Omega} \boldsymbol{\lambda}\cdot\boldsymbol{f} \,\mathrm{d}\,\Omega \\ &- \int_{\Gamma_D} \boldsymbol{u}\cdot(\boldsymbol{C} : \nabla\boldsymbol{\lambda})\boldsymbol{n} + \boldsymbol{\lambda}\cdot\boldsymbol{\sigma}(\boldsymbol{u})\boldsymbol{n} \,\mathrm{d}\,\Gamma - \int_{\Gamma_N} \boldsymbol{\lambda}\cdot\bar{\boldsymbol{t}} \,\mathrm{d}\,\Gamma\,, \end{aligned} \tag{22}$$

where $\boldsymbol{\lambda}$ is a vector-valued Lagrange parameter field. The stationarity condition for the functional $L(\Omega,\boldsymbol{u},\boldsymbol{\lambda})$, i.e.,

$$\delta L(\Omega,\boldsymbol{u},\boldsymbol{\lambda}) = 0 \tag{23}$$

provides the complete set of equations for shape optimisation. The variation of $L(\Omega,\boldsymbol{u},\boldsymbol{\lambda})$ with respect to the Lagrange parameter $\boldsymbol{\lambda}$ yields the original boundary value problem (17). The variation of $L(\Omega,\boldsymbol{u},\boldsymbol{\lambda})$ with respect to the displacement vector $\boldsymbol{u}$ yields the adjoint boundary value problem

$$\nabla\cdot\boldsymbol{\sigma}(\boldsymbol{\lambda}) - \boldsymbol{f} = \boldsymbol{0} \qquad \text{in } \Omega\,, \tag{24a}$$

$$\boldsymbol{\lambda} = \boldsymbol{0} \qquad \text{on } \Gamma_D\,, \tag{24b}$$

$$\boldsymbol{\sigma}(\boldsymbol{\lambda})\boldsymbol{n} = -\bar{\boldsymbol{t}} \qquad \text{on } \Gamma_N\,. \tag{24c}$$

By comparing the adjoint problem with the original boundary value problem (17) we deduce that $\boldsymbol{\lambda} = -\boldsymbol{u}$. This holds only when the structural compliance (21) is the cost functional.

Next, we consider the variation of $L(\Omega,\boldsymbol{u},\boldsymbol{\lambda})$ with respect to the domain Ω. In deriving it we make use of the domain deformation $\boldsymbol{x}_t = \boldsymbol{\varphi}_t(\boldsymbol{x}) = \boldsymbol{x} + t\boldsymbol{V}$ introduced in (11). In shape optimisation of linear elasticity problems usually only the parts of the Neumann boundary with zero tractions are allowed to move, i.e.,

$$\begin{aligned} &\boldsymbol{V} = \boldsymbol{0} \quad \text{on } \Gamma_D\,, \\ &\boldsymbol{V} = \boldsymbol{0} \quad \text{on } \Gamma_N \text{ with } \boldsymbol{\sigma}\boldsymbol{n} = \bar{\boldsymbol{t}}\,, \\ &\boldsymbol{V} \neq \boldsymbol{0} \quad \text{on } \Gamma_N \text{ with } \boldsymbol{\sigma}\boldsymbol{n} = \boldsymbol{0}\,. \end{aligned} \tag{25}$$

The variation of the Lagrangian (22) with respect to the domain perturbations yields the shape gradient. For structural compliance (21) as the cost functional (i.e., $\boldsymbol{\lambda} = -\boldsymbol{u}$) and with the result (14) at hand we obtain for the shape gradient

$$\frac{\partial L}{\partial \Omega}\boldsymbol{V} = \int_{\Gamma_N} (2\boldsymbol{u}\cdot\boldsymbol{f} - \nabla\boldsymbol{u} : \boldsymbol{\sigma}(\boldsymbol{u}))\,(\boldsymbol{V}\cdot\boldsymbol{n})\ \mathrm{d}\,\Gamma = \int_{\Gamma_N} g(\boldsymbol{u})(\boldsymbol{V}\cdot\boldsymbol{n})\ \mathrm{d}\,\Gamma\,, \tag{26}$$

where $g(\boldsymbol{u})$ is the shape kernel function. It is worth emphasising that without restricting $\boldsymbol{V}$ as stated in (25) the shape derivate would contain several more terms.

In iterative shape optimisation the shape kernel function $g(\boldsymbol{u})$ is used as gradient information. In order to achieve a decrease in the cost functional the boundary perturbation is chosen proportional to

$$\boldsymbol{V} = -g(\boldsymbol{u})\boldsymbol{n} \tag{27}$$

such that

$$\frac{\partial L}{\partial \Omega}\boldsymbol{V} = -\int_{\Gamma_N} g(\boldsymbol{u})^2\ \mathrm{d}\,\Gamma\,. \tag{28}$$

During the shape optimisation the boundary value problem (17) has to be repeatedly solved on a continuously changing domain. In a conventional finite element setting this would require frequent mesh smoothing or updating. Therefore, immersed, or embedded, finite element approaches that do not require remeshing have clear advantages in shape optimisation. In the present work, we use an immersed finite element technique previously developed in the context of incompressible fluid-structure interaction [28–30]. For more details see [3].

3.1.2 Boundary Flux Optimisation in Electrostatics

The motivation for our second optimisation problem comes from the design of high-voltage electrical devices. Avoidance of electrical breakdown is one of the key design considerations for such devices. In a first approximation, electrical breakdown can be avoided by reducing the electrical field strength on critical components. The aim of shape optimisation is to modify the geometry of components so that the maximum electric field strength on surfaces, i.e., the normal flux, is reduced.

The electrostatic field equation in absence of space charges is given by a Dirichlet boundary value problem for the Laplace equation

$$\begin{aligned} -\Delta u &= 0 \quad \text{in } \Omega, \\ u &= 0 \quad \text{on } \Gamma_{D_0}, \\ u &= \overline{u} \quad \text{on } \Gamma_{D_f}, \end{aligned} \tag{29}$$

where u is the scalar electric potential or voltage, Ω is a multiply connected domain with a Dirichlet boundary consisting of a (to be optimised) free part Γ_{D_f} and a fixed part Γ_{D_0}. It is assumed that the potentials on Γ_{D_f} and Γ_{D_0} are constant. The geometry of the free Dirichlet boundary Γ_{D_f} with the prescribed potential $\overline{u}$ is to be determined with shape optimisation. Notice that the electric potential u is denoted non-bold because it is a scalar field quantity.

In order to pointwise minimise the maximum of the normal flux on the free part of the boundary Γ_{D_f} we chose to minimise the cost functional

$$J(\Omega, u) = \frac{1}{2} \int_{\Gamma_{D_f}} \left(\frac{\partial u}{\partial \boldsymbol{n}} - Q \right)^2 \mathrm{d}\,\Gamma\,, \tag{30}$$

where $Q \geq 0$ is a user prescribed constant expected value.

The adjoint sensitivity analysis, as discussed in Sect. 3.1.1 for elasticity, is used for obtaining the adjoint boundary value problem and the shape gradient. The adjoint boundary value problem corresponding to the cost functional (30) reads

$$\begin{aligned} -\Delta\lambda &= 0 && \text{in } \Omega, \\ \lambda &= 0 && \text{on } \Gamma_{D_0}, \\ \lambda &= \frac{\partial u}{\partial \boldsymbol{n}} - Q && \text{on } \Gamma_{D_f}\,. \end{aligned} \tag{31}$$

Importantly, the adjoint solution λ is different from the solution u of the original problem so that in applications both problems (29) and (31) have to be solved.

Moreover, the shape gradient corresponding to the cost functional (30) reads

$$\frac{\partial L}{\partial \Omega} \boldsymbol{V} = \int_{\Gamma_{D_f}} \left(-\frac{\partial \lambda}{\partial \boldsymbol{n}} \frac{\partial u}{\partial \boldsymbol{n}} - \frac{H}{2} \right) (\boldsymbol{V} \cdot \boldsymbol{n}) \, \mathrm{d}\,\Gamma = \int_{\Gamma_{D_f}} g(\lambda, u)(\boldsymbol{V} \cdot \boldsymbol{n}) \, \mathrm{d}\,\Gamma\,, \tag{32}$$

where the shape kernel function $g(\lambda, u)$ now depends on the electric potential u and adjoint solution λ. In deriving the shape gradient we made use of (16) and H is the mean curvature of the boundary.

As discussed for elasticity, during the iterative shape optimisation the shape kernel function $g(\boldsymbol{u})$ is used as gradient information. In order to achieve a decrease in the cost functional the boundary perturbation is chosen proportional to $\boldsymbol{V} = -g(\lambda, \boldsymbol{u})\boldsymbol{n}$.

For solving the original and the adjoint boundary value problems (29) and (31), respectively, we use the boundary element method, see e.g. [36], which does not require a domain discretisation and is ideal for solving problems with unbounded domains that occur in electrostatic field analysis. For more details see [3].

3.2 Discrete Shape Sensitivity Formulation

3.2.1 Compliance Optimisation of Thin-Shells

We use the Kirchhoff-Love energy functional for modelling the mechanical response of thin shells. Since the related equations are classically formulated in curvilinear coordinates, it is usually easier to compute the shape sensitivities using a discrete approach. To this end, we first discretise the governing equations with finite elements and subsequently differentiate the cost function and the discrete equilibrium equations with respect to control vertex coordinates.

The Kirchhoff-Love energy functional depends on mid-surface displacements and their first and second derivatives. Therefore, the basis functions have to be smooth and their second order derivatives square-integrable. As originally introduced by Cirak et al. [10, 11] Kirchhoff-Love shells can be elegantly discretised with subdivision basis functions. In the resulting discrete equilibrium equations the control vertex displacements are the only degrees of freedom.

To begin with, we consider a shell with the undeformed mid-surface Ω, the position vector $\boldsymbol{x}(\xi_1, \xi_2)$ and the displacement vector $\boldsymbol{u}(\xi_1, \xi_2)$. It is assumed that the mid-surface is parameterised in terms of the curvilinear coordinates (ξ_1, ξ_2). The two corresponding (covariant) surface basis vectors follow from

$$\boldsymbol{a}_\alpha = \boldsymbol{x}_{,\alpha} \,. \tag{33}$$

From here onwards Greek indices take the values 1 and 2 and a comma denotes partial differentiation with respect to curvilinear coordinates.

According to [11] the linearised membrane and bending strains $\alpha_{\alpha\beta}$ and $\beta_{\alpha\beta}$, respectively, are given by

$$\alpha_{\alpha\beta} = \frac{1}{2}(\boldsymbol{a}_\alpha \cdot \boldsymbol{u}_{,\beta} + \boldsymbol{u}_{,\alpha} \cdot \boldsymbol{a}_\beta) \,, \tag{34}$$

$$\begin{aligned} \beta_{\alpha\beta} = &-\boldsymbol{u}_{,\alpha\beta} \cdot \boldsymbol{a}_3 + \frac{1}{\sqrt{a}} \left[\boldsymbol{u}_{,1} \cdot (\boldsymbol{a}_{\alpha,\beta} \times \boldsymbol{a}_2) + \boldsymbol{u}_{,2} \cdot (\boldsymbol{a}_1 \times \boldsymbol{a}_{\alpha,\beta})\right] \\ &+ \frac{\boldsymbol{a}_3 \cdot \boldsymbol{a}_{\alpha,\beta}}{\sqrt{a}} \left[\boldsymbol{u}_{,1} \cdot (\boldsymbol{a}_2 \times \boldsymbol{a}_3) + \boldsymbol{u}_{,2} \cdot (\boldsymbol{a}_3 \times \boldsymbol{a}_1)\right] . \end{aligned} \tag{35}$$

With the above strain expressions at hand, the potential energy of a linear-elastic Kirchhoff-Love shell reads

$$\Pi(\boldsymbol{u}) = \int_\Omega \left(W^m(\alpha_{\alpha\beta}) + W^b(\beta_{\alpha\beta})\right) \mu \, \mathrm{d}\, \Omega + \Pi^{ext}(\boldsymbol{u}) \,, \tag{36}$$

where Ω is the shell mid-surface, W^m is the membrane energy density, W^b is the bending energy density, μ is the jacobian associated with the integration over the thickness and $\Pi^{ext}(\boldsymbol{u})$ is the potential of the external forces. For a linear elastic

material the two energy densities are defined with

$$W^m(\alpha_{\alpha\beta}) = \frac{1}{2}\frac{Et}{1-\nu^2} H^{\alpha\beta\gamma\delta}\alpha_{\alpha\beta}\alpha_{\gamma\delta}\,, \tag{37a}$$

$$W^b(\beta_{\alpha\beta}) = \frac{1}{2}\frac{Et^3}{12(1-\nu^2)} H^{\alpha\beta\gamma\delta}\beta_{\alpha\beta}\beta_{\gamma\delta}\,, \tag{37b}$$

where t is the shell thickness, E is the Young's modulus, ν is the Poisson's ratio and $H^{\alpha\beta\gamma\delta}$ is an auxiliary fourth order tensor

$$H^{\alpha\beta\gamma\delta} = \nu\, a^{\alpha\beta}a^{\gamma\delta} + \frac{1}{2}(1-\nu)\,(a^{\alpha\gamma}a^{\beta\delta} + a^{\alpha\delta}a^{\beta\gamma})\,. \tag{38}$$

The contravariant metric tensor components $a^{\alpha\beta} = \boldsymbol{a}^\alpha \cdot \boldsymbol{a}^\beta$ are computed from the contravariant basis vectors defined according to $\boldsymbol{a}^\alpha \cdot \boldsymbol{a}_\beta = \delta^\alpha_\beta$, where δ^α_β is the Kronecker delta.

Next, the mid-surface position and displacement vector are discretised with subdivision basis functions

$$\boldsymbol{x}_h(\xi_1,\xi_2) = \sum_i N_i(\xi_1,\xi_2)\boldsymbol{x}_i\,, \qquad \boldsymbol{u}_h(\xi_1,\xi_2) = \sum_i N_i(\xi_1,\xi_2)\boldsymbol{u}_i\,, \tag{39}$$

where $N_i(\xi_1,\xi_2)$ is the basis function, $\boldsymbol{x}_i$ is the position vector and $\boldsymbol{u}_i$ is the displacement of a control vertex with the index i. Recall that on structured meshes the subdivision basis functions are equivalent to quartic box-splines in case of the Loop scheme and equivalent to tensor-product cubic b-splines in case of the Catmull-Clark scheme. As shown by Stam [34, 35] in the vicinity of extraordinary vertices the subdivision surfaces can be evaluated in the same way as regular patches after only few steps of refinement. For implementation details see [10, 11].

In order to obtain the discrete equilibrium equations, first the interpolation (39) is introduced into the energy functional (36) and subsequently the integrals are evaluated with Gauss quadrature. Computing the stationary points of the discretised energy functional yields the discrete equilibrium equations

$$\boldsymbol{K}\boldsymbol{u} = \boldsymbol{f}, \tag{40}$$

where $\boldsymbol{K}$ is the stiffness matrix, $\boldsymbol{f}$ is the array of vertex forces and, with a slight abuse of notation, $\boldsymbol{u}$ is the array of vertex displacements.

In discrete shape sensitivity formulation we aim to minimise a discretised cost function

$$\min_{\boldsymbol{x}} J(\boldsymbol{x},\boldsymbol{u}) \tag{41}$$

which depends on the array of vertex coordinates $\boldsymbol{x}$ and displacements $\boldsymbol{u}$. It is clear that the displacements have to satisfy (40). The compliance cost function corresponding to the discrete equilibrium equation reads

$$J(\boldsymbol{x}, \boldsymbol{u}) = \boldsymbol{f}^{\mathsf{T}}\boldsymbol{u} = \boldsymbol{u}^{\mathsf{T}}\boldsymbol{K}\boldsymbol{u}\,. \tag{42}$$

In order to compute the derivatives of the cost function with respect to the vertex coordinates we consider the adjoint formulation

$$L(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\lambda}) = J(\boldsymbol{x}, \boldsymbol{u}) + \boldsymbol{\lambda}^{\mathsf{T}}[\boldsymbol{f} - \boldsymbol{K}\boldsymbol{u}], \tag{43}$$

where $\boldsymbol{\lambda}$ is a Lagrange parameter vector. The stationarity condition for $L(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\lambda})$ with respect to the vertex displacements leads to the adjoint problem

$$\boldsymbol{K}\boldsymbol{\lambda} = \frac{\partial J(\boldsymbol{x}, \boldsymbol{u})}{\partial \boldsymbol{u}}\,. \tag{44}$$

Here, we made use of the symmetry of the stiffness matrix $\boldsymbol{K}$. The stationarity condition for $L(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\lambda})$ with respect to the vertex coordinates leads to the discrete shape gradients

$$\frac{\partial L(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\lambda})}{\partial \boldsymbol{x}} = \frac{\partial J(\boldsymbol{x}, \boldsymbol{u})}{\partial \boldsymbol{x}} + \boldsymbol{\lambda}^{\mathsf{T}}\left[\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} - \frac{\partial \boldsymbol{K}}{\partial \boldsymbol{x}}\boldsymbol{u}\right]. \tag{45}$$

By introducing the cost function (42) and the adjoint solution with $\boldsymbol{\lambda} = \boldsymbol{u}$ we obtain

$$\frac{\partial L(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\lambda})}{\partial \boldsymbol{x}} = 2\boldsymbol{u}^{\mathsf{T}}\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} - \boldsymbol{u}^{\mathsf{T}}\frac{\partial \boldsymbol{K}}{\partial \boldsymbol{x}}\boldsymbol{u}\,. \tag{46}$$

The derivatives of the stiffness matrix with respect to the vertex coordinates can be determined by consecutively differentiating the discretised Kirchhoff-Love energy functional (36) first with respect to vertex displacements and then with respect to the vertex positions. In gradient-based optimisation, in order achieve a decrease in the cost function the perturbation of the control vertex positions is chosen proportional to

$$\boldsymbol{V} = -2\boldsymbol{u}^{\mathsf{T}}\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} + \boldsymbol{u}^{\mathsf{T}}\frac{\partial \boldsymbol{K}}{\partial \boldsymbol{x}}\boldsymbol{u}\,. \tag{47}$$

Note that, with a slight abuse of notation, the perturbation vectors for the continuous and discrete shape sensitivity formulations are denoted with the same symbol $\boldsymbol{V}$. For more details we refer to [2].

4 Multiresolution Shape Optimisation

The introduced subdivision multiresolution editing technique enables to use two different resolutions of the same geometry for optimisation and analysis. A simplified two-level version of the proposed optimisation algorithm is shown Fig. 11. The optimisation and analysis meshes correspond to different refinement levels in a multiresolution hierarchy. The optimisation level is $\ell_o = 0$ and the analysis level is $\ell_c = n$. In our actual implementation the optimisation level is successively increased until $\ell_o = \ell_c$ and the analysis level ℓ_c is fixed. Crucially, in the spirit of isogeometric analysis the control meshes for analysis and optimisation represent the same geometry.

In shape optimisation it is usually necessary to use a coarse control mesh for geometry updating and a relatively fine control mesh for analysis. As is known, unwanted geometry oscillations may appear when the analysis and geometry representations have similar resolutions [8]. These geometry oscillations are usually a numerical artefact or a result of the ill-posedness of the considered optimisation problem. Moreover, in practical applications it might be desirable to optimise only a very coarse representation out of aesthetic or manufacturability reasons.

A more detailed description of the employed multiresolution technique is given in Algorithm 1. In the presented computations, different from Algorithm 1 we use instead of the steepest descent update on line 9 the method of moving asymptotes (MMA) as implemented in the NLopt library [20]. In addition, in practical computations there are usually constraints, such as bounds on vertex positions or area/volume constraints, which are not mentioned in Algorithm 1. For more details we refer to [3, 4].

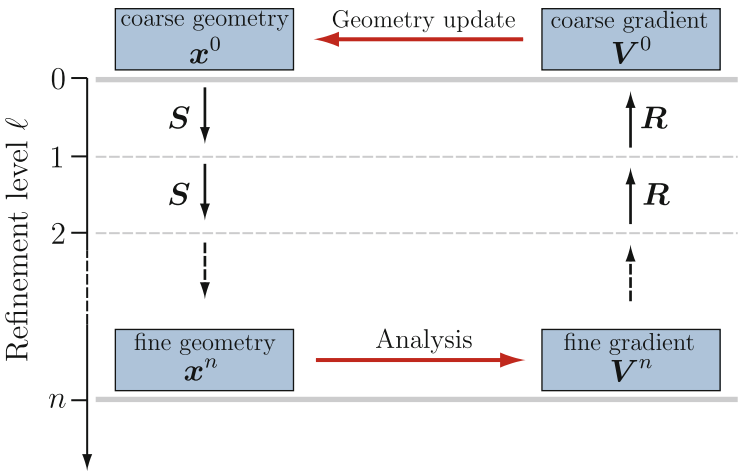


Fig. 11 Flowchart of the simplified multiresolution optimisation algorithm

Algorithm 1 Multiresolution shape optimisation

// choose computational level ℓ_c
// read coarse input control mesh $\boldsymbol{x}^0$
// Initialise optimisation level
$\ell_o = 0$
// Initialise cost functional
$J = \infty$
// iterate over optimisation levels
while $\ell_o \leq \ell_c$ **do**
  // update vertex coordinates $\boldsymbol{x}^{\ell_o}$ *while the cost functional decreases*
  repeat
    // subdivide optimisation level ℓ_o *up to analysis level* ℓ_c
    for $\ell \leftarrow \ell_o$ to ℓ_c **do**
      $\boldsymbol{x}^{\ell} \leftarrow \boldsymbol{S}\boldsymbol{x}^{\ell}$
    // compute cost functional $J = J(\boldsymbol{x}^{\ell_c}, \boldsymbol{u}(\boldsymbol{x}^{\ell_c}))$ *and the descent direction* $\boldsymbol{V}^{\ell_c}$
    // project the descent direction $\boldsymbol{V}^{\ell_c}$ *to the optimisation level*
    for $\ell \leftarrow \ell_c$ to ℓ_o **do**
      $\boldsymbol{V}^{\ell} \leftarrow \boldsymbol{R}\boldsymbol{V}^{\ell}$
    // update vertex coordinates of the optimisation level
    $\boldsymbol{x}^{\ell_o} \leftarrow (\boldsymbol{x}^{\ell_o} + \alpha \boldsymbol{V}^{\ell_o})$ with $\alpha > 0$
  until $J < J_{\text{previous}}$
  // increment optimisation level
  $\ell_o \leftarrow (\ell_o + 1)$

5 Examples

5.1 Immersed Finite Element Discretised Elasticity Problems

5.1.1 Simply Supported Plate with a Hole

This example highlights the advantages of multiresolution optimisation over classical approaches that use only one or two representation levels. The problem consists of a square plate with an edge length $L = 2$ and a circular hole with diameter $D = 1$, see Fig. 12. The plate is loaded with a line load of length 1. The Young's modulus and Poisson's ratio of the plate are $E = 100$ and $\nu = 0.4$, respectively. During optimisation the shape of the hole is to be modified so that the structural compliance of the plate is minimised.

Initially, at level $\ell = 0$ the hole is represented with a cubic spline with 8 control points. The immersed finite element grid has 100×100 cells of uniform size. Three cases referred to as C1, C2 and C3 with different geometry and analysis resolutions are studied:

- In C1 only one level with $\ell_o = \ell_c = 0$ is used for analysis and optimisation.
- In C2 a four times subdivided control mesh at refinement level $\ell_o = \ell_c = 4$ is used for analysis and optimisation.

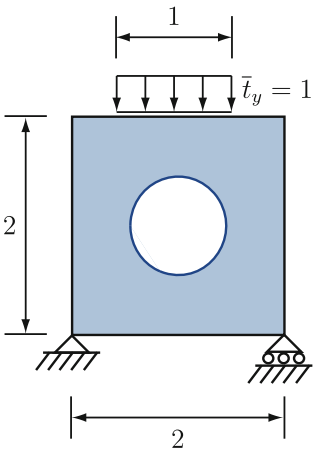


Fig. 12 Simply supported plate with a hole. Problem description

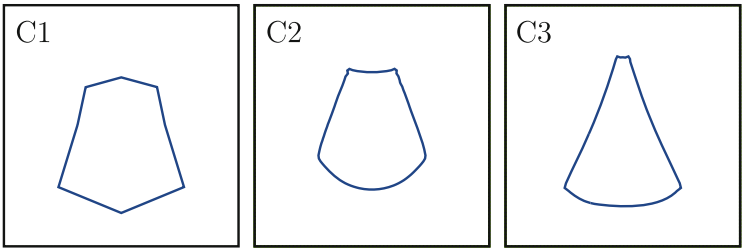


Fig. 13 Simply supported plate with a hole. Optimised hole shapes for cases C1, C2 and C3

– In C3 the optimisation level starts with $\ell_o = 0$ and increases until $\ell_o = \ell_c = 4$ is reached. Throughout the computations the analysis level is fixed to $\ell_c = 4$.

In case C1 the control mesh that is visible by the immersed finite element grid contains 8 elements and in cases C2 and C3 it contains 128 elements. It is clear that in case C1 the hole geometry is poorly resolved on the immersed finite element grid.

In Fig. 13 the optimised final hole shapes for the three cases are shown. In particular, the difference in optimal shapes for cases C2 and C3 which use the same analysis level $\ell_c = 4$ is striking. The case C1 is different from the other two cases because of the mentioned inadequately coarse analysis control mesh with $\ell_c = 0$. During optimisation the optimisation level ℓ_o is successively increased only for case C3. The optimisation level is always incremented when a minimum is reached, cf. Algorithm 1. For the three cases the reduction of the relative cost function over the number of iterations is shown in Fig. 14. The case C2 with fixed fine resolution achieves the smallest cost reduction while the case C3 with multiresolution achieves the largest cost reduction. The strong dependence of the optimisation results on geometry parameterisation is well known in structural optimisation and is often

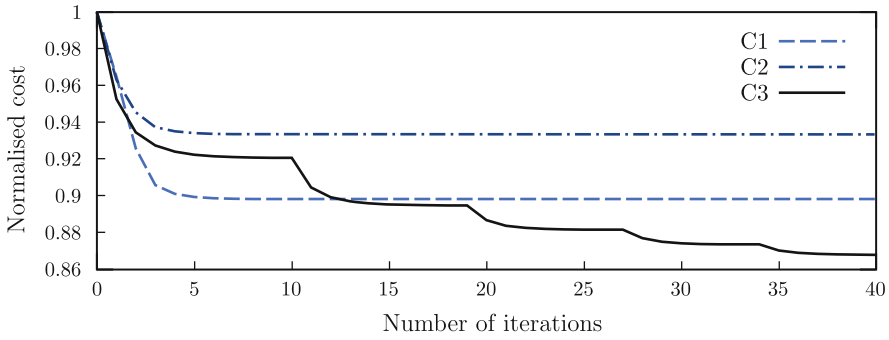


Fig. 14 Simply supported plate with a hole. Reduction of the normalised cost over the number of optimisation iterations. The initial cost for case C1 is 0.073 and for cases C2 and C3 is 0.065

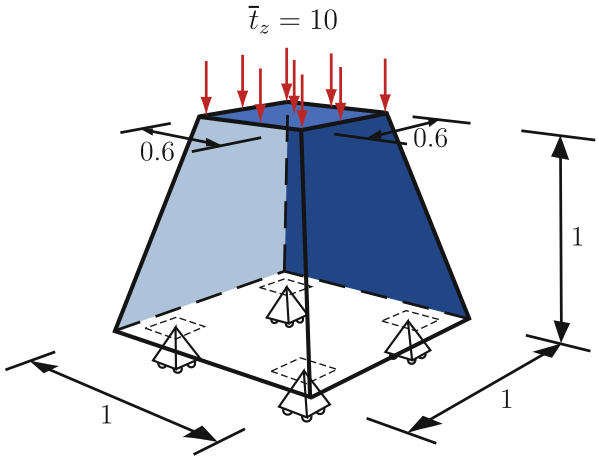


Fig. 15 Three-dimensional stool. Problem description. Roller supports are applied to all finite element nodes inside the regions of size 0.2×0.2 marked by *dashed squares*

associated with the non-convexity of the considered optimisation problem. We conjecture that by initially using a coarse control mesh for optimisation the possible number of local minima is significantly reduced which reduces the possibility of landing in a non-optimal local minimum. It appears that in case C2 the optimisation problem is caught in a local minimum which is significantly higher than the global minimum.

5.1.2 Three-Dimensional Stool

In this example we present the combined topology and shape optimisation of a three-dimensional solid, see Fig. 15. Our algorithm for combined topology and shape

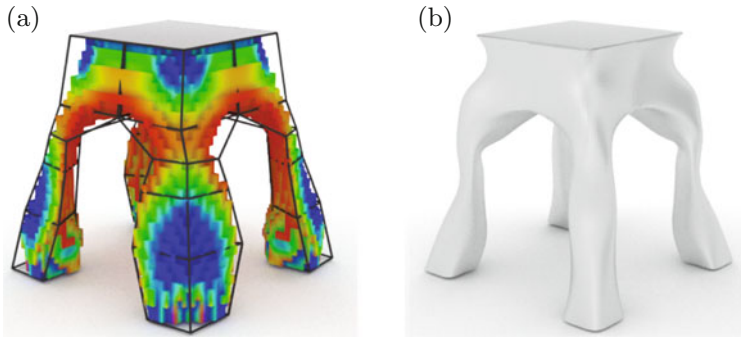


Fig. 16 Three-dimensional stool. First topology and subsequent shape optimisation step. (**a**) Topology optimised. (**b**) Shape optimised

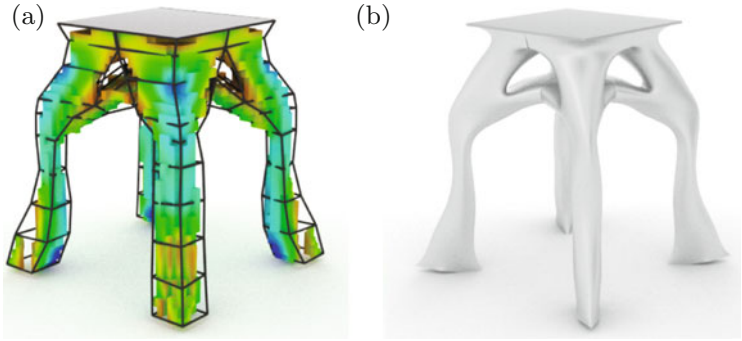


Fig. 17 Three-dimensional stool. Second topology and subsequent shape optimisation step. (**a**) Topology optimised. (**b**) Shape optimised

optimisation is described in [3]. The initial domain is a truncated pyramid and is at its top loaded with a uniform distributed load $\bar{t}_z = 10$. At its bottom it is supported by four distributed roller supports each of size 0.2×0.2. The Young's modulus and Poisson's ratio are chosen with $E = 100$ and $\nu = 0.4$, respectively. In the optimisation study only one quarter of the domain is considered and appropriate bounds and geometry tags are applied at the two planes of symmetry. The corresponding immersed finite element grid is of size $0.7 \times 0.7 \times 1$ and consists of $30 \times 30 \times 30$ cells.

The sequence of the performed topology and shape optimisation steps are shown in Figs. 16 and 17. In total two topology and two shape optimisation steps are performed. During each topology optimisation step we remove in one go a relatively large amount of material by deleting computational cells with the topology derivative below a given threshold. Figure 16a shows the result of the first topology optimisation step and the semi-manually generated coarse control mesh for representing the new topology. In the following shape optimisation step, see Fig. 16b, the generated control mesh serves as the optimisation level $\ell_o = 0$ and

the computation level is chosen with $\ell_c = 2$. During the shape optimisation the volume of the domain is constraint to remain constant. The result of the second topology optimisation step and the generated control mesh are shown in Fig. 17a. This is followed by the final shape optimisation step shown Fig. 17b.

5.2 *Boundary Element Discretised Electrostatic Problems*

5.2.1 Box in a Sphere

We optimise the shape of a box placed inside a sphere, see Fig. 18, with the expected normal flux density Q in (30) set to 20. It can be shown that the optimal shape for the inner box is a sphere with half the diameter of the outer sphere [15]. The box, representing the to be optimised boundary Γ_{D_f}, is of size $0.16 \times 0.2 \times 0.24$ and the outer sphere, representing the fixed boundary Γ_{D_0}, has radius 0.2. The coarse mesh for the box contains 48 elements which increases to 768 elements in the twice subdivided fine mesh at level $\ell_c = 2$. During the subdivision refinement, the creases in the coarse mesh are maintained as creases using the extended subdivision stencils mentioned in Sect. 2.2, see also [5, 10]. With the extended subdivision stencils the limit surface corresponding to the coarse box mesh is a box of the same geometry. Note that on the limit surface the creases are only C^0-continuous and not at least C^1-continuous. The resolution of the outer sphere remains fixed with 3840 elements. Hence, the meshes for the boundary element analysis of the cube and sphere consist of 768 and 3840 elements, respectively.

Figure 18a shows the initial coarse geometry yielding a cost functional value of $J(\Omega^{\ell_c}, u^{\ell_c}) = 15.38$. First we select this coarse geometry as optimisation level, i.e., $\ell_o = 0$. After consecutively selecting $\ell_o = 1$ and $\ell_o = 2$ and optimising we obtain the final optimised geometry shown in Fig. 18b. The final shape of the initial box is nearly a sphere of diameter 0.215 and the cost functional value is $J(\Omega^{\ell_c}, u^{\ell_c}) = 8.46 \cdot 10^{-3}$, which represents a reduction of 99.95 %. As to be

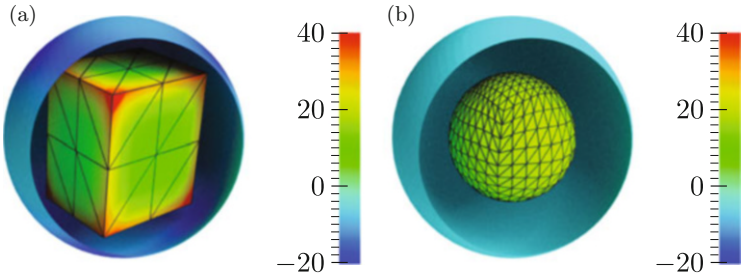


Fig. 18 Box in a sphere. Initial and optimised geometries with isocontours of the normal flux. The meshes indicate the optimisation level ℓ_o. The isocontours belong to the fine computational mesh at level $\ell_c = 2$. (**a**) Initial geometry at $\ell_o = 0$. (**b**) Final optimised geometry at $\ell_o = 2$

expected, the optimisation leads to a geometry with nearly uniform distribution of normal flux as seen in Fig. 18b. Since the extended subdivision scheme was used in this case, the marked creases were not smoothed out by the subdivision itself, but rather by the shape optimisation procedure.

5.2.2 Gas Insulated Switchgear

In this example we apply the proposed shape optimisation approach to the design of an electrode in a gas insulated switchgear, see Fig. 19a. Such devices are widely used as circuit-breakers in high-voltage power transmission. The objective of shape optimisation is to reduce the propensity for electrical breakdown with the ultimate aim of enabling more compact device geometries. This can be achieved by modifying the electrode geometries such that the cost functional $J(\Omega^{\ell_c}, u^{\ell_c})$ in (30) is minimised.

In Fig. 19a the gas insulated switchgear is shown with the electrode in the form of a primitive cylinder. The cylinder represents the electrode geometry to be optimised. The initial coarse mesh of the cylinder contains 264 elements. The creases on the cylinder are not tagged. Therefore, the geometry becomes smoother while it is refined by subdivision, see Fig. 19b. As a design constraint, the inner surface of the cylinder is required to have a constant radius for a bolt passing through it. Geometric constraints on the positions of vertices lying on the inner surface are applied to prevent any movement that would violate this design requirement.

In this example, a once subdivided electrode mesh with 1056 elements is chosen as the computational level ($\ell_c = 1$) and the geometry at level $\ell_o = 0$ is used for optimisation. In the initial design, Fig. 19b, the maximum normal flux is 81.63 before optimisation and reduces to 66.99 in the optimised shape shown in Fig. 20, corresponding to a reduction of 17.94 %. However, the reduction in the cost function $J(\Omega^{\ell_c}, u^{\ell_c})$ is much higher with 38.24 %. In Fig. 20b two electrode

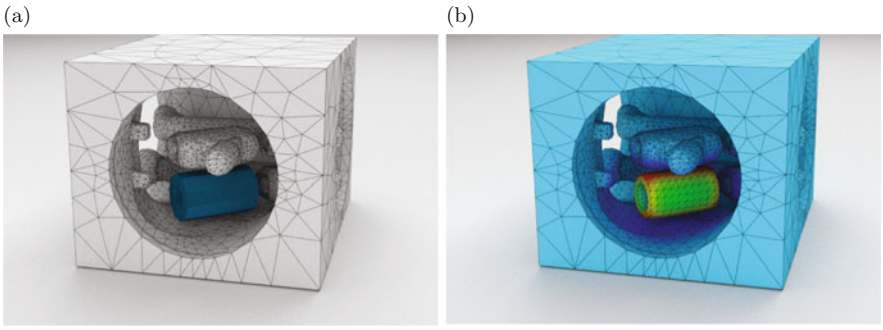


Fig. 19 Gas insulated switchgear with coarse and refined electrode meshes. The considered electrode is shown in *dark blue* in (**a**). The isocontours in (**b**) represent the normal flux

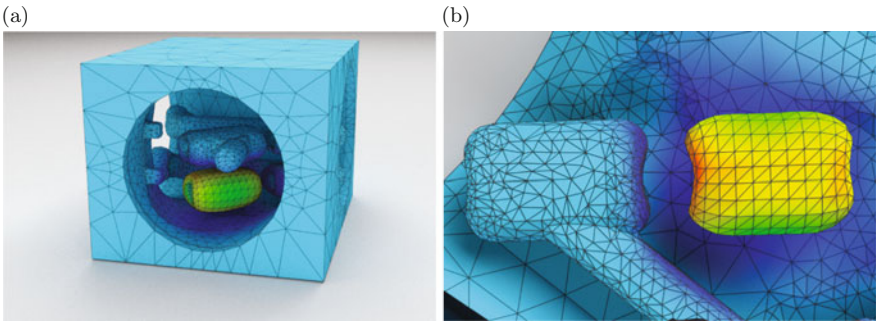


Fig. 20 Gas insulated switchgear with the optimised electrode geometry. The isocontours represent the normal flux. (**a**) Isometric view. (**b**) Top view

geometries are visible. The one in blue is an electrode geometry that has been obtained over the years by combining engineering intuition with simple calculations and testing. The similarities between the systematically shape optimised and the electrode geometry in production are striking. Notice in particular the saddle shapes at the two ends of the both electrodes, which help to reduce the large normal fluxes that are present at the sharp edges of the inner hole.

5.3 Subdivision Finite Element Discretised Thin-Shell Problems

5.3.1 Inverted Catenary Arch

In this verification example we optimise a thin strip pinned at both ends and subjected to a vertical distributed load. The optimal shape for such a strip is (in the limit of zero thickness) a catenary because for shells it is energetically more favourable to carry loads through membrane action instead of bending action. The catenary is the shape assumed by a loose string hung freely from two fixed points [24], its equation reads

$$y = \alpha \cosh\left(\frac{x+\beta}{\alpha}\right) + \gamma\,, \tag{48}$$

where the y-axis is parallel to the applied load vector and the three constants α, β and γ are determined based on the location of the supports and the string length l. Two cases are studied, one where the supports are at equal height and the other where one support is higher than the other. In both cases the horizontal distance between the supports is 1. The vertical offset between the supports in the second case is 0.2. Initially, the strip is a narrow flat plate connecting the supports with length equal to

the distance between the supports. Width and thickness are constant with 0.05 and 0.02, respectively. The magnitude of the vertical uniformly distributed load is 1000, the Young's modulus and Poisson's ratio are $E = 2 \times 10^8$ and $\nu = 0.3$, respectively. The length l of the optimised strip is chosen with 1.1 and 1.3.

The Catmull-Clark subdivision scheme is used for geometry representation and for discretising the thin-shell equations. The initial coarse mesh used for optimisation contains only 3 elements along the length and 1 element across the width of the strip. This increases to 48 in the twice subdivided fine mesh used for analysis. During optimisation the mesh resolution is increased starting $\ell_o = 0$ up to $\ell_o = 2$. Only the y coordinates of the control points are optimised. The comparison of the optimisation results with the corresponding catenary curve for different curve lengths and support positions is shown in Fig. 21. The reduction of the objective function is more than 99.9 % for all cases and the results show good visual agreement with catenary curves. The slight deviations from the catenary are possibly due to the finite width of the strip, which leads during optimisation to some curvature generation across the width of the shell (not visible in Fig. 21).

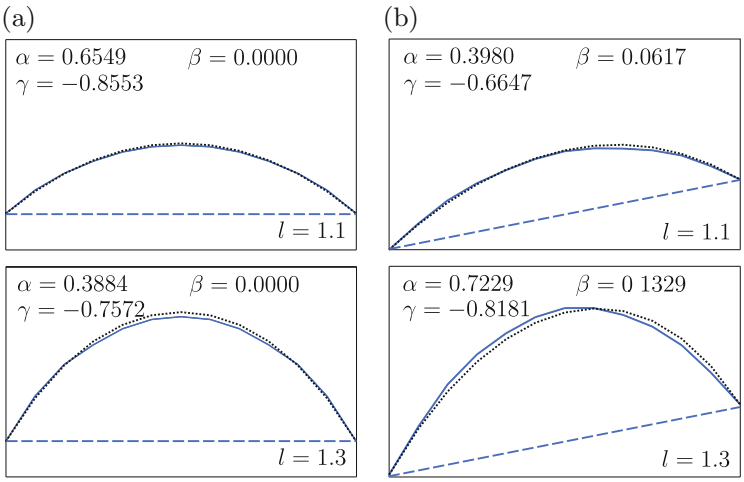


Fig. 21 Inverted catenary arch. The *dashed* and *solid blue lines* show the centre lines of the strip before and after optimisation, respectively. The *dotted black line* is the catenary curve corresponding to length l. The α, β and γ values are the parameters in (48). In the chosen coordinate system the left support has the coordinates $(-0.5, 0.0)$. (**a**) Supports at same height. (**b**) Supports at different heights

5.3.2 Architectural Roof

We consider the optimisation of an initial roof design shown in Fig. 22 with approximate dimensions of $2.31 \times 6.27 \times 0.75$. This design was created by an architect using the Autodesk Maya software. A vertical uniformly distributed load of 1000 is chosen as the design load and the shell thickness is $t = 0.02$. The Young's modulus and the Poisson's ratio are $E = 1 \times 10^{10}$ and $\nu = 0.2$, respectively.

During the shape optimisation only the vertical coordinates of control points are chosen as design variables. Moreover, an area constraint is applied in order to restrict the size of the optimised shape. An important architectural feature of the roof is the ridge profile, which is preserved by applying special extended subdivision stencils at the corner vertices and crease edges, see Figs. 22 and 23.

The coarse mesh at optimisation level $\ell_o = 0$ contains 26 nodes which is twice subdivided to obtain the computation mesh at level $\ell_c = 2$ with 272 nodes. The second optimisation stage is done at level $\ell_o = 1$ but no optimisation is done at level $\ell_o = 2$ as this results in oscillatory surface features. The optimised roof design is shown in Fig. 24. The initial value of the compliance cost function is 31.36 and is reduced by 79.13 % through optimisation.

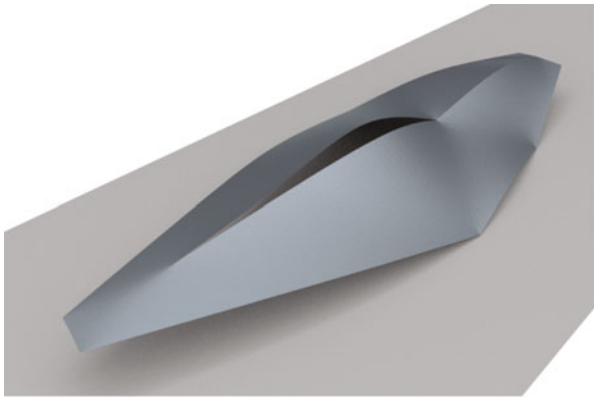


Fig. 22 Limit surface of the initial architectural roof design. The model has three supports, a central opening and a roof ridge. The corresponding coarse control mesh and selected vertex and edge tags are shown in Fig. 23

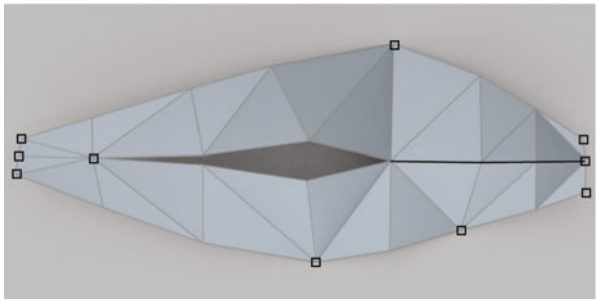


Fig. 23 Top view of the initial control mesh of the architectural roof. The nodes indicated by *black squares* are tagged as corner and the edges highlighted by the *black solid line* as crease

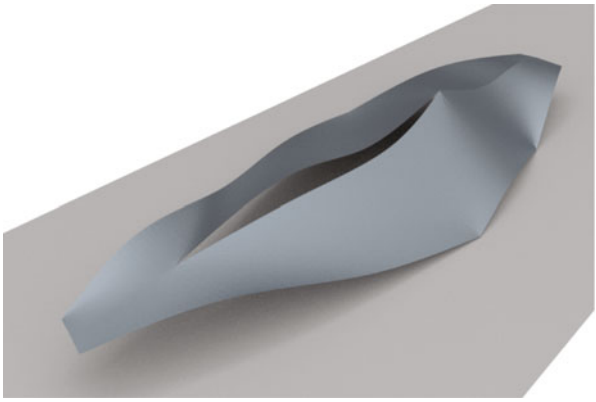


Fig. 24 Limit surface of the optimised architectural roof design

6 Summary and Conclusions

We reviewed the multiresolution optimisation technique based on subdivision surfaces and presented its application to elasticity, electrostatics and thin-shell problems. For discretising the governing partial differential equations we employed three different discretisation techniques, namely immersed finite elements for elasticity, boundary elements for electrostatics and subdivision finite elements for thin-shells. The inherent hierarchy of the subdivision surfaces allows us to consider the same surface at different resolutions and to take advantage of multiresolution editing techniques. Starting from the coarsest control mesh increasingly finer meshes are used for geometry updating and always a fine mesh is used for analysis. As demonstrated with the computed examples, this effectively inhibits the appearance of non-physical geometry oscillations that may occur in shape optimisation. Moreover, any pathological element distortions on the computational mesh are practically avoided. As a result, there is no need to regenerate or smooth the boundary mesh during the optimisation. Finally, it is evident that the multiresolution editing semantics is particularly appealing for isogeometric analysis because it enables the decoupling of geometry and analysis representations.

Acknowledgements The partial support of the EPSRC through grant # EP/G008531/1 and EC through the Marie Curie Actions (IAPP) program CASOPT project are gratefully acknowledged.

References

1. G. Allaire, F. Jouve, A.M. Toader, Structural optimization using sensitivity analysis and a level-set method. J. Comput. Phys. **194**, 363–393 (2004)
2. K. Bandara, F. Cirak, Multiresolution shape optimisation of thin-shell structures with subdivision finite elements. Technical report, University of Cambridge (2015)

3. K. Bandara, T. Rüberg, F. Cirak, Shape optimisation with multiresolution subdivision surfaces and immersed finite elements. arXiv:1510.02719 (2015)
4. K. Bandara, F. Cirak, G. Of, O. Steinbach, J. Zapletal, Boundary element based multiresolution shape optimisation in electrostatics. J. Comput. Phys. **297**, 584–598 (2015)
5. H. Biermann, A. Levin, D. Zorin, Piecewise smooth subdivision surfaces with normal control, in *SIGGRAPH 2000 Conference Proceedings*, New Orleans, 2000, pp. 113–120
6. K.U. Bletzinger, S. Kimmich, E. Ramm, Efficient modeling in shape optimal design. Comput. Syst. Eng. **2**, 483–495 (1991)
7. P.B. Bornemann, F. Cirak, A subdivision-based implementation of the hierarchical b-spline finite element method. Comput. Methods Appl. Mech. Eng. **253**, 584–598 (2013)
8. V. Braibant, C. Fleury, Shape optimal design using b-splines. Comput. Methods Appl. Mech. Eng. **44**, 247–267 (1984)
9. E. Catmull, J. Clark, Recursively generated b-spline surfaces on arbitrary topological meshes. Comput.-Aided Des. **10**, 350–355 (1978)
10. F. Cirak, Q. Long, Subdivision shells with exact boundary control and non-manifold geometry. Int. J. Numer. Methods Eng. **88**, 897–923 (2011)
11. F. Cirak, M. Ortiz, P. Schröder, Subdivision surfaces: a new paradigm for thin-shell finite-element analysis. Int. J. Numer. Methods Eng. **47**, 2039–2072 (2000)
12. F. Cirak, M. Scott, E. Antonsson, M. Ortiz, P. Schröder, Integrated modeling, finite-element analysis, and engineering design for thin-shell structures using subdivision. Comput.-Aided Des. **34**, 137–148 (2002)
13. M. Delfour, J.P. Zolésio, *Shapes and Geometries: Analysis, Differential Calculus, and Optimization*. Advances in Design and Control (SIAM, Philadelphia, 2001)
14. D. Doo, M. Sabin, Behavior of recursive division surfaces near extraordinary points. Comput.-Aided Des. **10**, 356–360 (1978)
15. M. Flucher, M. Rumpf, Bernoulli's free-boundary problem, qualitative theory and numerical approximation. Journal für die reine und angewandte Mathematik **486**, 165–204 (1997)
16. D. Fußeder, B. Simeon, A.V. Vuong, Fundamental aspects of shape optimization in the context of isogeometric analysis. Comput. Methods Appl. Mech. Eng. **286**, 313–331 (2015)
17. R.T. Haftka, R.V. Grandhi, Structural shape optimization – a survey. Comput. Methods Appl. Mech. Eng. **57**, 91–106 (1986)
18. J. Haslinger, R.A.E. Mäkinen, *Introduction to Shape Optimization: Theory, Approximation, and Computation* (SIAM, Philadelphia, 2003)
19. T. Hughes, J. Cottrell, Y. Bazilevs, Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. Comput. Methods Appl. Mech. Eng. **194**, 4135–4195 (2005)
20. S.G. Johnson, The NLopt nonlinear-optimization package. http://ab-initio.mit.edu/nlopt
21. J. Kiendl, R. Schmidt, R. Wüchner, K.U. Bletzinger, Isogeometric shape optimization of shells using semi-analytical sensitivity analysis and sensitivity weighting. Comput. Methods Appl. Mech. Eng. **274**, 148–167 (2014)
22. J.M. Lane, R.F. Riesenfeld, A theoretical development for the computer generation and display of piecewise polynomial surfaces. IEEE Trans. Pattern Anal. Mach. Intell. **PAMI-2**, 35–46 (1980)
23. N. Litke, A. Levin, P. Schröder, Fitting subdivision surfaces, in *IEEE Visualization 2001*, San Diego, 2001, pp. 319–324
24. E.H. Lockwood, *A Book of Curves* (Cambridge University Press, Cambridge, 1961)
25. C. Loop, Smooth subdivision surfaces based on triangles. Master's thesis, Department of Mathematics, University of Utah (1987)
26. M. Lounsbery, T.D. DeRose, J. Warren, Multiresolution analysis for surfaces of arbitrary topological type. ACM Trans. Graph. **16**, 34–73 (1997)
27. J. Peters, U. Reif, *Subdivision Surfaces*. Springer Series in Geometry and Computing (Springer, Berlin/Heidelberg, 2008)
28. T. Rüberg, F. Cirak, An immersed finite element method with integral equation correction. Int. J. Numer. Methods Eng. **86**, 93–114 (2011)

29. T. Rüberg, F. Cirak, Subdivision-stabilised immersed b-spline finite elements for moving boundary flows. Comput. Methods Appl. Mech. Eng. **209–212**, 266–283 (2012)
30. T. Rüberg, F. Cirak, A fixed-grid b-spline finite element technique for fluid–structure interaction. Int. J. Numer. Methods Fluids **74**(9), 623–660 (2014)
31. R. Sanches, P. Bornemann, F. Cirak, Immersed b-spline (i-spline) finite element method for geometrically complex domains. Comput. Methods Appl. Mech. Eng. **200**, 1432–1445 (2011)
32. D. Schillinger, L. Dede, M.A. Scott, J.A. Evans, M.J. Borden, E. Rank, T.J.R. Hughes, An isogeometric design-through-analysis methodology based on adaptive hierarchical refinement of NURBS, immersed boundary methods, and T-spline CAD surfaces. Comput. Methods Appl. Mech. Eng. **249–253**, 116–150 (2012)
33. J. Sokolowski, J.P. Zolesio, *Introduction to Shape Optimization: Shape Sensitivity Analysis* (Springer, Berlin/Heidelberg, 1992)
34. J. Stam, Exact evaluation of Catmull-Clark subdivision surfaces at arbitrary parameter values, in *SIGGRAPH 1998 Conference Proceedings*, Orlando, 1998, pp. 395–404
35. J. Stam, Evaluation of Loop subdivision surfaces, in *SIGGRAPH 1999 Course Notes*, Los Angeles (1999)
36. O. Steinbach, *Numerical Approximation Methods for Elliptic Boundary Value Problems. Finite and Boundary Elements* (Springer, New York, 2008)
37. W. Wall, M. Frenzel, C. Cyron, Isogeometric structural shape optimizisation. Comput. Methods Appl. Mech. Eng. **197**, 2976–2988 (2008)
38. J. Warren, H. Weimer, *Subdivision Methods for Geometric Design: A Constructive Approach* (Morgan Kaufmann, San Francisco, 2001)
39. L. Ying, D. Zorin, Nonmanifold subdivision, in *IEEE Visualization 2001*, San Diego, 2001, pp. 325–332
40. D. Zorin, P. Schröder, Subdivision for modeling and animation, in *SIGGRAPH 2000 Course Notes*, New Orleans (2000)
41. D. Zorin, P. Schröder, W. Sweldens, Interactive multiresolution mesh editing, in *Proceedings of the 24th annual conference on computer graphics and interactive techniques (SIGGRAPH)*, Los Angeles, (ACM Press/Addison-Wesley, New York, 1996), pp. 259–268

Atom Simplification and Quality T-mesh Generation for Multi-resolution Biomolecular Surfaces

Tao Liao, Guoliang Xu, and Yongjie Jessica Zhang

Abstract In this paper, we present an algorithm to simplify low-contributing atoms and generate quality T-meshes for multi-resolution biomolecular surfaces. The structure of biomolecules is first simplified using an error-bounded atom elimination method. An extended cross field-based parameterization method is then developed to adapt the parametric line spacings to different surface resolutions. Moreover, an anisotropy defined from an input scalar field can also be achieved. From the parameterization results, we extract adaptive and anisotropic T-meshes for the further T-spline surface construction. Finally, a gradient flow-based method is developed to improve the T-mesh quality, with the anisotropy preserved in the quadrilateral elements. The effectiveness of the presented algorithm has been verified using several large biomolecular complexes.

1 Introduction

Geometric modeling of biomolecules plays an important role in various applications [10, 22, 36, 37, 45]. New challenges arise in simulations involving extremely large biomolecular complexes. The computational cost for biomolecular surface construction increases as the atom number becomes progressively larger. Generally, biomolecular complexes are formed by a binding of several biomolecules, and a specific biomolecule can be contained in different complexes. For example in human cardiac muscle, there are many different types of filament lattice complexes, and they all contain the same biomolecule called filament protein. Simplifying the

T. Liao • G. Xu • Y.J. Zhang (✉)
Department of Mechanical Engineering, Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, PA 15213, USA

Institute of Computational Mathematics, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China
e-mail: taoliao@andrew.cmu.edu; xuguo@lsec.cc.ac.cn; jessicaz@andrew.cmu.edu

B. Jüttler, B. Simeon (eds.), *Isogeometric Analysis and Applications 2014*,
Lecture Notes in Computational Science and Engineering 107,
DOI 10.1007/978-3-319-23315-4_7

structure of filament protein can significantly improve the modeling efficiency in building different types of filament lattice complexes.

Various methods have been developed for biomolecular surface generation [11, 12]. Some of them construct biomolecular surface in an explicit way, such as alpha-shapes [1, 16], MSMS [35], advancing front and generalized Delaunay approaches [24], NURBS approximation [3], and PDE-based methods [19, 42, 49]. Other methods implicitly represent biomolecular geometry using a Gaussian density map, and then extract surface or volumetric meshes from it [44, 48]. Techniques were developed to improve the computational efficiency, such as PDE transform based on Fast Fourier Transform (FFT) [19, 49], Non-uniform Fast Fourier Transform (NFFT) [4], coarse-grained modeling technique [2], as well as CPU and GPU parallel computation based on neighboring search, KD-tree and Bounding Volume Hierarchy (BVH) [26].

In the context of isogeometric analysis [20, 38, 39], the T-spline surface provides a powerful basis for computation in different applications, which can also greatly benefit the analysis of biomolecules. Due to the high efficiency of the T-spline control mesh, namely T-mesh, the multi-resolution feature on the biomolecular surface can be represented efficiently. Various methods have been developed for T-mesh generation [28, 40, 41]. In recent years, the cross field-based global parameterization methods were introduced in surface quadrangulation [6], which capture surface features based on the principal curvature directions [7, 21, 23] or eigenfunctions of the Laplace-Beltrami operator [25]. Various methods were introduced to reduce distortion in the parameterization. For example, the distortion in the cross field was minimized in [30, 31], the mapping of the input triangle meshes was optimized in [27], and the local bijectivity was achieved in [29]. More contributions have been made in different aspects such as controlling the scale of captured features [15] and achieving the anisotropy in parametric lines, which has been fulfilled based on the metric tensor [23], adapted cross field [25] or surface deformation [32]. The integer-grid map [5] was introduced to achieve quadrangulation in the surface parameterization, and a robust method was developed to extract quadrilateral meshes [14]. Other than the cross field, a novel parameterization method was introduced based on the N-poly vector field [13], which can avoid the mixed-integer problem. These techniques provide a nice basis for us to build T-meshes for biomolecules with multi-resolution features preserved efficiently.

During Gaussian density map construction, faraway atoms have little contribution to the bimolecular surface due to the decay of Gaussian kernel function. In this paper, an atom simplification method is developed to eliminate these atoms from the biomolecular structure. Based on the simplified structure, a multi-resolution biomolecular surface can be built for quality T-mesh generation.The main

contributions in this paper include:

1. An error-bounded atom simplification algorithm is introduced for biomolecular complexes to improve the computational efficiency of density map construction;
2. An extended cross field-based method is developed for adaptive and anisotropic parameterization, which adapts the parametric line spacings to multi-resolution surface features; and
3. A new gradient flow-based method is introduced for T-mesh quality improvement, preserving the anisotropy in quadrilateral elements robustly.

The remainder of this paper is organized as follows. Section 2 describes our atom simplification method for biomolecular complexes. Section 3 explains the extended parameterization algorithm together with the new quality improvement method. Section 4 shows the results. Finally, Sect. 5 draws conclusions and points out the future work.

2 Algorithm Overview and Atom Simplification

Figure 1 shows the main steps of our algorithm. Biomolecular structures from the Protein Data Bank (PDB) are simplified to improve the modeling efficiency, which are used for multi-resolution triangular surface generation. Then T-meshes are generated based on surface parameterization. The two main components of our work (atom simplification and T-mesh construction) are marked in orange and green, respectively.

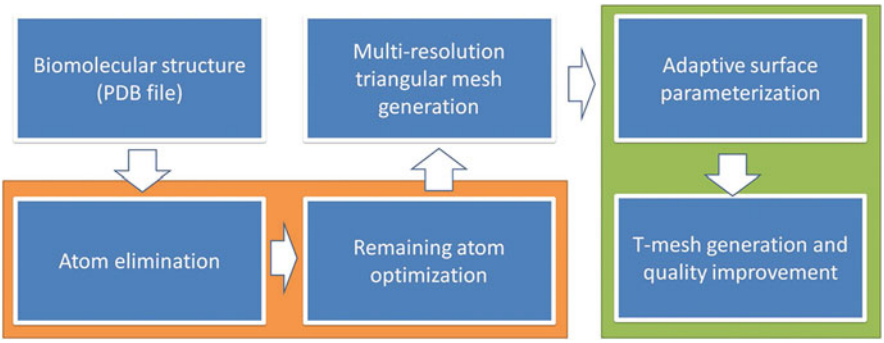


Fig. 1 The main steps of T-mesh generation for multi-resolution biomolecular surfaces. Note that the simplification process in the *orange* area is not required if the simplified structure is already prepared

In this paper, we simplify the structure of biomolecular complexes by eliminating low-contributing atoms. The efficiency of density map construction can be improved since the atom number is reduced. There are two main steps in our atom simplification: (1) eliminating the low-contributing atoms; and (2) optimizing the center location, the radius and the decay rate of the remaining atoms to minimize the error in the density map around the biomolecular surface.

2.1 Atom Elimination

Biomolecules, especially proteins, have hierarchical structures. They consist of several chains, each chain is formed by a set of residues, and each residue contains a group of atoms. The Gaussian density can be computed using a multi-level summation of Gaussian kernel functions [26, 48],

$$G(\mathbf{x}) = \sum_{k \in U_C} \left(\sum_{j \in U_R^k} \left(\sum_{i \in U_A^j} e^{-\kappa_i \left(\|\mathbf{x} - \mathbf{x}_i\|^2 - r_i^2 \right)} \right)^{P_R^{k,j}} \right)^{P_C^k}, \tag{1}$$

where U_C is the set of chains in the biomoleucle, U_R^k is the set of residues in Chain k, U_A^j is the set of atoms in Residue j, $\left(P_R^{k,j}, P_C^k\right)$ are coefficients controlling the surface resolution, and $(\mathbf{x}_i, r_i, \kappa_i)$ are the atom center, the radius and the decay rate of Atom i. The biomolecular surface is represented by a Gaussian density map implicitly, which can be extracted as an isosurface $\{M : G(\mathbf{x}) = 1\}$ in a rectilinear grid. In this paper, the grid spacings in x, y and z directions are set to be 0.3Å. Generally, smaller values for P_R and P_C result in lower resolution, while larger values of P_R and P_C result in higher resolution on the biomolecular surface. We choose various P_R and P_C values for different residues and chains, resulting in a multi-resolution surface [26].

Let S_A be the set of all the atoms, S_R be the set of remaining atoms after eliminating the low-contributing atoms, and $G_R^-(\mathbf{x}, i)$ be the Gaussian density from all the atoms in S_R except Atom i. During the atom elimination, the change in Gaussian density around the biomolecular surface should be sufficiently small. In a rectilinear grid, a region Ω_S around the surface consists of a set of grid points $\mathbf{x}$ with Gaussian density $g_l < G(\mathbf{x}) < g_u$, here we choose $(g_l, g_u) = (0.9, 1.1)$. Due to the decay of Gaussian kernel functions, eliminating Atom i only influences the

Gaussian density map in its neighboring region Ω_i, where all its grid points $\mathbf{x}$ satisfy

$$\|\mathbf{x} - \mathbf{x}_i\|^2 \leq -\frac{\ln \epsilon_r}{\kappa_i} + r_i^2, \tag{2}$$

and ϵ_r is a pre-defined threshold (e.g. $\epsilon_r = 10^{-6}$). The Gaussian density error e_G^i of ignoring Atom i can be defined as

$$e_G^i = \max_{\mathbf{x} \in \Omega_i \cap \Omega_S} \left| \frac{G(\mathbf{x}) - G_R^-(\mathbf{x}, i)}{G(\mathbf{x})} \right|. \tag{3}$$

In a rectilinear grid, the overall contribution C_i of Atom i to the surface is defined as

$$C_i = \left(\sum_{\mathbf{x} \in \Omega_i \cap \Omega_S} \left(\frac{G^-(\mathbf{x}, i)}{G(\mathbf{x})} \right)^2 \right)^{-1/2}, \tag{4}$$

where $G(\mathbf{x})$ and $G^-(\mathbf{x}, i)$ are the Gaussian density with and without considering Atom i, respectively. We sort all the atoms based on their overall contributions, and keep eliminating the one with the lowest contribution until we find an atom with the Gaussian density error defined in Eq. (3) greater than ϵ_G, where ϵ_G is an input threshold. After atom elimination, we build the Gaussian density map with the remaining atoms and extract a triangle mesh to represent the simplified surface. Let M be the triangle mesh. The surface error in M can be measured as

$$e_G^M = \max_{\mathbf{x} \in M} \left| \frac{G(\mathbf{x}) - 1}{G(\mathbf{x})} \right|, \tag{5}$$

where $\mathbf{x}$ is a vertex in the mesh. See Algorithm 1 for the details.

During Gaussian density map construction, a neighboring search and BVH-based GPU parallel method is employed for efficient computation [26]. In this paper biomolecules are represented by their PDB IDs. Figure 2 shows the simplification results of 2O53 with different ϵ_G values and coefficients (P_R, P_C). As ϵ_G increases, the Gaussian density error on the surface e_G^M tends to increase faster in concave and valley areas than the rest of the surface. As shown in Table 1, as the surface resolution is improved with larger P_R, less atoms can be eliminated. This is because more atoms are exposed on the high resolution surface than the low resolution one.

Algorithm 1 Atom elimination

$S_R := S_A$;
for each grid point **x do**
  Compute Gaussian density value $G(\mathbf{x})$;
end for
Identify the region Ω_S around the biomolecular surface, where $g_l < G(\mathbf{x}) < g_u$;
for each atom $i \in S_A$ **do**
  Identify its neighboring region Ω_i using Eq. (2);
  Compute the overall contribution C_i to the surface in $\Omega_i \cap \Omega_S$;
end for
Sort all the atoms in S_A based on C_i;
for the atom with the lowest contribution in S_R **do**
  Set this atom to be Atom i;
  Compute the Gaussian density error e_G^i using Eq. (3);
  if $e_G^i > \epsilon_G$ **then**
    STOP;
  else
    $S_R := S_R \setminus \{i\}$;
    Continue;
  end if
end for
Output the surface error e_G^M according to Eq. (5).

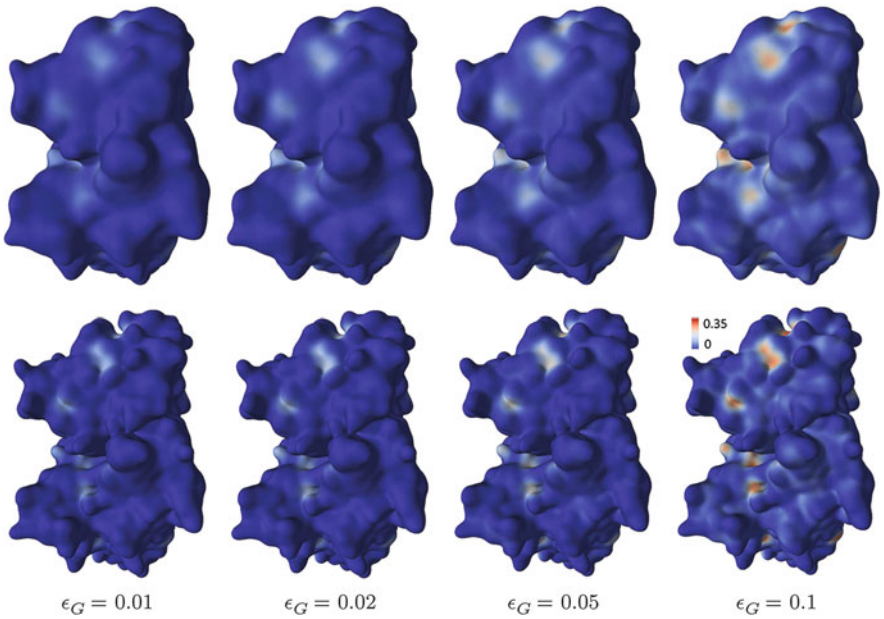


Fig. 2 Biomolecular surfaces for the simplified protein 2O53 with $P_C = 1.0$ and $\kappa = 0.3$. *Top row*: $P_R = 0.25$; *Bottom row*: $P_R = 0.5$. The color represents the distribution of e_G^M

Table 1 Atom Elimination Results for 2O53

		$\epsilon_G = 0.0$	$\epsilon_G = 0.01$	$\epsilon_G = 0.02$	$\epsilon_G = 0.05$	$\epsilon_G = 0.1$
Case 1	N_R	4,912(100 %)	3,316(67.5 %)	3,011(61.3 %)	2,810(57.2 %)	2,557(52.1 %)
	e_G^M	0.0	0.021	0.032	0.057	0.221
	Optimized e_G^M	–	0.019	0.030	0.053	0.215
	T_G	0.92	0.55	0.52	0.47	0.42
Case 2	N_R	4,912(100 %)	3,662(74.6 %)	3,513(71.5 %)	3,362(68.4 %)	3,102(63.2 %)
	e_G^M	0.0	0.023	0.028	0.060	0.210
	Optimized e_G^M	–	0.021	0.024	0.057	0.208
	T_G	0.98	0.81	0.79	0.76	0.72

Note: Case 1 – $P_R = 0.25$; Case 2 – $P_R = 0.5$; N_R – the number of remaining atoms; atoms; e_G^M – the surface error defined in Eq. (5); and T_G – time for Gaussian density map computation (unit: second). The percentage of remaining atoms is shown in parentheses

2.2 Optimization of Remaining Atoms

To reduce the surface error of the simplified biomolecule, the parameters of the remaining atoms can be optimized. Let N_R be the number of remaining atoms. The atoms in S_R are sorted, and their parameters are optimized using a gradient flow method. The objective function is defined as

$$\mathbf{F}(\mathbf{f}) = \int_{\mathbb{R}^3} (G(\mathbf{x}) - G_R(\mathbf{f}, \mathbf{x}))^2 \mathrm{d}\mathbf{x}, \tag{6}$$

and in a rectilinear grid we have

$$\mathbf{F}(\mathbf{f}) = \sum_{l=1}^{N_G} (G(\mathbf{x}_l) - G_R(\mathbf{f}, \mathbf{x}_l))^2, \tag{7}$$

where N_G is the number of grid points, and $\mathbf{f} = (x_1, y_1, z_1, r_1, \kappa_1, \ldots, x_{N_R}, y_{N_R}, z_{N_R}, r_{N_R}, \kappa_{N_R})^T$. The initial value $\mathbf{f}^0$ is given by the original centers, radii and decay rates of the remaining atoms. For the kth iteration of the gradient flow-based optimization, we have

$$\mathbf{f}^{k+1} = \mathbf{f}^k - \tau \nabla \mathbf{F}, \tag{8}$$

where τ is the step length, and

$$\nabla \mathbf{F} = \left(\frac{\partial \mathbf{F}}{\partial x_1}, \frac{\partial \mathbf{F}}{\partial y_1}, \frac{\partial \mathbf{F}}{\partial z_1}, \frac{\partial \mathbf{F}}{\partial r_1}, \frac{\partial \mathbf{F}}{\partial \kappa_1}, \ldots, \frac{\partial \mathbf{F}}{\partial x_{N_R}}, \frac{\partial \mathbf{F}}{\partial y_{N_R}}, \frac{\partial \mathbf{F}}{\partial z_{N_R}}, \frac{\partial \mathbf{F}}{\partial r_{N_R}}, \frac{\partial \mathbf{F}}{\partial \kappa_{N_R}} \right)^T.$$

As $\mathbf{F}$ is minimized step by step, the error in the Gaussian density map is minimized. In each step, the parameters of the remaining atoms are updated until the change in the parameters is less than a threshold ϵ_f,

$$\left\| \mathbf{f}^{k+1} - \mathbf{f}^k \right\|_\infty < \epsilon_f. \tag{9}$$

During the iteration in Eq. (8), the variation in $\mathbf{f}$ should be bounded. Let $\mathbf{x}_i^0$, r_i^0 and κ_i^0 be the original location, radius and decay rate of Atom i. If $\left\| \mathbf{x}_i^k - \mathbf{x}_i^0 \right\| > \rho_x$ (e.g., $\rho = 0.5$), then

$$\mathbf{x}_i^k = \mathbf{x}_i^0 + \rho_x \frac{\mathbf{x}_i^k - \mathbf{x}_i^0}{\left\| \mathbf{x}_i^k - \mathbf{x}_i^0 \right\|},$$

where ρ_x defines a sphere around $\mathbf{x}_i^0$ and the updated atom center should be inside the sphere. Similarly, the modifications in the radius $|r_i^k - r_i^0|$ and the decay rate $|\kappa_i^k - \kappa_i^0|$ are bounded to ρ_r and ρ_κ (e.g., $\rho_r = 0.5$ and $\rho_\kappa = 0.2$), respectively. As

shown in Table 1, the surface error e_G^M can be improved after the optimization of remaining atoms.

3 Multi-resolution T-mesh Construction

For the analysis of large biomolecular complexes, usually only a specific component is essential for accuracy. A multi-resolution surface can be used to represent the biomolecular surface, maintaining high resolution details on the surface for the emphasized components while providing low resolution for the rest. In this paper, the emphasized components are defined as specific chains in the biomolecular complexes based on their biological functions. For example in Fig. 3a, the chains belonging to actin (blue and orange) in the human Thin Filament protein should be emphasized because they are receptors for some inhibitors. For the emphasized chains, we keep all the atoms and also use larger coefficients (P_R, P_C) in Eq. (1) to obtain higher resolution; while for the rest of the biomolecule, we simplify atoms and choose smaller (P_R, P_C). Sometimes, sharp noises may happen in the Gaussian density map around the connection region of the lower and higher resolution surface, which can be removed by applying a low-pass filter based on the fast Fourier transform [17]. Then, the multi-resolution surface is extracted from the constructed Gaussian density map using the dual contouring method [47, 48], and adaptive triangular meshes are obtained. In the following, we will talk about how to construct surface parameterization and quality T-meshes using these triangular meshes.

3.1 Surface Parameterization and T-mesh Construction

In this paper, we extend the cross field-based parameterization method [7, 21, 23, 25, 33, 34] to T-mesh generation for biomolecular surfaces. Firstly, a cross field is built on the triangle mesh guided by the principal curvature directions. Then an adaptive parameterization is computed based on the cross field. Anisotropy can also be achieved during surface parameterization. Finally, T-meshes are constructed by connecting the nodes with integer parametric coordinates.

Review of Cross Field-based Parameterization Here we briefly review the cross field generation and parametric coordinate computation, more details can be found in [7]. A cross field is defined in each triangle T_i with four perpendicular vectors, which can be represented as an angle θ_i referring to an edge $\mathbf{e}_i$ of the triangle, namely the reference edge. These vectors can be initialized using different inputs, such as the principal curvature directions. Then the cross field is smoothed by minimizing

the smoothness energy [7]

$$\Gamma^S = \sum_{e_{ij} \in E} (\theta_i + \varphi_{ij} + \frac{\pi}{2} p_{ij} - \theta_j)^2, \tag{10}$$

where e_{ij} is the edge shared by triangles T_i and T_j, E_T is the set of edges in the mesh, φ_{ij} is the angle between the reference edges of triangle T_i and T_j, and p_{ij} is the integer valued period jump of the cross field across e_{ij}. The principal curvature directions in a set of sparsely distributed triangles are chosen as the guidance for the cross field. By minimizing the smoothness energy, θ_i and p_{ij} are updated in each triangle T_i with the constraints from the guidance, and a smooth cross field is obtained, see Fig. 3b. Singularities of the cross field (red dots) can be located with the optimized θ_i and p_{ij} in each triangle [7]. Then, the surface is cut into a disk-like planar region with all the singularities on its boundary (black lines). The parametric coordinates (u, v) of

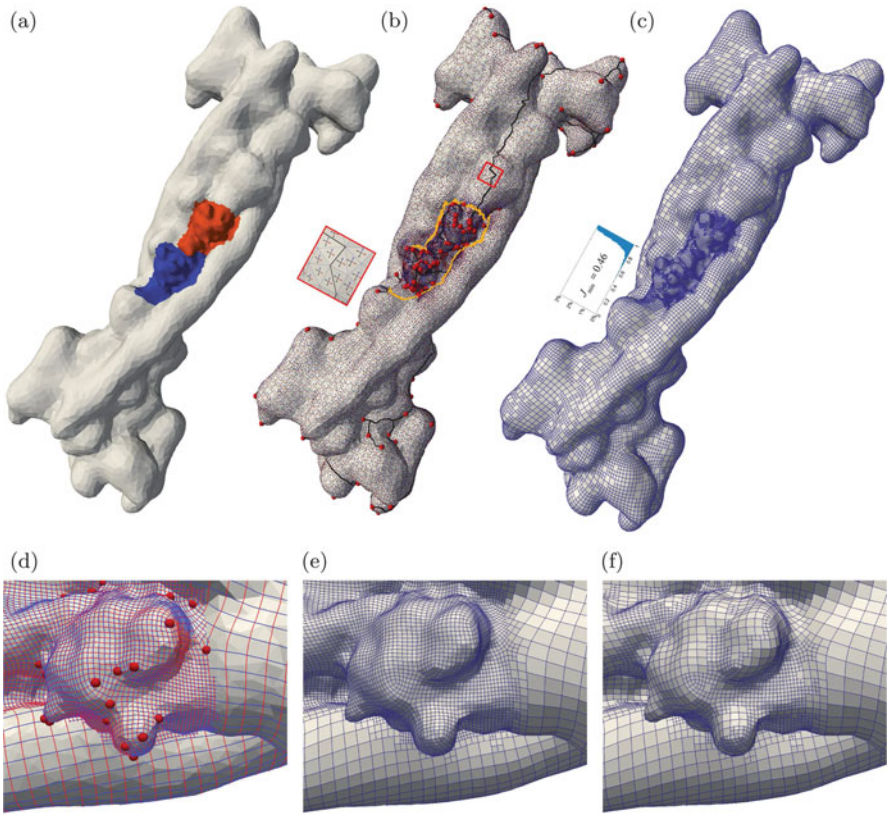


Fig. 3 Adaptive parameterization for the multi-resolution surface of 2W4U. (**a**) The multi-resolution triangle surface; (**b**) the cross field; (**c**) the adaptive T-mesh; (**d**) the parametric lines; and (**e**, **f**) the T-meshes corresponding to (**d**) before and after the removal of redundant vertices

each vertex behave as two piece-wise linear scalar fields, which can be obtained by minimizing an orientation energy [7, 25, 33],

$$\Gamma^O = \sum_{i=1}^{N_T} A_i \left(\left\| h_i^u \nabla_{T_i} u - \mathbf{u}_i \right\|^2 + \left\| h_i^v \nabla_{T_i} v - \mathbf{v}_i \right\|^2 \right), \tag{11}$$

where N_T is the number of triangles, A_i is the area of triangle T_i, (h_i^u, h_i^v) are parameters controlling the spacings of the parametric lines, $(\nabla_{T_i} u, \nabla_{T_i} v)$ are gradients of (u, v) in T_i, and $(\mathbf{u}_i, \mathbf{v}_i)$ are two perpendicular vectors chosen from the cross field (red and blue arrows in Fig. 3b). Details about the cutting process and the choice of $(\mathbf{u}_i, \mathbf{v}_i)$ directions can be found in [7]. The parametric line spacings in u and v directions equal to $1/h_i^u$ and $1/h_i^v$, respectively. Integer constraints are set on the planar region boundary to ensure consistent parametric lines, enabling a valid quadrangulation of the surface. Minimizing the smoothness energy in Eq. (10) and the orientation energy in Eq. (11) are two mixed-integer problems, which can be solved using the greedy mixed-integer optimizer introduced in [8].

Adaptive T-mesh Generation For the multi-resolution surface in Fig. 3a, we need to generate denser elements in the high resolution patches (blue and orange) to capture the detailed features. This can be achieved by adapting the line spacings in the orientation energy in Eq. (11): smaller (h_i^u, h_i^v) values are set for the higher resolution regions, while larger values are set for others. Different from [7], the multi-resolution surface is first cut along the patch boundaries. Note that different patches are not completely independent, and each patch is connected with one of its neighbors on the boundary. Similar to [7], each patch is cut into a disk-like region based on a Dijkstra tree, and all the singularities are on the boundary of the disk-like region. Then the $(\mathbf{u}_i, \mathbf{v}_i)$ directions are rotated to keep consistent inside each patch. As shown in Fig. 4 when a higher resolution patch i (blue) is connected with a lower resolution patch j (yellow), we restrict $h_j = 2^m h_i$, where m is a positive integer. In

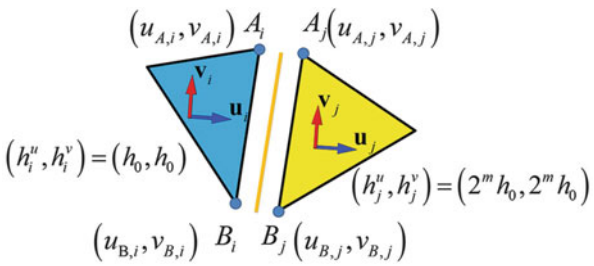


Fig. 4 Two triangles across the patch boundary (*orange*)

addition, the parametric coordinates at the patch boundaries should satisfy

$$\begin{cases} (u_{A,i}, v_{A,i})^T = 2^m \mathbf{R}_{AB} \left(u_{A,j}, v_{A,j}\right)^T + 2^m (I_u, I_v)^T, \\ (u_{B,i}, v_{B,i})^T = 2^m \mathbf{R}_{AB} \left(u_{B,j}, v_{B,j}\right)^T + 2^m (I_u, I_v)^T, \end{cases} \tag{12}$$

where $\mathbf{R}_{AB}$ is the rotation matrix across edge AB, (I_u, I_v) are integer-valued shift in the parametric coordinates, and m controls the difference of the parametric line spacing across the patch boundary. The parametric coordinates are computed by minimizing the orientation energy Γ^O with the constraints in Eq. (12). By connecting vertices with integer-valued parametric coordinates, a quadrilateral mesh can be built with T-junctions on the edges. To ensure the strongly-balanced structure in the T-mesh ($m \leq 1$ for any two neighboring elements), some quadrilateral elements in the transition region need to be subdivided.

Figure 3d, e shows the adaptive parameterization result for the multi-resolution surface (a local region) of Thin filament (2W4U). High resolution is set for two emphasized chains (Chains 18 and 20) with denser elements. To have a smooth transition from a higher resolution to a lower one, some T-mesh elements need to be modified. Figure 5 shows four different connections across the patch boundary (orange curve). Here $m = 2$, so some quadrilateral elements are subdivided to ensure a strongly-balanced structure. The obtained T-meshes may be too fine in some regions, therefore we identify redundant vertices based on their surface error and remove them from the T-meshes. Figure 3e, f shows a comparison between the T-meshes before and after removing redundant vertices.

Anisotropic T-mesh Generation Similar with [25], an anisotropy can be defined from an input scalar field f. Instead of using the principal curvatures, here we choose eigenfunction based cross-field guidance for anisotropic T-mesh generation.

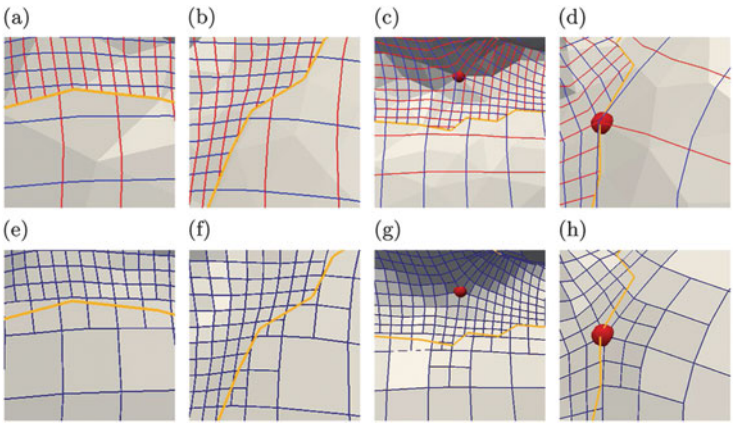


Fig. 5 Various parametric lines across the patch boundary and their resulting T-meshes. (**a**–**d**) The parametric lines; and (**e**–**h**) the corresponding T-meshes

In anisotropic T-meshes, quadrilateral elements are stretched in certain directions. The principal curvature directions usually vary intensively on the surface especially for regions with many detailed features, yielding a lot of singularities in the cross field, which makes it difficult to define the directions for stretching elements. On the contrary, low-mode eigenfunctions are very smooth and yield much fewer singularities. Their gradient directions in a set of sparsely distributed triangles are taken as the constraint for minimizing the smoothness energy in Eq. (10). After the cross field is built, singularities can be located using the same method in [7]. For the parametric lines following the gradient direction ∇f, h_i is set to be an uniform value h^0. For the perpendicular direction to ∇f, the line spacing $h_i^{\perp}$ is determined by the gradient magnitude,

$$\frac{1}{h_i^{\perp}} = \frac{\alpha \cdot \max\left(\|\nabla f\|\right)}{h^0 \, \|\nabla f_i\|}, \tag{13}$$

where α controls the minimum line spacing. If $\|\nabla f_i\| < \alpha \cdot \max\left(\|\nabla f\|\right)$, the line spacing is set to be $h_i^{\perp} = h^0$. As such, the quadrilateral elements are stretched along the direction perpendicular to the gradient, and dense elements are generated in the gradient direction. In Fig. 6, Mode 2 eigenfunction of the Laplace-Beltrami operator is used as the input scalar field. Similar with the isotropic parameterization in Fig. 3d, we adapt the parametric line spacings to the surface resolution by quadrupling the line spacings across the patch boundary.

3.2 *T-mesh Quality Improvement*

As shown in Fig. 7a, for each T-junction we extend it to make the surrounding elements form a local unstructured quadrilateral mesh. For a Vertex $\mathbf{x}_i$ in Element j, the other vertices in the element can be represented as $\mathbf{x}_{i+1}^j$, $\mathbf{x}_{i+2}^j$ and $\mathbf{x}_{i+3}^j$ in the counter-clockwise order, see Fig. 7b. The normal direction at Vertex $\mathbf{x}_i$ is computed as

$$\mathbf{n}_i = \sum_{j \in Q_i} \left(\mathbf{x}_{i+1}^j - \mathbf{x}_i\right) \times \left(\mathbf{x}_{i+3}^j - \mathbf{x}_i\right),$$

where Q_i is the set of neighboring elements of Vertex $\mathbf{x}_i$ and "$\times$" is the cross product of vectors. $\mathbf{n}_i$ is then normalized as $\mathbf{n}_i / \|\mathbf{n}_i\|$. In this way, $\mathbf{n}_i$ is estimated as the weighted average of normal directions in the neighboring elements, and the areas of these elements are taken as the weights. The *scaled Jacobian* [18, 46] at $\mathbf{x}_i$ in Element j equals to $\|\mathbf{v}_1 \times \mathbf{v}_2\| \cdot s$, where

$$\mathbf{v}_1 = \frac{\mathbf{x}_{i+1}^j - \mathbf{x}_i}{\left\|\mathbf{x}_{i+1}^j - \mathbf{x}_i\right\|}, \; \mathbf{v}_2 = \frac{\mathbf{x}_{i+3}^j - \mathbf{x}_i}{\left\|\mathbf{x}_{i+3}^j - \mathbf{x}_i\right\|}, \text{ and } s = \operatorname{sign}\left((\mathbf{v}_1 \times \mathbf{v}_2) \cdot \mathbf{n}_i\right).$$

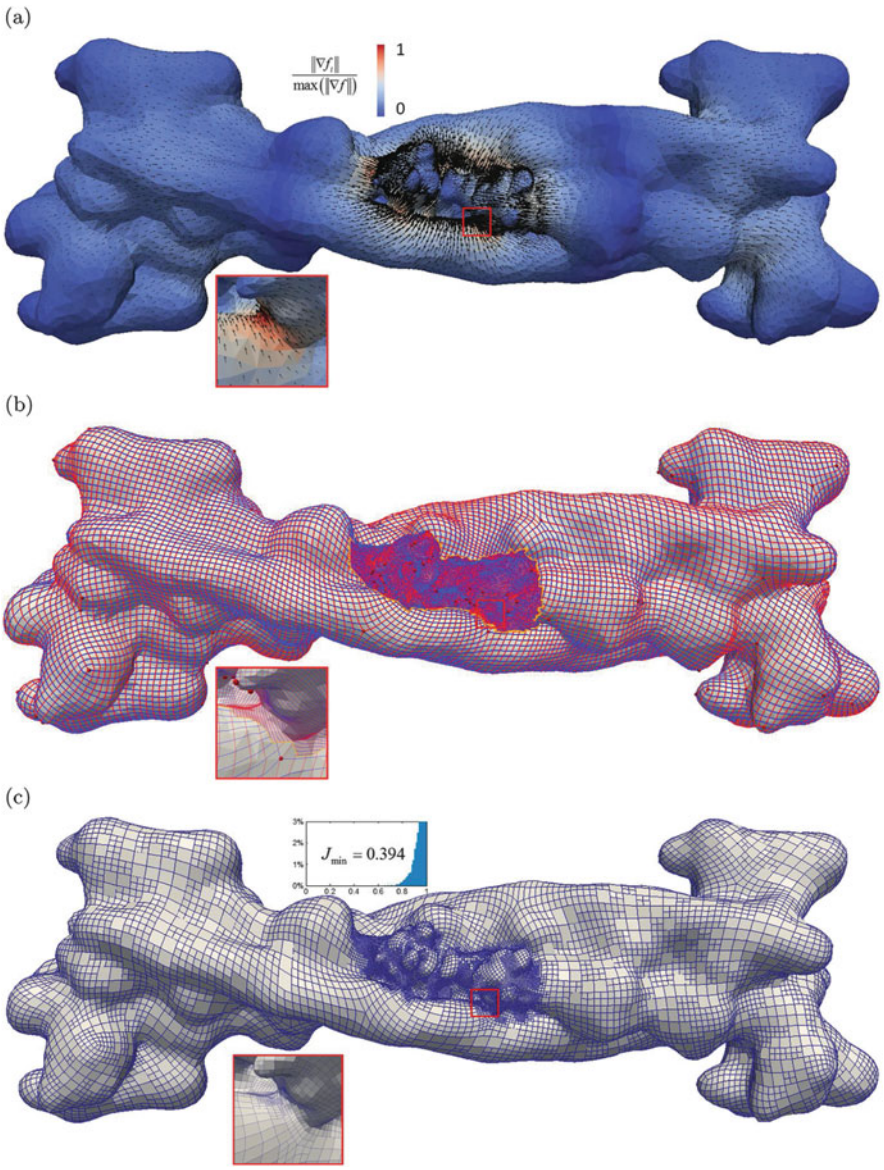


Fig. 6 Anisotropic T-mesh generation from the Mode 2 eigenfunction of LBO ($\alpha = 0.1$). (**a**) The gradient and gradient magnitude of the input field; (**b**) the adaptive and anisotropic parameterization; and (**c**) the corresponding T-mesh

The quality of Element j can be measured by the minimum Jacobian at its four vertices. Generally, the overall quality of the T-mesh from surface parameterizaion is good in term of Jacobian, except a few elements. To improve the Jacobian of

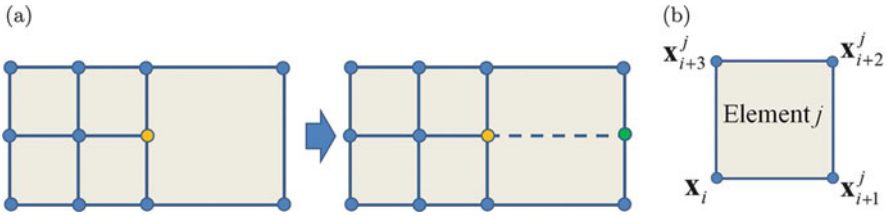


Fig. 7 (**a**) Extending the T-junction (*orange*) to form a local unstructured mesh; and (**b**) A neighboring element j surrounding Vertex $\mathbf{x}_i$

the T-mesh elements, we apply a new gradient flow-based quality improvement method, which considers both the original shape of elements and the orthogonality of edges. Contrast to other existing approaches [9, 43], our method can preserve the anisotropy of the T-mesh.

Suppose Element j in Fig. 7b is a neighboring element surrounding Vertex $\mathbf{x}_i$, the *cross distortion* at $\mathbf{x}_i$ is defined as

$$\Gamma^C(\mathbf{x}_i) = \sum_{j \in Q_i} \frac{\left\| \frac{\mathbf{x}^j_{i+1} - \mathbf{x}_i}{a_j} \right\|^2 + \left\| \frac{\mathbf{x}^j_{i+3} - \mathbf{x}_i}{b_j} \right\|^2}{\sigma_{ij}^2 + \sqrt{\sigma_{ij}^2 + 4\delta^2}}, \tag{14}$$

where $\sigma_{ij} = \mathbf{n}_i \cdot \left(\frac{\mathbf{x}^j_{i+1} - \mathbf{x}_i}{a_j} \times \frac{\mathbf{x}^j_{i+3} - \mathbf{x}_i}{b_j} \right)$, δ is an arbitrary small value (e.g. $\delta = 0.1$), and

$$(a_j, b_j) = \left(\frac{\left\| \mathbf{x}^j_{i+1} - \mathbf{x}_i \right\| + \left\| \mathbf{x}^j_{i+2} - \mathbf{x}^j_{i+3} \right\|}{2}, \frac{\left\| \mathbf{x}^j_{i+3} - \mathbf{x}_i \right\| + \left\| \mathbf{x}^j_{i+2} - \mathbf{x}^j_{i+1} \right\|}{2} \right)$$

are the *feature lengths* of Element j. The total cross distortion energy of the T-mesh is defined as

$$\Gamma^C_M = \sum_{i=1}^{N} \Gamma^C(\mathbf{x}_i), \tag{15}$$

where N is the number of vertices. Minimizing Γ^C_M not only makes the corners of each quadrilateral element close to 90°, but also makes the edge length adapt to the local surface feature, which improves the element quality and preserves its stretched shape. The minimization can be fulfilled via a gradient flow method.Let

$\mathbf{g} = (x_1, y_1, z_1, \ldots, x_N, y_N, z_N)^T$ be the vector containing the coordinates of all the vertices in the T-mesh.For the kth step of the gradient flow, we have

$$\mathbf{g}^{k+1} = \mathbf{g}^k - \tau \nabla \Gamma_M^C, \tag{16}$$

where

$$\nabla \Gamma_M^C = \left(\frac{\partial \Gamma_M^C}{\partial x_1}, \frac{\partial \Gamma_M^C}{\partial y_1}, \frac{\partial \Gamma_M^C}{\partial z_1}, \ldots, \frac{\partial \Gamma_M^C}{\partial x_N}, \frac{\partial \Gamma_M^C}{\partial y_N}, \frac{\partial \Gamma_M^C}{\partial z_N} \right)^T.$$

Note that in each step, vertices can only move on the tangent plane. Therefore, we update $\mathbf{x}_i^k$ with $\mathbf{x}_i^{k+1} - \left(\mathbf{n}_i \cdot \mathbf{p}_i^k\right) \mathbf{n}_i$, where $\mathbf{n}_i$ is the surface normal and $\mathbf{p}_i^k = \mathbf{x}_i^{k+1} - \mathbf{x}_i^k$. Our quality improvement method can also handle flat and concave elements although so far they were not observed in our T-mesh results.

We also implemented an isotropic T-mesh quality improvement method based on the triangle optimization [9, 43] and compared our method with it. For an isotropic T-mesh, the ideal shapes of the elements are squares or rectangles, as shown in Fig. 8, which can be decomposed into several *ideal triangles*. For a general quadrilateral element, we decompose it into several *real triangles* following the same splitting format. An affine mapping between a real triangle j and its corresponding ideal triangle j_I can be defined as

$$\begin{aligned} \mathbf{f}_\mathbf{K} : j_I &\to j, \\ \tilde{\mathbf{x}} &\mapsto \mathbf{x} = \mathbf{K}_j \tilde{\mathbf{x}} + \mathbf{v}, \end{aligned} \tag{17}$$

where $\mathbf{x}$ is a vertex in the real triangle, $\tilde{\mathbf{x}}$ is the corresponding vertex in the ideal triangle, and $\mathbf{v}$ is a constant term. The distortion of the T-mesh is defined as

$$\Gamma_M^T = \sum_{i=1}^{N_Q} \sum_{j \in R_i} \frac{\left\| K_j \right\|_F^2}{\det\left(\mathbf{K}_j\right) + \sqrt{\left(\det\left(\mathbf{K}_j\right)\right)^2 + 4\delta^2}}, \tag{18}$$

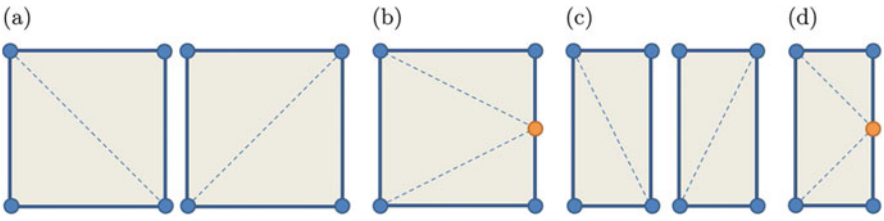


Fig. 8 Ideal triangles from a square (**a**–**b**) or rectangle (**c**–**d**). (**a**, **c**) No T-junction; and (**b**, **d**) with a T-junction. The *orange dots* are T-junctions

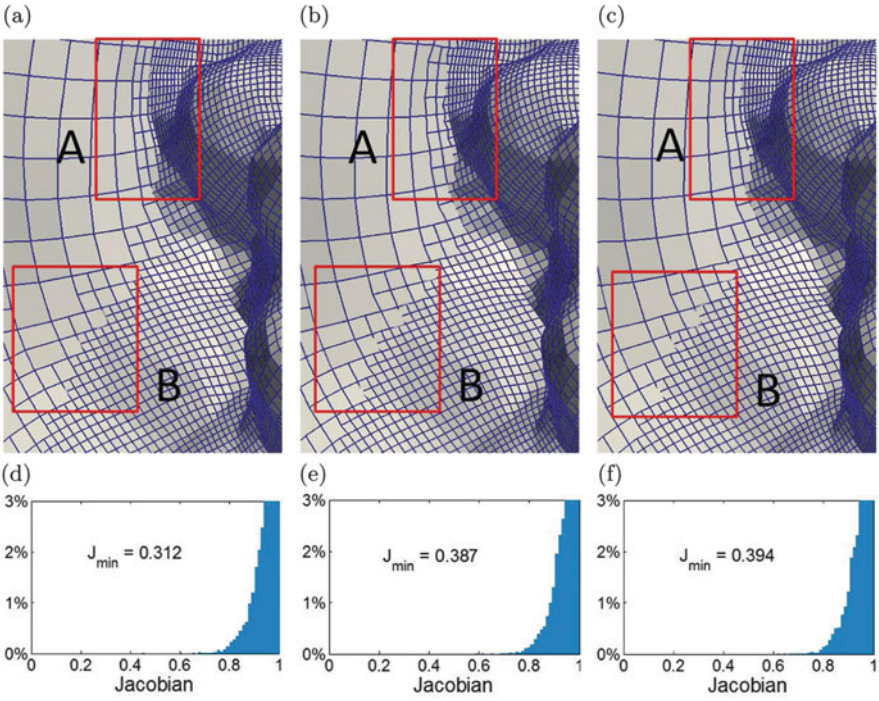


Fig. 9 Quality improvement results for the T-mesh in Fig. 6c using two different methods. (**a**) The original T-mesh; (**b–c**) the improved T-meshes using the triangle optimization method and our method, respectively; and (**d–f**) the Jacobian distribution corresponding to (**a–c**)

where N_Q is the number of quadrilateral elements, R_i is the set of real triangles generated from Element i, and $\|K_j\|_F = \sqrt{\text{tr}\left(\mathbf{K}_j{}^T\mathbf{K}_j\right)}$ is the Frobenius norm of $\mathbf{K}_j$. Similar with Eq. (16), a gradient flow method can be applied to minimize Γ_M^T.

Figure 9 shows a comparison between our method and the triangle optimization method. We can observe that both methods can improve the mesh quality. As shown in (b–c), for the isotropic elements in Region B, both methods yield similar results. But for the anisotropic elements in Region A, only our method preserves the anisotropic rectangle shape of the elements.

4 Results and Discussion

In this section, the biomolecule simplification and multi-resolution T-mesh generation algorithms are applied to various biomolecular complexes. All the results are generated from a computer with an Intel Xeon E5-1620 CPU, a Nvidia GeForce GTX680 graphic card, and 16GB of memory.

Table 2 Simplification results for biomolecules

	No simplification		Atom simplification (no emphasized chains)			Atom simplification (with emphasized chains)		
PDB ID	N_A	T_G	N_R	T_G	e_G^M	N_R	T_G	e_G^M
2O53	4,912	0.92	3,316 (67.5 %)	0.55	0.019	4,114 (83.8 %)	0.87	0.018
4KYT	7,908	1.35	5,377 (68.0 %)	0.97	0.018	5,501 (69.6 %)	0.99	0.018
4N78	22,843	3.26	14,436 (63.2 %)	2.13	0.023	15,107 (66.1 %)	2.33	0.021
4A7F	33,500	3.88	20,134 (60.1 %)	2.42	0.026	20,465 (62.7 %)	2.47	0.026
2W4U	0.14M	16.21	74,962 (54.4 %)	10.13	0.032	78,853 (57.2 %)	10.98	0.031
2KU2	1.23M	33.21	0.51M (39.3 %)	15.81	0.055	0.64M (49.3 %)	17.48	0.051

Note: N_A – the number of all the atoms; N_R – the number of remaining atoms; and T_G – time for Gaussian density map computation (unit: second). $\epsilon_G = 0.01$ for the atom elimination. The percentage of remaining atoms is shown in the parentheses

Table 2 shows the results of atom simplification for proteins with various sizes. Among them, 2O53 is a protein in human brain, and one of its two identical components is emphasized. 4KYT, 4N78, 4A7F and 2W4U are involved in the heart contractile process. Components with important biological functions are chosen to be emphasized. 2KU2 is one of the largest proteins in the PDB with 1.23M atoms. One of its seven symmetric components is chosen to be emphasized. As shown in Table 2, the percentage of remaining atoms varies from 39.3 % to 69.6 %, and tends to decrease as the protein size increases because most atoms are buried inside the biomolecular surface. The number of remaining atoms also depends on the size of emphasized components. For example in 2O53, when the emphasized components are considered, the percentage of remaining atoms increases by 16.3 %. Contrast to the other proteins, the emphasized components in 4A7F are exposed to the surface, and most of their atoms are kept during simplification. Therefore the number of remaining atoms does not change much when considering these emphasized components.

Figures 3, 10 and 11 show adaptive T-mesh generation results for the five proteins in Table 2, which can be used directly as the control mesh to build rational T-spline surfaces [41]. The principal curvatures are used to guide the parametric line directions, therefore the obtained T-meshes follow the local surface features. On the contrary in Fig. 12, eigenfunctions are used to define the input vector field. Therefore, the obtained anisotropic T-mesh follows the gradient direction of those eigenmodes. The variation of different modes is reflected by both the element orientations and the stretched shapes. We can also set various resolutions for different emphasized components. In Fig. 13, the parametric line spacings are $1 : 2 : 4$ for Chain 20 (orange), Chain 18 (blue) and the rest of the surface. Table 3 shows statistics of all the T-mesh generation results. We can observe that the number of T-junctions and the number of singularities vary for different proteins due to their complex surface features. In addition, the generated T-meshes are in good quality with $J_{\min} \geq 0.37$.

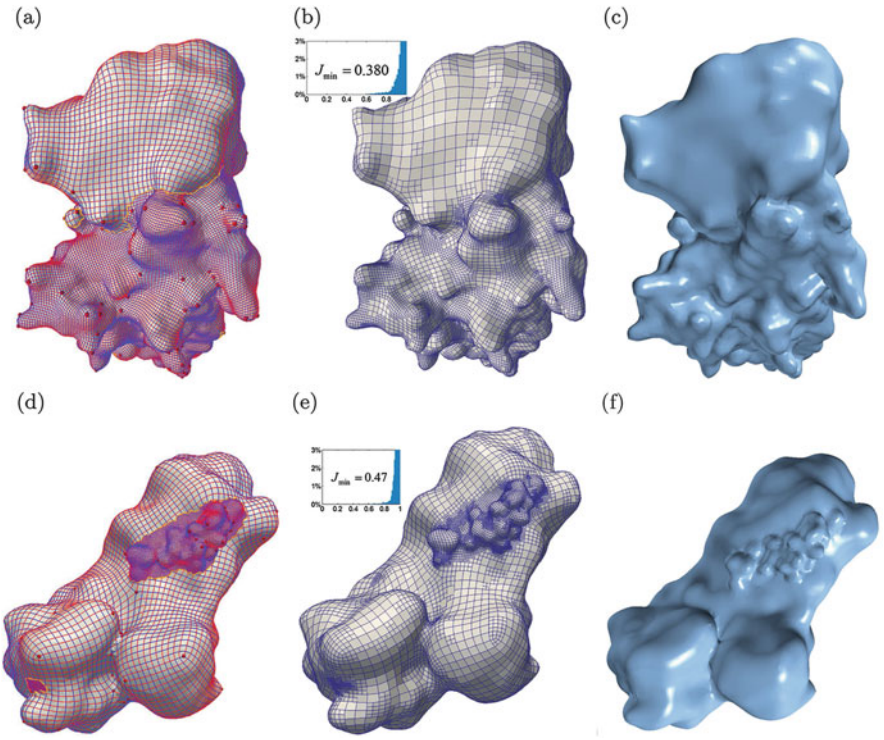


Fig. 10 Multi-resolution surfaces for 2O53 and 4KYT. (**a**–**c**) 2O53; and (**d**–**f**) 4KYT. *Left column*: adaptive parameterization; *Middle column*: T-mesh; *Right column*: T-spline surface

5 Conclusion and Future Work

In this paper, we have introduced a new approach to simplify low-contributing atoms and generate quality T-meshes for multi-resolution biomolecular surfaces. An error-bounded atom elimination algorithm is designed to reduce the atom number and preserve multi-resolution surface feature at the same time. An extended cross field-based parameterization is introduced to generate adaptive and anisotropic T-meshes, which can be used further for T-spline surface construction. In addition, a new gradient flow-based method is introduced for T-mesh quality improvement, preserving the anisotropy in the input T-mesh.

Isogeometric analysis has been applied in a lot of engineering fields, it also has a great potential for applications in computational biology to study biomolecular complexes or proteins. In the future we intend to explore along this direction.

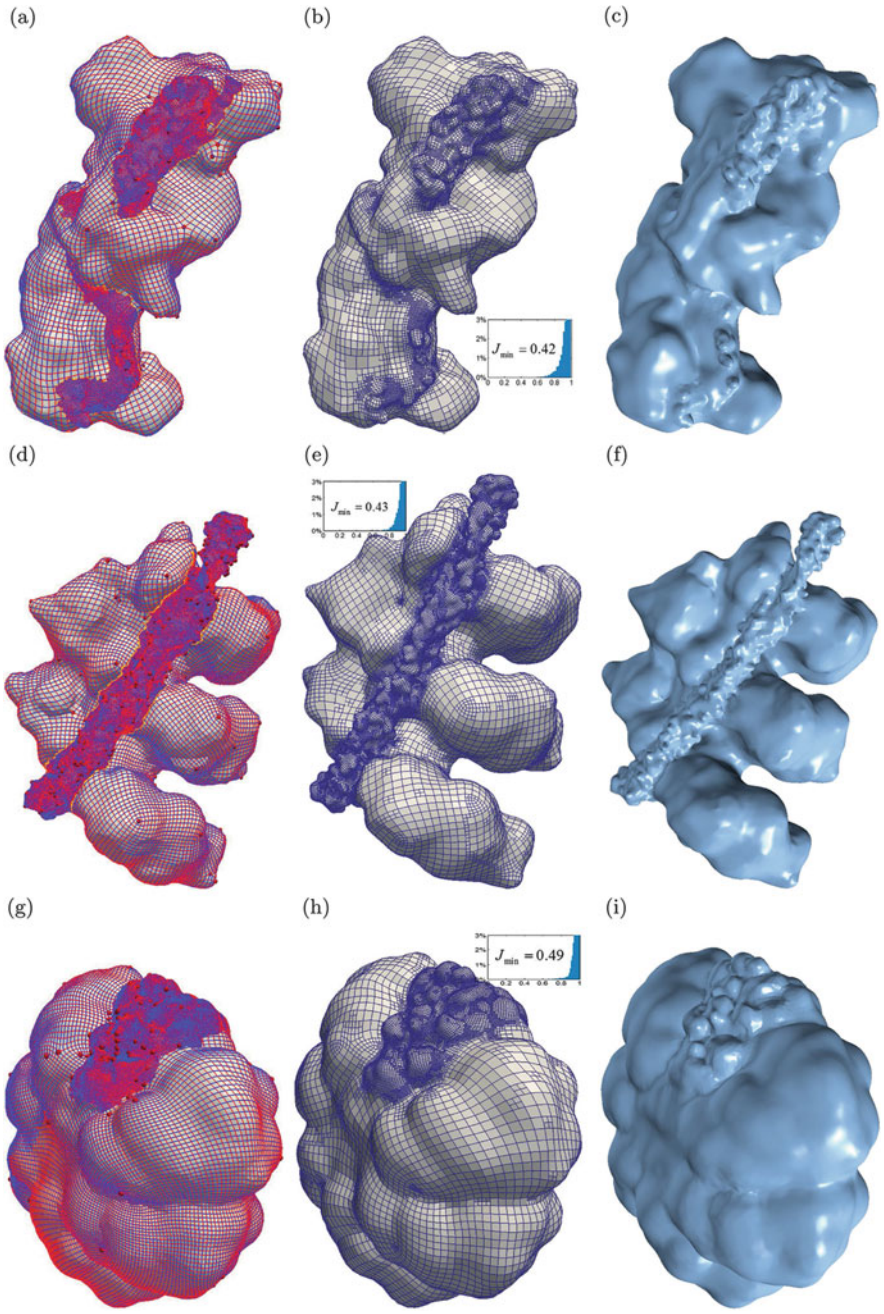


Fig. 11 Multi-resolution surfaces for 4N78, 4A7F and 2KU2. (**a**–**c**) 4KYT; (**d**–**f**) 4N78; and (**g**–**i**) 2KU2. *Left column*: adaptive parameterization; *Middle column*: T-mesh; *Right column*: T-spline surface

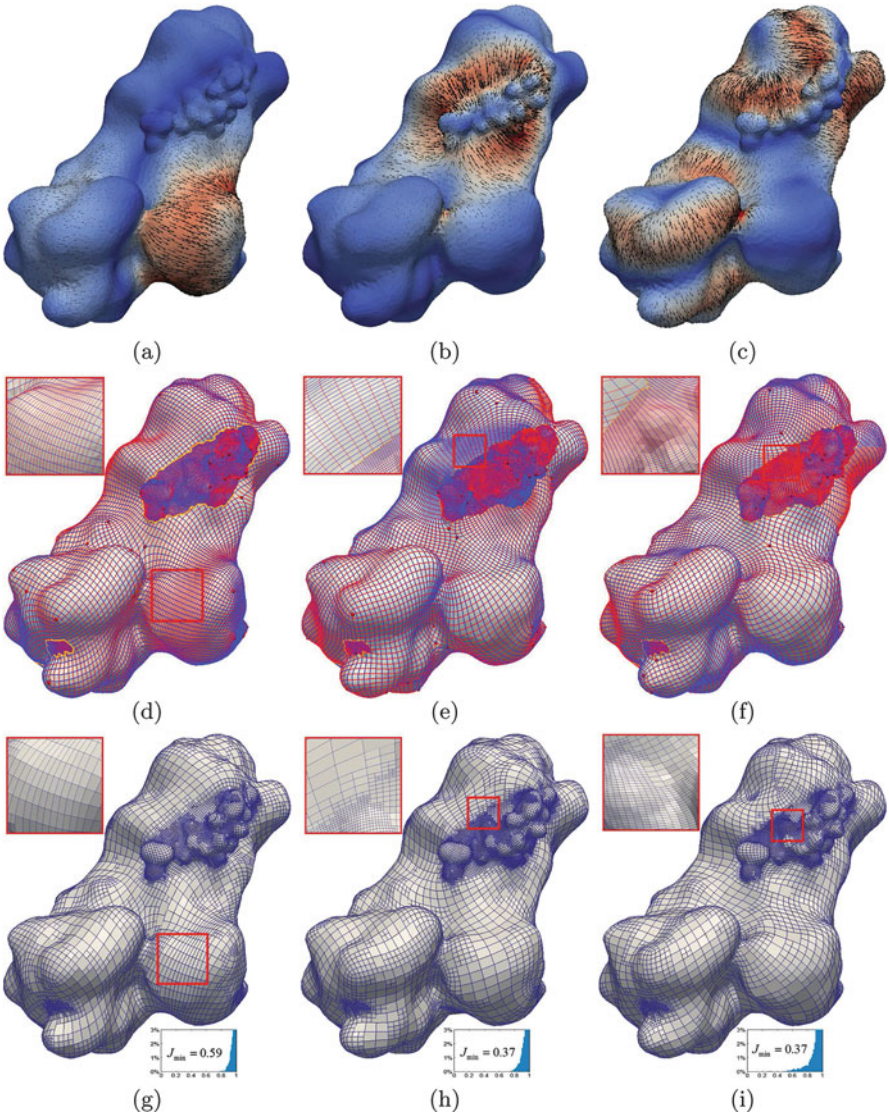


Fig. 12 Adaptive and anisotropic T-mesh construction of 4KYT from three different eigenmodes. (**a–c**) The gradient direction and magnitude of the eigenfunctions; (**d–f**) surface parameterization; and (**g–i**) adaptive and anisotropic T-meshes. *Left column*: results from Mode 2; *Middle column*: results from Mode 3; and *Right column*: results from Mode 6

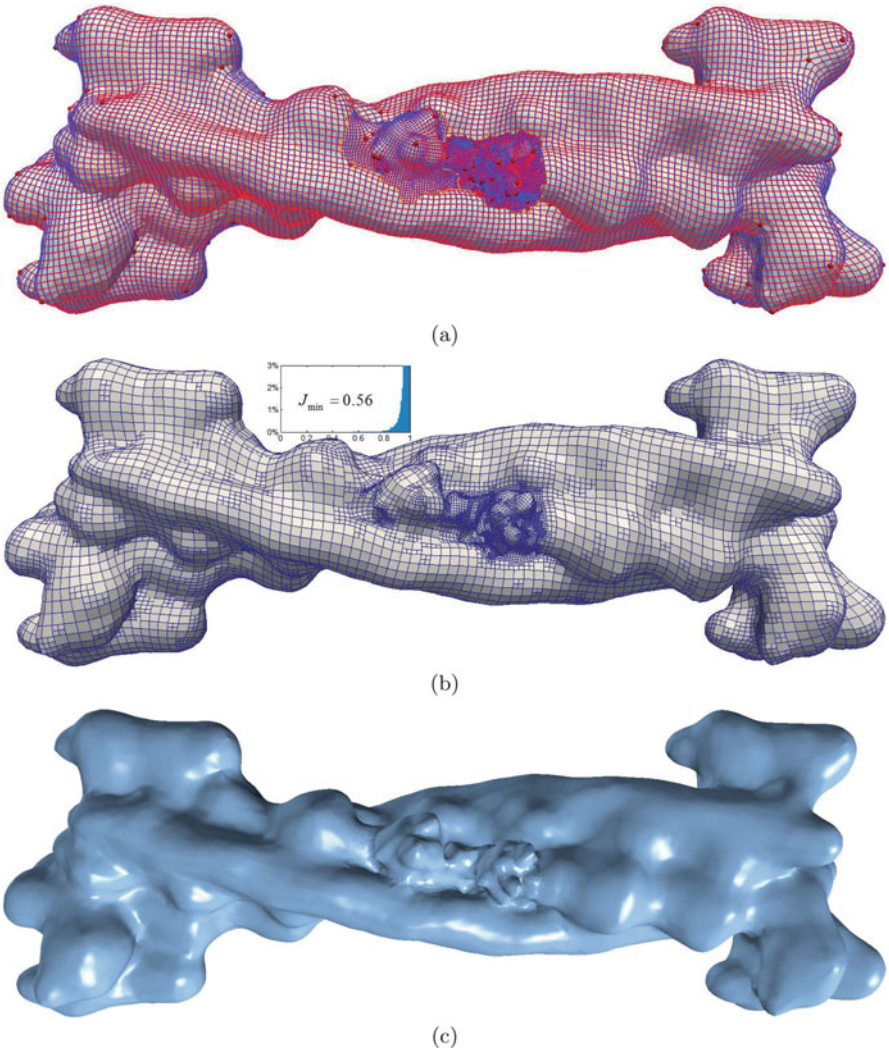


Fig. 13 Biomolecular surfaces for 2W4U emphasizing Chains 18 and 20 with different resolutions. (**a**) Adaptive parameterization; (**b–c**) the corresponding T-mesh and T-spline surface

Table 3 T-mesh generation results for biomolecules

PDB ID	(Vertex No., Element No.)	Number of T-junctions	Number of Singularities	Number of Levels	Number of J_{min}	min_Jacobian (s)	Time
2O53	Fig. 10b	(17,235 15,494)	3,766	156	4	0.47	0.38
4KYT	Fig. 10e	(11,815, 10,576)	3,272	136	4	0.47	13.3
	Fig. 12g	(13,557, 12,251)	2,770	56	4	0.59	13.6
	Fig. 12h	(16,432, 15,045)	2,270	62	4	0.37	13.9
	Fig. 12i	(13,212, 11,942)	2,582	56	4	0.37	13.5
4N78	Fig. 11b	(14,484, 12,843)	4,782	122	4	0.42	14.2
4A7F	Fig. 11e	(42,560 38,423)	3,272	172	4	0.43	15.6
2W4U	Fig. 3c	(30,142, 26,782)	5,214	240	4	0.46	16.3
	Fig. 6c	(28,993, 26,227)	2,956	226	4	0.39	16.6
	Fig. 13b	(21,908 19,298)	5,210	168	4	0.56	15.2
2KU2	Fig. 11h	(25,759, 23,774)	5,433	120	4	0.49	18.3

Acknowledgements T. Liao and Y. Zhang were supported in part by PECASE Award N00014-14-1-0234 and NSF CAREER Award OCI-1149591. G. Xu was supported in part by NSFC under the grants (11101401, 81173663), and NSFC Fund for Creative Research Groups of China (grant No. 11321061).

References

1. L. Albou, B. Schwarz, O. Poch, J. Wurtz, D. Moras, Defining and characterizing protein surface using alpha shapes. Proteins **76**(1), 1–12 (2009)
2. C. Bajaj, S. Chen, G. Xu, Q. Zhang, W. Zhao, Hierarchical molecular interfaces and solvation electrostatics, in *SIAM/ACM Joint Conference on Geometric and Physical Modeling*, San Francisco, 2009, pp. 283–288
3. C.L. Bajaj, V. Pascucci, A. Shamir, R.J. Holt, A.N. Netravali, Dynamic maintenance and visualization of molecular surfaces. Discret. Appl. Math. **127**(1), 23–51 (2003)
4. C.L. Bajaj, V. Siddavanahalli, Fast error-bounded surfaces and derivatives computation for volumetric particle data. Tech. rep., ICES 06-03, 2006
5. D. Bommes, M. Campen, H. Ebke, P. Alliez, L. Kobbelt, Integer-grid maps for reliable quad meshing. ACM Trans. Graphics **32**(4), 98:1–98:12 (2013)
6. D. Bommes, B. Lévy, N. Pietroni, E. Puppo, C. Silva, D. Zorin, Quad-mesh generation and processing: a survey. Comput. Graphics Forum 32(6), 51–76 (2013)
7. D. Bommes, H. Zimmer, L. Kobbelt, Mixed-integer quadrangulation. ACM Trans. Graphics **28**(3), 1–10 (2009)
8. D. Bommes, H. Zimmer, L. Kobbelt, Practical mixed-integer optimization for geometry processing, in *Curves and Surfaces*, ed. by J.-D. Boissonnat, P. Chenin, A. Cohen, C. Gout, T. Lyche, M.-L. Mazure, L. Schumaker (Springer, Berlin/Heidelberg, 2012), pp. 193–206
9. M. Brovka, J.I. López, J.M. Escobar, J.M. Cascón, R. Montenegro, A new method for T-spline parameterization of complex 2D geometries. Eng. Comput. **30**(4), 457–473 (2014)
10. Y. Cheng, C.A. Chang, Z. Yu, Y. Zhang, M. Sun, T.S. Leyh, M.J. Holst, J.A. Mccammon, Diffusional channeling in the sulfate activating complex: combined continuum modeling and coarse-grained Brownian dynamics studies. Biophys. J. **95**(10), 4659–4667 (2008)
11. M.L. Connolly, Analytical molecular surface calculation. J. Appl. Crystallogr. **16**(5), 548–558 (1983)
12. M.L. Connolly, *Molecular Surface: A Review* (Network Science, 1996). www.netsci.org
13. O. Diamanti, A. Vaxman, D. Panozzo, O. Sorkine-Hornung, Designing N-polyvector fields with complex polynomials. Comput. Graphics Forum **33**(5), 1–11 (2014)
14. H. Ebke, D. Bommes, M. Campen, L. Kobbelt, Qex: robust quad mesh extraction. ACM Trans. Graphics **32**(6), 168–177 (2013)
15. H. Ebke, M. Campen, D. Bommes, L. Kobbelt, Level-of-detail quad meshing. ACM Trans. Graphics **33**(6), 184:1–184:11 (2014)
16. Edelsbrunner, E.P. H., Mücke, Three-dimensional alpha shapes. ACM Trans. Graphics **13**(1), 43–72 (1994)
17. M. Frigo, S. Johnson, The design and implementation of FFTW3. Proc. IEEE **93**(2), 216–231 (2005)
18. R. Garimella, M. Shashkov, P. Knupp, Triangular and quadrilateral surface mesh quality optimization using local parametrization. Comput. Methods Appl. Mech. Eng. **193**(9), 913–928 (2004)
19. L. Hu, D. Chen, G. Wei, High-order fractional partial differential equation transform for molecular surface construction. Mol. Based Math. Biol. **1**, 1–25 (2013)
20. B. Jüttler, M. Kapl, D. Nguyen, Q. Pan, M. Pauley, Isogeometric segmentation: the case of contractible solids without non-convex edges. Comput.-Aided Des. **57**, 74–90 (2014)

21. F. Kälberer, M. Nieser, K. Polthier, Quadcover-surface parameterization using branched coverings. Comput. Graphics Forum **26**, 375–384 (2007)
22. P. Kekenes-Huskey, T. Liao, A. Gillette, J. Hake, Y. Zhang, A. Michailova, A. McCulloch, J. McCammon, Molecular and subcellular-scale modeling of nucleotide diffusion in the cardiac myofilament lattice. Biophys. J. **105**(9), 2130–2140 (2013)
23. D. Kovacs, A. Myles, D. Zorin, Anisotropic quadrangulation. Comput. Aided Geom. Des. **28**(8), 449–462 (2011)
24. P. Laug, H. Borouchaki, Molecular surface modeling and meshing. Eng. Comput. **18**, 199–210 (2002)
25. T. Liao, G. Xu, Y. Zhang, Structure-aligned guidance estimation in surface parameterization using eigenfunction-based cross field. Graphical Models **76**, 691–705 (2014)
26. T. Liao, Y. Zhang, P. Kekenes-Huskey, Y. Cheng, A. Michailova, A. McCulloch, M. Holst, J. McCammon, Multi-core CPU or GPU-accelerated multiscale modeling for biomolecular complexes. Mol. Based Math. Biol. **1**, 164–179 (2013)
27. Y. Lipman, Bounded distortion mapping spaces for triangular meshes. ACM Trans. Graphics **31**(4), 108:1–108:13 (2012)
28. A. Myles, N. Pietroni, D. Kovacs, D. Zorin, Feature-aligned T-meshes. ACM Trans. Graphics **29**(4), 98:1–98:12 (2010)
29. A. Myles, N. Pietroni, D. Zorin, Robust field-aligned global parametrization. ACM Trans. Graphics **33**(4), 135 (2014)
30. A. Myles, D. Zorin, Global parametrization by incremental flattening. ACM Trans. Graphics **31**(4), 109:1–109:11 (2012)
31. A. Myles, D. Zorin, Controlled-distortion constrained global parametrization. ACM Trans. Graphics **32**(4), 105:1–105:13 (2013)
32. D. Panozzo, E. Puppo, M. Tarini, O. Sorkine-Hornung, Frame fields: anisotropic and non-orthogonal cross fields. ACM Trans. Graphics **33**(4), 134–144 (2014)
33. N. Pietroni, M. Tarini, O. Sorkine, D. Zorin, Global parametrization of range image sets. ACM Trans. Graphics **30**(6), 149:1–149:10 (2011)
34. N. Ray, W. Li, B. Lévy, A. Sheffer, P. Alliez, Periodic global parameterization. ACM Trans. Graphics **25**(4), 1460–1485 (2006)
35. M.F. Sanner, A.J. Olson, J.C. Spehner, Reduced surface: an efficient way to compute molecular surfaces. Biopolymers **38**(3), 305–320 (1996)
36. Y. Song, Y. Zhang, C.L. Bajaj, N.A. Baker, Continuum diffusion reaction rate calculations of wild-type and mutant mouse acetylcholinesterase: adaptive finite element analysis. Biophys. J. **87**(3), 1558–1566 (2004)
37. Y. Song, Y. Zhang, T. Shen, C.L. Bajaj, J.A. McCammon, N.A. Baker, Finite element solution of the steady-state Smoluchowksi equation for rate constant calculations. Biophys. J. **86**(4), 2017–2029 (2004)
38. D. Thomas, M. Scott, J. Evans, K. Tew, E. Evans, Bézier projection: a unified approach for local projection and quadrature-free refinement and coarsening of NURBS and T-splines with particular application to isogeometric design and analysis. Comput. Methods Appl. Mech. Eng. **284**, 55–105 (2015)
39. A. Vuong, C. Giannelli, B. Jüttler, B. Simeon, A hierarchical approach to adaptive local refinement in isogeometric analysis. Comput. Methods Appl. Mech. Eng. **200**(49), 3554–3567 (2011)
40. W. Wang, Y. Zhang, M. Scott, T. Hughes, Converting an unstructured quadrilateral mesh to a standard T-spline surface. Comput. Mech. **48**(4), 477–498 (2011)
41. W. Wang, Y. Zhang, G. Xu, T. Hughes, Converting an unstructured quadrilateral/hexahedral mesh to a rational T-spline. Comput. Mech. **50**(1), 65–84 (2012)
42. G. Wei, Y. Sun, Y. Zhou, M. Feig, Molecular multiresolution surfaces. arXiv math-ph/0511001 (2005)
43. T. Wilson, J. Sarrate, X. Roca, R. Montenegro, J. Escobar, Untangling and smoothing of quadrilateral and hexahedral meshes, in *Proceedings of the 8th International Conference on Engineering Computational Technology*, Stirlingshire, Paper 36 (2012)

44. Z. Yu, M.J. Holst, Y. Cheng, J.A. McCammon, Feature-preserving adaptive mesh generation for molecular shape modeling and simulation. J. Mol. Graphics Model. **26**(8), 1370–1380 (2008)
45. D. Zhang, J. Suen, Y. Zhang, Y. Song, Z. Radic, P. Taylor, M.J. Holst, C.L. Bajaj, N.A. Baker, J.A. McCammon, Tetrameric mouse acetylcholinesterase: continuum diffusion rate calculations by solving the steady-state Smoluchowski equation using finite element methods. Biophys. J. **88**(3), 1659–1665 (2005)
46. Y. Zhang, C. Bajaj, Adaptive and quality quadrilateral/hexahedral meshing from volumetric data. Comput. Methods Appl. Mech. Eng. **195**(9), 942–960 (2006)
47. Y. Zhang, C.L. Bajaj, B. Sohn, 3D finite element meshing from imaging data. Comput. Methods Appl. Mech. Eng. **194**(48–49), 5083–5106 (2005)
48. Y. Zhang, G. Xu, C.L. Bajaj, Quality meshing of implicit solvation models of biomolecular structures. Comput. Aided Geom. Des. **23**(6), 510–530 (2006)
49. Q. Zheng, S. Yang, G. Wei, Biomolecular surface construction by PDE transform. Int. J. Numer. Methods Biomed. Eng. **28**(3), 291–316 (2012)

Algorithmic Aspects of Isogeometric Shape Optimization

Daniela Fußeder and Bernd Simeon

Abstract Shape optimization is concerned about finding optimal designs under the aspect of some cost criteria often involving the solution of a partial differential equation (PDE) over the afore said unknown shape.

In general, industrial cases involve a geometric model from Computer Aided Design (CAD). However, solving PDEs requires an analysis suitable working model, typically a Finite Element (FEM) triangulation. Hence, some of the geometric properties known from the CAD model may be lost during this format change. Therefore, we employ isogeometric analysis (IGA) instead, which has a tighter connection between geometry, simulation and shape optimization.

In this paper, we present a self-contained treatment of gradient based shape optimization method with isogeometric analysis, focusing on algorithmic and practical aspects like computation of shape gradients in an IGA formulation and updating B-spline and NURBS geometries.

1 Introduction

Isogeometric analysis was invoked in the seminal paper [8] to serve as a bridge between Computer Aided Design (CAD) and the Finite Element Method (FEM). With IGA, so the hope, a seamless work flow from designing CAD models to simulation is possible, saving man power from (manually) adapting FEM meshes to CAD models. Moreover, in IGA there is no information loss as is for such mesh adaptations of classical FEM. Shape optimization was targeted as an application which might benefit from IGA already in that first introduction to IGA. The reasons

D. Fußeder (✉) • B. Simeon
Technische Universität Kaiserslautern, Paul-Ehrlich-Str. Geb. 31, 67633 Kaiserslautern, Germany
e-mail: fusseder@mathematik.uni-kl.de; simeon@mathematik.uni-kl.de

B. Jüttler, B. Simeon (eds.), *Isogeometric Analysis and Applications 2014*,
Lecture Notes in Computational Science and Engineering 107,
DOI 10.1007/978-3-319-23315-4_8

are the same as above, namely that

- there is no loss of information between optimization and analysis model, and
- B-splines and NURBS practically are the toolbox of designing because of their nice approximation properties, so why not use them also for finding optimal designs numerically.

A general formulation of shape optimization problems, see for instance [22], reads

$$\min J(u, \Omega) \quad \text{s.t. } E(u, \Omega) = 0 \text{ and } g(\Omega) \leq 0, \quad h(\Omega) = 0 \tag{1}$$

where J is a real valued objective or cost function depending on a domain Ω and the solution u of a PDE, which is given by the term $E(u, \Omega) = 0$. In g and h we collect additional inequality and equality constraints on the shape, but, in this work, not on the state u.

A gradient-based optimization method is employed to solve the discretized version of the minimization problem (1). Typical gradient-based optimization, as in [16], are for instance a sequential quadratic programming method (SQP), method of moving asymptotes (MMA), or an interior point method. Using such a black box gradient-based optimization routine, e.g. in the form of MATLAB solver `fmincon`, we illustrate the steps towards an optimal shape in Algorithm 1.

Algorithm 1 Basic Black Box Shape Optimization Algorithm

Require: Initial geometry Ω
Require: PDE(Ω) ▷ PDE solver on Ω which yields solution u_h
Require: OBJECTIVE(Ω, u) ▷ evaluate objective function $J(\Omega, u)$ for domains Ω
Require: CONSTRAINTS(Ω) ▷ evaluate constraint functions $g(\Omega)$, $h(\Omega)$ for domains Ω
Require: SHAPE GRAD(Ω, u) ▷ shape gradient computation ∇J, $\nabla C = (\nabla g, \nabla h)^T$
Require: UPDATE(Ω, s) ▷ geometry update routine $\Omega_{new} =$ "$\Omega + s$" with a descent direction s

Black Box Optimization, for instance with `fmincon`

1: **repeat**
2: $u_h \leftarrow$PDE(Ω)
3: $J \leftarrow$ OBJECTIVE(Ω, u_h)
4: $C \leftarrow$CONSTRAINTS(Ω)
5: $(\nabla J, \nabla C) \leftarrow$SHAPE GRAD($\Omega, u_h$)
6: **if** Ω is not optimal, i.e. does not satisfy a stopping criteria **then**
7: compute a descent direction s involving ∇J, ∇g, ∇h
8: $\Omega \leftarrow$ UPDATE(Ω, s)
9: **end if**
10: **until** Ω is optimal

Of course, in this work we use a B-spline or NURBS representation of $\Omega \subset \mathbb{R}^2$ and solve the state equation PDE(Ω) in Line 2 with IGA. Here, $E(u, \Omega) = 0$ describes a linear elasticity or Poisson equation. One focus will be on the practical computation of shape gradients ∇J, ∇g and ∇h in SHAPE GRAD(Ω, u) in Line 5 which are formulated in an isogeometric way using classical methods

from [13, 22]. For that, we take an *optimize first–then discretized view* as opposed to *discretize first–optimize then*, see [7] for the first and [22] for the second approach. However, we delegate finding a descent direction in Line 7 to a black box nonlinear optimization routine, MATLAB `fmincon`, [10] or [26], which vitally depends on the quality of said shape gradients. Although checking the stopping criteria in Line 6 is part of the black-box optimizer we need to address it also for possible pitfalls. Last but not least updating the geometry in UPDATE(Ω) Line 8 falls again to our lot and consequently needs our attention to avoid infeasible representations.

A basic reference for IGA is [8]; B-splines and NURBS are treated in [17]. Isogeometric analysis has been applied to several fields of shape optimization problems already, for instance [14, 15] with application to electromagnetism and [2, 3, 20, 27] with application to solid mechanics and also shells [11]. As one of the basic references in the classical shape optimization field, we refer to [18], and a rigorous mathematical treatment is provided by [22]. Especially the definition of shape gradients is mainly attributed to [13, 18, 22]. Among many others, the introduction to optimization methods [16] treats SQP and interior-point methods. A popular optimizer in structural shape optimization is MMA by [23]. For our model problems, we use the packages [10] which includes for instance SQP and MMA, and [26], an interior-point solver. Also, SQP and interior-point method both are options for `fmincon` in [12].

Algorithm 1 provides the structure for this work: we introduce the core of isogeometric shape optimization – NURBS and B-splines – in Sect. 2 and apply it subsequently to obtain representations of domains in Sect. 2.2, find solutions of PDEs with IGA in Sect. 3, and formulate derivatives w.r.t. geometries in Sect. 4. For that, the space of admissible domains is provided with a vector space structure which allows us to define "summation of domains", i.e. updating shapes in Sect. 5. Finally in Sect. 6, we apply the theory from the previous sections to obtain the required procedures in Algorithm 1 for the abstract shape optimization problem, and realize them for some model examples. We conclude this work with the numerical solutions of these model problems in Sect. 7, a short summary, and an outlook.

2 Introduction to B-Splines, NURBS and Geometries in IGA

Isogeometric analysis starts and ends with B-splines and NURBS as basis functions for geometric modeling, as basis functions for the Galerkin projection in simulation and as design variables for shape optimization.

Therefore, we present the basic definitions subsequently in the first part, Sect. 2.1, but refer to various sources like [4, 21] for B-splines and for instance to the monograph [17] for a more elaborate treatment of NURBS than we can provide here. Equipped with the notation from this introduction we apply it to geometry representation in IGA in the following in Sect. 2.2.

2.1 Introduction to B-Splines and NURBS

From a given knot vector $\Xi = (\xi_1, \ldots, \xi_{n+p+1})$ we form n B-spline basis functions of degree p recursively

$$\text{for } p = 0: \quad N_{i,0}(\hat{x}) = \begin{cases} 1, & \text{if } \xi_i \le \hat{x} < \xi_{i+1} \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

$$\text{for } p > 0: \quad N_{i,p}(\hat{x}) = \frac{\hat{x} - \xi_i}{\xi_{i+p} - \xi_i} N_{i,p-1}(\hat{x}) + \frac{\xi_{i+p+1} - \hat{x}}{\xi_{i+p+1} - \xi_{i+1}} N_{i,p-1}(\hat{x}) \; . \tag{3}$$

Here, we always assume an open knot vector where the first and last knots are repeated $p+1$ times, i.e. $\xi_i = a$ for $i = 1, \ldots, p+1$ and $\xi_i = b$ for $i = n, \ldots, p+1$. This way, $N_{1,p}$ and $N_{n,p}$ are interpolative. We obtain non-uniform rational B-spline basis functions, in short NURBS, by weighting the B-spline basis

$$R_{i,p,W} = \frac{\omega_i N_{i,p}(\hat{x})}{\sum_{j=1}^{n} \omega_j N_{j,p}(\hat{x})} \tag{4}$$

with positive weights $\omega_i > 0$ collected in a weight vector $W = (\omega_1, \ldots, \omega_n)$.

A set of B-spline basis functions spans a spline space $\mathcal{S}(\Xi, p)$ of degree p which is determined uniquely by the underlying knot vector,

$$\mathcal{S}(\Xi, p) = \text{span}\{N_i(\hat{x}) := N_{i,p}(\hat{x}),\ i = 1, \ldots, n,\ \hat{x} \in [0,1]\}. \tag{5}$$

We reach higher space dimensions by forming tensor products: given two knot vectors Ξ_1, Ξ_2 and corresponding sets of basis functions $N_{i_1,p_1}(\hat{x}_1)$ and $N_{i_2,p_2}(\hat{x}_2)$ a tensor product space

$$\mathcal{S}(\Xi_1, p_1) \otimes \mathcal{S}(\Xi_2, p_2) = \text{span}\{N_i(\hat{x}) : i = 1, \ldots, n_1 n_2,\ \hat{x} \in [0,1]^2\} \tag{6}$$

is obtained with $n = n_1 n_2$ basis functions $N_i = N_{i_1,p_1}(\hat{x}_1) N_{i_2,p_2}(\hat{x}_2)$ for $i_1 = 1, \ldots, n_1$, $i_2 = 1, \ldots, n_2$.

Taking a NURBS basis by exchanging N_{i_1,p_1} with R_{i_1,p_1,W_1} and N_{i_2,p_2} with R_{i_2,p_2,W_2} leads to a NURBS space

$$\mathcal{S}(\Xi, p, W) = \text{span}\{N_i(\hat{x}) := R_{i,p,W}(\hat{x}),\ i = 1, \ldots, n,\ \hat{x} \in [0,1]\} \tag{7}$$

and analogously to (6) we have $\mathcal{S}(\Xi_1, p_1, W_1) \otimes \mathcal{S}(\Xi_2, p_2, W_2)$ with basis functions $N_i := R_{i_1,p_1,W_1} R_{i_2,p_2,W_2}$.

Of course, the tensor product can be extended to arbitrary space dimensions, however, for our examples two dimensions are sufficient. Since B-splines are a special case of NURBS with all weights equal to one, we from now on use the same symbols N_i for uni- and bivariate NURBS and B-splines, also dropping the product

notation and the indices for fixed degrees p and weight vectors W. When we speak of NURBS, we mean both NURBS and B-splines, except where we distinguish them explicitly.

With such B-splines and NURBS we are equipped for designing geometric forms, performing simulation and shape optimization in the following sections.

2.2 Geometries in Isogeometric Analysis

One key ingredient of isogeometric analysis is the representation G of geometries by linear combinations of NURBS with fixed weights. Moreover, in isogeometric shape optimization we search for designs also in such NURBS spaces, i.e. the linear coefficients of G are the optimization variables. In addition, one could also look for shapes where the weights are variable as well. However, for the definition of shape gradients we need a linear space which we can provide for NURBS with variable weights in homogeneous coordinates only.

In the following, we first introduce geometries from the easier case of B-spline and NURBS spaces with fixed weights, and then geometries built from NURBS with variable weights.

2.2.1 B-Splines and NURBS with Fixed Weights

A NURBS geometry in $\mathbb{R}^d$ is a linear combination of NURBS basis functions

$$G(\hat{x}) = \sum_{i=1}^{n} X_i N_i(\hat{x}) \in \mathcal{S}^d \text{ with control points } X_i \in \mathbb{R}^d \,, \tag{8}$$

typically $d = 1, 2, 3$. In our examples, G is defined over a parameter space $\hat{\Omega} = [0, 1]^d$.

For NURBS *curves*, $d = 1$, we have univariate NURBS $N_i \in \mathcal{S} = S(\Xi, p, W)$ and $\hat{x} \in [0, 1]$. For *surfaces*, $d = 2$ and the parameter domain is $[0, 1]^2$; we use bivariate NURBS $N_i \in \mathcal{S} = \mathcal{S}(\Xi_1, p_1, W_1) \otimes \mathcal{S}(\Xi_2, p_2, W_2)$.

In Algorithm 1 we assume that the geometries Ω are surfaces given by a NURBS representation as in (8), $\Omega = G(\hat{\Omega})$.

2.2.2 NURBS Space with Variable Weights

For the shape gradient derivation in the variational setting we have to restore linearity of the NURBS space with variable weights. For that, we resort to homogeneous coordinates and a perspective map, see [17]. A homogeneous coordinate vector $\tilde{X}$ in

$\mathbb{R}^{d+1}$ with $\tilde{X} := (X^w, X_{d+1})$ and $X^w \in \mathbb{R}^d$ is projected to $\mathbb{R}^d$ by

$$H(\tilde{X}) = \begin{cases} X^w/X_{d+1}, & X_{d+1} \neq 0 \\ X^w/|X^w|, & \text{else.} \end{cases}$$

A $d+1$ dimensional B-spline geometry $(G^w, w) \in \mathcal{S}^{d+1}$ with $G^w \in \mathcal{S}^d$ and $0 < w \in \mathcal{S}$ is mapped to a rational representation in $\mathbb{R}^d$ by

$$H\colon (G^w, w) \mapsto G := G^w/w\,. \tag{9}$$

Finally, $\mathcal{S}^d \oplus \mathcal{S}$ has a basis $\{N_i e_k \colon N_i \in \mathcal{S},\ i = 1, \ldots, n,\ k = 1, \ldots, d+1\}$ and is a linear space which is isomorphic to $\mathbb{R}^{(n+1)d}$.

The global geometry representations with fixed weights are a key ingredient of isogeometric analysis in Sect. 3. The case of variable weights is only important for shape optimization in Sect. 4.

3 Simulation in Isogeometric Analysis

Solving PDEs with isogeometric analysis works with the variational form and Galerkin projection the same way as in FEM. However, the PDE is transformed with a geometry function G such that it is posed over the parameter domain $\hat{\Omega} = [0,1]^d$.

Given a PDE in its strong form

$$Lu = f \tag{10}$$

with a linear elliptic partial differential operator L, its relation to the weak form $a(u,v) = l(v)$ is given by the dual pairing $a(u,v) = \langle Lu, v\rangle$ and by $l(v) = (f,v)$. As in FEM, we use the variational form of (10) to find its numerical solution. In general, we say that u solves the state equation $E(u,\Omega) = 0$ if it solves the variational form of (10) over the domain $\Omega \subset \mathbb{R}^d$

$$u \in H_0^1(\Omega)^m\colon \quad a(u,v) = l(v) \quad \forall v \in H_0^1(\Omega)^m\,. \tag{11}$$

We now first introduce the Galerkin projection according to the isogeometric paradigm and then state the PDE solver for Algorithm 1.

3.1 Galerkin Projection in IGA

The approximation space for the state equations in the form of (11) is in isogeometric analysis

$$\mathcal{V}_h := \{v_h \in H_0^1(\Omega)^m : v_h \circ G \in \mathcal{M}^m\} \tag{12}$$

where m is the dimension of $u: \Omega \to \mathbb{R}^m$ and $\mathcal{M}$ is a NURBS space of space dimension d. Typically in IGA, it is the same NURBS space as for the geometric representation in (8), i.e. $\mathcal{M} = \mathcal{S}$, but that is not mandatory.

The important point also for shape optimization later is, that we move the dependency on the domain from the Sobolev space to the bilinear and linear form:

$$a_G(\hat{v}, \hat{w}) := \int_{\hat{\Omega}} L_G \hat{v} \hat{w} |\det J_G| \, d\hat{\Omega} \,, \quad L_G := L \circ G \tag{13}$$

$$l_G(\hat{v}) := \int_{\hat{\Omega}} f \circ G \cdot \hat{v} |\det J_G| \, d\hat{\Omega} \,. \tag{14}$$

The variational form which we solve in IGA then is

$$\hat{u}_h \in \mathcal{M}_0^m : a_G(\hat{u}_h, \hat{v}_h) = l_G(\hat{v}_h) \quad \forall \hat{v}_h \in \mathcal{M}_0^m \,, \tag{15}$$

which is equivalent to (11) for $\hat{u}_h = u_h \circ G$. The NURBS space $\mathcal{M}_0^m := \mathcal{M}^m \cap H_0^1(\hat{\Omega})^m$ contains only those NURBS $\hat{v}_h \in \mathcal{M}^m$ which respect (possible) Dirichlet boundary conditions. Note, that through the transformed differential operator L_G we also introduce J_G^{-T} into the forms a_G and l_G besides $\det J_G$.

All in all, the solution of (15) has the form

$$\hat{u}_h = \sum_{i=1}^{n} Q_i M_i(\hat{x}) \text{ with solution coefficients } Q_i \in \mathbb{R}^m \text{ and } M_i \in \mathcal{M} \,. \tag{16}$$

The Poisson equation is an example for such a transformed notation. In that case, the dimension of the solution is $m = 1$ and

$$a_G(\hat{v}, \hat{w}) = \int_{\hat{\Omega}} \nabla \hat{v}^T J_G^{-1} J_G^{-T} \nabla \hat{w} |\det J_G| \mathrm{d}\hat{x} \,.$$

A vector field example with $m = 2$ is linear elasticity for which a weak IGA form is found in (52) in Sect. 6 about model problems.

3.2 PDE Solver

Going back to the Basic Shape Optimization Algorithm 1, line 2 corresponds to assembling and solving equation (15). This leads to the linear system

$$KQ = F, \quad K_{iljk} := a_G(M_i e_l, M_j e_k), \quad F_{il} := l_G(M_i e_l) \tag{17}$$

for the model problems with stiffness matrix K, right hand side F and coefficients of the solution $\hat{u} = \sum_{i=1}^{n} \sum_{l=1}^{m} Q_{il} M_i e_l(\hat{x})$. We silently assume that the coefficients $Q_{il} \in \mathbb{R}$ can be ordered lexicographical in a solution vector Q, for instance like

$$Q = \begin{pmatrix} Q_{11} & Q_{12} & \dots & Q_{n1} & \dots & Q_{nm} \end{pmatrix}^T$$

and in the same breath we write

$$Q_i = \begin{pmatrix} Q_{i1} & \dots & Q_{im} \end{pmatrix}^T .$$

Hence, for any admissible domain Ω given by a NURBS parameterization $G = \sum_{i=1}^{n_N} X_i N_i \in \mathcal{S}^d$ and a test function space of NURBS $\mathcal{M}^m$ the procedure PDE(Ω) is given by Algorithm 2.

Algorithm 2 PDE(Ω)

1: **for all** basis functions $M_i e_l$ **do**
2: **for all** basis functions $M_j e_k$ **do**
3: $K_{iljk} \leftarrow a_G(M_i e_l, M_j e_k)$
4: **end for**
5: $F_{il} \leftarrow l_G(M_i e_l)$
6: **end for**
7: Apply boundary conditions of state to K and F
8: Solve $KQ = F$
9: $u_h \leftarrow \sum_{i=1}^{m} Q_i M_i$.

4 Shape Gradients in Isogeometric Analysis

Gradient based optimization routines use gradients of the objective function and of the constraints to find a descent direction towards a (local) minimum and also as first order optimality condition. In case of shape optimization, the objective and constraints depend on a domain from the space of admissible domains and mostly also on a state u defined on that domain. Hence, a gradient would be w.r.t. a shape.

In classical shape optimization such gradients are obtained by defining perturbations of identity

$$F_t(x) = id(x) + th(x) \tag{18}$$

where a perturbed domain results from $\Omega_t := \{x_t = F_t(x) : x \in \Omega\}$. The vector fields h must be chosen in a way that respects certain regularity conditions, see for instance [22]. Then, for a function $\phi(\Omega)$ the derivative in direction of h is understood as

$$\phi'(\Omega; h) := \lim_{t \to 0+} \frac{1}{t}\big(\phi(\Omega_t) - \phi(\Omega)\big) = \mathrm{d}_t \phi(\Omega_t)|_{t=0} \,. \tag{19}$$

Next, we derive the corresponding formulas in IGA for NURBS with fixed weights in Sect. 4.1 and then extend them to variable weights in 4.2.

4.1 Isogeometric Shape Sensitivities

In isogeometric analysis we have for $x \in \Omega$ that $x = G(\hat{x})$ for some $\hat{x} \in \hat{\Omega}$. Applying this transformation to the perturbation of identity (18) yields

$$G_t(\hat{x}) := F_t \circ G(\hat{x}) = G(\hat{x}) + th \circ G(\hat{x}) \,. \tag{20}$$

Choosing h true to the isogeometric paradigm $\theta := h \circ G \in \mathcal{S}^d$, as shown in Fig. 1, this results in the isogeometric perturbation of identity

$$G_t(\hat{x}) = G(\hat{x}) + t\theta(\hat{x}) \text{ and } \Omega_t = G_t(\hat{\Omega}) \,. \tag{21}$$

Therefore it makes sense to notate $\phi(G)$ and the derivative in direction of θ as

$$\phi'(G; \theta) = \lim_{t \to 0+} \frac{1}{t}\big(\phi(G_t) - \phi(G)\big) = \mathrm{d}_t \phi(G_t)|_{t=0} \,. \tag{22}$$

The perturbations in isogeometric analysis fulfill the regularity conditions such that the directional derivatives constitute the shape gradient. As we have shown in [6]

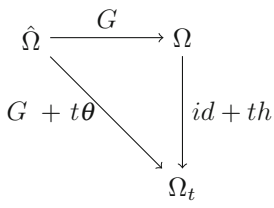


Fig. 1 Perturbation of identity with IGA

the directional derivatives of integrals in isogeometric analysis of the form

$$\phi(G) := \int_{\Omega} j_1(x)\mathrm{d}x + \int_{\partial\Omega} j_2(s)\mathrm{d}s \tag{23}$$

have the expression

$$\phi'(G;\theta) = \int_{\hat{\Omega}} \left(\nabla j_1 \circ G \cdot \theta + j_1 \circ G \,\mathrm{trace}(J_G^{-1} D\theta)\right) |\det J_G| \mathrm{d}\hat{x} \tag{24}$$

$$+ \int_{\partial\hat{\Omega}} \Big(\nabla j_2 \circ G \cdot \theta |J_G^{-T}\hat{n}| + j_2 \circ G \,\mathrm{trace}(J_G^{-1} D\theta) \tag{25}$$

$$- j_2 \circ G \frac{\hat{n}^T J_G^{-1} J_G^{-T} D\theta^T J_G^{-T} \hat{n}}{|J_G^{-T}\hat{n}|}\Big) |\det J_G| \mathrm{d}\hat{s}\,. \tag{26}$$

The isogeometric shape sensitivities in Eqs. (24), (25), and (26) can be obtained by using existing classical results and the transformation rule, or can be developed by solving at $t = 0$

$$d_t\phi(G_t) = d_t \int_{\hat{\Omega}} j_1 \circ G_t |\det J_{G_t}| \mathrm{d}\hat{x} + d_t \int_{\partial\hat{\Omega}} j_2 \circ G_t \|d_s G_t\|_2 \mathrm{d}\hat{s}\,. \tag{27}$$

Also in [6], we proved for linear elliptic PDEs of second order that the directional derivative from solving (24), (25), and (26) for $\theta = N_i e_k$ for a B-spline $N_i e_k \in \mathcal{S}^d$ is equal to the standard derivative $d_{X_{ik}}\phi$, i.e. taking the derivative of ϕ w.r.t. a control point component k of X_i

$$d_{X_{ik}}\phi(G) = d_{X_{ik}} \int_{\hat{\Omega}} j_1 \circ G |\det J_G| \mathrm{d}\hat{x} + d_{X_{ik}} \int_{\partial[0,1]^2} j_2 \circ G |\det J_G| \mathrm{d}\hat{s}\,. \tag{28}$$

Reversing the order of differentiation and integration shows that in (28) we, too, need $\mathrm{d}_{X_{ik}} G$ and $\mathrm{d}_{X_{ik}} |\det J_G|$. An explicit, efficient term has been derived in [20] and [27].

The two notions of derivatives stem from two ways of approaching optimal control problems: in the first case, the optimality conditions are derived for the continuous problem in an infinite dimensional function space setting, then all occurring function spaces and operators are approximated by finite dimensional ones, leading to the directional derivatives in (23). In the second case, the continuous problem is discretized by substituting all infinite dimensional function spaces with finite approximations, and then the optimality criteria are derived. It so happens, that the spline spaces are isomorphic to real vector spaces and therefore the directional derivatives are the standard derivatives in isogeometric shape optimization.

The above derivative (24) is a domain representation. In literature, one can also find a boundary representation by using the so called Hadamard structure theorem

[5, Theorem 3.6]. Basically, it says that a shape derivative in direction h of the form

$$\int_\Omega \operatorname{div}(j_1(x)h(x))\mathrm{d}x = \int_{\partial\Omega} j_1(x)h(x)\mathrm{d}s \tag{29}$$

with Stokes' divergence theorem. In this work, we only consider domain representations, but all steps can be applied to Hadamard sensitivities as well. In [2], such Hadamard representations have been used together with IGA in linear elasticity compliance minimization. In both cases, the directions θ of the derivatives are chosen such that θ is a basis function living on the *moving* boundary. This is, of course, natural for a boundary representation, but it also reduces the degrees of freedom for the domain representation significantly. By doing so, we speak of such basis functions $\theta = N_i e_k \neq 0$ on the moving boundary as well as their corresponding control points X_{ik} as *design variables* and denote their number by n_{des}.

4.2 Weight Optimization

For the following argument, we explicitly denote the NURBS representation of G with NURBS from Eq. (7) as

$$G(\hat{x}) = \sum_{i=1}^{n} X_i \frac{\omega_i N_i(\hat{x})}{\sum_{j=1}^{n} \omega_j N_j(\hat{x})} \,. \tag{30}$$

In the light of taking the derivative w.r.t. a control point in Eq. (28), it is not far fetched to get the derivative w.r.t. a weight ω_i for rational geometries G as in (30), which implies the need for $\mathrm{d}_{\omega_i} G$ and $\mathrm{d}_{\omega_i} |\det J_G|$, an explicit derivation can again be found in [20]. In [6] the authors have shown how to embed weight optimization in the function space setting using homogeneous coordinates. We emphasize here, that in the case of variable weights, the formulas for the two approaches of directional derivatives are not the same. One could say, that in the function space setting a term is missing. This is compensated by a different update rule, though. See also Sect. 5 for that.

We use this in the following sensitivity formula. For a shape functional ϕ like (23), we obtain the isogeometric shape sensitivities in direction $\tilde{\theta} \in \mathcal{S}^{d+1}$ for $G = H \circ \tilde{G}$ and $\theta = H \circ \tilde{\theta}$ as

$$\begin{aligned} &\phi'(G;\theta) \\ &\quad = \int_{\hat{\Omega}} \left(\nabla j_1 \circ G \cdot DH \circ \tilde{G}\tilde{\theta} + j_1 \circ G \operatorname{trace}(J_G^{-1}\dot{D})\right) |\det J_G| \mathrm{d}\hat{x} \end{aligned} \tag{31}$$

$$+\int_{\hat{\Gamma}}\Big(\nabla j_2 \circ G \cdot DH \circ \tilde{G}\tilde{\theta}|J_G^{-T}\hat{n}| + j_2 \circ G|J_G^{-T}\hat{n}|\,\mathrm{trace}(J_G^{-1}\dot{D}) + \tag{32}$$

$$-j_2 \circ G\frac{\hat{n}^T J_G^{-1} J_G^{-T}\dot{D}^T J_G^{-T}\hat{n}}{|J_G^{-T}\hat{n}|}\Big)|\det J_G|\mathrm{d}\hat{x} \tag{33}$$

with $\dot{D} := \mathrm{d}_t D(H \circ \tilde{G}_t)|_{t=0} = D^2H \circ \tilde{G} \odot \tilde{\theta} D\tilde{G} + DH \circ \tilde{G} D\tilde{\theta}$, where

$$D^2H \circ \tilde{G} \odot \tilde{\theta} = \frac{1}{w^2}\left((0_d, -e_1)\tilde{\theta}, \ldots, (0_d, -e_d)\tilde{\theta}, (-I_d, 2G)\tilde{\theta}\right) \tag{34}$$

$$= \begin{cases} \frac{1}{w^2}\left(0_d,\ -\theta^w\right) & , \text{if } \tilde{\theta}_{d+1} = 0 \\ \frac{-\tilde{\theta}_{d+1}}{w^2}\left(I_d,\ \frac{1}{\tilde{\theta}_{d+1}}\theta^w - 2G\right) & , \text{if } \tilde{\theta}_{d+1} \neq 0\ . \end{cases} \tag{35}$$

$$DH \circ \tilde{G} = \frac{1}{w}\left(I_d,\ -G\right). \tag{36}$$

In this formulation $\tilde{\theta} = N_i e_k$ for e_k a unit vector of $\mathbb{R}^{d+1}$ and $k = 1, \ldots, d$ corresponds to varying control point components X_{ik} and $k = d+1$ varies weights ω_i. However note, that $\phi'(G;\theta) \neq \mathrm{d}_X\phi(G)$ anymore, where $X = X_{ik}$ or ω_i.

4.3 Regularization

Though both interpretations of directional derivatives, (27) and (28), lead to the same system for the *directional derivative*, in the functional space setting the gradient is a member of a Hilbert space H and thus depending on the underlying scalar product $(\cdot,\cdot)$. For a Fréchet-differentiable real valued function ϕ, the gradient denoted by $\nabla\phi \in H$ is defined as the solution to

$$(\nabla\phi, \theta) = \phi'(G;\theta) \quad \forall\theta \in H\ . \tag{37}$$

This leaves some freedom in choosing the scalar product $(\cdot,\cdot)$ and hence the regularity of $\nabla\phi$. According to [19] such a regularization has the effect of preconditioning the optimization process.

As yet we have not used regularization in any of the model problems, however, in [2] an example can be found that combines regularization with extending the gradient to the inside, or what we call mesh movement and treat later in Sect. 5.1.

4.4 With State Equation

Typically, the cost functional J depends on the domain Ω but also on the solution u of a PDE over the domain, $u = u(\Omega)$. So formally, a shape derivative of J employs the chain rule and yields

$$\mathrm{d}_\Omega J = \partial_\Omega J + \partial_u J \mathrm{d}_\Omega u \ . \tag{38}$$

This sloppy notation can be justified by both approaches, *discretize first–optimize then* and vice versa, and realized with the adjoint state. All in all, we have the following necessary optimality system for an optimal shape $\Omega = G(\hat{\Omega})$ in IGA.

1. State equation (PDE): $u \in \mathcal{V}_h$: $u = \sum Q_i M_i$:

$$a_G(u, v) = l_G(v) \quad \forall v \in \mathcal{V}_h \qquad \Leftrightarrow \qquad KQ = F \tag{39}$$

2. Adjoint state (chain rule): $z \in \mathcal{V}_h$: $z = \sum P_i M_i$:

$$a_G(v, z) = \mathrm{d}_u J(\Omega, v) \quad \forall v \in \mathcal{V}_h \qquad \Leftrightarrow \qquad K^T P = \left(\mathrm{d}_u J(\Omega, M_k)\right)_k \tag{40}$$

3. Stationary point (shape gradient): For all design variables $N_k e_l$ and $X_{k,l}$ respectively

$$\begin{aligned} J'(\Omega, \hat{u}; N_k e_l) + l'_G(\hat{z}; N_k e_l) - a'_G(\hat{u}, \hat{z}; N_k e_l) = 0 \qquad \Leftrightarrow \\ \partial_{X_{k,l}} J + p^T(\partial_{X_{k,l}} F - \partial_{X_{k,l}} KQ) = 0 \end{aligned} \tag{41}$$

with $l'_G(\hat{v}; \theta) = \mathrm{d}_t l_{G+t\theta}(\hat{v})|_{t=0}$ and $a'_G(\hat{y}, \hat{v}; \theta) = \mathrm{d}_t a_{G+t\theta}(\hat{y}, \hat{v})|_{t=0}$.

An example of such shape derivatives in IGA notation can be found in Sect. 6.

4.5 Shape Gradient Computation

In this section we summarize in Algorithm 3 the previous points in one procedure for Algorithm 1. Moreover, we make a practical consideration on the relation between the (possible) two discretizations, simulation and optimization mesh. When using two meshes, we have shape sensitivities like $a'_G(u, z; N_k e_l)$ which involve integrals over basis functions $N_k e_l$ from geometry and $M_i e_j$ from simulation. Therefore, the domain of integration will be the mutual support $\mathrm{supp}(N_k) \cap \mathrm{supp}(M_j)$. For an easy implementation, it is favorable when the meshes match, i.e. if one mesh is the h-refinement of the other. We then can assemble over the finer mesh and the mutual support is found straight forward.

Algorithm 3 SHAPE GRAD(Ω, u)

Shape gradient of objective function ▷ **compute terms in (41) for objective function**

for all basis functions $M_i e_l$ **do** ▷ assemble right hand side of adjoint equation
  $B_{i,l} \leftarrow \mathrm{d}_u J(\Omega, M_i e_l)$
end for
Solve $K^T P = B$: adjoint state $z = \sum_{i=1}^{m} P_i M_i$ ▷ solve adjoint
for all design variables $N_k e_l$ **do** ▷ directional derivatives of objective function
  $\nabla J_{k,l} \leftarrow J'(\Omega, u; N_k e_l) + l'_G(z; N_k e_l) - a'_G(u, z; N_k e_l)$
end for

Shape gradient of constraint functions

for all design variables $N_k e_l$ **do** ▷ directional derivatives of constraints
  $\nabla C_{k,l} \leftarrow (g'(G; N_k e_l), h'(G; N_k e_l))$ derived from Eqs. (24), (25), and (26)
end for

Regularization ▷ **if needed**

for all design variables $N_k e_l$ **do** ▷ assemble system matrix
  for all design variables $N_r e_s$ **do**
    $R_{klrs} \leftarrow (N_k e_l, N_r e_s)$ with scalar product $(\cdot, \cdot)$ of Hilbert space H of wanted regularity
  end for
end for
Solve $Ry = \nabla J$, $\nabla J \leftarrow y$
Solve $Ry = \nabla C$, $\nabla C \leftarrow y$

5 Geometry Update

In Algorithm 1 we need a means how to update shapes. This is also given by the perturbation of identity (18) $\Omega_t = F_t(\Omega) = (id + th)(\Omega)$ or in IGA by (21)

$$\Omega_t = G_t(\hat{\Omega}) = G(\hat{\Omega}) + t\theta(\hat{\Omega}) \ . \tag{42}$$

Since in IGA θ is from the same NURBS space as the geometry function G, the update rule in IGA consists of adding control points.

A special case is updating geometries from NURBS spaces with variable weights. There we have

$$\Omega_t = H(\tilde{G} + t\tilde{\theta})(\hat{\Omega}) \tag{43}$$

where $\tilde{G}$ and $\tilde{\theta}$ are NURBS geometries in homogeneous coordinates.

Before we cast the update rule into a procedure, we have to consider another issue. Often, large steps result in infeasible parameterizations. However, it can partly be avoided with the right mesh movement algorithm.

5.1 Mesh Movement

In case of large deformation at the boundary of a geometry due to updates by shape gradients it frequently leads to infeasible parameterizations, for instance, when the control polygon overlaps with inner control lines, as in Fig. 2. Such tangled meshes are no phenomenon of IGA nor of shape optimization alone. Several techniques have been successfully applied in IGA:

- minimizing the Winslow functional in electromagnetic shape optimization, which aims for a positive determinant of the Jacobian of G [15],
- solving a linear elasticity problem with the displacement of control points on the boundary as Neumann boundary condition [2],
- Analysis-aware IGA meshes [28], which also optimize the magnitude of the determinant of the Jacobian of G.

We write here very generally that a mesh movement function $\Phi\colon \mathbb{R}^{n_{des}} \to \mathbb{R}^{nd}$ propagates a descent direction s to all control points, where s results from the sensitivity analysis and has the size n_{des} corresponding to the number ob design variables as in Sect. 4.1 The focus is here on an algorithmic framework for shape optimization, therefore we assume that Φ is given by a mesh movement method, for instance from above or the one we used in [6]. In the end, the update in IGA is carried out by changing the control points, with Φ respecting fixed boundaries and the movement on the optimization boundary, i.e.

$$\begin{cases} y_\ell = s_\ell, & \text{if } N_{i_\ell} e_{k_\ell} \text{ is design variable,} \\ y_\ell = 0, & \text{if } N_{i_\ell} e_{k_\ell} \text{ is on fixed boundary .} \end{cases} \tag{44}$$

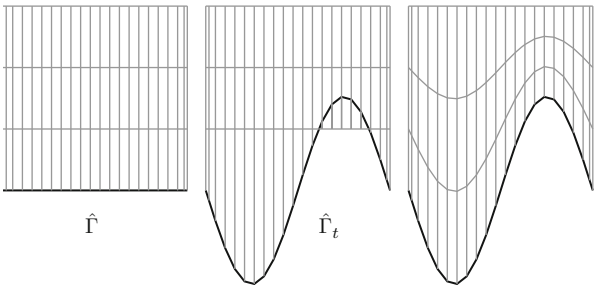


Fig. 2 Moving a boundary which results in non-convex domains can cause irregular parameterizations by overlapping control polygons, hence a suitable method to also move inner control points must be applied

5.2 Computation of New Geometry

Here, we give the procedure for updating a geometry in Algorithm 1. The descent direction s depends on what the optimization routine returns on the basis of ∇J and ∇C but it is up to the user how to update the geometry with it and possibly to move the mesh. We summarize the whole procedure in Algorithm 4. As mentioned above, the update rules differ for NURBS with variable weights in the discretize first and optimize first ansatz. For a comparison with line 6, we state the update rule of discretize first in the following code snippet

$X_i \leftarrow X_i + (s_{i,1}, \ldots, s_{i,d})^T$
$\omega_i \leftarrow \omega_i + s_{i,d+1}$.

Hence, for $S_i := (s_{i,1}, \ldots, s_{i,d})^T$ and $W_i := s_{i,d+1}$ an update rule with homogeneous coordinates results in a geometry

$$G_{new} = \sum_{i=1}^{n} (X_i \omega_i + S_i) \frac{N_i}{\sum_{j=1}^{n} (\omega_j + W_j) N_j} \tag{45}$$

as opposed to the new geometry from the discretize first approach

$$G_{new} = \sum_{i=1}^{n} (X_i + S_i)(\omega_i + W_i) \frac{N_i}{\sum_{j=1}^{n} (\omega_j + W_j) N_j} \ . \tag{46}$$

Algorithm 4 UPDATE(Ω, s)

Require: stiffness matrix A of mesh moving problem
Require: geometry function G of initial domain Ω_0
Require: descent direction s from black-box gradient descent optimizer
1: apply mesh movement function $y = \Phi(s)$ to propagate the change also to the inside control points, $s \leftarrow y$
2: **if** G is a B-spline geometry or a NURBS geometry with fixed weights **then**
3: **for all** geometry basis functions N_i **do**
4: $X_i \leftarrow X_i + (s_{i,1}, \ldots, s_{i,d})^T$
5: **end for**
6: **else** G is a NURBS geometry with variable weights
7: **for all** geometry basis functions N_i **do**
8: $X_i \leftarrow X_i + (s_{i,1}, \ldots, s_{i,d})^T / s_{i,d+1}$
9: $\omega_i \leftarrow \omega_i + s_{i,d+1}$
10: **end for**
11: **end if**

6 Model Problems

In this section we apply the complete Algorithm 1 with the procedures 2–8 to two model problems. These examples, though simple, illustrate the isogeometric shape optimization process and moreover, expose possible weaknesses, which have to be considered. Both examples were also treated by the authors in [6].

We treat two optimization problems: maximizing an area and increasing stiffness. The first one involves no PDE, but demonstrates the influence of quadrature errors in optimizing weights and control points simultaneously. We review it here under this aspect which sheds new light on using weights as optimization variables.

The second problem is well known in shape optimization and has been treated already with isogeometric analysis, also by others. We include it here, for two reasons. Firstly, it is posed in the discretize first setting in IGA literature, so here we view it also in the optimize first picture. Secondly, in this example we show practically the use of two meshes.

6.1 Area Maximization

Since NURBS are able to represent conics exactly, we treat first a simple example without PDE to show the use and also pitfalls of isogeometric shape optimization with variable weights.

Example 1 (Area maximization)

$$\begin{aligned} &\min J(\Omega) = -\int_{\Omega} 1 \mathrm{d}x \quad \text{subject to} \\ &h(\Omega) = \int_{\partial\Omega} 1 \mathrm{d}s - P_0 = 0 \text{ with a constant } P_0 \,. \end{aligned} \tag{47}$$

Shape gradients for J and h without variable weights are given by Eqs. (24), (25), and (26), where $j_1 = 1$ covers the volume and $j_2 = 1$ the perimeter functionals.

$$J'(\Omega;\theta) = -\int_{\hat{\Omega}} \operatorname{trace}(J_G^{-1} D\theta) |\det J_G| \mathrm{d}\hat{x} \tag{48}$$

$$h'(\Omega;\theta) = \int_{\partial\hat{\Omega}} \left(\operatorname{trace}(J_G^{-1} D\theta) - \frac{\hat{n}^T J_G^{-1} J_G^{-T} D\theta^T J_G^{-T} \hat{n}}{|J_G^{-T} \hat{n}|} \right) |\det J_G| \mathrm{d}\hat{s} \,. \tag{49}$$

Considering also variable weights, the shape derivative equations (31), (32), and (33) yield

$$J'(\Omega;\theta) = -\int_{\hat{\Omega}} \left(\operatorname{trace}(J_G^{-1}\dot{D})\right) |\det J_G| \mathrm{d}\hat{x} \tag{50}$$

$$h'(\Omega;\theta) = \int_{\hat{\Gamma}} \left(|J_G^{-T}\hat{n}| \operatorname{trace}(J_G^{-1}\dot{D}) + \right.$$
$$\left. - \frac{(J_G^{-T}\hat{n})^T J_G^{-T}\dot{D}^T J_G^{-T}\hat{n}}{|J_G^{-T}\hat{n}|} \right) |\det J_G| \mathrm{d}\hat{x} \,. \tag{51}$$

6.1.1 Results

We start the optimization process with a square $[-1,1]^2$ given by B-splines with degree $p = 1$ to 4 over knot vectors $(0,\ldots,0,1,\ldots,1)$ in both space dimensions. The first and last knot are repeated $p+1$ times and $V_0 = 4$ and $P_0 = 8$. A convergence study of Fig. 3 reveals, two things. First, we see that non-rational B-splines behave as one expects. Namely, since higher degree B-splines can approximate conics better, they should perform better, i.e. giving smaller errors than lower degree approximations. Second, one anticipates h-convergence for a fixed degree on the same grounds. Both is indicated in the plot. For $p = 4$ the flat behaviour after two refinements is due to reaching machine precision. A first

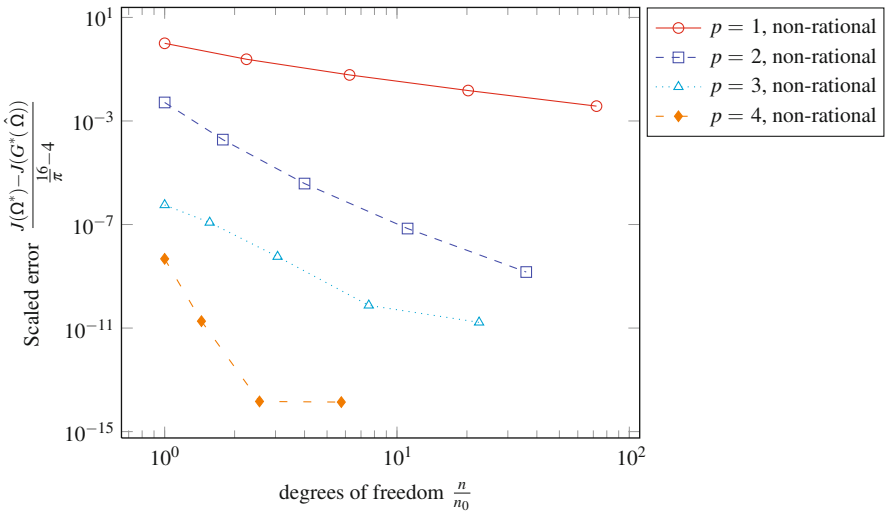


Fig. 3 Convergence study of Example 1 with non-rational B-splines of different degrees p

guess would assign the same behaviour to rational B-splines, even expect a much lower error for $p = 2$ already, because the disk has a representation in this space. We use again the knot vectors $(0, 0, 0, 1, 1, 1)$ in both space direction with 3×3 rational B-spline functions and 9 weights, all equal to 1; control points are such that we have the same initial square as above. Since rational B-splines are rational polynomials we have to take quadrature errors into account. We pursue this in the following before starting optimization with NURBS.

6.1.2 Influence of Quadrature Errors

When picking efficient quadrature rules for the assembly of a_G and l_G in IGA, transformation terms $\det J_G$ and J^{-T} are neglected in this search by [9] on the basis that these terms change slowly in comparison to other factors. This is in general valid since the geometry is fixed at a coarse level with degree p_0 and mesh size parameter h_0; for analysis a (several times) p- and h-refined version is used. Hence, G is a polynomial with degree $p_0 \leq p$ and almost constant on elements of sizes $h \ll h_0$. This carries over to the transformation terms. However, this assumption does not hold anymore when the geometry G has the same refinement level or worse, for cost functionals or for constraints on Ω without state, as for instance a volume constraint in the form of $h(\Omega) = \int_\Omega 1 \mathrm{d}x - const$. In IGA, this yields $h(G) = \int_{\hat{\Omega}} |\det J_G| \mathrm{d}\hat{x}$.

Using rational B-splines, this simple constraint already exhibits a need for higher order quadrature rules. That is, if we enforce the volume constraint by a tolerance smaller than 10^{-6} in the optimization routine, Table 1 shows that we need at least more than 5 quadrature points in each direction to get rid of integration errors.

6.1.3 Optimization with NURBS

For Example 1 with NURBS of degree $p = 2$ and variable weights the exact disk Ω^* is in this NURBS space, so we expect an error $J(\Omega^*) - J(G^*(\hat{\Omega}))$ dominated by quadrature errors and optimization parameters. In Fig. 4 a visually good result is obtained for the following setting. A Gauß quadrature with 10 quadrature points in each space direction is used. The perimeter constraint is enforce within a tolerance of 10^{-8} and as stopping criteria for the SQP optimizer in the package NLOPT [10] the same tolerance is chosen. In Fig. 4 no difference between exact and numerical

Table 1 Quadrature errors in area and perimeter of a circle with radius 1 given by exact representation with NURBS of degree $p = 2$

#quad. points	5	9	15
Area	$\approx 8 \cdot 10^{-06}$	$\approx 3 \cdot 10^{-11}$	$\approx 1 \cdot 10^{-15}$
Perimeter	$\approx 9 \cdot 10^{-07}$	$\approx 2 \cdot 10^{-12}$	$\approx 9 \cdot 10^{-16}$

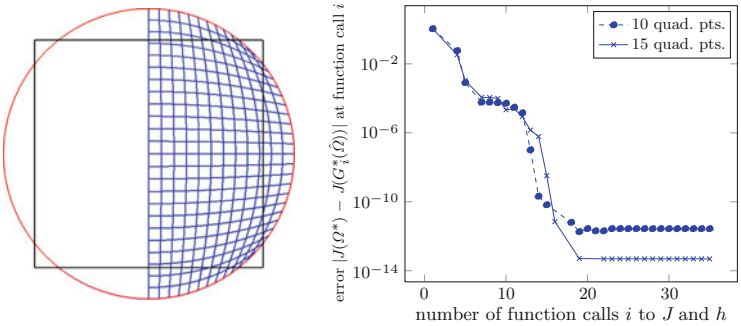


Fig. 4 Result for Example 1 with initial domain (*square*): The optimal numerical shape (plotted as IGA mesh on half disk) is a NURBS approximation of a disk (exact disk plotted as boundary *circle*). On the right, the history of (feasible) SQP optimization steps is shown

optimum can be detected. In fact, the true error in the cost functional is $|J(\Omega^*) - J(G^*(\hat{\Omega}))| \approx 2.1 \cdot 10^{-10}$ after 34 function calls, i.e. iterations. For an even smaller error we increase the quadrature rule to 15 points in each direction and require that the perimeter may not deviate more than 10^{-12} from $P_0 = 8$ and as stopping tolerance we give 10^{-9}. Then, the error $|J(\Omega^*) - J(G^*(\hat{\Omega}))|$ drops to $\approx 8.9 \cdot 10^{-15}$ after 34 function calls. It is unreasonable to expect anything less since from Table 1 we learn that the error in the objective is then dominated by quadrature errors.

Comparison with Non-rational B-Splines

We compare the results from rational and non-rational B-spline optimization and how many design variables we need in both cases for an error in the cost functional of order 10^{-15}. In case of NURBS with weight optimization we have 9 control points with 8 of them corresponding to basis functions with support on the boundary. Since they have two components, we end up with 16 optimization variables plus the 8 weights for each basis, amounting to 24 design variables in total. In case of B-splines without weights Fig. 3 shows that for the desired error we need B-splines of degree $p = 4$ which have been h-refined twice. This leaves us with 64 degrees of freedom of which 32 are nonzero at the boundary. Although we have less design variables using NURBS we pay for it by using higher quadrature rules. Together with a PDE constraint it might therefore be computationally more efficient to use B-splines of higher degrees if the optimal shape is expected to be conic.

6.2 Compliance Minimization

This prominent example has been treated already in [2, 6, 20, 27] with IGA. Our linear elasticity problem for plane stress has dimension $m = 2$ and the strong form

$$\begin{cases} -\operatorname{div}\sigma(u) &= 0 \text{ in } \Omega, \\ u &= 0 \text{ on } \Gamma_D, \\ \sigma(u)\cdot n &= g \text{ on } \Gamma_N, \\ \sigma(u)\cdot n &= 0 \text{ on } \Gamma, \end{cases} \tag{52}$$

with strain $\varepsilon(u) = \frac{1}{2}(\nabla u + \nabla u^T)$ and stress $\sigma(u) = 2\mu\varepsilon(u) + \lambda(\nabla\cdot u)I_2$. We use a constant vector field g in the Neumann boundary conditions on Γ_N, the domain is fixed on the boundary part Γ_D and allowed to move on Γ, where $\partial\Omega$ is the union of all closed boundaries $\bar{\Gamma}_D \cup \bar{\Gamma}_N \cup \bar{\Gamma}$. For this problem $E(u, \Omega) = 0$ is equivalent to

$$u \in H_0^1(\Omega)^2\colon \quad \int_\Omega \underline{\varepsilon}(u)^T C \underline{\varepsilon}(v)\mathrm{d}\Omega = \int_{\Gamma_N} g\cdot v\mathrm{d}\Gamma_N \quad \forall v \in H_0^1(\Omega)^2, \tag{53}$$

where $\underline{\varepsilon}(u) = \left(\varepsilon_{11}, \varepsilon_{22}, \frac{1}{2}(\varepsilon_{12}+\varepsilon_{21})\right)^T$ and elasticity tensor C characterized by the linearized stress-strain relation. Isogeometric Galerkin projection of the plane stress problem in Eq. (52) yields

$$a_G(M_i e_l, M_j e_k) = \int_{\hat{\Omega}} \underline{\varepsilon}_G(M_i e_l)^T C \underline{\varepsilon}_G(M_j e_k)|\det J_G|\mathrm{d}\hat{x}\,. \tag{54}$$

with $\underline{\varepsilon}_G$ obtained from the modified gradient $\nabla u \circ G = J_G^{-T}\nabla\hat{u}$. The linear form is transformed to $l_G(M_i e_l) = \int_{\hat{\Gamma}_N} g\circ G\cdot M_i e_l \|\mathrm{d}_{\hat{s}} G\|\mathrm{d}\hat{s}$ where $\hat{\Gamma}_N$ is the preimage of Neumann boundary Γ_N under G.

Example 2 (Compliance minimization) The shape optimization problem takes the form

$$\min J(\Omega, u) = \int_{\Gamma_N} g\cdot u\mathrm{d}x \tag{55}$$

$$\text{s.t. } h(\Omega) = \int_\Omega 1\mathrm{d}x - V_0 = 0\,. \tag{56}$$

In this example we have a volume constraint on the domain with V_0 a constant, such as initial volume. Moreover, our domain of interest is a square plate with a hole, where the form of the hole is to be optimized. In [24] it is argued that a circular hole has maximal stiffness, i.e. minimal compliance. Since the problem is symmetric we consider only a quarter of the plate which is depicted in Fig. 5 together with its boundary conditions.

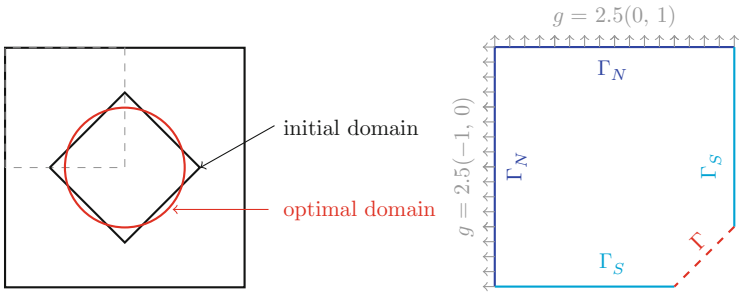


Fig. 5 Configuration of Example 2 where the quarter plate Ω is connected at boundary Γ_S to congruent plates symmetrically to give a closed hole, the moving boundary is given by Γ

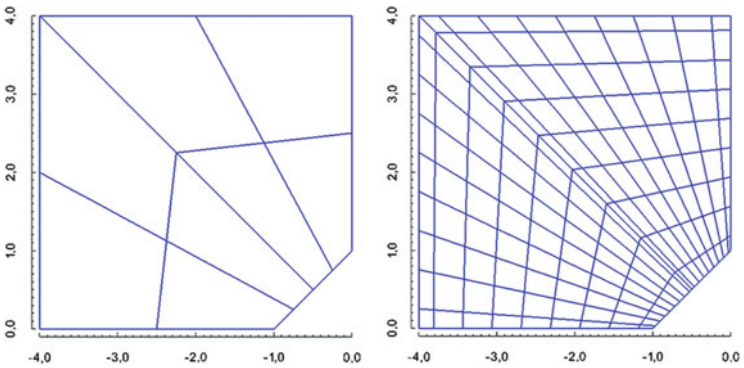


Fig. 6 *Left*: optimization mesh, *right*: simulation mesh, both degree 2, simulation mesh is 3 times h-refined, both depicted as control mesh

From Eq. (41) we deduce the directional shape derivative for the objective function in case of fixed weights using either a domain $J(\Omega, u) = a(u, u)$ or boundary representation $J(\Omega, u) = l(u)$ of the cost functional. This problem is self-adjoint, hence we have for the domain representation of the shape derivative of J

$$J'(\Omega, u; \theta) = \mathrm{d}_t \int_{\hat{\Omega}} \underline{\varepsilon}_{G+t\theta}(M_i e_l)^T C \underline{\varepsilon}_{G+t\theta}(M_j e_k) |\det J_{G+t\theta}| \mathrm{d}\hat{x} \Big|_{t=0} . \tag{57}$$

The shape derivative of the volume constraint h is given by Eq. (48). For this example, we assume two separate levels of h-refinement in the simulation and optimization mesh, the parameterizations are shown in Fig. 6. As predicted, the optimization process leads to a circular hole in Fig. 7 and stress peaks are much alleviated.

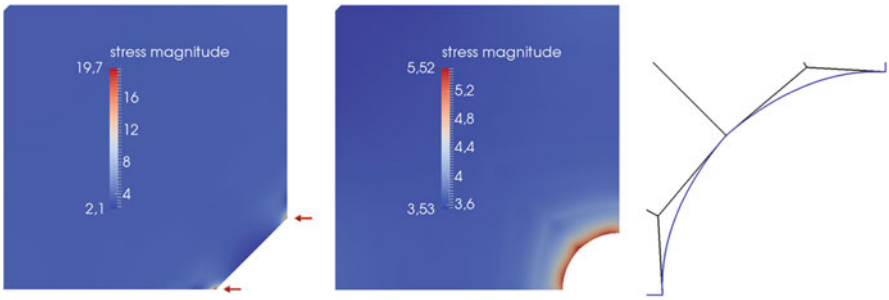


Fig. 7 *Left*: stress on initial physical domain with stress peaks at arrow tips, *middle*: stress on numerical optimal physical domain, *right*: control polygon and physical boundary of optimization mesh after end of optimization

7 Conclusion

In this work, we have treated shape optimization in isogeometric analysis in an algorithmic way. In particular, we formulated its pieces on a discretized level. Model examples verified the implementation. We relied on third party libraries for optimization like [10]. Of course, stopping tolerances of such optimization routines only make sense if we can provide reasonably good approximations of all occurring terms. Usually, quadrature errors go without saying in this discussion as they are easily diminished. Instead of neglecting them here, in the first problem we show that optimization with NURBS requires more carefully chosen quadrature rules than with B-splines. However, in future we mostly are interested in the less obvious influence of interpolation errors for state and controls, which we want to address separately. For that, the second example showcased the use of two different meshes originated from one initial parameterization. From a priori error estimates, also valid for IGA [1], interpolation power of state ansatz spaces are known. With such a detached simulation grid, the error in the state equation can be controlled by finer grids and, together with a posteriori error estimates, locally adaptive refinement [25] for simulation is possible. Lacking at the moment is a means to determine the refinement level of the optimization grid and the influence of its approximation error in the cost term. Our next steps involve first hierarchical B-splines for locally refined state solutions on a static optimization mesh. Next, a successive refinement at the boundary of interest of the optimization mesh is intended.

Acknowledgements We thank Utz Wever from Siemens AG, Corporate Technology, for many helpful discussions on the subject.

This work was supported by the European Union within the Project 284981 “TERRIFIC” (7th Framework Program).

References

1. Y. Bazilevs, L. Beirão da Veiga, J. Cottrell, T.J.R. Hughes, G. Sangalli, Isogeometric analysis: approximation, stability and error estimates for *h*-refined meshes. Math. Models Methods Appl. Sci. **16**(7), 1031–1090 (2006)
2. L. Blanchard, R. Duvigneau, A.V. Vuong, B. Simeon, Shape gradient for isogeometric structural design. J. Optim. Theory Appl. **161**(2), 1–7 (2013)
3. P.B. Bornemann, F. Cirak, A subdivision-based implementation of the hierarchical B-spline finite element method. Comput. Methods Appl. Mech. Eng. **253**, 584–598 (2013)
4. C. De Boor, *A Practical Guide to Splines*, vol. 27 (Springer, New York, 2001)
5. M.C. Delfour, J.P. Zolésio, *Shapes and Geometries: Metrics, Analysis, Differential Calculus, and Optimization*. Advances in Design and Control, 2nd edn. (SIAM, Philadelphia, 2011)
6. D. Fußeder, A.V. Vuong, B. Simeon, Fundamental aspects of shape optimization in the context of isogeometric analysis. Comput. Methods Appl. Mech. Eng. **286**, 313–331 (2015)
7. J. Haslinger, P. Neittaanmäki, *Finite Element Approximation for Optimal Shape Design: Theory and Applications* (Wiley, Chichester/New York, 1988)
8. T.J.R. Hughes, J.A. Cottrell, Y. Bazilevs, Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. Comput. Methods Appl. Mech. Eng. **194**(39), 4135–4195 (2005)
9. T.J.R. Hughes, A. Reali, G. Sangalli, Efficient quadrature for NURBS-based isogeometric analysis. Comput. Methods Appl. Mech. Eng. **199**, 301–313 (2010)
10. S.G. Johnson, The NLOPT nonlinear-optimization package (2014), http://ab-initio.mit.edu/nlopt. [Online; Accessed 10 Oct 2014]
11. J. Kiendl, R. Schmidt, R. Wüchner, K.U. Bletzinger, Isogeometric shape optimization of shells using semi-analytical sensitivity analysis and sensitivity weighting. Comput. Methods Appl. Mech. Eng. **274**, 148–167 (2014)
12. MATLAB, Release 2012a (2012), "The MathWorks Inc."
13. F. Murat, J. Simon, Etude de problèmes d'optimal design, in *Optimization Techniques Modeling and Optimization in the Service of Man Part 2* (Springer, Berlin/Heidelberg, 1976), pp. 54–62
14. D.M. Nguyen, Isogeometric analysis and shape optimization in electromagnetism. Ph.D. thesis, Technical University of Denmark, Feb 2012
15. D.M. Nguyen, A. Evgrafov, J. Gravesen, Isogeometric shape optimization for electromagnetic scattering problems. Prog. Electromagn. Res. B **45**, 117–146 (2012)
16. J. Nocedal, S.J. Write, *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering, 2nd edn. (Springer, New York, 2006)
17. L. Piegl, W. Tiller, *The NURBS Book* (Springer, New York, 1995)
18. O. Pironneau, *Optimal Shape Design for Elliptic Systems* (Springer, New York, 1983)
19. B. Protas, T.R. Bewley, G. Hagen, A computational framework for the regularization of adjoint analysis in multiscale PDE systems. J. Comput. Phys. **195**, 49–89 (2004)
20. X. Qian, Full analytical sensitivities in NURBS based isogeometric shape optimization. Comput. Methods Appl. Mech. Eng. **199**, 2059–2071 (2010)
21. L.L. Schumaker, *Spline Functions: Basic Theory* (Wiley, New York, 1981)
22. J. Sokolowski, J.P. Zolésio, *Introduction to Shape Optimization: Shape Sensitivity Analysis*. Springer Series in Computational Mathematics, vol. 16 (Springer, Berlin/New York, 1992)
23. K. Svanberg, A class of globally convergent optimization methods based on conservative convex separable approximations. SIAM J. Optim. **12**(2), 555–573 (2002)
24. S. Timoshenko, J.N. Goodier, *Theory of Elasticity* (McGraw-Hill, New York, 1951)
25. A.V. Vuong, *Adaptive Hierarchical Isogeometric Finite Element Methods* (Springer, Wiesbaden, 2012)

26. A. Wächter, L.T. Biegler, On the implementation of a primal-dual Interior point filter line search algorithm for large-scale nonlinear programming. Math. Program. **106**(1), 25–57 (2006)
27. W.A. Wall, M.A. Frenzel, C. Cyron, Isogeometric structural shape optimization. Comput. Methods Appl. Mech. Eng. **197**, 2976–2988 (2008)
28. G. Xu, B. Mourrain, R. Duvigneau, A. Galligo, Optimal analysis-aware parameterization of computational domain in 3D isogeometric analysis. Comput. Aided Des. **45**(4), 812–821 (2013)

Effective Integration of Sophisticated Operators in Isogeometric Analysis with `igatools`

Nicola Cavallini, Oliver Weeger, M. Sebastian Pauletti, Massimiliano Martinelli, and Pablo Antolín

Abstract `igatools` is a newly released library for operators assembly in isogeometric analysis. The library, which is object oriented designed and written in `C++11`, is general purpose, therefore it is not devoted to any specific application. In this paper we show that such a design makes `igatools` an effective tool in assembling isogeometric discretizations of sophisticated differential operators. This effectiveness will be demonstrated showing code snippets relating one-to-one with the operators written on paper. To embrace a wide audience, applications from nonlinear incompressible solid and fluid mechanics will be addressed. In both cases we are going to deal with mixed isogeometric formulations. The applicative nature of this paper will be stressed solving industrially relevant tests cases.

1 Introduction

Scientific computing is an area where expertise from several backgrounds such as computer science, engineering, mathematics, and physics come together. Contributions to this scientific field range from most theoretical to the most applied ones.

N. Cavallini (✉)
Scuola Internazionale Superiore di Studi Avanzati, Via Bonomea 265, Trieste, Italy
e-mail: nicola.cavalli@sissa.it

O. Weeger
Faculty of Mathematics, TU Kaiserslautern, P.O. Box 3049, 67653 Kaiserslautern, Germany
e-mail: weeger@rhrk.uni-kl.de

M.S. Pauletti
Instituto de Matemática Aplicada del Litoral (IMAL), Consejo Nacional de Investigaciones científicas y técnicas (CONICET), Santa Fe, Argentinia

M. Martinelli
Istituto di Matematica Applicata e Tecnologie Informatiche (IMATI), Consiglio Nazionale delle Ricerche (CNR), Pavia, Italy

P. Antolín
Dipartimento di Ingegneria Civile ed Architettura, Università degli Studi di Pavia, Via Ferrata 3, 27100 Pavia, Italy

B. Jüttler, B. Simeon (eds.), *Isogeometric Analysis and Applications 2014*,
Lecture Notes in Computational Science and Engineering 107,
DOI 10.1007/978-3-319-23315-4_9

With this work we address the numerical solution of nonlinear partial differential equations arising from industrial applications in solid and fluid mechanics. The aim of the paper is to demonstrate that the newly released software library `igatools` allows an effective implementation of isogeometric finite element discretizations of rather complex differential operators, here in particular mixed formulations of nonlinear incompressible elasticity and Navier-Stokes equations, defined on complex domains.

As the software prefix `iga` suggests, we are dealing with isogeometric type of spaces. Isogeometric analysis has been introduced in [24] with the aim of bridging scientific computing and computational geometry. The basic idea here, is to use the shape functions that describe the geometry, as basis functions for a Galerkin method. Several aspects in computational modeling benefit from this linking. In particular, an appealing aspect of this approach is the higher inter element continuity that characterizes the shape functions describing the geometry and numerical solution. Among other contributions we would mention [13] for structural vibration, [27] for shell structures and [35] in non-linear vibration analysis. In fluid dynamics we would like to mention [6] for a variational multi scale modeling approach to turbulent flows and [19] for divergence conforming B-splines.

`igatools` is an object oriented library for discretization of partial differential equations, using isogeometric type of spaces. It has been first presented in [29] and it is available under GPL conditions at www.igatools.org. The idea that underlays the software design is encapsulating mathematical concepts used in isogeometric method into objects, namely classes. Mapping the interaction between classes we get methods used in the actual integration. The software is implemented in `C++11` [7, 26] and it makes an extensive use of generic programming, templates in `C++`. In this context, generic programming is particularly useful to obtain dimension independent code, a very interesting feature of the software.

This paper is devoted to the community of scientists curious about numerical solution implementation of partial differential equations. In this respect we thought at two model readers. Those who might be skeptic about object oriented programming in scientific computing, and to those who already have an acquaintance with this programming style. We aim at involving the first class of readers showing that a carefully designed software can help in implementing non elementary operators. In the second case we would like to capture their attention in a collaborative way. To pursue this twofold objective, we think there is no better way rather than presenting a simple way of implementing sophisticated operators. The implementation ease will be demonstrated providing code snippets, that prove a one-to-one correspondence between the way we write the operators on paper and the way these operators are coded in assembly loops.

Validation of the software will be provided with respect to literature test cases and, industrial applications will be presented to prove feasibility of the software in applicative contexts. In order to embrace the widest possible audience, we present two systems of partial differential equations, one arising from solid mechanics and one from fluid dynamics. In both cases the problems are nonlinear. Though this paper is not focused on numerical treatment of nonlinear systems, we would like to

remark that the nonlinearities treatment is a matter of linear algebra strategies, that do not interfere with operators assembly.

One of the goals of this paper is to show that `igatools` can be an effective tool in industrially relevant applications. `igatools` development has been supported by several institutions and projects. In the context of this paper, a special mention goes to the TERRIFIC project, as the presented applications are partial fulfillment of the project itself. TERRIFIC (Towards Enhanced Integration of Design and Production in the Factory of the Future through Isogeometric Technologies) is a project within the seventh framework program of the European Union, that lasted from 1st of September 2011 to the 31st of August 2014. The aim of the project was to inject isogeometric analysis in an industrial context. The geometries presented are courtesy of ALENIA aeronautica, in particular the authors are grateful to L. Morrone and G. Mirra. The other supporting institutions and projects will be listed in the acknowledgments.

In Sect. 2 we continue with a brief summary of `igatools` design. As the applications we are going to deal with require mixed formulations, we recall the isogeometric formulation of Taylor-Hood elements in Sect. 3. In Sect. 4 we detail our computational mechanics applications and in Sect. 5 our fluid dynamics applications are addressed. Finally, in Sect. 6 we conclude the paper.

2 `igatools` Design Description

The software prefix `iga` restricts the discussion to isogeometric methods. Isogeometric methods commonly address Galerkin approximations, characterized by high inter element continuity of basis functions. Originally, in historical sense, basis functions for isogeometric methods are B-Splines and NURBS, the interested reader can refer to [30, 31] for a precise definition, and to [29] for `igatools` related notation. Other than these, new reference spaces have been introduced lately. The reader can refer to [5, 11, 15, 32] for T-splines, to [14] for locally refined splines, and to [20, 22, 34] for hierarchical splines. In this paper, we are going to use B-splines. B-splines of degree d and regularity r ($0 \leq r < d$) will be addressed as Splines S_r^d. In this section we briefly review the objects, or classes, that build the backbone of the software. As extensively explained in [29], these objects are designed to resemble the key mathematical concepts in isogeometric methods. A visual sketch of the major players is outlined in Fig. 1.

2.1 *Reference Space (`BSplineSpace` and `NURBSSpace`)*

The classes `BSPlineSpace` and `NURBSSpace` represent the shape functions defined on the parametric domain (usually the unit square in two dimensions and the unit cube in three dimensions). These spaces can be non-homogeneous, in the

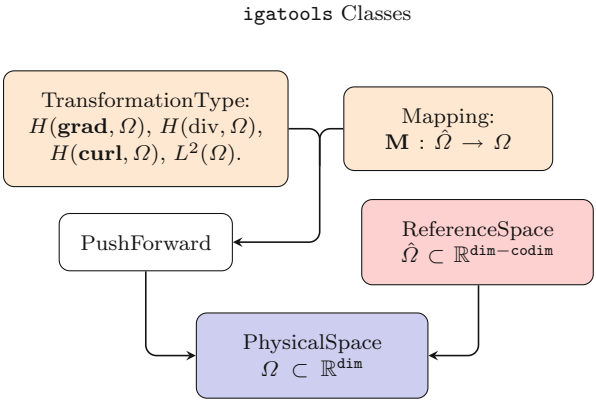


Fig. 1 In this sketch we picture the major object we isolated in isogeometric analysis, together with their mathematical symbols. We outlined that in general the dimension of the reference space can differ from the physical space, this difference is addressed as codimension `codim`

sense that they can be characterized by different degrees and regularities in each dimension.

2.2 *Mapping (`Mapping`)*

The mathematical mapping is the object that maps the parametric domain into the physical one. Notice that, in principle, there is no reason why this mapping should be restricted to isogeometric type of maps. Moreover, in the desired case of an isogeometric type of map, `Mapping` is not supposed to be refined together with the reference space to achieve convergence.

2.3 *Push Forward (`PushForward`)*

The `PushForward` object combines the `Mapping` together with the `TransformationType`. The transformation type defines how to transform functions. Depending on the different transformations, different operators can be preserved throughout the transformation itself. Example transformations are $H(\mathbf{grad}, \Omega)$, $H(\mathrm{div}, \Omega)$, $H(\mathbf{curl}, \Omega)$ and $L^2(\Omega)$.

2.4 *Physical Space (`PhysicalSpace`)*

The combination of the `ReferenceSpace` with the `PushForward` gives the `PhysicalSpace`. This class contains all the information necessary to recover point values of functions and derivatives on the physical domain. This point values are the ones used to assemble the desired operators.

Element Iterator (`ElementIterator`) One of the similarities between standard finite elements and the isogeometric method, is the decomposition of the domain in a collection of elements where a small number of functions have support. The global matrix, is built *iterating* through all the elements in the grid. In object oriented programming the mechanism that points to an element, and modifies itself to point to the next element is called iterator. In [29] we pointed out that we consider the previously mentioned classes as *grid-li ke* containers, this why the element level information is accessed via the `ElementIterator` class.

For an immediate flavor on how a simple $\int \nabla u \nabla v$ operator is assembled, we sketch a sort of "Hello World" program, in Listing 1. In the first four lines we define the presented classes. We start with a `CartesianGrid`, namely the knots without repetitions. This class is used to build the corresponding `BSplineSpace`. The transformation type (`h_grad`, in this case) couples with a `Mapping` to build the `PushForward`. Finally it only remains to instantiate the `PhysicalSpace` as the combination of `BSplineSpace` and `PushForward`. The mechanism that implies the initialization of the cache `elem->init_values()` and its filling is detailed in [29]. In this paragraph we limit ourselves to emphasize that we access

```
1   auto grid = CartesianGrid<dim>::create(box, n_knots);
2   auto ref_space = BSplineSpace<dim>::create(grid, deg);
3   map       = BallMapping<dim>::create(grid);
4   space     = PhysicalSpace<BSplineSpace<dim>,
5       PushForward<h_grad, dim>>::create(ref_space,
6        PushForward<h_grad, dim>::create(map));
7
8   auto elem = space->begin();
9   const auto elem_end = space->end();
10  ValueFlags fill_flags = ...
11  elem->init_values(fill_flags, elem_quad);
12
13  for (; elem != elem_end; ++elem){
14      elem->fill_values();
15      for (int i = 0; i < n_basis; ++i){
16          for (int j = 0; j < n_basis; ++j){
17              for (int qp = 0; qp < n_qp; ++qp){
18                  loc_mat(i,j) += scalar_product(
19                      elem->get_basis_gradient(i,qp),
20                      elem->get_basis_gradient(j,qp))*
21                      w_meas[qp];}}}
```

Listing 1 A first "Hello World" like code. Intended to assemble a simple $\int \nabla u \nabla v$ operator

the shape function values, using `PhysicalSpace` as a grid-like container, to assemble the local matrix.

3 Mixed Methods with Isogeometric Finite Elements

The mathematical description of physical phenomena involves several mechanisms interplaying at the same time. It often happens that different independent variables are necessary to describe a given mechanism. When such independent variables are discretized using different solution spaces we have a mixed method.

A typical case is an equation of motion coupled with an incompressibility constrain. In the applications we are going to address, namely incompressible elasticity (Sect. 4) and incompressible Navier-Stokes (Sect. 5), the independent variables are two. The first one will be referred to as $\mathbf{u}$ and will be used to describe velocities or displacements. The second one is going to be p that represents the Lagrangian multiplier of the incompressibility condition. In both mechanical and fluid dynamical contexts the discretization and linearization of the continuous operators will lead to a block linear system of the form:

$$\begin{pmatrix} A & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ p \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ 0 \end{pmatrix}.$$

If we consider the two discrete spaces $\mathcal{V}_h$ and $\mathcal{Q}_h$ for $\mathbf{u}$ and p respectively, it is well known that these two spaces have to satisfy the inf-sup condition, see [1, 8, 9]. In standard finite elements, an established choice is the Taylor-Hood element. Here the discrete spaces for $\mathbf{u}$ and p share the same triangulation and the inf-sup condition is fulfilled when the degree of the shape functions for $\mathbf{u}$ is $d + 1$, being d the shape functions degree for p.

The isogeometric counterpart of standard Taylor-Hood elements is constructed choosing $\mathcal{V}_h = (S_{d-1}^{d+1})^{\mathtt{dim}}$ and $\mathcal{Q}_h = S_{d-1}^{d}$, being `dim` the dimension. This type of element has been first used in [4] and a stability proof can be found in [10]. For sake of clearness we represent in one dimension the shape functions corresponding to S_1^3/S_1^2 pair in Fig. 2.

As mentioned in Sect. 2, the software design should resemble the mathematical concepts needed to integrate the boundary value problem we are interested in, in a one-to-one correspondence. As different spaces are required to solve for $\mathbf{u}$ and p, the same situation will be reproduced in the code.

Listing 2 reports the code necessary to define $\mathcal{Q}_h$ and $\mathcal{V}_h$. We first focus on the solution space for p, namely `BSplineSpace<dim>`. As previously mentioned, one of the most appealing features of `igatools` is its code dimension independence. In the present case of a scalar space, we only need to define the first template parameter as `dim`, the other template parameters are default defined to give a scalar space. A bit different is the situation for the vector space $\mathcal{V}_h$. In this case mathematics is elegantly resembled by specifying the second template

Fig. 2 In this figure we sketch one dimensional shape functions for the Taylor-Hood type of spaces. In particular we represent S_1^3 splines for the velocity, and S_1^2 splines for the pressure. The corresponding knot vectors are: $\Xi_{\mathbf{u}} = \{0, 0, 0, 0, 0.25, 0.25, 0.5, 0.5, 0.75, 0.75, 1, 1, 1, 1\}$ for the velocity and $\Xi_p = \{0, 0, 0, 0.25, 0.5, 0.75, 1, 1, 1\}$ for the pressure. (**a**) Space $\mathcal{Q}_h = \mathcal{S}_1^2$ for the Lagrangian multiplier p. (**b**) Space $\mathcal{V}_h = \mathcal{S}_1^3$ of primal variable $\mathbf{u}$

parameter, the equations range is the same as the problem dimension, this is why we get `BSplineSpace<dim,dim>`.

In code snippet in Listing 2 the reader can also notice how two different vectors, `mult_u` and `mult_p`, are defined to assign the right multiplicity to each knot. Following the Taylor-Hood element definition, the two spaces must have the same regularity and different degree, and we need to set the knots multiplicity to obtain compatible spaces.

4 Computational Mechanics: Nonlinear Incompressible Elasticity

As a first application for the `igatools` implementation of a mixed method, we present the isogeometric discretization of nonlinear elasticity, i.e. large deformation kinematics and a hyperelastic constitutive law, subject to an incompressibility constraint.

```
using PreSpace = BSplineSpace<dim>;
using VelSpcae = BSplineSpace<dim, dim>;

deg_u = deg_p + 1;

shared_ptr<BSplineSpace<dim>> pre_space;
shared_ptr<BSplineSpace<dim, dim>> vel_space;

vector<int> mult_p(n_knots, deg_p - reg);
vector<int> mult_u(n_knots, deg_u - reg);
mult_p[0] = mult_p[n_knots-1] = deg_p + 1;
mult_u[0] = mult_u[n_knots-1] = deg_u + 1;

pre_mult.fill(mult_p);
vel_mult.fill(mult_u);

pre_deg.fill(deg_p);
vel_deg.fill(deg_u);

auto grid = CartesianGrid<dim>::create(n_knots);
pre_space = PreSpace::create(grid, pre_mult, pre_deg);
vel_space = VelSpace::create(grid, vel_mult, vel_deg);
```

Listing 2 Definition of to different spaces for the variables $\mathbf{u}$ and p. In this listing we detail the construction of spaces with different degrees and multiplicites

Incompressible elasticity has been studied already by a few researchers in the context of isogeometric formulations. Elguedj et al. investigated the use of $\bar{B}$- and $\bar{F}$-projection methods for linear and nonlinear incompressible elasticity. Taylor [33] and Mathisen et al. [28] employed a three-field mixed approach for nonlinear incompressible elasticity with independent approximation of $\mathbf{u}$, p and θ. Here we use a classical mixed method with independent variables $\mathbf{u}$ and p, as presented in the previous section, and outline its implementation using `igatools`. The developed solver is validated using a well-known benchmark example and then applied to an industrial test case.

In this paper we employ the Langrangian or material description of the large deformation elasticity problem in `dim` $= 2, 3$ dimensions, as for instance introduced in [1, 36]. Thus, a body in its undeformed, Lagrangian configuration is given by its reference or material domain $\Omega \in \mathbb{R}^{\texttt{dim}}$ and its deformed, current configuration is expressed with the mapping $\hat{\varphi} : \Omega \to \mathbb{R}^{\texttt{dim}}$:

$$\hat{\varphi}(\mathbf{X}) = \mathbf{X} + \hat{\mathbf{u}}(\mathbf{X}). \tag{1}$$

Following the notation of [1] from the displacement field $\hat{\mathbf{u}}$ we can derive the kinematic quantities such as the deformation gradient

$$\hat{\mathbf{F}} = \mathbf{F}(\hat{\mathbf{u}}) = \mathbf{I} + \nabla\hat{\mathbf{u}}, \tag{2}$$

and the right Cauchy-Green strain tensor

$$\hat{\mathbf{C}} = \mathbf{C}(\hat{\mathbf{u}}) = \hat{\mathbf{F}}^T\hat{\mathbf{F}}. \tag{3}$$

$\hat{J} = \det\hat{\mathbf{F}}$ is the determinant of the deformation gradient and for a purely incompressible material it must hold $\hat{J} = 1$.

Using Lamé parameter μ and a pressure-like variable $\hat{p}$ we can introduce the elastic energy functional for a purely incompressible Neo-Hookean material:

$$\Pi_{elast}(\hat{\mathbf{u}}, \hat{p}) = \int_\Omega \left\{ \frac{1}{2}\mu \left[\mathbf{I} : \hat{\mathbf{C}} - d \right] - \mu \ln \hat{J} + \hat{p} \ln(\hat{J}) \right\} d\Omega - \mathcal{F}(\hat{\mathbf{u}}, \gamma). \tag{4}$$

Here $\mathcal{F}(\hat{\mathbf{u}}, \gamma)$ denotes the external energy functional and has the form

$$\mathcal{F}(\hat{\mathbf{u}}, \gamma) = \gamma \left(\int_\Omega \mathbf{f} \cdot \hat{\mathbf{u}} d\Omega + \int_{\Gamma_N} \mathbf{p} \cdot \hat{\mathbf{u}} d\Omega \right), \tag{5}$$

where $\mathbf{f}$ is a body load, $\mathbf{p}$ a traction force acting on the Neumann boundary of the domain $\Gamma_N \subset \delta\Omega$ and $\gamma \in \mathbb{R}$ a load parameter.

For the minimization of the energy functional (4) it follows from the calculus of variations that the total differential of Π_{elast} must be zero,

$$\mathrm{d}\Pi_{elast}(\hat{\mathbf{u}}, \hat{p})[\mathbf{v}, q] = 0,$$

for all generic virtual displacement fields $\mathbf{v}$ and pressure fields q. Explicitly this reads:

$$\begin{cases} \mu \int_\Omega [\hat{\mathbf{F}} - \hat{\mathbf{F}}^{-T}] : \nabla\mathbf{v} + \int_\Omega \hat{p}\, \hat{\mathbf{F}}^{-T} : \nabla\mathbf{v} - \mathcal{F}(\mathbf{v}; \gamma) = 0 \quad \forall \mathbf{v}, \\ \int_\Omega \ln(\hat{J}) q = 0 \quad \forall q. \end{cases} \tag{6}$$

The nonlinear system of equations (6) is then discretized using isogeometric Taylor-Hood elements in order to obtain displacements $\hat{\mathbf{u}} \in \mathcal{V}_h$ and pressures $\hat{p} \in \mathcal{Q}_h$. We also need to evaluate the second derivative of the energy functional (4), in order to setup a Newton's method for our solver:

$$\mathrm{d}^2\Pi_{elast}(\hat{\mathbf{u}}, \hat{p})[(\mathbf{u}, p), (\mathbf{v}, q)] = a_\gamma(\mathbf{u}, \mathbf{v}) + b_\gamma(\mathbf{v}, p) + b_\gamma(\mathbf{u}, q), \tag{7}$$

where

$$\begin{cases} a_\gamma(\mathbf{u},\mathbf{v}) := \mu \int_\Omega \nabla\mathbf{u} : \nabla\mathbf{v} + \int_\Omega [\mu - \hat{p}]\, (\hat{\mathbf{F}}^{-1}\nabla\mathbf{u})^T : \hat{\mathbf{F}}^{-1}\nabla\mathbf{v}, \\ b_\gamma(\mathbf{v},q) := \int_\Omega q\, \hat{\mathbf{F}}^{-T} : \nabla\mathbf{v}. \end{cases} \tag{8}$$

This formulation is a good example of what the authors intend as "sophisticated formulation". If we were to give a precise definition of "sophisticated" in this context, we would define it as an open scale, where grade zero is the classical $\int_\Omega \nabla u \cdot \nabla v$ arising from a Poisson's problem. Of course we do not aim grading every possible operator, but we think that a good measure of complexity is the number of manipulations beyond the classical grad − grad. It is in such cases that `igatools` can help.

In particular we focus on the term

$$\int_\Omega q\, \hat{\mathbf{F}}^{-T} : \nabla\mathbf{v}$$

from (7), which serves as a good example for the difficulties in this formulation. Its implementation is sketched in Listing 3. We first start by noticing that this is a mixed element. As we have two different spaces for $\mathbf{u}$ and p we need to iterate trough both of them and therefor we need two different iterators: the fist one is `defo_elem`, which iterates over the elements of the deformation space, and the second one is `prex_elem`, which iterates over the elements of the pressure like space. The single terms such as $\hat{\mathbf{F}}$, $\nabla\mathbf{v}$, or q are defined as tensors, and we refer to `igatools` documentation of the `Tensor` class for details [25]. Two classical for loops are used to evaluate $\nabla\mathbf{u}$ and p at quadrature points. Then $\hat{\mathbf{F}}^{-T}$ can be easily evaluated combing the inverse and the transpose functions. Once all the variables have been evaluated, they can be combined to obtain the local contribution to the tangent matrix evaluation. The remaining terms of the residual and tangent matrix are assembled in an analogous way. Once all the operators are in place, the setup of a Newton's method is a matter of linear algebra.

For validation of the implementation we chose the well-known Cook's membrane problem as benchmark [12]. This is a well established test in the context of finite element and also isogeometric methods [16, 28]. It is used by a number of researchers to validate solvers and benchmark performance of discretizations. Cook's membrane is a 2D panel, clamped on its side and subject to a shear load on the right side, causing combined bending and shear deformation. Geometry, dimensions and values of parameters can be found in Fig. 3. The quantity of interest is the horizontal deformation of the top right corner of the panel u_y, which is used to study p/k- and h-convergence and locking-free behavior of mixed elements.

In Fig. 4 convergence behavior of u_y with respect to the total number of degrees of freedom is shown. We compare the isogeometric Taylor-Hood elements with pure displacement isogeometric formulations. The Poisson's ratio ν is set to 0.4998

```
for (; defo_elem != defo_end ; ++defo_elem, ++prex_elem)
{
    ...
  for (Index q = 0; q < n_qp; ++q)
  {
    for(Index i = 0; i < defo_loc_ndofs; ++i)
      defo_grad_q += defo_vec(dof) * grad_phi_q[i];

    for(Index i = 0; i < prex_loc_ndofs; ++i)
      prex_q += prex_vec(dof) * prex_phi_q[i];

    defgrad_q = unit_defgrad + defo_grad_q;

    inverse (defgrad_q,defgrad_inv_q);
    defgrad_invT_q = transpose(defgrad_inv_q);

    for (Index i = 0; i < defo_loc_ndofs; ++i)
      defo_loc_res(i) += mat_mu * prex_q[0] *
                  scalar_product(defgrad_invT_q,
                                 grad_phi_q[i]) *w_meas[q];
    }
}
```

Listing 3 Assembly of the mixed term in the second variation of the energy functional

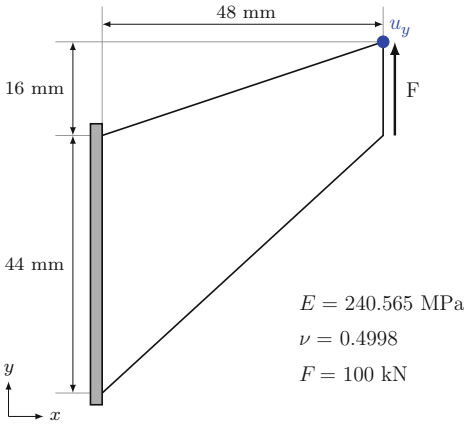


Fig. 3 Cook's membrane problem. A 2D panel clamped on the left side, subject to shear load on right side

that correspond to nearly incompressible material. The displacement-based show a rather slow convergence gradient due to locking. On the other hand, as expected, the Taylor-Hood elements converge quickly for p/k- and h-refinement. These results also match very well with the ones obtained by [16], where an isogeometric $\bar{F}$-method was used instead of a mixed formulation.

Even though the purely displacement formulation also converges towards the correct value of u_y for higher polynomial degree and number of degrees of freedom, the approximation of stresses exhibits spurious oscillations, which are typical for

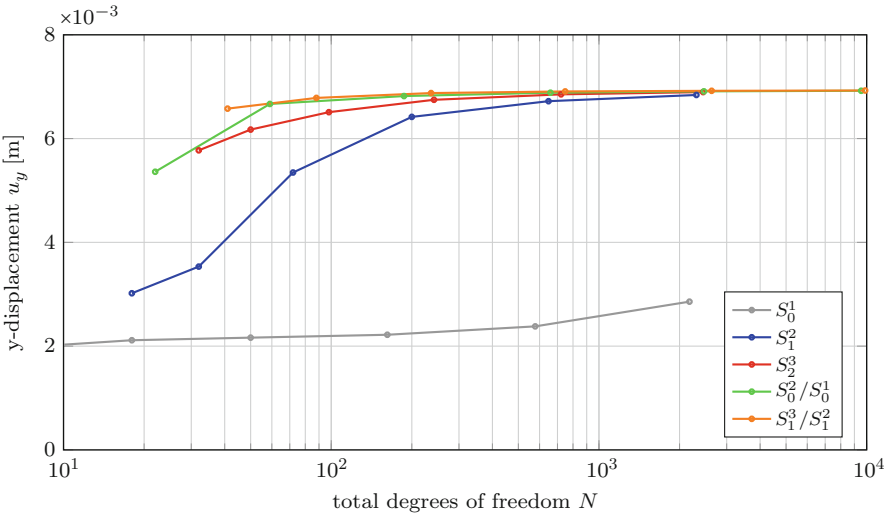


Fig. 4 Convergence test for the Cook's membrane. The reader can notice the fast convergence the Taylor Hood elements and a slower convergence for pure displacements formulation. In the following, the plots of the stresses, will demonstrate that volumetric locking is preventing from a fast convergence

locking phenomena. In Fig. 5 we plot the stress component σ_{xx}. The non mixed elements show a highly oscillatory behavior of volumetric stress components – a typical sign of volumetric locking – while the stress distributions for mixed elements are smooth.

The geometry of a pneumatic pipe, used in aircraft manufacturing, is one of the industrial applications within the TERRIFIC project. Using symmetry boundary conditions we can parametrize only half of the pipe as a B-Spline volume with degrees $d = (1, 2, 2)$ and $n = (2, 34, 3)$ control points, i.e. 204 control points of the volume. The first parameter direction is the thickness of the pipe, the second the circumferential direction and the third its length. For numerical computations we degree-elevate to $d = (2, 2, 2)$ and perform four steps of uniform h-refinement in the length-direction. Thus we have the following Taylor-Hood spaces for pressure and displacements:

$$\begin{aligned} d_p &= (2, 2, 2), & n_p &= (3, 34, 18), & N_p &= 1836, \\ d_{\mathbf{u}} &= (3, 3, 3), & n_{\mathbf{u}} &= (4, 66, 34), & N_{\mathbf{u}} &= 8976. \end{aligned} \tag{9}$$

here n stands for degrees of freedom per direction, while N indicates the total number of degrees of freedom. The material parameters of the pipe are:

$$E = 240.565\,\text{MPa}, \qquad \nu = 0.5, \qquad \rho = 1.0 \cdot 10^3\,\text{kg/m}^3. \tag{10}$$

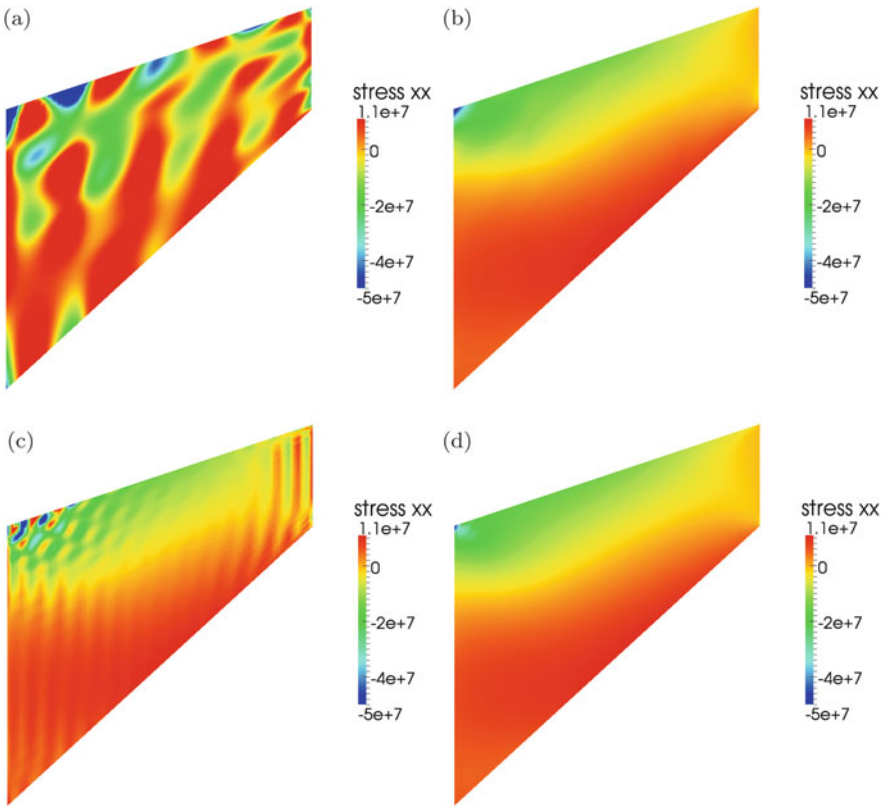


Fig. 5 Stress σ_{xx} for purely displacement-based and mixed formulations. The reader can notice how purely displacement based formulations are affected from volumetric locking, while, as expected, mixed methods are not. (**a**) S_2^3 with 4 elements per side. (**b**) S_1^3/S_1^2 with 4 elements per side. (**c**) S_2^3 with 16 elements per side. (**d**) S_1^3/S_1^2 with 16 elements per side

Boundary conditions are symmetry, zero displacement at long ends and an outwarded-directed Neumann load. We run 10 load steps increasing the magnitude of the surface load from 1.0 to 10.0 kN.

Figure 6 shows the deformed pipe after 10 load steps. Large displacements occur and no volume change is visible in the figure. In Fig. 7 the evolution of displacements u_x, u_y and pressure-like Lagrangian multiplier p, evaluated at the center point where maximum x-displacement and pressure occur, is plotted over the 10 load steps. The nonlinear behaviour can be noticed in the values of the displacements and pressure.

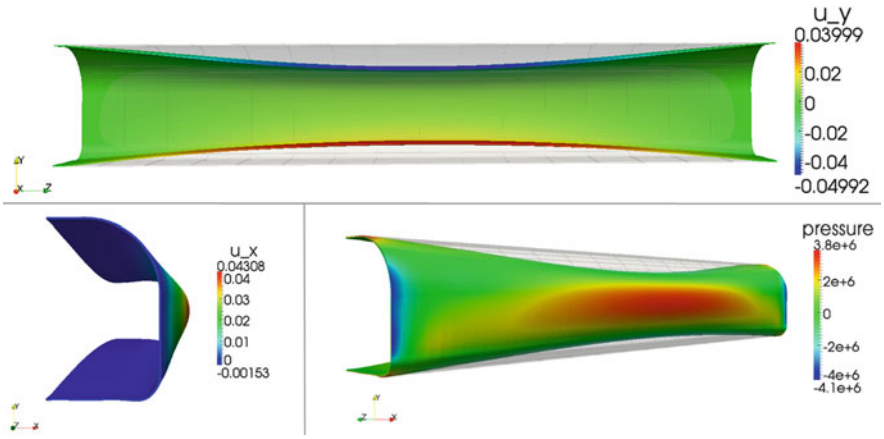


Fig. 6 Displacement and pressure of the pneumatic pipe after 10 load steps. Deformed pipe is colored by displacement resp. pressure, initial un-deformed configuration is shown in translucent grey

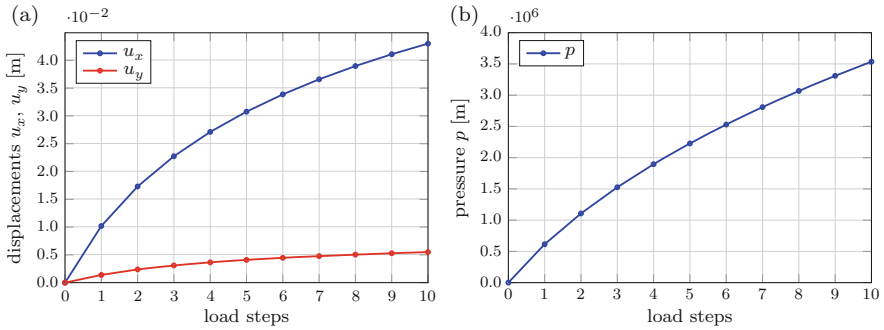


Fig. 7 Evaluation of point values over load steps for pneumatic pipe provided by ALENIA. (**a**) Displacements u_x, u_y. (**b**) Pressure-like Langrangian multiplier p

5 Computational Fluid Dynamics

The laminar motion of an incompressible fluid is modeled by the well known Navier-Stokes equations. We first write the momentum conservation equation:

$$\partial_t \mathbf{u} + (\nabla \mathbf{u})\,\mathbf{u} = -\nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{f},$$

then the mass preservation one:

$$\nabla \cdot \mathbf{u} = 0.$$

Here $\mathbf{u}$ is the fluid velocity, p is the fluid pressure and ν is the kinematic viscosity. The external forces are denoted by $\mathbf{f}$. As a matter of clearness, since it is going to be useful for the method implementation, we explicitly write the component wise formulation of the momentum equation:

$$\partial_t \begin{pmatrix} u_x \\ u_y \end{pmatrix} + \begin{pmatrix} \partial_x u_x & \partial_y u_x \\ \partial_x u_y & \partial_y u_y \end{pmatrix} \begin{pmatrix} u_x \\ u_y \end{pmatrix} = - \begin{pmatrix} \partial_x p \\ \partial_y p \end{pmatrix} + \nu \begin{pmatrix} \partial_{xx} u_x + \partial_{yy} u_x \\ \partial_{xx} u_y + \partial_{yy} u_y \end{pmatrix} + \begin{pmatrix} f_x \\ f_y \end{pmatrix}.$$

In the following, we focus on the stationary version of the problem:

$$(\nabla \mathbf{u})\,\mathbf{u} - \nu \nabla^2 \mathbf{u} + \nabla p = \mathbf{f}, \tag{11}$$

The spatial integration of equation (11) is again performed using the Galerkin approach. Given the computational domain $\Omega \subset \mathbb{R}^{\mathtt{dim}}$, we consider $\mathcal{V}_h \subset (H^1(\Omega))^{\mathtt{dim}}$ the solution space for the velocity, and $\mathcal{Q}_h \subset L^2(\Omega)$ the solution space for the pressure. We consider all the $\mathbf{v}_h \in \mathcal{V}_h$ the test functions for the velocity, then the assembled viscous term $\nabla^2 \mathbf{u}$ correspond to the matrix A such that:

$$A_{ij} = \int_\Omega (\nabla \mathbf{u}_j) : (\nabla \mathbf{v}_i) = \int_\Omega \begin{pmatrix} \partial_x u_x & \partial_y u_x \\ \partial_x u_y & \partial_y u_y \end{pmatrix}_j : \begin{pmatrix} \partial_x v_x & \partial_y v_x \\ \partial_x v_y & \partial_y v_y \end{pmatrix}_i.$$

The convective term is a nonlinear one. For a given value of the velocity field we obtain the matrix N:

$$N_{ij} = \int_\Omega (\nabla \mathbf{u}_j)\,\mathbf{u} \cdot \mathbf{v}_i = \int_\Omega \begin{pmatrix} \partial_x u_x & \partial_y u_x \\ \partial_x u_y & \partial_y u_y \end{pmatrix}_j \begin{pmatrix} u_x \\ u_y \end{pmatrix} \cdot \begin{pmatrix} v_x \\ v_y \end{pmatrix}_i. \tag{12}$$

The operator that couples the momentum equation and the preservation of mass is B^T:

$$B^T_{ij} = \int_\Omega p_j \nabla \cdot \mathbf{v}_i = \int_\Omega p_j (\partial_x v_x + \partial_y v_y)_i.$$

These operators when combined together result in the following block nonlinear system:

$$\begin{pmatrix} A + N(\mathbf{u}) & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ p \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ 0 \end{pmatrix}.$$

The nonlinearity is treated with a classical, residual-tangent strategy. Given an initial guess for the velocity $\mathbf{u}_0$ we first obtain the tentative solution $\mathbf{x}_k = (\mathbf{u}, p)_k$ by solving:

$$\begin{pmatrix} A + N(\mathbf{u}_0) & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ p \end{pmatrix}_k = \begin{pmatrix} \mathbf{f} \\ 0 \end{pmatrix}.$$

For $k > 0$ we need to form the residual $\mathbf{F}(\mathbf{x}_k)$:

$$\mathbf{F}(\mathbf{x}_k) = \begin{pmatrix} A + N(\mathbf{u}_k) & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ p \end{pmatrix}_k - \begin{pmatrix} \mathbf{f} \\ 0 \end{pmatrix},$$

and correct the tentative solution with an approximation of the tangent. The correction at the k-th step will be denoted as $(\delta\mathbf{u}, \delta p)_k$. The most popular strategies to approximate the tangent are the fixed point and the Newton approximation. In the first case we approximate the tangent with the function itself:

$$\begin{pmatrix} A + N(\mathbf{u}_k) & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} \delta\mathbf{u} \\ \delta p \end{pmatrix}_k = -\mathbf{F}(\mathbf{x}_k).$$

In the second case we consider a first order expansion of the operator:

$$\begin{pmatrix} A + N(\mathbf{u}_k) + D(\mathbf{u}_k) & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} \delta\mathbf{u} \\ \delta p \end{pmatrix}_k = -\mathbf{F}(\mathbf{x}_k).$$

In this case we need to assemble the function Jacobian as well:

$$D_{ij} = \int_\Omega (\nabla\mathbf{u})\,\mathbf{u}_j \cdot \mathbf{v}_i = \int_\Omega \begin{pmatrix} \partial_x u_x & \partial_y u_x \\ \partial_x u_y & \partial_y u_y \end{pmatrix} \begin{pmatrix} u_x \\ u_y \end{pmatrix}_j \cdot \begin{pmatrix} v_x \\ v_y \end{pmatrix}_i. \tag{13}$$

Finally, the solution is updated with its correction:

$$\begin{pmatrix} \mathbf{u} \\ p \end{pmatrix}_{k+1} = \begin{pmatrix} \mathbf{u} \\ p \end{pmatrix}_k + \begin{pmatrix} \delta\mathbf{u} \\ \delta p \end{pmatrix}_k,$$

the procedure stops when the residual satisfies a given tolerance.

Now we intend to focus on the two terms involved in the nonlinearity treatment. Looking at the convective term, we have the velocity gradient applied to the fluid velocity. This mathematical formulation has a pretty simple implementation in `igatools`. In Listing 4 we have the gradient of the trial function that acts, through the function `action`, on a point evaluation of the velocity `vel_q`. The result of this `action` is scalar multiplied with the test functions, by using a `scalar_product` function. In this way we obtain the local contribution to the global matrix. Almost the same procedure applies to the evaluation of the operator (13). In this case the quantity that has to be evaluated is the velocity gradient `grad_vel_q`. We evaluate the `action` of the velocity gradient on the trial functions, and then we perform the `scalar_product` with the test functions to get the local contribution to the global matrix, see Listing 5. Both `vel_q` and `grad_vel_q` are considered as tensors. We refer to the `Tensor`'s class documentation for further details.

```
for (dof_index q = 0; q < quad.get_num_points(); q++)
{
   for (dof_index i = 0; i <local_ndofs; i ++){
      auto phi = element->get_value(i,q);
      vel_q = vel_q+vel[local_dofs[i]]*phi;}

   for (dof_index i = 0; i <local_ndofs; i ++){
      for (dof_index j = 0; j <local_ndofs; j ++){
         adv_ij = scalar_product(action(
                                  element->get_gradient(j,q),
                                  vel_q),
                                  element->get_value(i,q) )*
                                  element->get_w_measures()[q];]}}
}
```

Listing 4 Code snippet for the advection operator assembly. In this case the reader can notice the point evaluation of the velocity, the action of the velocity gradient on the velocity value, and the scalar product with the velocity test functions

```
for (dof_index q = 0; q < quad.get_num_points(); q++)
{

   for (dof_index i = 0; i <local_ndofs; i++){
      auto grad_phi = element->get_gradient(i,q);
      grad_vel_q = grad_vel_q + vel[local_dofs[i]] * grad_phi;}

   for (dof_index i = 0; i <local_ndofs; i++){
      for (dof_index j = 0; j <local_ndofs; j++){
         jac_ij = scalar_product(action(grad_vel_q,
                                  element->get_value(j,q)),
                                  element->get_value(i,q) )*
                                  element->get_w_measures()[q];}}
}
```

Listing 5 Code snippet for the Jacobian assembly. In this case we have the point evaluation of the velocity gradient. Then the velocity gradient acts on the trial functions. The result of this multiplication is scalar multiplied with the test functions, to get the local contribution to the global matrix

The solver is tested using the cavity flow test case. This is a typical test case for stationary Navier-Stokes equations [21, 23]. The horizontal velocity u_x is imposed at the top of a square domain. Its value is:

$$u_x(x) = \begin{cases} \sin(\pi x/2a) & \text{if } 0 \le x \le a \\ 1 & \text{if } a \le x \le 1-a \\ \sin(\pi(1-x)/2a) & \text{if } 1-a \le x \le 1 \end{cases}$$

with $a = 1/32$. No-slip boundary conditions are imposed elsewhere. It is known that isogeometric shape functions are not interpolatory to the degrees of freedom. It is

```
class CavityVel : public Function
{
public:
    CavityVel () : Function() {}

    void evaluate(const ValueVector & points,
                  ValueVector & values)       const;
}
...
CavityVel cv;
project_boundary_values(cv,velocity_space,face_q,
direchelet_id,dof_values);
apply_boundary_values(dof_values,matrix,rhs,solution)
```

Listing 6 In this code snippet we present how to impose non constant boundary conditions for our problem. The user needs to create a derived class that represents the function to be evaluated. Implementation for the `evaluate` virtual function has to be provided. `igatools` will take care of L^2 projecting the function on the velocity boundary trace space. The resulting values will be applied to the global matrix, right hand side and solution

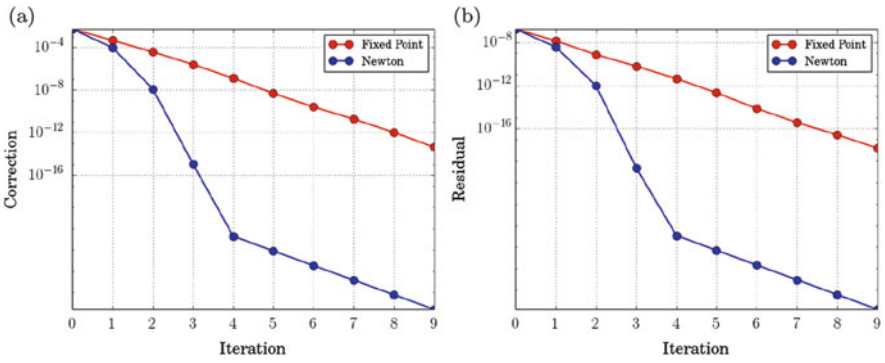


Fig. 8 Correction and residual for the cavity flow test case with Re = 100. (**a**) Solution correction. (**b**) Residual

then necessary to define values for the imposed degrees of freedom. In `igatools`, Listing 6, this is done defining a `CavityVel` class that inherits form the base class `Function` and providing implementation for the virtual function function `evaluate`. At this point the user is let to define an instance of his own class `CavityVel` and `igatools` will take care of evaluating an L^2 projection of the boundary function on the velocity trace space. The resulting `dof_values` will be applied to global matrix, right hand side and solution.

Once the solver is implemented we start with the experiments. In Fig. 8 we represent the residual and the correction for Reynolds's number Re = 100 for each iteration of the nonlinear solver. As expected, the residual and correction drop linearly with the fixed point type of iterations, and quadratically with the Newton's solver.

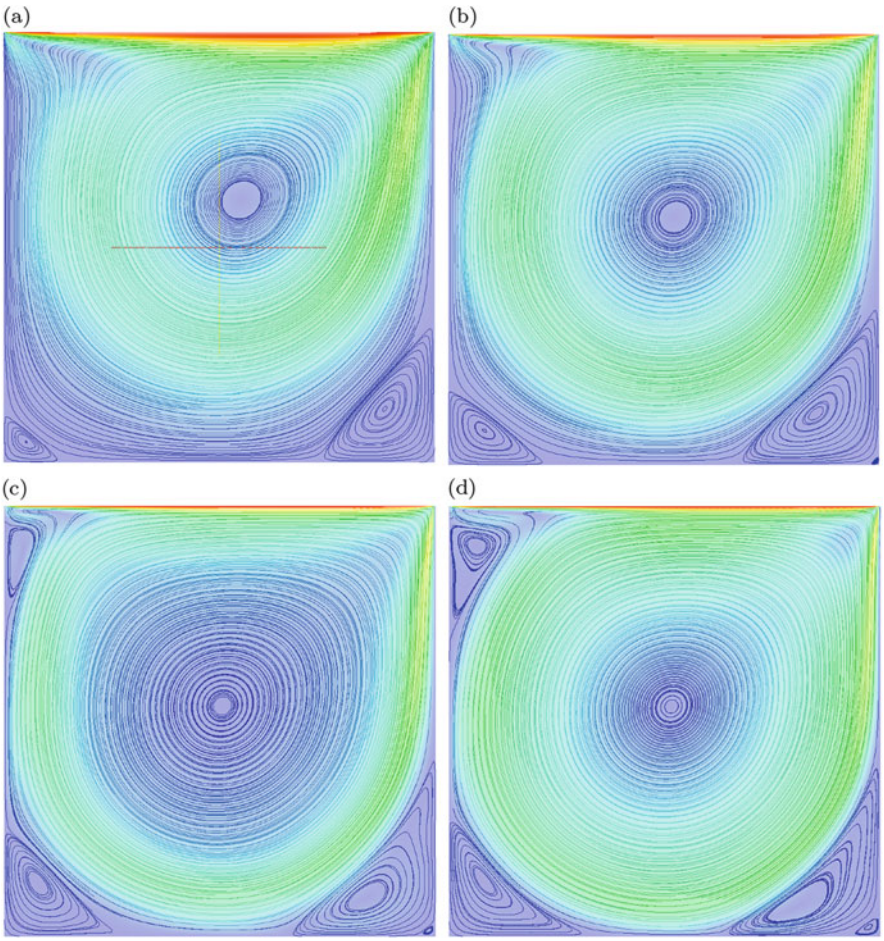


Fig. 9 Cavity flow test case for the Navier-Stokes solver. Colormap represents the velocity magnitude in a scale the ranges from zero to one. (**a**) Re = 400. (**b**) Re = 1000. (**c**) Re = 3200. (**d**) Re = 5000

In the present case of the cavity flow experiment we used S_1^3 splines for the velocity and S_1^2 splines for the pressure with mesh size $h = 1/32$. Efficient linear algebra solvers for this system is still an active research area, the reader can refer to [17, 18] as example references. In this experiment and in the ones that follow case we set for a standard direct solver.

In a second set of experiments we consider increasing Re numbers. The solution space is still S_3^1/S_2^1, but the mesh size is 64×64 elements. In Fig. 9 we present the velocity streamlines. The colormap pictures the velocity magnitude that ranges from zero to one. The agreement with results from [21] is good and the solver can be considered as validated.

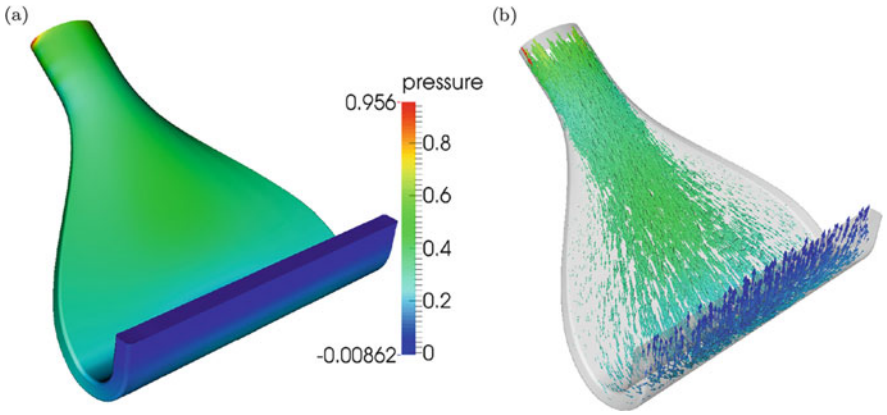


Fig. 10 Pressure map and velocity quiver for the aircraft cabin ventilation outlet. In this simulation Re = 10,000, shape functions are S_1^3/S_1^2, the number of knots, without repetitions, is $17 \times 33 \times 9$. Inlet velocity is imposed, stress free on the outlet, and no slip at the boundary. (**a**) Pressure map. (**b**) Velocity quiver

The industrial test case we are considering is a ventilation outlet of an aircraft cabin, see Fig. 10a for the geometry. Dirichlet inlet boundary conditions and Neumann boundary conditions at the outlet are applied with $\mathrm{Re} = 10,000$. Shape functions spaces are S_1^3 for the velocity and S_1^2 for the pressure. The mesh is $16 \times 32 \times 8$ resulting in approximately 200,000 degrees of freedom for the velocity. The final results are visualized in Fig. 10, where Fig. 10a shows the pressure map and the velocity is represented in Fig. 10b. At this Reynolds number the fluid velocity does not diffuse uniformly along the outlet section and higher velocities are concentrated in the middle of the section.

6 Conclusions

In this paper we showed how `igatools` is an effective tool for assembling isogeometric discretizations of sophisticated operators. `igatools` has been recently released and its design has been presented in [29]. Only time will say if its design will be capable to attract new users and developers. We intended to demonstrate that we are on a good track in this direction. We picked sophisticated operators arising from industrial applications, and showed that a one to one relationship exists between how we write operators on paper, and how we implement them using `igatools`. We believe this attempt has been successful both in computational mechanics and computational fluid dynamics. In order to demonstrate a realistic effectiveness of the software we solved industrially relevant applications.

Acknowledgements The authors are grateful to C. Lovadina for the useful discussion. N. Cavallini and O. Weeger have been supported by the TERRIFIC project, European Community's Seventh Framework Programme, Grant Agreement 284981 Call FP7-2011-NMP-ICT-FoF. In all the experiments in this paper we used linear algebra packages from `deal.II` [2, 3].

References

1. F. Auricchio, L. Beirão da Veiga, C. Lovadina, A. Reali, R.L. Taylor, P. Wriggers, Approximation of incompressible large deformation elastic problems: some unresolved issues. Comput. Mech. **52**(5), 1153–1167 (2013)
2. W. Bangerth, R. Hartmann, G. Kanschat, deal.II – a general purpose object oriented finite element library. ACM Trans. Math. Softw. **33**(4), 24/1–24/27 (2007)
3. W. Bangerth, T. Heister, G. Kanschat, *`deal.II` Differential Equations Analysis Library, Technical Reference*, http://www.dealii.org
4. Y. Bazilevs, L. Beirão da Veiga, J.A. Cottrell, T.J.R. Hughes, G. Sangalli, Isogeometric analysis: approximation, stability and error estimates for h-refined meshes. Math. Models Methods Appl. Sci. **16**(7), 1031–1090 (2006)
5. Y. Bazilevs, V.M. Calo, J.A. Cottrell, J.A. Evans, T.J.R. Hughes, S. Lipton, M.A. Scott, T.W. Sederberg, Isogeometric analysis using T-splines. Comput. Methods Appl. Mech. Eng. **199**(5–8), 229–263 (2010)
6. Y. Bazilevs, C. Michler, V.M. Calo, T.J.R. Hughes, Isogeometric variational multiscale modeling of wall-bounded turbulent flows with weakly enforced boundary conditions on unstretched meshes. Comput. Methods Appl. Mech. Eng. **199**(13–16), 780–790 (2010)
7. P. Becker, Working draft, standard for programming language C++. Technical Report N3242=11-0012, ISO/IEC JTC 1, Information Technology, Subcommittee SC 22, Programming Language C++, Feb 2011
8. D. Boffi, F. Brezzi, M. Fortin, *Mixed Finite Element Methods and Applications*. Springer Series in Computational Mathematics (Springer, London, 2013)
9. D. Boffi, C. Lovadina, Analysis of new augmented lagrangian formulations for mixed finite element schemes. Numerische Mathematik **75**(4), 405–419 (1997)
10. A. Bressan, G. Sangalli, Isogeometric discretizations of the Stokes problem: stability analysis by the macro element technique. IMA J. Numer. Anal. (2012)
11. A. Buffa, D. Cho, G. Sangalli, Linear independence of the T-spline blending functions associated with some particular T-meshes. Comput. Methods Appl. Mech. Eng. **199**(23–24), 1437–1445 (2010)
12. R.D. Cook, Improved two-dimensional finite element. J. Struct. Div. **100**, 1851–1863 (1974)
13. J.A. Cottrell, A. Reali, Y. Bazilevs, T.J.R. Hughes, Isogeometric analysis of structural vibrations. Comput. Methods Appl. Mech. Eng. **195**(41–43), 5257–5296 (2006)
14. T. Dokken, T. Lyche, K.F. Pettersen, Locally refinable splines over box-partitions. Technical report, SINTEF, Feb 2012
15. M. Dörfel, B. Jüttler, B. Simeon, Adaptive isogeometric analysis by local h-refinement with T-splines. Comput. Methods Appl. Mech. Eng. **199**(5–8), 264–275 (2009)
16. T. Elguedj, Y. Bazilevs, V.M. Calo, T.J.R. Hughes, B-bar and f-bar projection methods for nearly incompressible linear and non-linear elasticity and plasticity based on higher-order NURBS elements. Comput. Methods Appl. Mech. Eng. **197**, 2732–2762 (2008)
17. H.C. Elman, D.J. Silvester, A.J. Wathen, Performance and analysis of saddle point preconditioners for the discrete steady-state Navier-Stokes equations. Numer. Math. **90**, 665–688 (2000)
18. H.C. Elman, D.J. Silvester, A.J. Wathen, *Finite Elements and Fast Iterative Solvers: With Applications in Incompressible Fluid Dynamics*. Numerical Mathematics and Scientific Computation (Oxford University Press, Oxford, 2005)

19. J.A. Evans, T.J.R. Hughes, Isogeometric divergence-conforming B-spline for the steady Navier-Stokes equations. Math. Models Methods Appl. Sci. **23**(8), 1421–1478 (2013)
20. D.R. Forsey, R.H. Bartels, Hierarchical B-spline refinement, in *Proceedings of the 15th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH'88)*, Atlanta, 1988, pp. 205–212
21. U. Ghia, K.N. Ghia, C.T. Shin, High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method. J. Comput. Phys. **48**, 387–411 (1982)
22. C. Giannelli, B. Jüttler, H. Speleers, THB–splines: the truncated basis for hierarchical splines. Comput. Aided Geom. D. **29**, 485–498 (2012)
23. R. Glowinski, P.G. Ciarlet, J.L. Lions, *Numerical Methods for Fluids*. Handbook of Numerical Analysis, vol. 3 (Elsevier, Amsterdam, 2002)
24. T.J.R. Hughes, J.A. Cottrell, Y. Bazilevs, Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. Comput. Methods Appl. Mech. Eng. **194**(39–41), 4135–4195 (2005)
25. igatools 0.3.0, An isogeometric analisys tool library – documentation and manual, Oct 2014
26. ISO/IEC 14882:2011 – Information technology – programming languages – C++, 2011
27. J. Kiendl, Y. Bazilevs, M.-C. Hsu, R. Wüchner, K.-U. Bletzinger, Kirchhoff-Love shell structures comprised of multiple patches. Comput. Methods Appl. Mech. Eng. **199**, 2403–2416 (2010)
28. K.M. Mathisen, K.M. Okstad, T. Kvamsdal, S.B. Raknes, Isogeometric analysis of finite deformation nearly incompressible solids. Rakenteiden Mekaniikka (J. Struct. Mech.) **44**(3), 260–278 (2011)
29. M.S. Pauletti, M. Martinelli, N. Cavallini, P. Antolin, Igatools: an isogeometric analysis library. *I.M.A.T.I.-C.N.R.*, 2014, pp. 1–27
30. L.A. Piegl, W. Tiller, *The NURBs Book*. Monographs in Visual Communication Series (Springer, Berlin/New York, 1997)
31. L.L. Schumaker, *Spline Functions: Basic Theory*. Cambridge Mathematical Library, 3rd edn. (Cambridge University Press, Cambridge, 2007)
32. M.A. Scott, X. Li, T.W. Sederberg, T.J.R. Hughes, Local refinement of analysis-suitable T-splines. Comput. Methods Appl. Mech. Eng. **213–216**, 206–222 (2012)
33. R. Taylor, Isogeometric analysis of nearly incompressible solids. Int. J. Numer. Methods Eng. **87**(1–5), 273–288 (2010)
34. A.-V. Vuong, C. Giannelli, B. Jüttler, B. Simeon, A hierarchical approach to adaptive local refinement in isogeometric analysis. Comput. Methods Appl. Mech. Eng. **200**, 3554–3567 (2011)
35. O. Weeger, U. Wever, B. Simeon, Isogeometric analysis of nonlinear Euler Bernoulli beam vibrations. Nonlinear Dyn. **72**(4), 813–835 (2013)
36. P. Wriggers, *Nonlinear Finite Element Methods* (Springer, Berlin, 2008)

Two-Sided Robust and Sharp a Posteriori Error Estimates in Isogeometric Discretization of Elliptic Problems

Stefan K. Kleiss and Satyendra K. Tomar

Abstract We present two-sided a posteriori error estimates for isogeometric discretization of elliptic problems. These estimates, derived on functional grounds, provide robust, guaranteed and sharp two-sided bounds of the exact error in the energy norm. Moreover, since these estimates do not contain any unknown/generic constants, they are fully computable, and thus provide quantitative information on the error. The numerical realization and the quality of the computed error distribution are addressed. The potential of the proposed estimates are illustrated using several computational examples.

1 Introduction

Interchange of geometry information between the engineering design process, e.g., computer aided design (CAD), and simulation/analysis, e.g., using finite element methods (FEM), is generally not only costly, but also prone to approximation errors, and may also require manual input. *Isogeometric analysis* (IGA), introduced by Hughes et al. [22], see also [13] and the references therein, aims at facilitating the transformation of geometry data between these two rather separately evolved directions. In IGA, instead of transforming the geometry data to a conventional FEM representation, the original geometry description is used directly, and the functions representing the geometry (typically non-uniform rational B-splines, abbreviated hereinafter as NURBS) are also used as basis for the discrete solution. This way, the geometry obtained from CAD is not changed, and the need for data transformation is eliminated. Furthermore, the exact representation from the coarsest mesh is preserved throughout the refinement process. IGA has been thoroughly studied and

S.K. Kleiss (✉)
Johannes Kepler University, Altenbergerstrasse 69, 4040 Linz, Austria
e-mail: stefan.kleiss@jku.at; Stefan_Ken.Kleiss@jku.at

S.K. Tomar
Cardiff School of Engineering, Queen's Buildings, The Parade, Cardiff CF24 3AA, UK
e-mail: tomar.sk@iitkalumni.org

B. Jüttler, B. Simeon (eds.), *Isogeometric Analysis and Applications 2014*,
Lecture Notes in Computational Science and Engineering 107,
DOI 10.1007/978-3-319-23315-4_10

its potential has been shown by successful applications to a wide range of problems, see, e.g., [3–5, 7, 9, 12, 18, 23, 30].

Since naive mesh refinement in a tensor-product setting (which is the hallmark of NURBS meshes) has global effects, the development of local refinement strategies for isogeometric analysis is a subject of current active research. Such local refinement techniques include, e.g., T-splines [6, 29, 36–38], truncated hierarchical B-splines (THB-splines) [20, 21], polynomial splines over hierarchical T-meshes (PHT-splines) [15, 40], and locally-refinable splines (LR-splines) [16, 25]. These approaches are mainly guided by the a-priori knowledge of the solution behavior or geometry and does not take in to account any information from the computed solution. To the best of the authors' knowledge, the issue of a-posteriori error estimation in isogeometric analysis is still in an infancy stage, and the only published results are [14, 17, 24, 27, 39–43].

The first (and often only) practical application of a posteriori error estimation is the issue of adaptive local refinement (see, e.g., [1, 35] for a general overview on error estimators). A good a posteriori error estimation method can accurately identify local area(s) where further refinement is needed due to the local error being significantly larger than in the rest of the domain. Thereby, computing resources can be more efficiently used than uniform global distribution.

A posteriori error estimates based on hierarchical bases, proposed by Bank and Smith [2], have been used in [17, 39]. The reliability and efficiency of this approach is subjected to the saturation assumption on the (enlarged) underlying space and the constants in the strengthened Cauchy inequality. As the authors remarked, the first assumption is critical and its validity depends on the considered example. Moreover, an accurate estimation of constants in the strengthened Cauchy inequality requires the solution of generalized minimum eigenvalue problem. As noted in [24, Page 41], this approach delivers *less than satisfactory* results. Residual-based a posteriori error estimates have been used in [24, 40–42]. This approach requires the computation of constants in Clement-type interpolation operators. Such constants are mesh (element) dependent, often generic/unknown or incomputable for general element shape; and the global constant often over-estimates the local constants, and thus the exact error. This fact has been explicitly stated by the authors in [24, Pages 42–43] and in [40, Remark 1]. Goal-oriented error estimation approach has been studied in [14, 27, 43]. The results presented in these studies show that neither the estimates of this approach are *guaranteed* to be an upper bound, nor the efficiency indices of the estimates are sharp. Moreover, this approach also requires the solution of an adjoint problem, the cost of which can not be entirely neglected. The approach of Zienkiewicz-Zhu type a posteriori error estimates is based on post-processing of approximate solutions, and depends on the superconvergence properties of the underlying basis. The superconvergence properties for B-splines (NURBS) functions are not yet well-known. Summarily, in general situations, the reliability and efficiency of these methods often depend on undetermined constants, which is not suitable for quality assurance purposes.

A posteriori error estimates have another important application, namely the issue of *quality assurance*, i.e., to quantify the error in the computed solution with certain degree of *guarantee*. IGA has shown great potential in many real life engineering applications. Therefore, even if in a niche domain of applications where one can obtain excellent a posteriori error estimates, authors believe that the issue of *quality assurance* of the computed solution deserves more attention than it is given today. In [26], authors addressed the issue of *guaranteed, fully computable and sharp* a posteriori error estimates for the isogeometric discretizations of elliptic problems. The estimates presented in [26] are for the upper bound of the error in the energy norm. In this paper we extend this study to cover the lower bound of the error in the energy norm. These *functional-type* a posteriori error estimates, which were introduced in [32, 33], and have been studied for various fields (see [35] and the references therein), provide guaranteed, sharp and fully computable bounds without any generic undetermined constants. These estimates are derived on purely functional grounds (based on integral identities or functional analysis) and are thus applicable to any conforming approximation in the respective space.

For elliptic problems considered in this paper and in [26], the weak solution of the problem lies in $H_0^1(\Omega)$. For such problems, the proposed upper bound of the error involve computing an auxiliary function $y \in H(\Omega, \mathrm{div})$, and the proposed lower bound of the error involve computing an auxiliary function $w \in H_0^1(\Omega)$. To get sharp two-sided estimates, the functions y and w are computed by solving global problems. As briefly explained above, we reiterate that our interest in a posteriori error estimates is not only for adaptivity, but also on quantifying the error in the computed solution (and thus guaranteeing the quality of the computed solution). To the best of authors' knowledge, there is no other, particularly cheaper, method available which can fulfill these objectives in general situations.

Two aspects motivate the application of functional-type error estimates in IGA. Firstly, unlike the standard Lagrange basis functions, NURBS basis functions of degree p are, in general, globally C^{p-1}-continuous. Hence, NURBS basis functions of degree $p \geq 2$ are, in general, at least C^1-continuous, and therefore, their gradients are automatically in $H(\Omega, \mathrm{div})$. Thereby, a sharp upper error bound can be computed without the need of constructing complicated functions in $H(\Omega, \mathrm{div})$, in particular for higher degrees (see, e.g., [8, 10, 11, 19]). Secondly, since the considered problems are solved in an isogeometric setting, an efficient implementation of NURBS basis functions is readily available, which can be used to construct the above mentioned functions y and w. Hence, applying the technique of functional-type a posteriori error estimation in a setting that relies only on the use of already available NURBS basis functions is greatly appealing.

The remainder of this paper is organized as follows. In Sect. 2, we define the model problem, and recall the definition of B-spline and NURBS basis functions. In Sect. 3, we recall functional-type two-sided a posteriori error estimates. Numerical examples are presented in Sect. 4, and some conclusions are drawn in Sect. 5.

2 Notation and Problem Setting

Let $\Omega \subset \mathbb{R}^2$ be an open, bounded and connected Lipschitz domain with boundary $\partial\Omega$. We shall consider the following model problem:

Find the scalar function $u : \overline{\Omega} \to \mathbb{R}$ such that

$$\begin{aligned} -\mathrm{div}(A\nabla u) &= f \qquad \text{in } \Omega, \\ u &= u_D \qquad \text{on } \Gamma_D = \partial\Omega, \end{aligned} \tag{1}$$

where A, f and u_D are given data. We assume that A is a symmetric positive definite matrix and has a positive inverse A^{-1}, and that there exist constants $c_1, c_2 > 0$ such that

$$c_1|\xi|^2 \leq A\xi \cdot \xi \leq c_2|\xi|^2, \quad \forall \xi \in \mathbb{R}^2. \tag{2}$$

Then, the norms

$$\|v\|_A^2 = \int_\Omega Av \cdot v \, dx, \quad \|v\|_{\bar{A}}^2 = \int_\Omega A^{-1}v \cdot v \, dx, \tag{3}$$

are equivalent to the L^2-norm $\|v\|^2 = \int_\Omega v \cdot v \, dx$. The weak form of problem (1) can be written as follows:

Find $u \in V_g$, such that

$$a(u, v) = f(v), \quad \forall v \in V_0, \tag{4}$$

where $V_0 \subset H^1(\Omega)$ contains the functions which vanish on Γ_D, and $V_g \subset H^1(\Omega)$ contains the functions satisfying the Dirichlet boundary conditions $u = u_D$ on Γ_D. We assume that the problem data A, f and u_D are given such that the bilinear form $a(\cdot, \cdot)$ is bounded, symmetric and positive definite, and that $f(\cdot)$ is a bounded linear functional. The energy norm of a function v is given by $\|\nabla v\|_A = \sqrt{a(v, v)}$. Note that we have considered the Dirichlet problem only for the sake of simplicity. Functional-type error estimates can be easily generalized to problems with mixed boundary conditions, see, e.g., [28, 35].

The problem (4) is discretized by choosing a finite-dimensional manifold $V_h \subset V_g$ and looking for an *approximate solution* $u_h \in V_h$. This leads to a linear system of equations of the form

$$\underline{K}_h \underline{u}_h = \underline{f}_h, \tag{5}$$

where $\underline{K}_h$ is the stiffness matrix induced by the bilinear form $a(\cdot, \cdot)$, $\underline{f}_h$ is the load vector, and $\underline{u}_h$ is the coefficient vector of the discrete solution u_h.

For brevity reasons, we recall only the IGA related notations which are used in this paper. For detailed discussions of NURBS basis functions, geometry mappings

and their properties, we refer to, e.g., [13, 22, 31] and the references therein. Let p be a non-negative *degree* and let $s = (s_1, \ldots, s_m)$ be a *knot vector* with $s_i \leq s_{i+1}$ for all i. We consider only *open knot vectors*, i.e., knot vectors s where the multiplicity of a knot is at most p, except for the first and last knot which have multiplicity $p+1$. For simplicity, we assume that $s_1 = 0$ and $s_m = 1$, which can be easily achieved by a suitable scaling. For $i = 1, \ldots, n$, and $n = m - p - 1$, we denote the univariate *B-spline basis functions* by $B^s_{i,p} : (0,1) \to \mathbb{R}$. For open knot vectors, the first and the last basis functions are interpolatory at the first and the last knot, respectively. B-spline basis functions of degree p are, in general, globally C^{p-1}-continuous. In the presence of repeated knots, the continuity reduces according to the multiplicity, i.e., if a knot appears k times, the continuity of a B-spline basis function of degree p at that knot is C^{p-k}. Let $\{B^s_{i,p}\}_{i=1}^{n_1}$ and $\{B^t_{j,q}\}_{j=1}^{n_2}$ be two families of B-spline basis functions defined by the degrees p and q, and the open knot vectors

$$s = (s_1, \ldots, s_{n_1+p+1}),\ t = (t_1, \ldots, t_{n_2+q+1}),$$

respectively. We denote the set of all double-indices (i,j) by

$$\mathcal{I}_R = \{(i,j) :\ i \in \{1, \ldots, n_1\}, j \in \{1, \ldots, n_2\}\}.$$

By $R_{(i,j)}(\xi_1, \xi_2)$, $(i,j) \in \mathcal{I}_R$ we denote the *bivariate NURBS basis functions* constructed from the above-mentioned families of B-spline basis functions and positive *weights* $w_{(i,j)}$, $(i,j) \in \mathcal{I}_R$ (see, e.g., [13, 22] for details). The continuity of the B-spline basis functions is inherited by the NURBS basis functions. Note that B-splines can be seen as a special case of NURBS with all weights being equal to one. Hence, we will not distinguish between these two and we will only use the term *NURBS* in the remainder of the paper.

Associated with the *parameter domain* $\hat{\Omega} = (0,1)^2$, the set of functions

$$\hat{V}_h = \operatorname{span}\{R_{(i,j)},\ (i,j) \in \mathcal{I}_R\},$$

is uniquely determined by the degrees p and q, the knot vectors s and t, and the weights w. Given the set of functions $\hat{V}_h$ and a *control net* of *control points* $P_{(i,j)} \in \mathbb{R}^2$, where $(i,j) \in \mathcal{I}_R$, the two-dimensional *NURBS-surface* $G : \hat{\Omega} \to \Omega$ is defined by

$$G(\xi_1, \xi_2) = \sum_{(i,j) \in \mathcal{I}_R} R_{(i,j)}(\xi_1, \xi_2)\, P_{(i,j)}. \tag{6}$$

We refer to $\Omega = G(\hat{\Omega})$ as the *physical domain*. We assume that the geometry mapping is continuous and bijective (i.e., not self-penetrating), which are natural assumptions for CAD-applications.

In IGA, the isoparametric principle is applied by using the same basis functions for the discrete solution u_h which are used for representing the geometry. For

detailed discussion, we refer the reader to, e.g., [13, 22]. The discrete solution u_h on the physical domain Ω is represented as follows:

$$u_h(x) = \sum_{(i,j)\in\mathcal{I}_R} u_{(i,j)} \left(R_{(i,j)} \circ G^{-1}\right)(x), \tag{7}$$

where $u_{(i,j)} \in \mathbb{R}$ are real-valued coefficients which form the coefficient vector $\underline{u}_h$. The discrete function space is thus defined by

$$V_h = \operatorname{span}\{R_{(i,j)} \circ G^{-1},\ (i,j) \in \mathcal{I}_R\}.$$

The initial mesh, and thereby the basis functions on this initial mesh, are assumed to be given via the geometry representation of the computational domain, i.e., the initial discretization is already determined by the problem domain. The exact representation of the geometry on the initial (coarsest) level is preserved in the process of mesh refinement.

As mentioned in the introduction, the straightforward definition of NURBS basis functions leads to a tensor-product structure of the discretization, which is the focus of this paper. Nevertheless, since the presented error estimators do not exploit any information from the underlying discretization method, the beauty of them is that they are also applicable to local refinement techniques (e.g., T-splines, THB-splines, PHT-splines, LR-splines, see Sect. 1) without any additional work.

3 Guaranteed Two-Sided Bounds of the Error

In this section, we recall the well-known theoretical bounds for the error in the energy norm (see, e.g., [32–35]). For a detailed theoretical exposition the reader is referred to [35].

For an upper bound of the error in the energy norm we have the following estimate.

Theorem 1 *Let C_Ω be the constant in the inequality $\|v\| \le C_\Omega \|\nabla v\|_A$ (Friedrich's type), for all $v \in V_0$, where $\|\cdot\|$ denotes the standard L_2-norm. Let u be the exact solution of the problem (4), and let $u_h \in V_h$ be an approximate solution. Then, the following estimate holds:*

$$\|\nabla u - \nabla u_h\|_A \le \|A\nabla u_h - y\|_{\bar{A}} + C_\Omega \|\operatorname{div} y + f\|, \tag{8}$$

where y is an arbitrary vector-valued function in $H(\Omega, \operatorname{div})$, and the norms are as defined in (3).

The constant C_Ω depends only on the domain Ω and the coefficient matrix A (but not on the underlying mesh), see, e.g., [26, 28, 35]. Note that C_Ω can be computed either numerically or, if one can find a domain $\Omega_\square \supset \Omega$, where $\Omega_\square$ is a square

domain with side-length ℓ, then $C_\Omega \leq c_2 \frac{\ell}{\pi\sqrt{d}}$, where d is the dimension and c_2 is the constant in (2). Therefore, the estimate given in Theorem 1 is fully computable upper bound for any conforming approximation $u_h \in V_g$. Note that, if we choose y via the (unknown) exact solution $y = A\nabla u$, both sides of (8) coincide. Hence, the estimate is sharp in the sense that, for any fixed u_h, we can find a function y such that the upper bound is as close to the exact error as desired.

To obtain a sharp estimate (and not just an indicator), one can find a function y which minimizes the right-hand-side of (8). To achieve this, we first rewrite the estimate in the following form

$$\|\nabla u - \nabla u_h\|_A^2 \leq (1+\beta)\|A\nabla u_h - y\|_A^2 + (1+\tfrac{1}{\beta})C_\Omega^2 \|\operatorname{div} y + f\|^2 \;=:\; M_\oplus^2(y,\beta), \tag{9}$$

where $\beta > 0$ is a free parameter, see e.g. [26, 28, 35]. Note that the upper bound in (9) holds true for *any* fixed $y \in H(\Omega, \operatorname{div})$ and $\beta > 0$. Hereinafter, for simplicity, we will refer to $M_\oplus^2(y,\beta)$ as the *majorant*. Introducing

$$\begin{aligned} a_1 &= 1+\beta, & a_2 &= (1+\tfrac{1}{\beta})C_\Omega^2, \\ B_1 &= \|A\nabla u_h - y\|_A^2, & B_2 &= \|\operatorname{div} y + f\|^2, \end{aligned} \tag{10}$$

we can briefly write the majorant as

$$M_\oplus^2(y,\beta) = a_1 B_1 + a_2 B_2. \tag{11}$$

Therefore, a *sharp* estimate can be obtained by finding $y \in H(\Omega, \operatorname{div})$ and $\beta > 0$ as solutions to the global minimization problem

$$\min_{y \in H(\Omega,\operatorname{div}),\ \beta>0} M_\oplus^2(y,\beta). \tag{12}$$

As explained in detail in [26], to find the minimizing parameter y, we set $M_\oplus^2(y)' = 0$, and solve

$$a_1 \int_\Omega A^{-1} y \cdot \tilde{y}\, dx + a_2 \int_\Omega \operatorname{div} y \operatorname{div} \tilde{y}\, dx = a_1 \int_\Omega \nabla u_h \cdot \tilde{y}\, dx - a_2 \int_\Omega f \operatorname{div} \tilde{y}\, dx \tag{13}$$

for a solution $y_h \in Y_h \subset H(\Omega, \operatorname{div})$. Note that the solution of the linear system of equations (13) incurs non-negligible cost as we need to assemble the matrix on the left hand side and the vector on the right hand side, and then solve the linear system. Minimization with respect to β results in $\beta = C_\Omega \sqrt{\frac{B_2}{B_1}}$, which requires only the evaluation of the integrals, and thus involves negligible cost.

For a lower bound of the error in the energy norm we have the following estimate, which can be found in [35].

Theorem 2 *Let u be the exact solution of the problem (4). Then, the following estimate holds.*

$$M_{\ominus}^2(w) \leq \|\nabla u - \nabla u_h\|_A^2, \tag{14}$$

where the minorant $M_{\ominus}^2(w)$ is defined as

$$M_{\ominus}^2(w) := \int_{\Omega} (2fw - 2A\nabla u_h \cdot \nabla w)dx - \|\nabla w\|_A^2, \tag{15}$$

and where w is an arbitrary function in V_0.

Note that, in contrast to the upper bound $M_{\oplus}^2(y, \beta)$, we have only one free parameter in the lower bound, namely the scalar-valued function w. To maximize $M_{\ominus}^2(w)$ with respect to w, we proceed as follows (see, e.g., [28]). Let $M_{\ominus}^2(w)'$ denote the derivative of $M_{\ominus}^2(w)$ with respect to w. Setting $M_{\ominus}^2(w)' = 0$, we obtain

$$\int_{\Omega} A\nabla w \cdot \nabla \tilde{w} dx = \int_{\Omega} (f\tilde{w} - A\nabla u_h \cdot \nabla \tilde{w})dx. \tag{16}$$

To solve (16), we choose a finite-dimensional subspace W_h of V_0 and search for a solution $w_h \in W_h$.

Together with the minorant $M_{\ominus}^2$ and the majorant $M_{\oplus}^2$, we have a two-sided estimate of the true error in the form

$$M_{\ominus}^2 \leq \|\nabla u - \nabla u_h\|_A^2 \leq M_{\oplus}^2. \tag{17}$$

Thereby, it is possible to provide a guaranteed, fully computable, and as sharp as desired interval containing the true error. To measure the efficiency of the computed lower bound and upper bound, the *efficiency indices* are defined by

$$I_{\ominus} = \frac{M_{\ominus}(w)}{\|\nabla u - \nabla u_h\|_A}, \qquad I_{\oplus} = \frac{M_{\oplus}(y, \beta)}{\|\nabla u - \nabla u_h\|_A}. \tag{18}$$

Clearly, the closer $I_{\ominus}$ or $I_{\oplus}$ is to 1, the better the estimate is.

Remark 1 To obtain sharp bounds it is necessary to choose the spaces Y_h and W_h sufficiently rich such that they have better approximation properties than V_h.

4 Numerical Results

In this section, we present numerical results of several examples with varying complexity. These examples include simple Poisson problem with solution as a sinusoidal function on a unit square domain, also with varying continuity of basis

functions, a problem with non-identity coefficient matrix where the PDE operator coefficients are of exponential form, a problem with curved boundary domain, and a problem with solution having two sharp peaks.

To reflect the associated polynomial degrees in respective dimensions, we will use the notation $\mathcal{S}_h^{p,q}$ for $\hat{V}_h$, which denotes the NURBS function of degrees p and q in the first and second coordinate, respectively, and where the parameter h is the characteristic cell size (non-vanishing knot-span) of the mesh for $\hat{V}_h$. Unless indicated otherwise, we assume that the NURBS functions in $\mathcal{S}_h^{p,q}$ are of C^{p-1}-continuity in the first coordinate, and C^{q-1}-continuity in the second coordinate. Since for $p \geq 2$, we have $(C^{p-2})^d \subset H(\Omega, \mathrm{div})$, we choose the following choice for $\hat{Y}_h$ (which is the *Case 1* studied in detail in [26])

$$\hat{Y}_h = \mathcal{S}_h^{p+1,p+1} \otimes \mathcal{S}_h^{p+1,p+1}. \tag{19}$$

Thereby, we choose a function space $\hat{Y}_h$ on the parameter domain and, analogously to the relation of $\hat{V}_h$ and V_h, we define the function space Y_h by the push-forward

$$Y_h = \hat{Y}_h \circ G^{-1}.$$

Cost efficient computation of the majorant are discussed in [26]. We shall not discuss a cost-efficient computation of the minorant in detail, and only present first results as a proof-of-concept. We consider a straightforward approach where W_h is obtained from V_0 by applying one step of p-refinement (degree elevation while keeping the continuity at the knots). In this setting, the size of the discretized problem for computing w_h is larger than the original problem for u_h. Since this set-up is costlier, the presented results are not computed on very fine meshes.

Example 1 (Sinus function in a unit square) In this numerical example, the computational domain is the unit square $\Omega = (0,1)^2$ and $u_h \in \mathcal{S}_h^{2,2}$, i.e., a piecewise quadratic function in both directions. The coefficient matrix is the identity matrix, i.e., $A = I$, and the exact solution is given by

$$u = \sin(6\pi x)\sin(3\pi y).$$

The right-hand-side f and the (homogeneous) boundary conditions u_D are determined by the prescribed exact solution u.

The computed efficiency indices for this example are presented in Table 1. We see that with successively refined meshes the efficiency indices approaches the academic value of 1, which means that the obtained upper and lower bounds are practically same as the true error. Note that the lack of sharpness of the bounds on coarser meshes is due to the oscillations in the solution of the problem (with 3 full wavelengths in the x direction and 1.5 wavelengths in the y direction).

Example 2 (Reduced continuity of basis functions) We consider the same exact solution and the same physical domain as in Example 1. However, we now use

Table 1 Efficiency indices for Example 1, $\hat{V}_h = \mathcal{S}_h^{2,2}$

$1/h$	$I_{\ominus}$	$I_{\oplus}$
16×16	0.9918	1.7887
32×32	0.9966	1.3762
64×64	0.9990	1.1883
128×128	0.9997	1.0947
256×256	0.9999	1.0476

Table 2 Efficiency indices for Example 2, $\hat{V}_h = \mathcal{S}_h^{4,4}$, with only C^1-continuity at $x = 0.5$ and $y = 0.5$

$1/h$	$I_{\ominus}$	$I_{\oplus}$
16×16	0.9984	2.3040
32×32	0.9974	1.5029
64×64	0.9987	1.2399
128×128	0.9996	1.1188
256×256	0.9999	1.0595

B-splines of degree $p = q = 4$ to represent Ω, and we add a triple knot at the coordinates $x = 0.5$ and $y = 0.5$. The initial knot vectors are thus given by

$$s = t = (0, 0, 0, 0, 0, 0.5, 0.5, 0.5, 1, 1, 1, 1, 1),$$

and the geometry mapping is only C^1-continuous at the coordinate 0.5. This is to mimic a multi-patch situation where the solution has full continuity everywhere except along certain lines with reduced continuity. Note that, in this case, we also reduced the continuity of $\hat{Y}_h$ at $x = 0.5$ and $y = 0.5$ by one order.

The computed efficiency indices for this example are presented in Table 2. We see that the results are qualitatively same as the results of Example 1, and with successively refined meshes the obtained upper and lower bounds are practically same as the true error.

Example 3 (Varying coefficients of PDE operator) Let the matrix A be of the form of

$$\begin{pmatrix} e^{b_{11}x+b_{12}y} & 0 \\ 0 & e^{b_{21}x+b_{22}y} \end{pmatrix},$$

which is positive definite for $b_{ij} \in \mathbb{R}^+$, $i, j = \{1, 2\}$. This will result in the PDE operator to be of the form of

$$e^{b_{11}x+b_{12}y}\frac{\partial^2}{\partial x^2} + e^{b_{21}x+b_{22}y}\frac{\partial^2}{\partial y^2} + b_{11}e^{b_{11}x+b_{12}y}\frac{\partial}{\partial x} + b_{22}e^{b_{21}x+b_{22}y}\frac{\partial}{\partial y}.$$

To have this PDE operator with full generality, we take $b_{11} = 0.1, b_{12} = 0.8, b_{21} = 0.4, b_{22} = 0.7$. With this generality, to have a good comparison of the efficiency indices with the examples considered so far, we again choose the exact solution to

Table 3 Efficiency indices for Example 3, $\hat{V}_h = \mathcal{S}_h^{2,2}$

$1/h$	$I_{\ominus}$	$I_{\oplus}$
16×16	0.9918	5.9954
32×32	0.9966	2.4968
64×64	0.9990	1.7377
128×128	0.9997	1.3699
256×256	0.9999	1.1856

be $u = \sin(6\pi x)\sin(3\pi y)$. The domain and the polynomial space for this example are the same as of Example 1. The right hand side function is accordingly calculated and the solution has homogeneous Dirichlet boundary values.

Note that in this case the constant C_Ω has to be accordingly modified. For the unit square domain, and the matrix A given above, its value is taken as $\dfrac{c_2}{\pi\sqrt{2}}$, where

$$c_2 = \max\{e^{b_{11}x+b_{12}y}, e^{b_{21}x+b_{22}y}\}.$$

The computed efficiency indices for this example are presented in Table 3. We see that the proposed estimator is robust with respect to the non-trivial PDE coefficient matrix A, and its performance is asymptotically similar to the case with the matrix A being identity. Note that some deviation could be attributed to the fact that we used same number of quadrature points for the evaluation of the matrices in both the cases, which is not sufficient when the PDE coefficients are of exponential form.

Example 4 (Domain with curved boundary) Consider the domain of a quarter annulus. In polar coordinates, Ω is defined by $(r, \phi) \in (1, 2) \times (0, \frac{\pi}{2})$. The circular parts of the domain boundary are represented exactly by the NURBS geometry mapping of degree 2, i.e., we have $p = q = 2$. We set $A = I$, and we prescribe the exact solution

$$u = (r-1)(r-2)\phi(\phi - \tfrac{\pi}{2})e^{-\alpha(r\cos\phi - 1)^2}.$$

We test our method with two values of α, namely,

$$\text{Example 4.a:} \quad \alpha = 20, \qquad \text{Example 4.b:} \quad \alpha = 50.$$

In both examples, this function has zero Dirichlet boundary values and a peak at $x = 1$, the sharpness of which is determined by the value of α. The exact solutions are depicted in Fig. 1.

The computed efficiency indices for this example are presented in Table 4. We again observe a similar pattern as in earlier examples with the unit square domain, with sharper peak demanding more resolution. This confirms the robustness of the two-sided bounds with respect to the boundary shape. From Figs. 2 and 3, we see that the marking based on the majorant (9) shows an excellent agreement with the marking based on the true error.

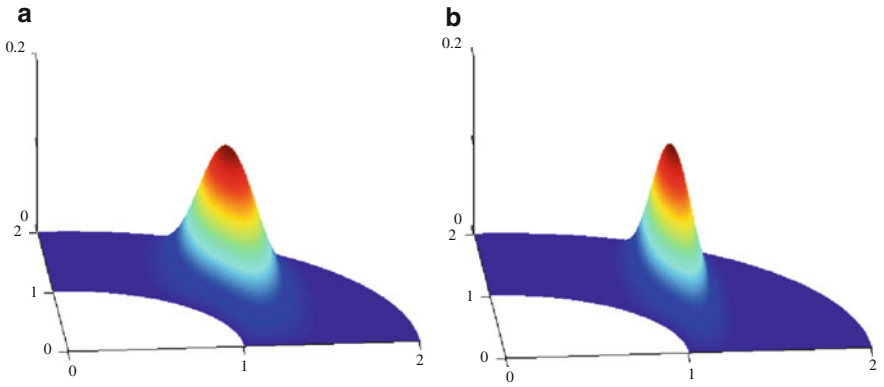


Fig. 1 Exact solutions u on Ω, Example 4. (**a**) Example 4.a ($\alpha = 20$). (**b**) Example 4.b ($\alpha = 50$)

Table 4 Efficiency indices for Example 4, $\hat{V}_h = \mathcal{S}_h^{2,2}$

	$\alpha = 20$		$\alpha = 50$	
$1/h$	$I_{\ominus}$	$I_{\oplus}$	$I_{\ominus}$	$I_{\oplus}$
16×8	0.9922	1.8614	0.9928	3.0624
32×16	0.9953	1.3357	0.9926	1.9711
64×32	0.9985	1.1596	0.9967	1.3923
128×64	0.9996	1.0792	0.9990	1.1891
256×128	0.9999	1.0396	0.9997	1.0930

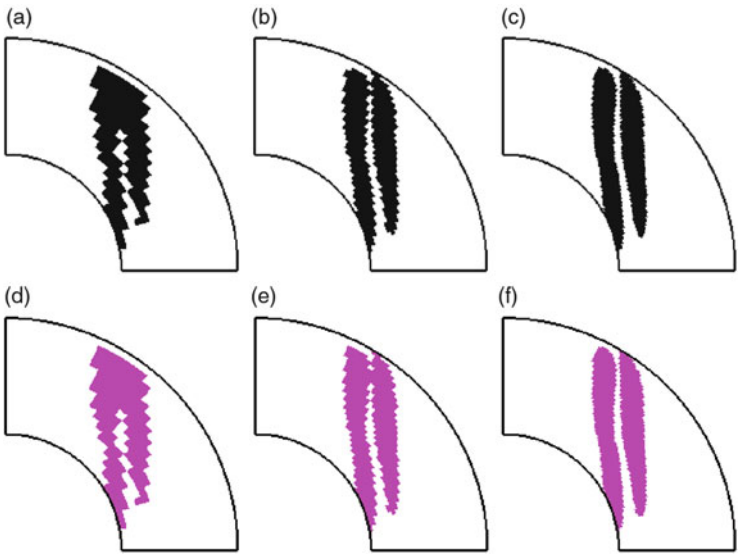


Fig. 2 Marked cells with $\psi = 20\,\%$ in Example 4.a ($\alpha = 20$), $\hat{V}_h = \mathcal{S}_h^{2,2}$. See [26] for details and the definition of Cases. (**a**) Exact, mesh 32×16. (**b**) Exact, mesh 64×32. (**c**) Exact, mesh 128×64. (**d**) Case 1, 32×16. (**e**) Case 2, 64×32. (**f**) Case 3, 128×64

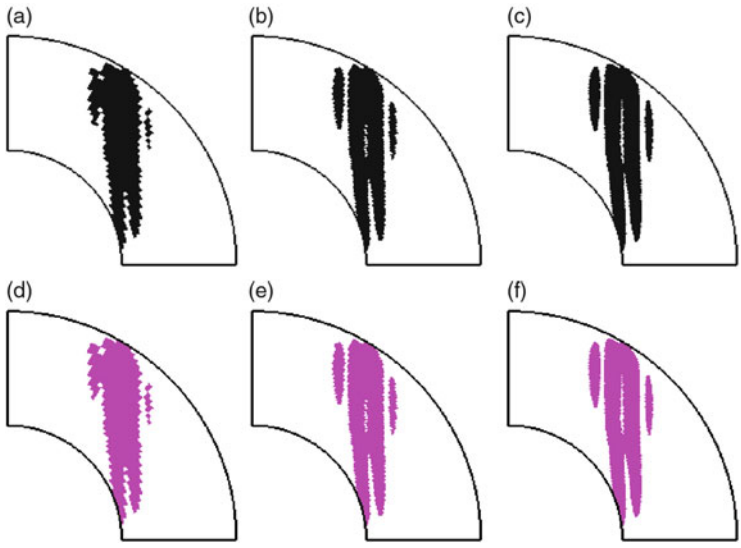


Fig. 3 Marked cells with $\psi = 20\,\%$ in Example 4.b ($\alpha = 50$), $\hat{V}_h = \mathcal{S}_h^{2,2}$. See [26] for details and the definition of Cases. (**a**) Exact, mesh 64×32. (**b**) Exact, mesh 128×64. (**c**) Exact, mesh 256×128. (**d**) Case 1, 64×32. (**e**) Case 2, 128×64. (**f**) Case 3, 256×128

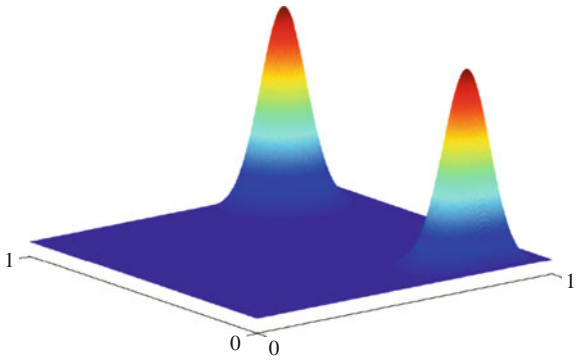


Fig. 4 Exact solution, Example 5

The next example is to test a basic adaptive refinement scheme based on the proposed upper bound.

Example 5 (Adaptive Refinement) The exact solution for this example is given by

$$u = (x^2 - x)(y^2 - y)e^{-100|(x,y)-(0.8,0.05)|^2 - 100|(x,y)-(0.8,0.95)|^2}.$$

The computational domain is again the unit square $\Omega = (0, 1)^2$, and is represented by B-splines of degree $p = q = 2$. The function u, which is illustrated in Fig. 4, has

Table 5 Efficiency indices for Example 5, $\hat{V}_h = \mathcal{S}_h^{2,2}$

$1/h$	$I_{\ominus}$	$I_{\oplus}$
16×16	0.9911	3.9972
25×26	0.9947	2.2161
38×44	0.9971	1.7824
64×74	0.9983	1.5383
92×136	0.9989	1.4599
184×256	0.9991	1.4058

zero Dirichlet boundary values and has two peaks at the coordinates $(0.8, 0.05)$ and $(0.8, 0.95)$.

The computed efficiency indices for this example are presented in Table 5. We see that the estimator is able to nicely direct the refinement process.

5 Conclusions

We have presented two-sided fully computable, guaranteed and sharp a posteriori error estimates in IGA. The computation of the presented estimates relies only on the use of NURBS basis functions, without the need for constructing complicated basis functions in $H(\Omega, \mathrm{div})$. For the computation of these estimates within the context of IGA, two properties of NURBS basis functions are useful. Firstly, the basis functions are, in general, automatically in $H(\Omega, \mathrm{div})$ due to their high continuity. Secondly, increasing the polynomial degree of NURBS basis functions adds only few DOFs. It is important to note that none of these properties are possible in FEM discretizations based on C^0 basis functions. Apart from the topical interest of quality assurance and adaptivity, the presented estimates can also be used in other topics in IGA, such as parametrization of computational domain, e.g., to optimize the placement of inner control points, the proposed estimator can be used to accurately detect the regions with large error and then use the optimization algorithm to reposition the control points. Finally, though in this paper we have only considered tensor-product NURBS discretizations, these estimates can be easily adapted to other type of basis functions as they are valid for *any* conforming approximation.

Acknowledgements The support from the Austrian Science Fund (FWF) through the project P21516-N18 is gratefully acknowledged.

References

1. M. Ainsworth, J.T. Oden, *A Posteriori Error Estimation in Finite Element Analysis* (Wiley Interscience, New York, 2000)

2. R.E. Bank, R.K. Smith A posteriori error estimates based on hierarchical bases. SIAM J. Numer. Anal. **30**(4), 921–935 (1993)
3. Y. Bazilevs, L. Beirão da Veiga, J.A. Cottrell, T.J.R. Hughes, G. Sangalli, Isogeometric analysis: approximation, stability and error estimates for *h*-refined meshes. Math. Models Methods Appl. Sci. **16**(7), 1031–1090 (2006)
4. Y. Bazilevs, V.M. Calo, J.A. Cottrell, T.J.R. Hughes, A. Reali, G. Scovazzi, Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows. Comput. Methods Appl. Mech. Eng. **197**(1–4), 173–201 (2007)
5. Y. Bazilevs, V.M. Calo, T.J.R. Hughes, Y. Zhang, Isogeometric fluid-structure interaction: theory, algorithms, and computations. Comput. Mech. **43**, 3–37 (2008)
6. Y. Bazilevs, V.M. Calo, J.A. Cottrell, J.A. Evans, T.J.R. Hughes, S. Lipton, M.A. Scott, T.W. Sederberg, Isogeometric analysis using T-splines. Comput. Methods Appl. Mech. Eng. **199**(5–8), 229–263 (2010)
7. L. Beirão da Veiga, A. Buffa, J. Rivas, G. Sangalli, Some estimates for *h*- *p*- *k*- refinement in isogeometric analysis. Numer. Math. **118**, 271–305 (2011)
8. D. Boffi, F. Brezzi, M. Fortin, *Mixed Finite Element Methods and Applications* (Springer, Berlin, 2013)
9. A. Buffa, G. Sangalli, R. Vázquez, Isogeometric analysis in electromagnetics: B-splines approximation. Comput. Methods Appl. Mech. Eng. **199**(17–20), 1143–1152 (2010)
10. A. Buffa, C. de Falco, G. Sangalli, IsoGeometric analysis: stable elements for the 2D Stokes equation. Int. J. Numer. Meth. Fluids **65**(11–12), 1407–1422 (2011)
11. A. Buffa, J. Rivas, G. Sangalli, R. Vázquez, Isogeometric discrete differential forms in three dimensions. SIAM J. Numer. Anal. **49**(2), 818–844 (2011)
12. J.A. Cottrell, T.J.R. Hughes, A. Reali, Studies of refinement and continuity in isogeometric structural analysis. Comput. Methods Appl. Mech. Eng. **196**, 4160–4183 (2007)
13. J. Cottrell, T.J.R. Hughes, Y. Bazilevs, *Isogeometric Analysis: Toward Integration of CAD and FEA* (Wiley, Chichester, 2009)
14. L. Dede, H.A.F.A. Santos, B-spline goal oriented error estimators for geometrically nonlinear rods. Comput. Mech. **49**, 35–52 (2012)
15. J. Deng, F. Chen, X. Li, C. Hu, W. Tong, Z. Yang, Y. Feng, Polynomial splines over hierarchical T-meshes. Graph. Models **70**, 76–86 (2008)
16. T. Dokken, T. Lyche, K.F. Pettersen, Polynomial splines over locally refined box-partitions Comput. Aided Geom. Des. **30**(3), 331–356 (2013)
17. M.R. Dörfel, B. Jüttler, B. Simeon, Adaptive isogeometric analysis by local *h*-refinement with T-splines. Comput. Methods Appl. Mech. Eng. **199**(5–8), 264–275 (2010)
18. T. Elguedj, Y. Bazilevs, V.M. Calo, T.J.R. Hughes, $\bar{B}$ and $\bar{F}$ projection methods for nearly incompressible linear and nonlinear elasticity and plasticity using higher-order NURBS elements. Comput. Methods Appl. Mech. Eng. **197**(33–40), 2732–2762 (2008)
19. J.A. Evans, T.J.R. Hughes, Isogeometric divergence-conforming B-splines for the Darcy-Stokes-Brinkman equations. Math. Models Methods Appl. Sci. **23**(04), 671–741 (2013)
20. C. Giannelli, B. Jüttler, H. Speleers, THB-splines: the truncated basis for hierarchical splines. Comput. Aided Geom. Des. **29**(7), 485–498 (2012)
21. C. Giannelli, B. Jüttler, H. Speleers, Strongly stable bases for adaptively refined multilevel splines spaces. Adv. Comput. Math. **40**, 459–490 (2014)
22. T.J.R. Hughes, J. Cottrell, Y. Bazilevs, Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. Comput. Methods Appl. Mech. Eng. **194**(39–41), 4135–4195 (2005)
23. T.J.R. Hughes, A. Reali, G. Sangalli, Efficient quadrature for NURBS-based isogeometric analysis. Comput. Methods Appl. Mech. Eng. **199**(5–8), 301–313 (2010)
24. K.A. Johannessen, An adaptive isogeometric finite element analysis. Master's thesis, Norwegian University of Science and Technology, 2009
25. K.A. Johannessen, T. Kvamsdal, T. Dokken, Isogeometric analysis using LR B-splines. Comput. Methods Appl. Mech. Eng. **269**, 471–514 (2014)

26. S.K. Kleiss, S.K. Tomar, Guaranteed and sharp a posteriori error estimates in isogeometric analysis. Comput. Math. Appl. **70**(3), 167–190 (2015)
27. G. Kuru, C.V. Verhoosel, K.G. van der Zee, E.H. van Brummelen, Goal-oriented isogeometric analysis with hierarchical splines. Comput. Methods Appl. Mech. Eng. **270**, 270–292 (2014)
28. R. Lazarov, S.I. Repin, S.K. Tomar, Functional a posteriori error estimates for discontinuous Galerkin approximations of elliptic problems. Numer. Methods Partial Differ. Equ. **25**(4), 952–971 (2009)
29. X. Li, J. Zheng, T.W. Sederberg, T.J.R. Hughes, M.A. Scott, On linear independence of T-spline blending functions. Comput. Aided Geom. Des. **29**(1), 63–76 (2012)
30. P.N. Nielsen, A.R. Gersborg, J. Gravesen, N.L. Pedersen, Discretizations in isogeometric analysis of Navier-Stokes flow. Comput. Methods Appl. Mech. Eng. **200**(45–46), 3242–3253 (2011)
31. L. Piegl, W. Tiller, *The NURBS Book*. Monographs in Visual Communications, 2nd edn. (Springer, Berlin/Heidelberg, 1997)
32. S.I. Repin, A posteriori error estimation for nonlinear variational problems by duality theory. J. Math. Sci. **99**, 927–935 (2000). See also: Zapiski Nauchnych Seminarov POMI **243**, 201–214 (1997)
33. S.I. Repin, A posteriori error estimation for variational problems with uniformly convex functionals. Math. Comput. **69**(230), 481–500 (2000)
34. S.I. Repin, Two-sided estimates of deviation from exact solutions of uniformly elliptic equations, in *Proceedings of the St. Petersburg Mathematical Society*, vol. IX, ed. by N. Uraltseva. Am. Math. Soc. Transl.: Ser. 2 **209**, 143–171 (2003)
35. S.I. Repin, *A Posteriori Estimates for Partial Differential Equations* (Walter de Gruyter, Berlin, 2008)
36. M.A. Scott, M.J. Borden, C.V. Verhoosel, T.W. Sederberg, T.J.R. Hughes, Isogeometric finite element data structures based on Bézier extraction of T-splines. Int. J. Numer. Meth. Eng. **88**, 126–156 (2011)
37. M.A. Scott, X. Li, T.W. Sederberg, T.J.R. Hughes, Local refinement of analysis-suitable T-splines. Comput. Methods Appl. Mech. Eng. **213–216**, 206–222 (2012)
38. T.W. Sederberg, D.L. Cardon, G.T. Finnigan, N.S. North, J. Zheng, T. Lyche, T-spline simplification and local refinement. ACM Trans. Graph. **23**(3), 276–283 (2004)
39. A.-V. Vuong, C. Giannelli, B. Jüttler, B. Simeon, A hierarchical approach to adaptive local refinement in isogeometric analysis. Comput. Methods Appl. Mech. Eng. **200**(49–52), 3554–3567 (2011)
40. P. Wang, J. Xu, J. Deng, F. Chen, Adaptive isogeometric analysis using rational PHT-splines. Comput.-Aided Des. **43**(11), 1438–1448 (2011)
41. G. Xu, B. Mourrain, R. Duvigneau, A. Galligo, Parameterization of computational domain in isogeometric analysis: methods and comparison. Comput. Methods Appl. Mech. Eng. **200**(23–24), 2021–2031 (2011)
42. G. Xu, B. Mourrain, R. Duvigneau, A. Galligo, Optimal analysis-aware parameterization of computational domain in 3D isogeometric analysis. Comput.-Aided Des. **45**, 812–821 (2013)
43. K.G. van der Zee, C.V. Verhoosel, Isogeometric analysis-based goal oriented error estimation for free boundary problems. Finite Elem. Anal. Des. **47**, 600–609 (2011)

Multilevel Preconditioning for Variational Problems

Angela Kunoth

Abstract For the numerical solution of linear variational problems involving elliptic partial differential operators in $n \geq 2$ space dimensions, iterative solution schemes are indispensable on account of their problem size. Our guiding principle is to devise iterative solvers which are optimal in the number of arithmetic operations, i.e., which are of linear complexity in the total number of unknowns. For these algorithms, asymptotically optimal preconditioners are required. The class of preconditioners for which this can be shown are of multilevel type, requiring nested approximation spaces to approximate the solution of the system on a fine user-specified grid. For smooth solutions of standard second and fourth order elliptic PDEs (partial differential equations) in variational form, approximations based on tensor products of higher-order B-splines yield high accuracy.

For such problem classes, this survey collects the main ingredients for multilevel preconditioners in terms of higher order B-splines. There are three types of multilevel preconditioners for which asymptotic optimality can be shown. Two of them, the so-called additive preconditioners, are specified for isogeometric analysis involving linear elliptic partial differential operators in terms of variants of the BPX (Bramble-Pasciak-Xu) preconditioner and wavelet preconditioners. The third class are the so-called multiplicative preconditioners, specifically, multigrid methods.

An essential ingredient for all these multilevel preconditioners are intergrid operators which transform vectors or matrices between grids of different grid spacing. For higher order B-splines, these intergrid operators can be derived from their refinement relations. In addition to a presentation of the theoretical ingredients, the performance of the different preconditioners will be demonstrated by some numerical examples.

A. Kunoth (✉)
Mathematisches Institut, Universität zu Köln, Weyertal 86–90, 50931 Köln, Germany
e-mail: kunoth@math.uni-koeln.de

B. Jüttler, B. Simeon (eds.), *Isogeometric Analysis and Applications 2014*,
Lecture Notes in Computational Science and Engineering 107,
DOI 10.1007/978-3-319-23315-4_11

1 Introduction

For variational systems involving linear elliptic partial differential equations (PDEs) in $n \geq 2$ space dimensions, a standard finite element or finite difference discretization on a uniform grid with grid spacing $0 < h < 1$ leads to the problem to numerically solve a large ill-conditioned system of linear equations. This is due to the fact that PDE operators have positive order $2r$, i.e., $r = 1$ for second order or $r = 2$ for fourth order operators. Ill-conditioned means that the system matrix $\mathbf{A}_h$ exhibits a spectral condition number $\kappa_2(\mathbf{A}_h)$ which is proportional to h^{-2r}, i.e., $\kappa_2(\mathbf{A}_h) \sim h^{-2r}$. Here and in the following, the relation $a \sim b$ stands for $a \lesssim b$ and $b \lesssim a$ where the latter inequality means that b can be bounded by some constant times a uniformly in all parameters on which a and b may depend. Since the convergence speed of any iterative solution scheme depends on the spectral condition number, the scheme will therefore become prohibitively slow. This effect becomes even worse when h is chosen smaller, in order to obtain more accurate approximations; the number of unknowns N increases like $N \sim h^{-n}$ and, thus, the system size also increases accordingly.

On the other hand, solutions of elliptic PDEs typically exhibit a multiscale behaviour. Enhancing iterative methods by multilevel ingredients, therefore, enables one to achieve much more efficient solution schemes. Ultimately, one strives for an 'optimally efficient scheme'. This means that one can solve the problem with fine grid accuracy with an amount of arithmetic operations that is proportional to the number of unknowns on this grid. The first such methods which were proven to provide an asymptotically optimal iterative scheme were geometric multigrid algorithms [9]. The basic idea of these schemes is to successively solve smaller approximations of the linear system on the finest grid. These can often be interpreted as discretizations with respect to coarser grids. Their iterative solutions can be seen as approximating the inverses of the system matrices applied to the right hand side on the different grids, thereby reducing the spectral condition number of the original system matrix. This idea has, therefore, suggested the term 'preconditioner'. We call a linear operator $\mathbf{C}_h$ an *(asymptotically) optimal preconditioner* if its set-up, storage and application is of optimal linear complexity in the number of unknowns N and if $\kappa_2(\mathbf{C}_h\mathbf{A}_h) \sim 1$ independent of h.

The search for such optimal preconditioners was a major topic for numerical elliptic PDE solvers in the 1980s. The goal was to better understand the ingredients which made a preconditioner optimal. Specifically, one aimed at finding directly applicable versions which could be interpreted as a change of basis. With the arrival of the hierarchical basis preconditioner [54], extending an idea of Babuška from the univariate case, a simple preconditioner became available. Although it is not optimal—one still has logarithmic growth in the grid spacing, $\kappa_2(\mathbf{C}_h\mathbf{A}_h) \sim \log h^{-2r}$, in the bivariate case, and exponential growth for $n = 3$—its simplicity still makes it popular up to now [42]. Another multilevel preconditioner was presented in [3, 4]. Within the last years, (tensor product) hierarchical B-spline discretizations were increasingly employed in the area of isogeometric analysis, mainly in the context

of deriving discretizations which can be locally adapted to singularities, see, e.g., [36, 51], the survey [5] and the references therein.

At the end of the 1980s, another methodology to derive preconditioners via space decompositions and subspace corrections was developed, see the review [52]. The BPX (Bramble-Pasciak-Xu) preconditioner proposed first in [10] was numerically observed to be optimal; it is based on a hierarchical generator system with grid-dependent weights. With techniques from approximation theory, its optimality was theoretically established in [21, 45]. Since then, its range of application has been widened extensively. For second and fourth order elliptic problems on the sphere a BPX-type preconditioner was developed and its optimality proved for triangular finite elements in [43]. The survey article [53] records extensions of the BPX and of multigrid preconditioners to $H(\mathrm{grad})$, $H(\mathrm{curl})$, and $H(\mathrm{div})$ systems on adaptive and unstructured grids; to name just two extensions.

Multigrid preconditioners for isogeometric analysis were presented in [27], whereas domain decomposition type preconditioners were proposed in [6]. Within the class of domain decomposition methods, also tearing and interconnecting methods (FETI-BETI) are important [34]. Note that not all of these preconditioners are or have been proven to be asymptotically optimal.

In this survey, we present the main ideas of the BPX preconditioner from [11] in the context of isogeometric analysis, employing tensor products of higher order B-splines in Sect. 2. We will see that the main theoretical tool to prove optimality of the BPX preconditioner are multilevel characterizations of the underlying energy space, so-called norm equivalences between Sobolev space norms and weighted sequence norms, describing their subspace contributions. We will also see that the main computational ingredients are linear intergrid operators which map vectors and matrices between grids of different grid spacings.

At about the same time, wavelets as a special example of a multiscale orthogonal basis of $L_2(\mathbb{R})$ with compact support were constructed [25]. While initially mainly developed and used for signal analysis and image compression, wavelets were soon discovered to provide optimal preconditioners in the above sense for second order elliptic boundary value problems [21, 33]. However, the fact that one cannot exploit L_2-orthogonality for elliptic boundary value problems together with the difficulty that the L_2-orthogonal Daubechies wavelets are only given implicitly led to the search for variants which are more practical for the numerical solution of PDEs. Soon, it was realized that biorthogonal spline-wavelets as developed in [17] are better suited since they allow one to work with piecewise polynomials for the discretization.

The principal and crucial property to prove optimality of a wavelet preconditioner are norm equivalences between Sobolev norms and sequence norms of weighted wavelet expansion coefficients. I mentioned before that multilevel characterizations for energy spaces are the crucial tool for proving optimality of the BPX preconditioner. Once a wavelet basis is available, one can represent the subspace contributions appearing there explicitly in terms of weighted wavelet coefficients. On this basis, optimal conditioning of the resulting linear system of equations can be achieved by applying the FWT (Fast Wavelet Transform) to a single-scale

discretization on a uniform grid. While again multilevel characterizations of the underlying energy space play the crucial theoretical role for proving optimality, the intergrid operators which perform bases changes between different levels of resolution are the main practical ingredient for the efficiency of the FWT.

Nowadays, the terminology 'wavelets' is used in a more general sense that originally in [25]; we consider classes of multiscale bases with three main features: (R) they provide Riesz bases for the underlying function spaces, (L) the basis functions are local, and (CP) they exhibit cancellation properties. These will be detailed in Sect. 3.

After the initial results concerning optimal preconditioning with wavelets of local support in [21], research on employing wavelets for numerically solving elliptic PDEs went into different directions. One problem was that the original constructions in [17, 25] and many others were based on using the Fourier transform so that these constructions provide bases only for function spaces on all of $\mathbb{R}$, on the torus or, by tensorization, on $\mathbb{R}^n$. In contrast, PDEs naturally live on a bounded domain $\Omega \subset \mathbb{R}^n$. In order for wavelets to be employed for numerical PDEs, there arose the need for constructions of wavelets on bounded intervals and domains without, of course, loosing the crucial properties (R), (L) and (CP). The first such systematic construction of biorthogonal spline-wavelets on $[0, 1]$ of arbitrary order and, by tensor products, on $[0, 1]^n$, was provided in [22]. Different domain decomposition approaches yield constructions of biorthogonal wavelets on domains which can be represented as unions of parametric mappings of $[0, 1]^n$ [13, 23, 24, 40], see also [50] for details. Once such bases are available, the absolute value of the condition numbers of (systems of) elliptic PDEs can be ameliorated significantly by further inexpensive linear transformations taking into account a setup of the system matrices on the coarsest grid, a so-called operator-based preconditioning [12, 46]. A more recent survey on the results of wavelet-based preconditioning with extensions to PDE-constrained control problems can be found in [39].

Aside from optimal preconditioning, the built-in potential of local adaptivity for wavelets plays a prominent role when solving elliptic or parabolic PDEs with non-smooth solutions, on account of the fact that wavelets provide a locally supported Riesz basis for a whole function space. This issue is extensively addressed in the more recent survey paper [48]. In addition to the material in [26], there are at least four extensive surveys on wavelet and multiscale methods for more general PDEs addressing, among other things, the connection between adaptivity and nonlinear approximation and the evaluation of nonlinearities [15, 18–20].

In this article, I want to remain focused on multilevel preconditioning with higher-order discretizations for smooth solutions for which uniform grids provide a user-specified accuracy.

Isogeometric analysis is an the increasingly popular field in which higher order B-Splines are employed to reach this accuracy. Another such area is mathematical finance; specifically, option prizing problems. The fair pricing of an American option can in a standard model be formulated as a parabolic boundary value problem involving Black–Scholes' equation with a free boundary. The aim is to compute the free boundary, the optimal exercise price of the option, together with the solution of

the PDE, the value of the option. For American put options, one generally does not have closed form solutions so that one has to resort to numerical schemes. In the simplest Black–Scholes' model, the volatility is assumed to be constant.

Numerical schemes for American option pricing are typically based on finite difference approaches, see, e.g., [1] and the references therein. In [14] and [44], corresponding multigrid methods are developed. However, here one is not only interested in the solution of the PDE, the option prize. For developing hedging strategies with options, it is of even more importance to accurately compute first and second derivatives of the solution, the so-called Greek letters. The availability of a smooth approximation enables to compute pointwise derivatives and, therefore, avoids additional numerical approximations of derivatives. Therefore, the idea in [30, 31] was to employ higher order B-splines for the solution of the variational inequality derived from Black-Scholes equation for American options. This approach enabled us to pointwise differentiate the option prize and, therefore, achieve a high accuracy approximation for the Greek letters. For the numerical solution, we extended the ideas about monotone multigrid methods with linear finite elements to solve variational inequalities from [35] to higher order B-splines. Here the intergrid operators mentioned before which stem from the refinement relations for B-splines had to be adapted in order not to violate the variational inequality; a task which is for higher order B-splines much more difficult than for linear finite elements since they are not interpolatory any more. Specific care has to be taken when deriving these intergrid operators in order not to violate the inequality constraints. The key feature was to replace function values by B-spline expansion coefficients which remain of the same sign because of the positivity of B-splines. Naturally, these ideas can be extended to obstacle problems which lead to variational inequalities and for which higher order approximations are sought.

While these results were very satisfying from a numerical point of view, the underlying Black-Scholes model has a severe deficiency in assuming that the volatility is constant. Particularly, an effect called volatility smile was observed in [32]. There are several approaches to estimate the volatility from observed stock data. Therefore, we adopted Heston's approach [28] to model the volatility to satisfy a stochastic differential equation. For Heston's model and one asset, the standard Itō approach yields a variational inequality in two space variables with a 'mildly' nonsymmetric parabolic differential operator. The ideas presented in [41] was as follows. A variational inequality for the American option pricing problem with Heston's model was discretized in terms of linear finite elements with respect to space. The resulting linear inequality system was solved in each time step with optimal linear computational complexity using a projective Gauss-Seidel scheme together with a monotone multigrid method. For this, again appropriate intergrid operators were constructed. Unfortunately, due to page limitations, I am not able to present these approaches here and have to refer to [30, 31, 41].

Once optimal preconditioners are available, for any of the problems described above, one can construct for a standard elliptic PDE optimal iterative schemes to achieve discretization error accuracy on the finest level with optimal linear complexity as follows. The idea, the so-called *nested iteration*, has been employed

together with multigrid methods for a long time: starting on a coarse grid, one iteratively solves the preconditioned linear system of equations up to discretization error accuracy on this level. The resulting solution is transformed to the next finer grid, employing again the linear intergrid operator from the coarse to the finer grid. In this way, only a fixed number of iterations are necessary on each grid, independent of that grid spacing, see, e.g., [39]. One should note that, in contrast, iterating only on the finest grid introduces an additional $\log N$ factor, where N denotes the amount of unknowns on the finest grid.

The structure of this survey paper is as follows. Section 2 is devoted to the description of the BPX preconditioner for isogeometric analysis together with the proof of its optimality and corresponding numerical results. Section 3 is concerned with wavelet approximations of solutions of PDEs, and the derivation of Fast Wavelet Transforms for optimal preconditioning. We conclude in Sect. 4 with a short summary and some outlook.

2 BPX Preconditioning for Isogeometric Analysis

In this section, we consider linear elliptic PDEs in the framework of isogeometric analysis, combining modern techniques from computer aided design with higher order approximations of the solution. We treat the physical domain by means of a regular B-spline mapping from the parametric domain $\hat{\Omega} = (0,1)^n$, $n \geq 2$, to the physical domain Ω. The numerical solution of the PDE is computed by means of tensor product B-splines mapped onto the physical domain. We will construct additive BPX-type multilevel preconditioners and show that they are asymptotically optimal. This means that the spectral condition number of the resulting preconditioned stiffness matrix is independent of the grid spacing h. Together with a nested iteration scheme, this enables an iterative solution scheme of optimal linear complexity. The theoretical results are substantiated by numerical examples in two and three space dimensions. The results of this section are essentially contained in [11].

We consider linear elliptic partial differential operators of order $2r = 2, 4$ on the domain Ω in variational form: for given $f \in H^{-r}(\Omega)$, find $u \in H_0^r(\Omega)$ such that

$$a(u,v) = \langle f, v\rangle \quad \text{for all } v \in H_0^r(\Omega) \tag{1}$$

holds. Here the energy space is $H_0^r(\Omega)$, a subset of the Sobolev space $H^r(\Omega)$, the space of square integrable functions with square integrable derivatives up to order r, containing homogeneous Dirichlet boundary conditions for $r = 1$ and homogeneous Dirichlet and Neumann derivatives for $r = 2$. The bilinear form $a(\cdot,\cdot)$ is derived from the linear elliptic PDE operator in a standard fashion, see, e.g., [8]. For example, the Laplacian is represented as $a(v,w) = \int_\Omega \nabla v \cdot \nabla w \, dx$. In order for the problem to be well-posed, we require the bilinear form $a(\cdot,\cdot) : H_0^r(\Omega) \times H_0^r(\Omega) \to \mathbb{R}$ to be symmetric, continuous and coercive on $H_0^r(\Omega)$. With $\langle\cdot,\cdot\rangle$, we denote on the right hand side of (1) the dual form between $H^{-r}(\Omega)$ and $H_0^r(\Omega)$. Our

model problem (1) covers the second order Laplacian with homogeneous boundary conditions

$$-\Delta u = f \quad \text{on } \Omega, \qquad u|_{\partial\Omega} = 0, \tag{2}$$

as well as fourth order problems with corresponding homogeneous Dirichlet boundary conditions,

$$\Delta^2 u = f \quad \text{on } \Omega, \qquad u|_{\partial\Omega} = \mathbf{n} \cdot \nabla u|_{\partial\Omega} = 0 \tag{3}$$

where $\partial\Omega$ denotes the boundary of Ω and $\mathbf{n}$ the outward normal derivative at $\partial\Omega$. These PDEs serve as prototypes for more involved PDEs like Maxwell's equation or PDEs for linear and nonlinear elasticity. The reason we formulate these model problems of order $2r$ involving the parameter r is that this exhibits more clearly the order of the operator and the scaling in the subsequently used characterization of Sobolev spaces $H^r(\Omega)$. Thus, for the remainder of this section, the parameter $2r$ denoting the order of the PDE operator is fixed.

The assumptions on the bilinear form $a(\cdot,\cdot)$ entail that there exist constants $0 < c_A \leq C_A < \infty$ such that the induced self-adjoint operator $\langle Av, w\rangle := a(v,w)$ satisfies the isomorphism relation

$$c_A \|v\|_{H^r(\Omega)} \leq \|Av\|_{H^{-r}(\Omega)} \leq C_A \|v\|_{H^r(\Omega)}, \quad v \in H_0^r(\Omega). \tag{4}$$

If the precise format of the constants in (4) does not matter, we abbreviate this relation as $\|v\|_{H^r(\Omega)} \lesssim \|Av\|_{H^{-r}(\Omega)} \lesssim \|v\|_{H^r(\Omega)}$, or shortly as

$$\|Av\|_{H^{-r}(\Omega)} \sim \|v\|_{H^r(\Omega)}. \tag{5}$$

Under these conditions, Lax-Milgram's theorem guarantees that, for any given $f \in H^{-r}(\Omega)$, the operator equation derived from (1)

$$Au = f \quad \text{in } H^{-r}(\Omega) \tag{6}$$

has a unique solution $u \in H_0^r(\Omega)$, see, e.g., [8].

In order to approximate the solution of (1) or (6), we choose a finite-dimensional subspace $V_h \subset H_0^r(\Omega)$. We will construct these approximation spaces by using tensor products of B-splines as specified next.

2.1 B-Spline Discretizations

Our construction of optimal multilevel preconditioners will rely on tensor products so that principally any space dimension $n \in \mathbb{N}$ is permissible as long as storage permits; the examples cover the cases $n = 2, 3$. As discretization space, we choose

in each spatial direction B-splines of the same degree p on uniform grids and with maximal smoothness. We begin with the univariate case and define B-splines on the interval $[0, 1]$ recursively with respect to their degree p. Given this positive integer p and some $m \in \mathbb{N}$, we call $\Xi := \{\xi_1, \ldots, \xi_{m+p+1}\}$ a p-open knot vector if the knots are chosen such that

$$0 = \xi_1 = \ldots = \xi_{p+1} < \xi_{p+2} < \ldots < \xi_m < \xi_{m+1} = \ldots = \xi_{m+p+1} = 1, \tag{7}$$

i.e., the boundary knots 0 and 1 have multiplicity $p + 1$ and the interior knots are single. For Ξ, B-spline functions of degree p are defined following the well-known Cox-de Boor recursive formula, see [7]. Starting point are the piecewise constants for $p = 0$ (or characteristic functions)

$$N_{i,0}(\zeta) = \begin{cases} 1, & \text{if } 0 \le \xi_i \le \zeta < \xi_{i+1} < 1, \\ 0, & \text{otherwise,} \end{cases} \tag{8}$$

with the modification that the last B-spline $N_{m,0}$ is defined also for $\zeta = 1$. For $p \ge 1$ the B-splines are defined as

$$N_{i,p}(\zeta) = \frac{\zeta - \xi_i}{\xi_{i+p} - \xi_i} N_{i,p-1}(\zeta) + \frac{\xi_{i+p+1} - \zeta}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\zeta), \quad \zeta \in [0, 1], \tag{9}$$

with the same modification for $N_{m,p}$. Alternatively, one can define the B-splines explicitly by applying divided differences to truncated powers [7]. This gives a set of m B-splines that form a basis for the space of *splines*, that is, piecewise polynomials of degree p with $p - 1$ continuous derivatives at the internal knots ξ_ℓ for $\ell = p + 2, \ldots, m$. (Of course, one can also define B-splines on a knot sequence with multiple internal knots which entails that the spline space is not of maximal smoothness.) For $p = 1$, the B-splines are at least $C^0([0, 1])$ which suffices for the discretization of elliptic PDEs of order 2, and for $p = 2$ they are $C^1([0, 1])$ which suffices for $r = 2$. By construction, the B-spline $N_{i,p}$ is supported in the interval $[\xi_i, \xi_{i+p+1}]$.

These definitions are valid for an arbitrary spacing of knots in Ξ (7). Recall that smooth solutions of elliptic PDEs can be approximated best with discretizations on a uniform grid. Therefore, we assume from now on that the grid is *uniform*, i.e., $\xi_{\ell+1} - \xi_\ell = h$ for all $\ell = p + 1, \ldots, m$.

For n space dimensions, we employ tensor products of the one-dimensional B-splines. We take in each space dimension a p-open knot vector Ξ and define on the closure of the parametric domain $\hat{\Omega} = (0, 1)^n$ (which we also denote by $\hat{\Omega}$ for simplicity of presentation) the spline space

$$\begin{aligned} S_h(\hat{\Omega}) &:= \operatorname{span} \left\{ B_i(\mathbf{x}) := \prod_{\ell=1}^{n} N_{i_\ell,p}(x_\ell),\ i = 1, \ldots, N := mn,\ \mathbf{x} \in \hat{\Omega} \right\} \\ &=: \operatorname{span} \left\{ B_i(\mathbf{x}), i \in \mathbb{I},\ \mathbf{x} \in \hat{\Omega} \right\}. \end{aligned} \tag{10}$$

In the spirit of isogeometric analysis, we suppose that the computational domain Ω can also described in terms of B-splines. We assume that the computational domain Ω is the image of a mapping $\mathbf{F} : \hat{\Omega} \to \Omega$ with $\mathbf{F} := (F_1, \ldots, F_n)^T$ where each component F_i of $\mathbf{F}$ belongs to $S_{\bar{h}}(\hat{\Omega})$ for some given $\bar{h}$. In many applications, the geometry can be described in terms of a very coarse mesh, namely, $\bar{h} \gg h$. Moreover, we suppose that $\mathbf{F}$ is invertible and satisfies

$$\|D^\alpha \mathbf{F}\|_{L_\infty(\hat{\Omega})} \sim 1 \quad \text{for} \quad |\alpha| \le r. \tag{11}$$

This assumption on the geometry can be weakened in that the mapping $\mathbf{F}$ can be a piecewise C^∞ function on the mesh with respect to $\bar{h}$, independent of h, or the domain Ω may have a multi-patch representation. This means that one can allow Ω also to be the union of domains Ω_k where each one parametrized by a spline mapping of the parametric domain $\hat{\Omega}$.

We now define the approximation space for (6) as

$$V_h^r := \{v_h \in H_0^r(\Omega) : \ v_h \circ \mathbf{F} \in S_h(\hat{\Omega})\}. \tag{12}$$

We will formulate three important properties of this approximation space which will play a crucial role later for the construction of the BPX-type preconditioners. The first one is that we suppose from now on that the B-spline basis is *normalized* with respect to L_2, i.e.,

$$\|B_i\|_{L_2(\hat{\Omega})} \sim 1, \ \text{and, thus, also} \ \|B_i \circ \mathbf{F}^{-1}\|_{L_2(\Omega)} \sim 1 \ \text{for all} \ i \in \mathbb{I}. \tag{13}$$

Then one can derive the following facts [11].

Theorem 1 *Let $\{B_i\}_{i\in\mathbb{I}}$ be the B-spline basis defined in (10) and normalized as in (13), $N = \#\mathbb{I}$ and V_h^r as in (12). Then we have*

(S) Uniform stability with respect to $L_2(\Omega)$
For any $\mathbf{c} \in \ell_2(\mathbb{I})$,

$$\left\| \sum_{i=1}^N c_i \, B_i \circ \mathbf{F}^{-1} \right\|_{L_2(\Omega)}^2 \sim \sum_{i=1}^N |c_i|^2 =: \|\mathbf{c}\|_{\ell_2}^2, \qquad \mathbf{c} := (c_i)_{i=1,\ldots,N}; \tag{14}$$

(J) Direct or Jackson estimates

$$\inf_{v_h \in V_h^r} \|v - v_h\|_{L_2(\Omega)} \lesssim h^s \, |v|_{H^s(\Omega)} \ \textit{for any} \ v \in H^s(\Omega), \ 0 \le s \le r, \tag{15}$$

where $|\cdot|_{H^s(\Omega)}$ denotes the Sobolev seminorm of highest weak derivatives s;

(B) Inverse or Bernstein estimates

$$\|v_h\|_{H^s(\Omega)} \lesssim h^{-s} \|v_h\|_{L_2(\Omega)} \ \textit{for any} \ v_h \in V_h^r \ \textit{and} \ 0 \le s \le r. \tag{16}$$

In all these estimates, the constants are independent of h but may depend on **F***, i.e., Ω, the polynomial degree p and the spatial dimension n.*

In the next section, we construct BPX-type preconditioners for (6) in terms of approximations with (12) and show their optimality.

2.2 Additive Multilevel Preconditioners

The construction of optimal preconditioners are based on a *multiresolution analysis* of the underlying energy function space $H_0^r(\Omega)$. As before, $2r \in \{2, 4\}$ stands for the order of the PDEs we are solving, and is always kept fixed.

We first describe the necessary ingredients within an abstract basis-free framework, see, e.g., [18]. Afterwards, we specify the realization for the parametrized tensor product spaces in (12).

Let $\mathcal{V}$ be a sequence of strictly nested spaces V_j, starting with some fixed coarsest index $j_0 > 0$, determined by the polynomial degree p which determines the support of the basis functions (which also depends on Ω), and terminating with a highest resolution level J,

$$V_{j_0} \subset V_{j_0+1} \subset \cdots \subset V_j \subset \cdots \subset V_J \subset H_0^r(\Omega). \tag{17}$$

The index j denotes the level of resolution defining approximations on a grid with dyadic grid spacing $h = 2^{-j}$, i.e., we use from now on the notation V_j instead of V_h to indicate different grid spacings. Then, V_J will be the space relative to the finest grid 2^{-J}. We associate with $\mathcal{V}$ a sequence of linear projectors $\mathcal{P} := \{P_j\}_{j \geq j_0}$ with the following properties.

Properties 2 *We assume that*

(P1) P_j *maps* $H_0^r(\Omega)$ *onto* V_j,
(P2) $P_j P_\ell = P_j$ *for* $j \leq \ell$,
(P3) $\mathcal{P}$ *is uniformly bounded on* $L_2(\Omega)$, *i.e.,* $\|P_j\|_{L_2(\Omega)} \lesssim 1$ *for any* $j \geq j_0$ *with a constant independent of j.*

These conditions are satisfied, for example, for $L_2(\Omega)$-orthogonal projectors, or, in the case of splines, for the quasi-interpolant proposed and analyzed in [47, Chapter 4]. The second condition (P2) ensures that the differences $P_j - P_{j-1}$ are also projectors for any $j > j_0$. We define next a sequence $\mathcal{W} := \{W_j\}_{j \geq j_0}$ of complement spaces

$$W_j := (P_{j+1} - P_j) V_{j+1} \tag{18}$$

which then yields the direct (but not necessarily orthogonal) decomposition

$$V_{j+1} = V_j \oplus W_j. \tag{19}$$

Thus, for the finest level J, we can express V_J in its *multilevel decomposition*

$$V_J = \bigoplus_{j=j_0-1}^{J-1} W_j \tag{20}$$

upon setting $W_{j_0-1} := V_{j_0}$. Setting also $P_{j_0-1} := 0$, the corresponding multilevel representation of any $v \in V_J$ is then

$$v = \sum_{j=j_0}^{J} (P_j - P_{j-1})v. \tag{21}$$

We now have the following result which will be used later for the proof of the optimality of the multilevel preconditioners.

Theorem 3 *Let $\mathcal{P}, \mathcal{V}$ be as above where, in addition, we require that for each V_j, $j_0 \le j \le J$, a Jackson and Bernstein estimate as in Theorem 1 (J) and (B) hold with $h = 2^{-j}$. Then one has the function space characterization*

$$\|v\|_{H^r(\Omega)} \sim \left(\sum_{j=j_0}^{J} 2^{2rj} \|(P_j - P_{j-1})v\|^2_{L_2(\Omega)} \right)^{1/2} \quad \text{for any } v \in V_J. \tag{22}$$

Such a result holds for much larger classes of function spaces, Sobolev or even Besov spaces which are subsets of $L_q(\Omega)$ for general q, possibly different from 2 and for any function $v \in H^r(\Omega)$, then with an infinite sum on the right hand side, see, e.g., [18]. The proof of Theorem 3 for such cases heavily relies on tools from approximation theory and can be found in [21, 45].

Next we demonstrate how to exploit the norm equivalence (22) in the construction of an optimal multilevel preconditioner. Define for any $v, w \in V_J$ the linear self-adjoint positive-definite operator $C_J : V_J \to V_J$ given by

$$(C_J^{-1} v, w)_{L_2(\Omega)} := \sum_{j=j_0}^{J} 2^{2rj} \left((P_j - P_{j-1})v, (P_j - P_{j-1})w \right)_{L_2(\Omega)}, \tag{23}$$

which we call a multilevel *BPX-type preconditioner*. Let $A_J : V_J \to V_J$ be the finite-dimensional operator defined by $(A_J v, w)_{L_2(\Omega)} := a(v, w)$ for all $v, w \in V_J$, the approximation of A in (6) with respect to V_J.

Theorem 4 *With the same prerequisites as in Theorem 3, C_J is an asymptotically optimal symmetric preconditioner for A_J, i.e., $\kappa_2(C_J^{1/2} A_J C_J^{1/2}) \sim 1$ with constants independent of J.*

Proof For the parametric domain $\hat{\Omega}$, the result was proved independently in [21, 45] and is based on the combination of (22) together with the well-posedness of the

continuous problem (6). The result on the physical domain follows then together with (11). □

Concrete realizations of the preconditioner defined in (23) based on B-splines lead to representations of the complement spaces W_j whose bases are called *wavelets*. For these, efficient implementations of optimal linear complexity involving the Fast Wavelet Transform can be derived explicitly, see Sect. 3.

However, since the order of the PDE operator r is positive, one can use here the argumentation from [10] which will allow to work with the same basis functions as for the spaces V_j. The first part of the argument relies on the assumption that the P_j are L_2-*orthogonal* projectors. For a clear distinction, we shall use the notation O_j for L_2-orthogonal projectors and reserve the notation P_j for the linear projectors with Properties 2. Then, the BPX-type preconditioner (23) (using the same symbol C_J for simplicity) reads as

$$C_J^{-1} := \sum_{j=j_0}^{J} 2^{2jr}(O_j - O_{j-1}), \tag{24}$$

which is by Theorem 4 a BPX-type preconditioner for the self-adjoint positive definite operator A_J. By the orthogonality of the projectors O_j, we can immediately derive from (24) that

$$C_J = \sum_{j=j_0}^{J} 2^{-2jr}(O_j - O_{j-1}). \tag{25}$$

Since $r > 0$, by rearranging the sum, the exponentially decaying scaling factors allow one to replace C_J by the spectrally equivalent operator

$$C_J = \sum_{j=j_0}^{J} 2^{-2jr} O_j. \tag{26}$$

Recall that two linear operators $\mathcal{A} : V_J \to V_J$ and $\mathcal{B} : V_J \to V_J$ are *spectrally equivalent* if they satisfy

$$(\mathcal{A}v, v)_{L_2(\Omega)} \sim (\mathcal{B}v, v)_{L_2(\Omega)}, \quad v \in V_J, \tag{27}$$

with constants independent of J. Thus, the realization of the preconditioner is reduced to a computation in terms of the bases of the spaces V_j instead of W_j. The orthogonal projector O_j can, in turn, be replaced by a simpler local operator which is spectrally equivalent to O_j, see [37] and the derivation below.

Up to this point, the introduction to multilevel preconditioners has been basis-free. We now show how this framework can be used to construct a BPX-preconditioner for the linear system (6). Based on the definition (12), we construct

a sequence of spaces satisfying (17) such that $V_J = V_h^r$. In fact, we suppose that for each space dimension we have a sequence of p-open knot vectors $\Xi_{j_0,\ell}, \ldots, \Xi_{J,\ell}$, $\ell = 1, \ldots, n$, which provide a uniform partition of the interval $[0,1]$ such that $\Xi_{j,\ell} \subset \Xi_{j+1,\ell}$ for $j = j_0, j_0+1, \ldots, J$. In particular, we assume that $\Xi_{j+1,\ell}$ is obtained from $\Xi_{j,\ell}$ by dyadic refinement, i.e., the grid spacing for $\Xi_{j,\ell}$ is proportional to 2^{-j} for each $\ell = 1, \ldots, n$. In view of the assumptions on the parametric mapping $\mathbf{F}$, we assume that $\bar{h} = 2^{-j_0}$, i.e., $\mathbf{F}$ can be represented in terms of B-splines on the coarsest level j_0. By construction, we have now achieved that

$$S_{j_0}(\hat{\Omega}) \subset S_{j_0+1}(\hat{\Omega}) \subset \ldots \subset S_J(\hat{\Omega}).$$

Setting $V_j^r := \{v \in H_0^r(\Omega) : \ v \circ \mathbf{F} \in S_j(\hat{\Omega})\}$, we arrive at a sequence of nested spaces

$$V_{j_0}^r \subset V_{j_0+1}^r \subset \ldots \subset V_J^r.$$

Setting $\mathbb{I}_j := \{1, \ldots, \dim S_j(\hat{\Omega})\}$, we denote by B_i^j, $i \in \mathbb{I}_j$, the set of L_2-normalized B-spline basis functions for the space $S_j(\hat{\Omega})$. Define now the positive definite operator $P_j : L_2(\Omega) \to V_j^r$ as

$$P_j := \sum_{i \in \mathbb{I}_j} (\cdot\,, B_i^j \circ \mathbf{F}^{-1})_{L_2(\Omega)}\, B_i^j \circ \mathbf{F}^{-1}. \tag{28}$$

Corollary 5 *For the basis $\{B_i^j \circ \mathbf{F}^{-1},\ i \in \mathbb{I}_j\}$, the operators P_j and the L_2-projectors O_j are spectrally equivalent for any j.*

Proof The assertion follows by combining (11), (14), with Remark 3.7.1 from [37], see [10] for the main ingredients. □

Finally, we obtain an explicit representation of the preconditioner C_J in terms of the mapped spline bases for $V_j^r, j = j_0, \ldots, J$,

$$C_J = \sum_{j=j_0}^{J} 2^{-2jr} \sum_{i \in \mathbb{I}_j} (\cdot\,, B_i^j \circ \mathbf{F}^{-1})_{L_2(\Omega)}\, B_i^j \circ \mathbf{F}^{-1}. \tag{29}$$

Note that this preconditioner involves *all* B-splines from all levels j with an appropriate scaling, i.e., in fact a properly scaled *generating system* for V_J^r.

Remark 6 The *hierarchical basis (HB) preconditioner* introduced for $n = 2$ in [54] for piecewise linear B-splines fits into this framework by choosing *Lagrangian interpolants* in place of the projectors P_j in (23). However, since these operators do not satisfy (P3) in Properties 2, they do not yield an asymptotically optimal preconditioner for $n \geq 2$. Specifically, for $n = 3$, this preconditioner does not have an effect at all.

So far we have not explicitly addressed the dependence of the preconditioned system on p. Since all estimates in Theorem 1 which enter the proof of optimality depend on p, it is to be expected that the absolute values of the condition numbers, i.e., the values of the constants, depend on and increase with p. Indeed, in the next section, we show some numerical results which also aim at studying this dependence.

2.3 Realization of the BPX Preconditioner and Numerical Results

Now we are in the position to describe the concrete implementation of the BPX preconditioner. Its main ingredient are linear *intergrid operators* which map vectors and matrices between different grids. Specifically, we need to define prolongation and restriction operators.

Since $V_j^r \subset V_{j+1}^r$, each B-spline B_i^j on level j can be represented by a linear combination of B-splines B_k^{j+1} on level $j+1$. Arranging the B-splines in the set $\{B_i^j, i \in \mathbb{I}_j\}$ into a vector $\mathbf{B}^j$ in a fixed order, this relation denoted as *refinement relation* can be written as

$$\mathbf{B}^j = \mathbf{I}_j^{j+1} \mathbf{B}^{j+1} \tag{30}$$

with *prolongation operator* $\mathbf{I}_j^{j+1}$ from the trial space V_j^r to the trial space V_{j+1}^r. The restriction $\mathbf{I}_{j+1}^j$ is then simply defined as the transposed operator, i.e., $\mathbf{I}_{j+1}^j = (\mathbf{I}_j^{j+1})^T$. In case of piecewise linear B-splines, this definition coincides with the well known prolongation and restriction operators from finite element textbooks obtained by interpolation, see, e.g., [8].

We will exemplify the construction in case of quadratic and cubic B-splines on the interval, see, e.g., [7], as follows. We equidistantly subdivide the interval $[0, 1]$ into 2^j subintervals and obtain 2^j and $2^j + 1$, respectively, B-splines for $p = 2, 3$ and the corresponding quadratic and cubic spline space V_j^r which is given on this partition, respectively, see Fig. 1 for an illustration. Note that the two boundary functions which do not vanish at the boundary were removed in order to guarantee that $V_j^r \subset H_0^r(\Omega)$. Moreover, recall that the B-splines are L_2 normalized according to (13) which means that B_i^j is of the form $B_i^j(\zeta) = 2^{j/2} B(2^j \zeta - i)$ if B_i^j is an interior function, and correspondingly for the boundary functions.

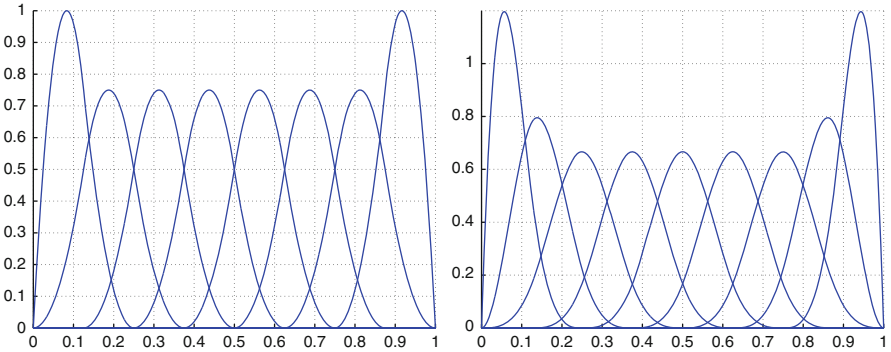


Fig. 1 Quadratic ($p = 2$) (*left*) and cubic ($p = 3$) (*right*) L_2-normalized B-splines (see (13)) on level $j = 3$ on the interval $[0, 1]$, yielding basis functions for $V_j^r \subset H_0^r(\Omega)$

In case of quadratic B-splines ($p = 2$), the restriction operator $\mathbf{I}_{j+1}^j$ reads

$$\mathbf{I}_{j+1}^j = 2^{-1/2} \begin{bmatrix} \frac{1}{2} & \frac{9}{8} & \frac{3}{8} & & & & & & \\ & \frac{1}{4} & \frac{3}{4} & \frac{3}{4} & \frac{1}{4} & & & & \\ & & \frac{1}{4} & \frac{3}{4} & \frac{3}{4} & \frac{1}{4} & & & \\ & & & \ddots & \ddots & & & & \\ & & & \frac{1}{4} & \frac{3}{4} & \frac{3}{4} & \frac{1}{4} & & \\ & & & & & \frac{3}{8} & \frac{9}{8} & \frac{1}{2} \end{bmatrix} \in \mathbb{R}^{2^j \times 2^{j+1}}.$$

For cubic B-splines ($p = 3$), it has the form

$$\mathbf{I}_{j+1}^j = 2^{-1/2} \begin{bmatrix} \frac{1}{2} & \frac{9}{8} & \frac{3}{8} & & & & & & & \\ & \frac{1}{4} & \frac{11}{12} & \frac{2}{3} & \frac{1}{6} & & & & & \\ & & \frac{1}{8} & \frac{1}{2} & \frac{3}{4} & \frac{1}{2} & \frac{1}{8} & & & \\ & & & \ddots & \ddots & \ddots & & & & \\ & & & & \frac{1}{8} & \frac{1}{2} & \frac{3}{4} & \frac{1}{2} & \frac{1}{8} & \\ & & & & & & \frac{1}{6} & \frac{2}{3} & \frac{11}{12} & \frac{1}{4} \\ & & & & & & & & \frac{3}{8} & \frac{9}{8} & \frac{1}{2} \end{bmatrix} \in \mathbb{R}^{(2^j+1)\times(2^{j+1}+1)}.$$

The normalization factor $2^{-1/2}$ stems from the L_2-normalization (13). The matrix entries are scaled in the usual fashion such that their rows sum to two. From these restriction operators for one dimensions, one obtains the related restriction operators on arbitrary unit cubes $[0, 1]^n$ via tensor products. Finally, we set $\mathbf{I}_j^J := \mathbf{I}_{J-1}^J \mathbf{I}_{J-2}^{J-1} \cdots \mathbf{I}_j^{j+1}$ and $\mathbf{I}_J^j := \mathbf{I}_j^{j+1} \mathbf{I}_{j+1}^{j+2} \cdots \mathbf{I}_{J-1}^J$ to define prolongations and restrictions between arbitrary levels j and J.

In order to derive the explicit form of the discretized BPX-preconditioner, for given functions $u_J, v_J \in V_J$ with expansion coefficients $u_{J,k}$ and $v_{J,\ell}$, respectively, we conclude from (29) that

$$\begin{aligned}(C_J u_J, v_J)_{L_2(\Omega)} &= \sum_{k,\ell\in\mathbb{I}_J} u_{J,k} v_{J,\ell} (C_J(B_k^J \circ \mathbf{F}^{-1}), B_\ell^J \circ \mathbf{F}^{-1})_{L_2(\Omega)} \\ &= \sum_{k,\ell\in\mathbb{I}_J} u_{J,k} v_{J,\ell} \sum_{j=j_0}^{J} 2^{-2jr} \sum_{i\in\mathbb{I}_j} (B_k^J \circ \mathbf{F}^{-1}, B_i^j \circ \mathbf{F}^{-1})_{L_2(\Omega)} \\ &\qquad \times (B_i^j \circ \mathbf{F}^{-1}, B_\ell^J \circ \mathbf{F}^{-1})_{L_2(\Omega)}.\end{aligned}$$

Next, one can introduce the mass matrix $\mathbf{M}_J = [(B_k^J \circ \mathbf{F}^{-1}, B_\ell^J \circ \mathbf{F}^{-1})_{L_2(\Omega)}]_{k,\ell}$ and obtains by the use of restrictions and prolongations

$$(C_J u_J, v_J)_{L_2(\Omega)} = \sum_{j=j_0}^{J} 2^{-2jr} \mathbf{u}_J^T \mathbf{M}_J \mathbf{I}_j^J \mathbf{I}_J^j \mathbf{M}_J \mathbf{v}_J.$$

The mass matrices which appear in this expression can be further suppressed since $\mathbf{M}_J$ is spectrally equivalent to the identity matrix. Finally, the *discretized BPX-preconditioner* to be implemented is of the simple form

$$\mathbf{C}_J = \sum_{j=j_0}^{J} 2^{-2jr} \mathbf{I}_j^J \mathbf{I}_J^j, \tag{31}$$

involving only restrictions and prolongations. A further simple improvement can be obtained by replacing the scaling factor 2^{-2jr} by $\operatorname{diag}(\mathbf{A}_j)^{-1}$, where $\operatorname{diag}(\mathbf{A}_j)$ denotes the diagonal matrix built from the diagonal entries of the stiffness matrix $\mathbf{A}_j$. This diagonal scaling has the same effect as the levelwise scaling by 2^{-2jr} but improves the condition numbers considerably, particularly if parametric mappings are involved. Thus, the discretized BPX-preconditioner takes on the form

$$\mathbf{C}_J = \sum_{j=j_0}^{J} \mathbf{I}_j^J \operatorname{diag}(\mathbf{A}_j)^{-1} \mathbf{I}_J^j \tag{32}$$

which we will use in the subsequent computations presented in Tables 1 and 2. If the condition number $\kappa(\mathbf{A}_{j_0})$ is already high in absolute numbers on the coarsest level j_0, it is worth to use its exact inverse on the coarse grid, i.e., to apply instead of (32) the operator

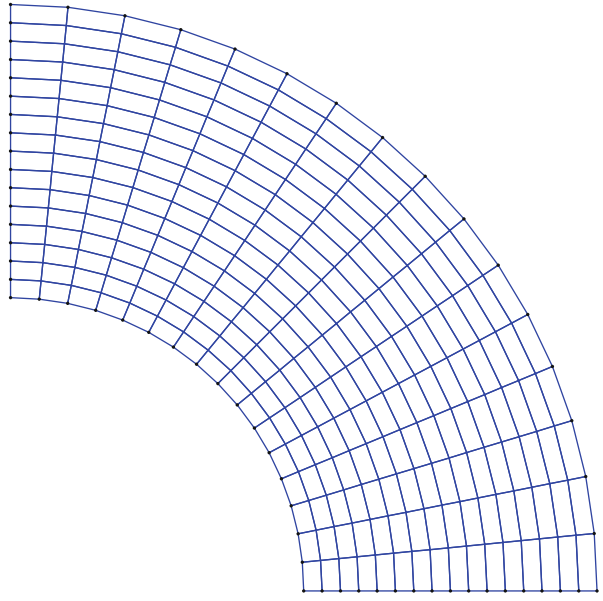
$$\mathbf{C}_J = \mathbf{I}_{j_0}^J \mathbf{A}_{j_0}^{-1} \mathbf{I}_J^{j_0} + \sum_{j=j_0+1}^{J} \mathbf{I}_j^J \operatorname{diag}(\mathbf{A}_j)^{-1} \mathbf{I}_J^j,$$

Table 1 Condition numbers of the BPX-preconditioned Laplacian on $\hat{\Omega} = (0, 1)^n$ for $n = 1, 2, 3$

	Interval ($n = 1$)				Square ($n = 2$)				Cube ($n = 3$)			
Level	$p = 1$	$p = 2$	$p = 3$	$p = 4$	$p = 1$	$p = 2$	$p = 3$	$p = 4$	$p = 1$	$p = 2$	$p = 3$	$p = 4$
3	7.43	3.81	7.03	5.93	5.93	7.31	22.8	133	3.49	39.5	356	5957
4	8.87	4.40	9.47	7.81	5.00	9.03	40.2	225	4.85	50.8	624	9478
5	10.2	4.67	11.0	9.36	5.70	9.72	51.8	293	5.75	56.6	795	11,887
6	11.3	4.87	12.1	10.7	6.27	10.1	58.7	340	6.40	59.7	895	13,185
7	12.2	5.00	12.7	11.5	6.74	10.4	63.1	371	6.91	61.3	961	13,211
8	13.0	5.10	13.0	11.9	7.14	10.5	66.0	391	7.34	62.2	990	13,234
9	13.7	5.17	13.2	12.1	7.48	10.6	68.0	403	7.70	62.6	1016	13,255
10	14.2	5.22	13.4	12.2	7.77	10.6	69.3	411	7.99	62.9	1040	–

Table 2 Condition numbers of the BPX-preconditioned Laplacian on the analytic arc seen on the right hand side. The bracketed numbers are the related condition numbers without preconditioning

Level	$p = 1$	$p = 2$	$p = 3$	$p = 4$
3	5.04 (21.8)	12.4 (8.64)	31.8 (31.8)	184 (184)
4	11.1 (90.2)	16.3 (34.3)	54.7 (32.9)	291 (173)
5	25.3 (368)	19.0 (139)	70.1 (98.9)	376 (171)
6	31.9 (1492)	21.4 (560)	79.2 (401)	436 (322)
7	37.4 (6015)	23.1 (2255)	84.4 (1620)	471 (1297)
8	42.1 (241,721)	24.3 (9062)	87.3 (6506)	490 (5217)
9	45.7 (969,301)	25.2 (36,353)	89.0 (26,121)	500 (20,945)
10	48.8 (388,690)	25.9 (145,774)	90.1 (104,745)	505 (83,975)



see [12, 46]. Another substantial improvement of the BPX-preconditioner can be achieved by replacing the diagonal scaling on each level by, e.g., a SSOR preconditioning as follows. We decompose the system matrix as $\mathbf{A}_j = \mathbf{L}_j + \mathbf{D}_j + \mathbf{L}_j^T$ with the diagonal matrix $\mathbf{D}_j$, the lower triangular part $\mathbf{L}_j$, and the upper triangular part $\mathbf{L}_j^T$. Then we replace the diagonal scaling on each level of the BPX-preconditioner (32) by the SSOR preconditioner, i.e., instead of (32) we apply the preconditioner

$$\mathbf{C}_J = \sum_{j=j_0}^{J} \mathbf{I}_j^J (\mathbf{D}_j + \mathbf{L}_j)^{-T} \mathbf{D}_j (\mathbf{D}_j + \mathbf{L}_j)^{-1} \mathbf{I}_J^j. \tag{33}$$

In doing so, the condition numbers can be improved impressively. In Table 3, we list the ℓ_2-condition numbers for the BPX-preconditioned Laplacian in case of cubic B-splines in two spatial dimensions. By comparing the numbers with those found in Tables 1 and 2 one can infer that the related condition numbers are all reduced by a factor about five. Note that the setup, storage and application of the operator defined in (33) is still of optimal linear complexity.

Finally, we provide numerical results in order to demonstrate the preconditioning and to specify the dependence on the spatial dimension n and the spline degree p. We consider an approximation of the homogeneous Dirichlet problem for the Poisson equation on the n-dimensional unit cube $\hat{\Omega} = (0,1)^n$ for $n = 1, 2, 3$. The mesh on level j is obtained by subdividing the cube j-times dyadically into 2^n subcubes of mesh size $h_j = 2^{-j}$. On this subdivision, we consider the B-splines of degree $p = 1, 2, 3, 4$ as defined in Sect. 2.1. The ℓ_2-condition numbers of the related stiffness matrices, preconditioned by the BPX-preconditioner (32), are shown in Table 1. The condition numbers seem to be independent of the level j, but they depend on the spline degree p and the space dimension n for $n > 1$. For fourth order problems on the sphere, corresponding results for the bi-Laplacian with and without BPX preconditioning were presented in [43].

We study next the dependence of the condition numbers on the parametric mapping $\mathbf{F}$. We consider the case $n = 2$ in case of a smooth mapping (see the plot on the right hand side of Table 2 for an illustration of the mapping). As one can see from Table 2, the condition numbers are at most about a factor of five higher than the related values in Table 1. Nearly the same observation holds if we replace the parametric mapping by a $\mathcal{C}^0$-parametrization which maps the unit square onto an L-shaped domain, see [11].

If we consider a singular map $\mathbf{F}$, that is, a mapping that does not satisfy (11), the condition numbers grow considerably as expected, see [11]. But even in this case, the BPX-preconditioner with SSOR acceleration (33) is able to drastically reduce the condition numbers of the system matrix in all examples, see Table 3.

Table 3 Condition numbers of the BPX-preconditioned Laplacian for cubic B-splines on different geometries in case of using a BPX-SSOR preconditioning on each level

Level	Square		C^0-map of the L-shape	Singular C^1-map of the L-shape
3	3.61	3.65	3.67	3.80
4	6.58	6.97	7.01	7.05
5	8.47	10.2	10.2	14.8
6	9.73	13.1	13.2	32.2
7	10.5	14.9	15.2	77.7
8	11.0	15.9	16.3	180
9	11.2	16.5	17.0	411
10	11.4	16.9	17.7	933

3 Wavelets and the FWT (Fast Wavelet Transform)

Returning to the abstract setting at the beginning of Sect. 2.2, it will now be specified how to construct and realize the FWT (fast wavelet transform) for preconditioning. I would like to emphasize at this point that wavelets are, more importantly than for preconditioning, an appropriate theoretical tool to derive adaptive discretizations in case of singularities in the domain or data. In fact, they enable one to not only prove convergence of a corresponding adaptive scheme but also optimal complexity of these schemes, see [16] and several subsequent papers or [48] for a more recent survey.

3.1 Some Basic Notions

In view of the problem formulation (6), we need to have a wavelet basis for the space $H_0^r(\Omega)$ and its dual. We formulate these basic properties for a general Hilbert space H, following [18] or [38]. A *wavelet basis* for H, shortly *wavelets*, is understood here as a collection of functions

$$\Psi_H := \{\psi_{H,\lambda} : \lambda \in \mathbb{I}_H\} \subset H \tag{34}$$

which are indexed by elements λ from an infinite index set $\mathbb{I}_H$. Each of the indices λ comprises different information $\lambda = (j, \mathbf{k}, \mathbf{e})$ such as the *refinement scale* or *level of resolution* j and a spatial location $\mathbf{k} = \mathbf{k}(\lambda) \in \mathbb{Z}^n$. In more than one space dimension, the basis functions are built from taking tensor products of certain univariate functions, and in this case the third index $\mathbf{e}$ contains information on the *type* of wavelet. We will frequently use the symbol $|\lambda| := j$ to access the resolution level j. In the univariate case on all of $\mathbb{R}$, $\psi_{H,\lambda}$ is typically generated by means of shifts and dilates of a single function ψ, i.e., $\psi_\lambda = \psi_{j,k} = 2^{j/2}\psi(2^j \cdot -k)$, $j, k \in \mathbb{Z}$, normalized with respect to $\|\cdot\|_{L_2(\mathbb{R})}$. On bounded domains which can be represented

as unions of mappings of tensor product domains, the structure of the functions is essentially the same up to modifications near the boundary.

We assume that the wavelets satisfy the following crucial properties.

Riesz basis property (R): Every $v \in H$ has a unique expansion in terms of Ψ_H,

$$v = \sum_{\lambda \in \mathbb{I}_H} v_\lambda \, \psi_{H,\lambda} =: \mathbf{v}^T \Psi_H, \quad \mathbf{v} := (v_\lambda)_{\lambda \in \mathbb{I}_H}, \tag{35}$$

and its expansion coefficients satisfy the *norm equivalence*

$$\|\mathbf{v}\| \sim \|\mathbf{v}^T \Psi_H\|_H, \quad \mathbf{v} \in \ell_2(\mathbb{I}_H), \tag{36}$$

where $\|\cdot\| := \|\cdot\|_{\ell_2(\mathbb{I}_H)}$.

Locality (L): The functions $\psi_{H,\lambda}$ have compact support which decreases with increasing level $j = |\lambda|$, i.e.,

$$\operatorname{diam}(\operatorname{supp} \psi_{H,\lambda}) \sim 2^{-|\lambda|}. \tag{37}$$

Cancellation property (CP): There exists an integer $\tilde{d} = \tilde{d}_H$ such that

$$\langle v, \psi_{H,\lambda} \rangle \lesssim 2^{-|\lambda|(n/2-n/p+\tilde{d})} |v|_{W_p^{\tilde{d}}(\operatorname{supp} \psi_{H,\lambda})}. \tag{38}$$

This means that integrating against a wavelet has the effect of taking an $\tilde{m}$th order difference which annihilates the smooth part of v. This property is for wavelets defined on Euclidean domains typically realized by constructing Ψ_H in such a way that it possesses a *dual* or *biorthogonal* basis $\tilde{\Psi}_H \subset H'$ such that the multiresolution spaces $\tilde{S}_j := \operatorname{span}\{\tilde{\psi}_{H,\lambda} : |\lambda| < j\}$ contain all polynomials of order $\tilde{m}$. Here *dual basis* means that $\langle \psi_{H,\lambda}, \tilde{\psi}_{H,\nu} \rangle = \delta_{\lambda,\nu}$, $\lambda, \nu \in \mathbb{I}_H$.

A few remarks on these properties should be made. In (R), the norm equivalence (36) is crucial since it means control over a function measured in $\|\cdot\|_H$ from above and below by its expansion coefficients: small changes in the coefficients only cause small changes in the function. Together with the locality (L), this also means that local changes stay local. This stability is an important feature which is used for deriving optimal preconditioners. Finally, the cancellation property (CP) entails that smooth functions have small wavelet coefficients which, on account of (36) may be neglected in a controllable way. Moreover, (CP) can be used to derive quasi-sparse representations of a wide class of operators which is important for compression purposes, see, e.g., [48].

By duality arguments one can show that (36) is equivalent to the existence of a biorthogonal collection which is *dual* or *biorthogonal* to Ψ_H,

$$\tilde{\Psi}_H := \{\tilde{\psi}_{H,\lambda} : \lambda \in \mathbb{I}_H\} \subset H', \quad \langle \psi_{H,\lambda}, \tilde{\psi}_{H,\mu} \rangle = \delta_{\lambda,\mu}, \qquad \lambda, \mu \in \mathbb{I}_H, \tag{39}$$

which is a Riesz basis for H', that is, for any $\tilde{v} = \tilde{\mathbf{v}}^T \tilde{\Psi}_H \in H'$ one has

$$\|\tilde{\mathbf{v}}\| \sim \|\tilde{\mathbf{v}}^T \tilde{\Psi}_H\|_{H'} \tag{40}$$

see [19]. Here the tilde expresses that the collection $\tilde{\Psi}_H$ is a dual basis to a primal one for the space identified by the subscript, so that $\tilde{\Psi}_H = \Psi_{H'}$. Here and in the following, we will view Ψ_H both as in (34) as a *collection* of functions as well as a (possibly infinite) column *vector* containing all functions always assembled in some fixed unspecified order. For a countable collection of functions Θ and some single function σ, the term $\langle \Theta, \sigma \rangle$ is then the column vector with entries $\langle \theta, \sigma \rangle$, $\theta \in \Theta$, and correspondingly $\langle \sigma, \Theta \rangle$ the row vector. For two collections Θ, Σ, the quantity $\langle \Theta, \Sigma \rangle$ is then a (possibly infinite) matrix with entries $(\langle \theta, \sigma \rangle)_{\theta \in \Theta,\, \sigma \in \Sigma}$ for which $\langle \Theta, \Sigma \rangle = \langle \Sigma, \Theta \rangle^T$. This also implies for a (possibly infinite) matrix $\mathbf{C}$ that $\langle \mathbf{C}\Theta, \Sigma \rangle = \mathbf{C}\langle \Theta, \Sigma \rangle$ and $\langle \Theta, \mathbf{C}\Sigma \rangle = \langle \Theta, \Sigma \rangle \mathbf{C}^T$.

In this notation, the *biorthogonality* or *duality conditions* (39) can be expressed shortly as

$$\langle \Psi, \tilde{\Psi} \rangle = \mathbf{I} \tag{41}$$

with the infinite identity matrix $\mathbf{I}$.

Wavelets with the above properties can be obtained in the following way. We start from an *anchor basis* $\Psi = \{\psi_\lambda : \lambda \in \mathbb{I} = \mathbb{I}_H\}$ which is a Riesz basis for $L_2(\Omega)$, meaning that Ψ is scaled such that $\|\psi_\lambda\|_{L_2(\Omega)} \sim 1$. Moreover, its dual basis $\tilde{\Psi}$ is also a Riesz basis for $L_2(\Omega)$. Ψ and $\tilde{\Psi}$ are constructed in such a way that rescaled versions of *both bases* $\Psi, \tilde{\Psi}$ form Riesz bases for a whole range of (closed subspaces of) Sobolev spaces $H^s := H^s(\Omega)$, for $0 < s < \gamma, \tilde{\gamma}$, respectively, or subspaces of these. Consequently, one can derive that for each $s \in (-\tilde{\gamma}, \gamma)$ the collection

$$\Psi_s := \{2^{-s|\lambda|}\psi_\lambda : \lambda \in \mathbb{I}\} =: \mathbf{D}^{-s}\Psi \tag{42}$$

is a Riesz basis for H^s [18], i.e., it holds

$$\|\mathbf{v}\| \sim \|\mathbf{v}^T \Psi_s\|_{H^s}, \quad \mathbf{v} \in \ell_2(\mathbb{I}), \tag{43}$$

for each $s \in (-\tilde{\gamma}, \gamma)$. Such a scaling represented by a diagonal matrix $\mathbf{D}^s$ introduced in (42) will play an important role later on. Concrete constructions of wavelet bases with the above properties for parameters $\gamma, \tilde{\gamma} \leq 5/2$ can be found in [22] which cover the situation in (6) for $r = 1, 2$.

3.2 Multiscale Decomposition of Function Spaces

In this section, the basic construction principles of the biorthogonal wavelets with properties (R), (L) and (CP) are summarized, see, e.g., [18]. Their cornerstones are *multiresolution analyses* of the function spaces under consideration as in Sect. 2.2 and the concept of *stable completions*. These concepts are free of Fourier techniques and can therefore be applied to derive constructions of wavelets on domains or manifolds which are subsets of $\mathbb{R}^n$.

Multiresolution of L_2 (univariate case). Practical constructions of wavelets typically start out with multiresolution analyses of function spaces. Consider a *multiresolution* $\mathcal{S}$ of L_2 which consists of closed subspaces S_j of L_2, called *trial spaces*, such that they are nested and their union is dense in L_2,

$$S_{j_0} \subset S_{j_0+1} \subset \ldots \subset S_j \subset S_{j+1} \subset \ldots L_2, \qquad \operatorname{clos}_{L_2}\Big(\bigcup_{j=j_0}^{\infty} S_j\Big) = L_2. \tag{44}$$

The index j is the refinement level which appeared already in the elements of the index set $\mathbb{I}$ in (34), starting with some coarsest level $j_0 \in \mathbb{N}_0$. We abbreviate for a finite subset $\Theta \subset L_2$ the linear span of Θ as

$$S(\Theta) = \operatorname{span}\{\Theta\}.$$

Typically the multiresolution spaces S_j have the form

$$S_j = S(\Phi_j), \qquad \Phi_j = \{\phi_{j,k} : k \in \Delta_j\}, \tag{45}$$

for some finite index set Δ_j, where the set $\{\Phi_j\}_{j=j_0}^{\infty}$ is *uniformly stable* in the sense that

$$\|\mathbf{c}\|_{\ell_2(\Delta_j)} \sim \|\mathbf{c}^T\Phi_j\|_{L_2}, \qquad \mathbf{c} = \{c_k\}_{k\in\Delta_j} \in \ell_2(\Delta_j), \tag{46}$$

holds uniformly in j. Here we have used again the shorthand notation

$$\mathbf{c}^T\Phi_j = \sum_{k\in\Delta_j} c_k\phi_{j,k}$$

and Φ_j denotes both the (column) vector containing the functions $\phi_{j,k}$ as well as the set of functions (45).

The collection Φ_j is called *single scale basis* since all its elements live only on one scale j. In the present context of multiresolution analysis, Φ_j is also called *generator basis* or shortly *generators* of the multiresolution. We assume that the $\phi_{j,k}$ are compactly supported with

$$\operatorname{diam}(\operatorname{supp}\phi_{j,k}) \sim 2^{-j}. \tag{47}$$

It follows from (46) that they are scaled such that

$$\|\phi_{j,k}\|_{L_2} \sim 1 \tag{48}$$

holds. It is known that nestedness (44) together with stability (46) implies the existence of matrices $\mathbf{M}_{j,0}$ such that the two-scale relation

$$\Phi_j = \mathbf{M}_{j,0}^T \Phi_{j+1} \tag{49}$$

holds. Any set of functions satisfying an equation of this form, the *refinement* or *two-scale relation*, will be called *refinable*. The matrices $\mathbf{M}_{j,0}$ are up to boundary modifications exactly the intergrid operators occurring in (30).

Denoting by $[X, Y]$ the space of bounded linear operators from a normed linear space X into the normed linear space Y, one has that

$$\mathbf{M}_{j,0} \in [\ell_2(\Delta_j), \ell_2(\Delta_{j+1})]$$

is *uniformly sparse* which means that the number of entries in each row or column is uniformly bounded. Furthermore, one infers from (46) that

$$\|\mathbf{M}_{j,0}\| = \mathcal{O}(1), \quad j \geq j_0, \tag{50}$$

where the corresponding operator norm is defined as

$$\|\mathbf{M}_{j,0}\| := \sup_{\mathbf{c} \in \ell_2(\Delta_j),\ \|\mathbf{c}\|_{\ell_2(\Delta_j)}=1} \|\mathbf{M}_{j,0}\mathbf{c}\|_{\ell_2(\Delta_{j+1})}.$$

Since the union of $\mathcal{S}$ is dense in L_2, a basis for L_2 can be assembled from functions which span any complement between two successive spaces S_j and S_{j+1}, i.e.,

$$S(\Phi_{j+1}) = S(\Phi_j) \oplus S(\Psi_j) \tag{51}$$

where

$$\Psi_j = \{\psi_{j,k} : k \in \nabla_j\}, \qquad \nabla_j := \Delta_{j+1} \setminus \Delta_j. \tag{52}$$

The functions Ψ_j are called *wavelet functions* or shortly *wavelets* if, among other conditions detailed below, the union $\{\Phi_j \cup \Psi_j\}$ is still uniformly stable in the sense of (46). Since (51) implies $S(\Psi_j) \subset S(\Phi_{j+1})$, the functions in Ψ_j must also satisfy a matrix-vector relation of the form

$$\Psi_j = \mathbf{M}_{j,1}^T \Phi_{j+1} \tag{53}$$

with a matrix $\mathbf{M}_{j,1}$ of size $(\#\Delta_{j+1}) \times (\#\nabla_j)$. Furthermore, (51) is equivalent to the fact that the linear operator composed of $\mathbf{M}_{j,0}$ and $\mathbf{M}_{j,1}$,

$$\mathbf{M}_j = (\mathbf{M}_{j,0}, \mathbf{M}_{j,1}), \tag{54}$$

is *invertible* as a mapping from $\ell_2(\Delta_j \cup \nabla_j)$ onto $\ell_2(\Delta_{j+1})$. One can also show that the set $\{\Phi_j \cup \Psi_j\}$ is uniformly stable if and only if

$$\|\mathbf{M}_j\|, \|\mathbf{M}_j^{-1}\| = \mathcal{O}(1), \quad j \to \infty. \tag{55}$$

The particular cases that will be important for practical purposes are when not only $\mathbf{M}_{j,0}$ and $\mathbf{M}_{j,1}$ are uniformly sparse but also the inverse of $\mathbf{M}_j$. We denote this inverse by $\mathbf{G}_j$ and assume that it is split into

$$\mathbf{G}_j = \mathbf{M}_j^{-1} = \begin{pmatrix} \mathbf{G}_{j,0} \\ \mathbf{G}_{j,1} \end{pmatrix}. \tag{56}$$

A special situation occurs when $\mathbf{M}_j$ is an orthogonal matrix,

$$\mathbf{G}_j = \mathbf{M}_j^{-1} = \mathbf{M}_j^T$$

which corresponds to the case of L_2 *orthogonal wavelets* [25]. A systematic construction of more general $\mathbf{M}_j$, $\mathbf{G}_j$ for spline-wavelets can be found in [22], see also [18] for more examples, including the hierarchical basis.

Thus, the identification of the functions Ψ_j which span the complement of $S(\Phi_j)$ in $S(\Phi_{j+1})$ is equivalent to completing a given refinement matrix $\mathbf{M}_{j,0}$ to an invertible matrix $\mathbf{M}_j$ in such a way that (55) is satisfied. Any such completion $\mathbf{M}_{j,1}$ is called *stable completion* of $\mathbf{M}_{j,0}$. In other words, the problem of the construction of compactly supported wavelets can equivalently be formulated as an algebraic problem of finding the (uniformly) sparse completion of a (uniformly) sparse matrix $\mathbf{M}_{j,0}$ in such a way that its inverse is also (uniformly) sparse. The fact that inverses of sparse matrices are usually dense elucidates the difficulties in the constructions. Constructions that yield compactly supported wavelets are particularly suited for computations in numerical analysis.

Combining the two-scale relations (49) and (53), one can see that $\mathbf{M}_j$ performs a change of bases in the space S_{j+1},

$$\begin{pmatrix} \Phi_j \\ \Psi_j \end{pmatrix} = \begin{pmatrix} \mathbf{M}_{j,0}^T \\ \mathbf{M}_{j,1}^T \end{pmatrix} \Phi_{j+1} = \mathbf{M}_j^T \Phi_{j+1}. \tag{57}$$

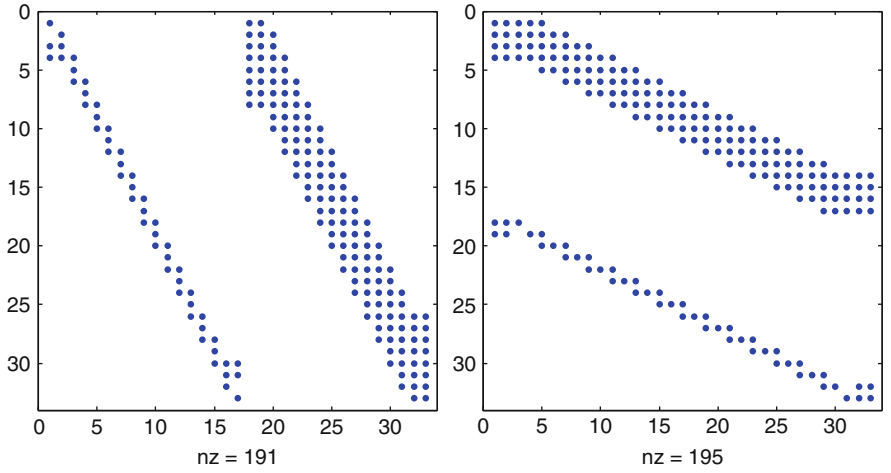


Fig. 2 Nonzero pattern of matrices $\mathbf{M}_j$ (*left*) and $\mathbf{G}_j$ (*right*) for boundary-adapted B-splines of order $d = 2$ (degree $p = 1$) as generators and duals of order $\tilde{d} = 4$

Conversely, applying the inverse of $\mathbf{M}_j$ to both sides of (57) results in the *reconstruction identity*

$$\Phi_{j+1} = \mathbf{G}_j^T \begin{pmatrix} \Phi_j \\ \Psi_j \end{pmatrix} = \mathbf{G}_{j,0}^T \Phi_j + \mathbf{G}_{j,1}^T \Psi_j. \tag{58}$$

An example of the structure of the matrices $\mathbf{M}_j$ and $\mathbf{G}_j$ is given in Fig. 2.

Fixing a *finest resolution level* J, one can repeat the decomposition (51) so that $S_J = S(\Phi_J)$ can be written in terms of the functions from the coarsest space supplied with the complement functions from all intermediate levels,

$$S(\Phi_J) = S(\Phi_{j_0}) \oplus \bigoplus_{j=j_0}^{J-1} S(\Psi_j). \tag{59}$$

Thus, every function $v \in S(\Phi_J)$ can be written in its *single-scale representation*

$$v = (\mathbf{c}_J)^T \Phi_J = \sum_{k \in \Delta_J} c_{J,k} \phi_{J,k} \tag{60}$$

as well as in its *multi-scale form*

$$v = (\mathbf{c}_{j_0})^T \Phi_{j_0} + (\mathbf{d}_{j_0})^T \Psi_{j_0} + \cdots + (\mathbf{d}_{J-1})^T \Psi_{J-1} \tag{61}$$

with respect to the *multiscale* or *wavelet basis*

$$\Psi^J := \Phi_{j_0} \cup \bigcup_{j=j_0}^{J-1} \Psi_j =: \bigcup_{j=j_0-1}^{J-1} \Psi_j \tag{62}$$

Often the single-scale representation of a function may be easier to compute and evaluate while the multi-scale representation allows one to separate features of the underlying function characterized by different length scales. Since therefore both representations are advantageous, it is useful to determine the transformation between the two representations, commonly referred to as the *Wavelet Transform*,

$$\mathbf{T}_J : \ell_2(\Delta_j) \to \ell_2(\Delta_j), \qquad \mathbf{d}^J \mapsto \mathbf{c}_J, \tag{63}$$

where

$$\mathbf{d}^J := (\mathbf{c}_{j_0}, \mathbf{d}_{j_0}, \ldots, \mathbf{d}_{J-1})^T.$$

The previous relations (57) and (58) indicate that this will involve the matrices $\mathbf{M}_j$ and $\mathbf{G}_j$. In fact, $\mathbf{T}_J$ has the representation

$$\mathbf{T}_J = \mathbf{T}_{J,J-1} \cdots \mathbf{T}_{J,j_0}, \tag{64}$$

where each factor has the form

$$\mathbf{T}_{J,j} := \begin{pmatrix} \mathbf{M}_j & \mathbf{0} \\ \mathbf{0} & \mathbf{I}^{(\#\Delta_J - \#\Delta_{j+1})} \end{pmatrix} \in \mathbb{R}^{(\#\Delta_J)\times(\#\Delta_J)}. \tag{65}$$

Schematically $\mathbf{T}_J$ can be visualized as a pyramid scheme

$$\begin{array}{ccccccccc} & \overset{\mathbf{M}_{j_0,0}}{} & & \overset{\mathbf{M}_{j_0+1,0}}{} & & & & \overset{\mathbf{M}_{J-1,0}}{} & \\ \mathbf{c}_{j_0} & \longrightarrow & \mathbf{c}_{j_0+1} & \longrightarrow & \mathbf{c}_{j_0+2} & \longrightarrow \cdots & \mathbf{c}_{J-1} & \longrightarrow & \mathbf{c}_J \\ & \overset{\mathbf{M}_{j_0,1}}{\nearrow} & & \overset{\mathbf{M}_{j_0+1,1}}{\nearrow} & & \nearrow \cdots & & \overset{\mathbf{M}_{J-1,1}}{\nearrow} & \\ \mathbf{d}_{j_0} & & \mathbf{d}_{j_0+1} & & \mathbf{d}_{j_0+2} & & \mathbf{d}_{J-1} & & \end{array} \tag{66}$$

Accordingly, the inverse transform $\mathbf{T}_J^{-1}$ can be written also in product structure (64) in reverse order involving the matrices $\mathbf{G}_j$ as follows:

$$\mathbf{T}_J^{-1} = \mathbf{T}_{J,j_0}^{-1} \cdots \mathbf{T}_{J,J-1}^{-1}, \tag{67}$$

where each factor has the form

$$\mathbf{T}_{J,j}^{-1} := \begin{pmatrix} \mathbf{G}_j & \mathbf{0} \\ \mathbf{0} & \mathbf{I}^{(\#\Delta_J - \#\Delta_{j+1})} \end{pmatrix} \in \mathbb{R}^{(\#\Delta_J)\times(\#\Delta_J)}. \tag{68}$$

The corresponding pyramid scheme is then

$$\begin{array}{ccccccccc} & & \mathbf{G}_{J-1,0} & & \mathbf{G}_{J-2,0} & & & & \mathbf{G}_{j_0,0} \\ \mathbf{c}_J & \longrightarrow & \mathbf{c}_{J-1} & \longrightarrow & \mathbf{c}_{J-2} & \longrightarrow & \cdots & \longrightarrow & \mathbf{c}_{j_0} \\ & & \mathbf{G}_{J-1,1} & & \mathbf{G}_{J-2,1} & & & & \mathbf{G}_{j_0,1} \\ & & \searrow & & \searrow & & \searrow \ \cdots \ \searrow & & \\ & & \mathbf{d}_{J-1} & & \mathbf{d}_{J-2} & & \mathbf{d}_{J-1} & & \mathbf{d}_{j_0} \end{array} \tag{69}$$

Property (55) and the fact that $\mathbf{M}_j$ and $\mathbf{G}_j$ can be applied in $(\#\Delta_{j+1})$ operations uniformly in j entails that the complexity of applying $\mathbf{T}_J$ or $\mathbf{T}_J^{-1}$ using the pyramid scheme is of order $\mathcal{O}(\#\Delta_J) = \mathcal{O}(\dim S_J)$ uniformly in J. For this reason, $\mathbf{T}_J$ is called the *Fast Wavelet Transform* (FWT). Note that one should *not* explicitly assemble $\mathbf{T}_J$ or $\mathbf{T}_J^{-1}$. In fact, due to the particular band structure of $\mathbf{M}_j$ and $\mathbf{G}_j$, this would result in matrices with $\mathcal{O}(J\,\#\Delta_J)$ entries. In Table 4 at the end of this section, spectral condition numbers for the FWT for different constructions of biorthogonal wavelets on the interval computed in [46] are displayed.

Since $\cup_{j\geq j_0} S_j$ is dense in L_2, a basis for the whole space L_2 is obtained when letting $J \to \infty$ in (62),

$$\begin{aligned} \Psi &:= \bigcup_{j=j_0-1}^{\infty} \Psi_j = \{\psi_{j,k} : (j,k) \in \mathbb{I}\}, \qquad \Psi_{j_0-1} := \Phi_{j_0} \\ \mathbb{I} &:= \{j_0\} \times \Delta_{j_0} \cup \bigcup_{j=j_0}^{\infty} \{j\} \times \nabla_j \,. \end{aligned} \tag{70}$$

Theorem 7 ([18]) *The multiscale transformations $\mathbf{T}_J$ are well-conditioned in the sense*

$$\|\mathbf{T}_J\|, \|\mathbf{T}_J^{-1}\| = \mathcal{O}(1), \quad J \geq j_0, \tag{71}$$

if and only if the collection Ψ defined by (70) is a Riesz basis for L_2.

A detailed construction of the *dual wavelets* $\tilde{\Psi}$, can be found in [18, 39].

Multiresolution of Sobolev spaces. Let now $\mathcal{S}$ be a multiresolution sequence consisting of closed subspaces of H^s with the property (44) whose union is dense in H^s. The following result from [18] ensures under which conditions norm equivalences hold for the H^s-norm.

Theorem 8 *Let* $\{\Phi_j\}_{j=j_0}^{\infty}$ *and* $\{\tilde{\Phi}_j\}_{j=j_0}^{\infty}$ *be uniformly stable, refinable, biorthogonal collections. If the* Jackson-type estimate

$$\inf_{v_j \in S_j} \|v - v_j\|_{L_2} \lesssim 2^{-sj} \|v\|_{H^s}, \quad v \in H^s,\ 0 < s \leq \bar{d}, \tag{72}$$

and the Bernstein inequality

$$\|v_j\|_{H^s} \lesssim 2^{sj} \|v_j\|_{L_2}, \quad v_j \in S_j,\ s < \bar{t}, \tag{73}$$

hold for

$$S_j = \begin{Bmatrix} S(\Phi_j) \\ S(\tilde{\Phi}_j) \end{Bmatrix} \text{ with order } \bar{d} = \begin{Bmatrix} d \\ \tilde{d} \end{Bmatrix} \text{ and } \bar{t} = \begin{Bmatrix} t \\ \tilde{t} \end{Bmatrix}, \tag{74}$$

then for

$$0 < \sigma := \min\{d, t\}, \qquad 0 < \tilde{\sigma} := \min\{\tilde{d}, \tilde{t}\}, \tag{75}$$

we have the norm equivalence

$$\|v\|_{H^s}^2 \sim \sum_{j=j_0-1}^{\infty} 2^{2sj} \|\langle \tilde{\Psi}_j, v \rangle\|_{\ell_2(\nabla_j)}^2, \quad s \in (-\tilde{\sigma}, \sigma). \tag{76}$$

For many applications it suffices to have established (76) only for certain $s > 0$ for which one only needs to require (72) and (73) for $\{\Phi_j\}_{j=j_0}^{\infty}$. The Jackson estimates (72) of order $\tilde{d}$ for $S(\tilde{\Phi}_j)$ imply the cancellation properties (CP) (38), see, e.g., [20].

Remark 9 When the wavelets live on $\Omega \subset \mathbb{R}^n$, (72) means that all polynomials up to order $\tilde{d}$ are contained in $S(\tilde{\Phi}_j)$. One also says that $S(\tilde{\Phi}_j)$ is *exact* of order $\tilde{d}$. Because of the biorthogonality conditions, this implies that the wavelets $\psi_{j,k}$ are orthogonal to polynomials up to order $\tilde{d}$ or have $\tilde{d}$th order *vanishing moments*. By Taylor expansion, this in turn yields (38).

For a summary of different constructions of biorthogonal wavelets on $\mathbb{R}$ and bounded domains in $\mathbb{R}^n$, see, e.g., [39]. We display spectral condition numbers for the FWT for two different constructions of biorthogonal wavelets on the interval in Table 4. The first column denotes the finest level on which the spectral condition numbers of the FWT are computed. The next column contains the numbers for the construction of biorthogonal spline-wavelets on the interval from [22] for the case $d = 2, \tilde{d} = 4$ while the last column displays the condition numbers for a scaled version derived in [12]. We observe that the absolute numbers stay constant and low even for high levels j. We will see later in Sect. 3.3 how the transformation $\mathbf{T}_J$ is used for preconditioning.

Table 4 Computed spectral condition numbers for the FWT on [0, 1] for different constructions of biorthogonal wavelets on the interval [46]

j	$\kappa_2(\mathbf{T}_{\mathrm{DKU}})$	$\kappa_2(\mathbf{T}_{\mathrm{B}})$	j	$\kappa_2(\mathbf{T}_{\mathrm{DKU}})$	$\kappa_2(\mathbf{T}_{\mathrm{B}})$
4	4.743e+00	4.640e+00	11	1.097e+01	8.011e+00
5	6.221e+00	6.024e+00	12	1.103e+01	8.034e+00
6	8.154e+00	6.860e+00	13	1.106e+01	8.046e+00
7	9.473e+00	7.396e+00	14	1.107e+01	8.051e+00
8	1.023e+01	7.707e+00	15	1.108e+01	8.054e+00
9	1.064e+01	7.876e+00	16	1.108e+01	8.056e+00
10	1.086e+01	7.965e+00			

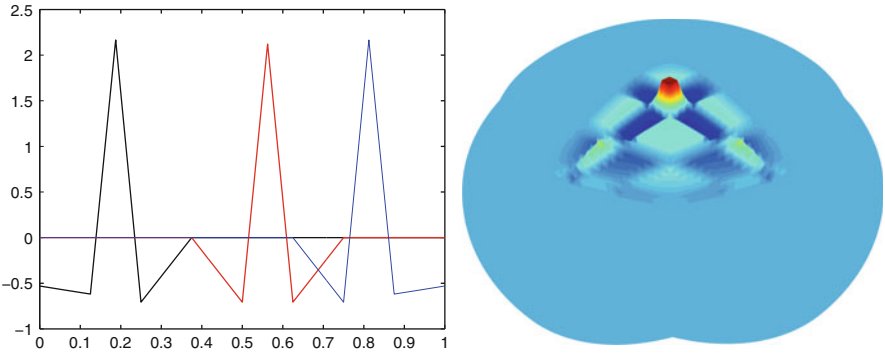


Fig. 3 Primal wavelets for $d = 2$ on [0, 1] (*left*) and on a sphere as constructed in [40] (*right*)

We see in Fig. 3 some primal biorthogonal wavelets of order $d = 2$ which consist of piecewise linear B-splines. These can be employed in the construction in wavelets on manifolds [24] which were optimized and implemented to construct biorthogonal wavelet bases on the sphere in [40], see the right graphic in Fig. 3.

3.3 *Elliptic Boundary Value Problems in Wavelet Coordinates*

We now derive a representation of the elliptic boundary value problem (6) in terms of (initially infinite) wavelet coordinates.

Let for $H = H_0^r(\Omega)$ the collection Ψ_H be a wavelet basis with corresponding dual $\tilde{\Psi}_H$ which satisfies the properties (R), (L) and (CP) from Sect. 3.1. Expanding the solution $u = \mathbf{u}^T \Psi_H$, the right hand side $f = \mathbf{f}^T \tilde{\Psi}_H$ and recalling the definition of A in (6), the wavelet representation of the elliptic boundary value problem (6) is given by

$$\mathbf{A}\mathbf{u} = \mathbf{f} \tag{77}$$

where

$$\mathbf{A} := a(\Psi_H, \Psi_H), \qquad \mathbf{f} := \langle \Psi_H, f \rangle. \tag{78}$$

Then the mapping property (5) and the Riesz basis property (R) yield the following fact.

Proposition 10 *The infinite matrix* $\mathbf{A}$ *is a boundedly invertible mapping from* $\ell_2 = \ell_2(\mathbb{I}_H)$ *into itself, and one has*

$$\|\mathbf{v}\| \sim \|\mathbf{A}\mathbf{v}\|, \qquad \mathbf{v} \in \ell_2. \tag{79}$$

The proof follows by combining the isomorphism relation (5) for the elliptic operator with the Riesz basis property (43). This entails the following consequence with respect to *preconditioning*. Let for $\mathbb{I} = \mathbb{I}_H$ the symbol Λ denote *any* finite subset of the index set $\mathbb{I}$. For the corresponding set of wavelets $\Psi_\Lambda := \{\psi_\lambda : \lambda \in \Lambda\}$, denote by $S_\Lambda := \operatorname{span} \Psi_\Lambda$ the respective finite-dimensional subspace of H. For the wavelet representation of A in terms of Ψ_Λ, $\mathbf{A}_\Lambda := a(\Psi_\Lambda, \Psi_\Lambda)$, we then have that $\kappa_2(\mathbf{A}_\Lambda) \sim 1$ independent of Λ, on account of the ellipticity of the operator. In other words, representations of A with respect to properly scaled wavelet bases for H entail well-conditioned system matrices $\mathbf{A}_\Lambda$ independent of Λ.

Fast wavelet transform. We briefly explain now in the situation of uniform refinements, i.e., when $S(\Phi_J) = S(\Psi^J)$, how the FWT $\mathbf{T}_J$ from (64) can be used for preconditioning linear elliptic operators, together with a diagonal scaling induced by the norm equivalence (76) [21]. We recall the notation from Sect. 3.2 where the wavelet basis is in fact the (unscaled) anchor basis from Sect. 3.1. Thus, the norm equivalence (36) using the scaled wavelet basis Ψ_H is the same as (76) in the anchor basis. Recall that the norm equivalence (76) implies that every $v \in H^s$ can be expanded uniquely in terms of the Ψ and its expansion coefficients $\mathbf{v}$ satisfy

$$\|v\|_{H^s} \sim \|\mathbf{D}^s \mathbf{v}\|_{\ell_2}$$

where $\mathbf{D}^s$ is a diagonal matrix with entries $\mathbf{D}^s_{(j,k),(j',k')} = 2^{sj}\delta_{j,j'}\delta_{k,k'}$. Depending on the order of the elliptic operator, we have $H \subset H^r(\Omega)$ for $r = 1$ or $r = 2$. We have, therefore, already identified the diagonal (scaling) matrix $\mathbf{D}_J$ consisting of the finite portion of the matrix $\mathbf{D} = \mathbf{D}^1$ for which $j_0 - 1 \le j \le J - 1$. The representation of A with respect to the (unscaled) wavelet basis Ψ^J can be expressed in terms of the FWT $\mathbf{T}_J$, that is,

$$\langle \Psi^J, A\Psi^J \rangle = \mathbf{T}_J^T \langle \Phi_J, A\Phi_J \rangle \mathbf{T}_J, \tag{80}$$

Table 5 Optimized spectral condition numbers of the operator $\mathbf{A}$ using tensor products of biorthogonal wavelets on the interval for space dimensions $n = 1, 2, 3$ [12]

j	$n = 1$	$n = 2$	$n = 3$
3	22.3	9.6	18.3
4	23.9	11.8	37.1
5	25.0	14.3	39.8
6	25.7	16.0	40.9
8	26.6	18.4	
10	27.1		
12	27.3		

where Φ_J is the single-scale basis for $S(\Psi^J)$. Thus, we first set up the operator equation as in finite element settings in terms of the single-scale basis Φ_J. Applying the FWT $\mathbf{T}_J$ together with $\mathbf{D}_J$ yields that the operator

$$\mathbf{A}_J := \mathbf{D}_J^{-1}\, \mathbf{T}_J^T\, \langle \Phi_J, A\Phi_J \rangle\, \mathbf{T}_J\, \mathbf{D}_J^{-1} \tag{81}$$

has uniformly bounded condition numbers independent of J. This can be seen by combining the properties of A according to (5) with the norm equivalences (36) and (40).

It is known that the boundary adaptations of the generators and wavelets aggravate the absolute values of the condition numbers. Nevertheless, these constants can be substantially reduced by an operator-adapted transformation which takes into account only the coarsest discretization level and, thus, is inexpensive [12]. Numerical tests confirm that the absolute constants can further be improved by taking instead of $\mathbf{D}_J^{-1}$ the inverse of the diagonal of $\langle \Psi^J, A\Psi^J \rangle$ for the scaling in (81) [12, 46].

In Table 5 we display the condition numbers for discretizations using the weak form of the elliptic operator $-\Delta + I$ on $(0,1)^n$ in up to three dimensions using boundary adapted biorthogonal spline-wavelets in the case $d = 2$, $\tilde{d} = 4$ with such a scaling and additional shifts of small eigenvalues which is an inexpensive operation [12].

4 Conclusion and Outlook

The central theme of this paper was to present optimal multilevel preconditioners which enable us to reduce the spectral condition number of the system matrix to be independent of the grid spacing h. Specifically in the context of isogeometric analysis in Sect. 2, the issue arises how the spectral condition numbers depend on the polynomial degree p; we have seen a corresponding behaviour already in Table 1. From a theoretical point of view, it is not clear yet how to estimate the dependence on p and how to remedy its influence.

Some recent results in this directions may be promising. In [29], a full geometric multigrid method was proposed for problems in isogeometric analysis with an increased number of smoothing steps to account for the dependence of the spectral condition number on the polynomial degree p. A different approach is presented with different multigrid approaches involving high-and low-order variants in [49].

I expect that the intergrid operators used in Sect. 2 will also be efficient for isogeometric collocation methods as presented in [2] although it is not clear how to prove optimality of the corresponding preconditioner.

This paper dealt with preconditioners for discretizations on uniform grids which provide best approximations for PDEs with smooth solutions. In case when the solution is not smooth, adaptivity may capture the optimal degrees of freedom when compared to a best N-term approximation. Also in an adaptive method one has to iteratively solve linear systems of equations which again requires a preconditioner. For a BPX-type or FWT preconditioner, the same principles can immediately be applied as long as the spaces generated in the adaptive process are nested as in (17). Of course, then the intergrid operators in Sect. 2.3 have to be adapted accordingly.

References

1. Y. Achdou, O. Pironneau, *Computational Methods for Option Pricing* (SIAM, Philadelphia, 2005)
2. F. Auriccio, L. Beirão da Veiga, T.J.R. Hughes, A. Reali, C. Sangalli, Isogeometric collocation methods. Math. Models Methods Appl. Sci. **20**(11), 2075–2107 (2010)
3. O. Axelsson, P.S. Vassilevski, Algebraic multilevel preconditioning methods I. Numer. Math. **56**, 157–177 (1989)
4. O. Axelsson, P.S. Vassilevski, Algebraic multilevel preconditioning methods II. SIAM J. Numer. Anal. **27**, 1569–1590 (1990)
5. L. Beirão da Veiga, A. Buffa, C. Sangalli, R. Vázquez, Mathematical analysis of variational isogeometric methods. Acta Numer. **23**, 157–287 (2014)
6. L. Beirão da Veiga, D. Cho, L.F. Pavarino, S. Scacchi, Overlapping Schwarz methods for isogeometric analysis. SIAM J. Numer. Anal. **50**(3), 1394–1416 (2012)
7. C. de Boor, *A Practical Guide to Splines*, rev. edn. (Springer, New York, 2001)
8. D. Braess, *Finite Elements: Theory, Fast Solvers and Applications in Solid Mechanics*, 2nd edn. (Cambridge University Press, Cambridge, 2001)
9. D. Braess, W. Hackbusch, A new convergence proof for the multigrid method including the V-cycle. SIAM J. Numer. Anal. **20**, 967–975 (1983)
10. J.H. Bramble, J.E. Pasciak, J. Xu, Parallel multilevel preconditioners. Math. Comput. **55**, 1–22 (1990)
11. A. Buffa, H. Harbrecht, A. Kunoth, G. Sangalli, BPX preconditioning for isogeometric analysis. Comput. Methods Appl. Mech. Eng. **265**, 63–70 (2013)
12. C. Burstedde, Fast optimised wavelet methods for control problems constrained by Elliptic PDEs, PhD Dissertation, Mathematisch-Naturwissenschaftliche Fakultät, Universität Bonn, Bonn, 2005, http://hss.ulb.uni-bonn.de/2005/0682/0682.htm
13. C. Canuto, A. Tabacco, K. Urban, The wavelet element method, part I: construction and analysis. Appl. Comput. Harm. Anal. **6**, 1–52 (1999)
14. N. Clarke, K. Parrot, Multigrid for American option pricing with stochastic volatility. Appl. Math. Financ. **6**, 177–195 (1999)

15. A. Cohen, *Numerical Analysis of Wavelet Methods*. Studies in Mathematics and its Applications, vol. 32 (Elsevier, Amsterdam/Boston, 2003)
16. A. Cohen, W. Dahmen, R. DeVore, Adaptive wavelet methods for elliptic operator equations—convergence rates. Math. Comput. **70**, 27–75 (2001)
17. A. Cohen, I. Daubechies, J.-C. Feauveau, Biorthogonal bases of compactly supported wavelets. Commun. Pure Appl. Math. **45**, 485–560 (1992)
18. W. Dahmen, Wavelet and multiscale methods for operator equations. Acta Numer. **6**, 55–228 (1997)
19. W. Dahmen, Wavelet methods for PDEs—some recent developments. J. Comput. Appl. Math. **128**, 133–185 (2001)
20. W. Dahmen, Multiscale and wavelet methods for operator equations, in *Multiscale Problems and Methods in Numerical Simulation*, ed. by C. Canuto. C.I.M.E. Lecture Notes in Mathematics, vol. 1825 (Springer, Heidelberg, 2003), pp. 31–96
21. W. Dahmen, A. Kunoth, Multilevel preconditioning. Numer. Math. **63**, 315–344 (1992)
22. W. Dahmen, A. Kunoth, K. Urban, Biorthogonal spline wavelets on the interval—stability and moment conditions. Appl. Comput. Harm. Anal. **6**, 132–196 (1999)
23. W. Dahmen, R. Schneider, Composite wavelet bases for operator equations. Math. Comput. **68**, 1533–1567 (1999)
24. W. Dahmen, R. Schneider, Wavelets on manifolds I: construction and domain decomposition. SIAM J. Math. Anal. **31**, 184–230 (1999)
25. I. Daubechies, Orthonormal bases of compactly supported wavelets. Commun. Pure Appl. Math. **41**, 909–996 (1988)
26. R.A. DeVore, A. Kunoth, *Multiscale, Nonlinear and Adaptive Approximations* (Springer, Heidelberg/New York, 2009)
27. K.P.S. Gahalaut, J.K. Kraus, S.K. Tomar, Multigrid methods for isogeometric discretization. Comput. Methods Appl. Mech. Eng. **253**(1), 413–425 (2013)
28. S.L. Heston, A closed-form solution for options with stochastic volatility with applications to bond and currency options. Rev. Financ. Stud. **6**(2), 327–343 (1993)
29. C. Hofreither, W. Zulehner, On full multigrid schemes for isogeometric analysis, NuMa-Report No. 2014–03, Johannes-Kepler-Universität Linz, May 2014, http://www.numa.uni-linz.ac.at/Publications/
30. M. Holtz, A. Kunoth, B-spline based monotone multigrid methods with an application to the pricing of American options, in *Multigrid, Multilevel and Multiscale Methods*, ed. by P. Wesseling, C.W. Oosterlee, P. Hemker. (Electr.) Proc. EMG, 2005
31. M. Holtz, A. Kunoth, B-spline based monotone multigrid methods. SIAM J. Numer. Anal. **45**(1), 1175–1199 (2007)
32. J.C. Hull, *Options, Futures and Other Derivatives* (Prentice Hall International Editions, Upper Saddle River, 2000)
33. S. Jaffard, Wavelet methods for fast resolution of elliptic problems. SIAM J. Numer. Anal. **29**, 965–986 (1992)
34. S.K. Kleiss, C. Pechstein, B. Jüttler, S. Tomar, IETI—isogeometric tearing and interconnecting. Comput. Methods Appl. Mech. Eng. **247–248**, 201–215 (2012)
35. R. Kornhuber, *Adaptive Monotone Multigrid Methods for Nonlinear Variational Problems* (Teubner, Stuttgart, 1997)
36. R. Kraft, Adaptive and linearly independent multilevel B-splines, in *Surface Fitting and Multiresolution Methods*, ed. by A. Le Méhauté, C. Rabut, L.L. Schumaker (Vanderbilt University Press, Nashville, 1997), pp. 209–218
37. A. Kunoth, *Multilevel Preconditioning* (Shaker-Verlag, Aachen, 1994)
38. A. Kunoth, *Wavelet Methods—Elliptic Boundary Value Problems and Control Problems*. Advances in Numerical Mathematics (Teubner, Stuttgart, 2001)
39. A. Kunoth, Optimized wavelet preconditioning, in *Multiscale, Nonlinear and Adaptive Approximations*, ed. by R.A. DeVore, A. Kunoth (Springer, Heidelberg/New York, 2009), pp. 325–378
40. A. Kunoth, J. Sahner, Wavelets on manifolds: an optimized construction. Math. Comput. **75**, 1319–1349 (2006)

41. A. Kunoth, Chr. Schneider, K. Wiechers, Multiscale methods for the valuation of American options with stochastic volatility. Int. J. Comput. Math. **89**(9), 1145–1163 (2012)
42. J. Maes, A. Bultheel, A hierarchical basis preconditioner for the biharmonic equation on the sphere. IMA J. Numer. Anal. **26**(3), 563–583 (2006)
43. J. Maes, A. Kunoth, A. Bultheel, BPX-type preconditioners for 2nd and 4th order elliptic problems on the sphere. SIAM J. Numer. Anal. **45**(1), 206–222 (2007)
44. C.W. Oosterlee, On multigrid for linear complementary problems with application to American-style options. Electron. Trans. Numer. Anal. **15**, 165–185 (2003)
45. P. Oswald, On discrete norm estimates related to multilevel preconditioners in the finite element method, in *Constructive Theory of Functions*, ed. by K.G. Ivanov, P. Petrushev, B. Sendov. Proceedings of the International Conference Varna 1991, Bulgarian Academy of Sciences, Sofia, 1992, pp. 203–214
46. R. Pabel, Wavelet methods for PDE constrained elliptic control problems with dirichlet boundary control, Diploma Thesis, Institut für Angewandte Mathematik, Universität Bonn (2006), Shaker-Verlag Aachen (2007). doi:10.2370/236_232
47. L.L. Schumaker, *Spline Functions: Basic Theory*. Cambridge Mathematical Library, 3rd edn. (Cambridge University Press, Cambridge, 2007)
48. R. Stevenson, Adaptive wavelet methods for solving operator equations: an overview, in *Multiscale, Nonlinear and Adaptive Approximations*, ed. by R.A. DeVore, A. Kunoth (Springer, Heidelberg/New York, 2009), pp. 543–598
49. H. Sundar, G. Stadler, G. Biros, Comparison of multigrid algorithms for high-order continuous finite element discretizations, submitted, 2014. [arXiv:1402.5938]
50. K. Urban, *Wavelet Methods for Elliptic Partial Differential Equations* (Oxford University Press, Oxford, 2009)
51. A.-V. Vuong, C. Giannelli, B. Jüttler, B. Simeon, A hierarchical approach to adaptive local refinement in isogeometric analysis. Comput. Methods Appl. Mech. Eng. **200**(49–52), 3554–3567 (2011)
52. J. Xu, Iterative methods by space decomposition and subspace correction. SIAM Rev. **34**(4), 581–613 (1992)
53. J. Xu, L. Chen, R.H. Nochetto, Optimal multilevel methods for H(grad), H(curl), and H(div) systems on graded and unstructured grids, in *Multiscale, Nonlinear and Adaptive Approximations*, ed. by R.A. DeVore, A. Kunoth (Springer, Heidelberg/New York, 2009), pp. 599–659.
54. H. Yserentant, On the multilevel splitting of finite element spaces. Numer. Math. **49**, 379–412 (1986)

Editorial Policy

1. Volumes in the following three categories will be published in LNCSE:

i) Research monographs
ii) Tutorials
iii) Conference proceedings

Those considering a book which might be suitable for the series are strongly advised to contact the publisher or the series editors at an early stage.

2. Categories i) and ii). Tutorials are lecture notes typically arising via summer schools or similar events, which are used to teach graduate students. These categories will be emphasized by Lecture Notes in Computational Science and Engineering. **Submissions by interdisciplinary teams of authors are encouraged.** The goal is to report new developments – quickly, informally, and in a way that will make them accessible to non-specialists. In the evaluation of submissions timeliness of the work is an important criterion. Texts should be well-rounded, well-written and reasonably self-contained. In most cases the work will contain results of others as well as those of the author(s). In each case the author(s) should provide sufficient motivation, examples, and applications. In this respect, Ph.D. theses will usually be deemed unsuitable for the Lecture Notes series. Proposals for volumes in these categories should be submitted either to one of the series editors or to Springer-Verlag, Heidelberg, and will be refereed. A provisional judgement on the acceptability of a project can be based on partial information about the work: a detailed outline describing the contents of each chapter, the estimated length, a bibliography, and one or two sample chapters – or a first draft. A final decision whether to accept will rest on an evaluation of the completed work which should include

– at least 100 pages of text;
– a table of contents;
– an informative introduction perhaps with some historical remarks which should be accessible to readers unfamiliar with the topic treated;
– a subject index.

3. Category iii). Conference proceedings will be considered for publication provided that they are both of exceptional interest and devoted to a single topic. One (or more) expert participants will act as the scientific editor(s) of the volume. They select the papers which are suitable for inclusion and have them individually refereed as for a journal. Papers not closely related to the central topic are to be excluded. Organizers should contact the Editor for CSE at Springer at the planning stage, see *Addresses* below.

In exceptional cases some other multi-author-volumes may be considered in this category.

4. Only works in English will be considered. For evaluation purposes, manuscripts may be submitted in print or electronic form, in the latter case, preferably as pdf- or zipped ps-files. Authors are requested to use the LaTeX style files available from Springer at http://www.springer.com/gp/authors-editors/book-authors-editors/manuscript-preparation/5636 (Click on LaTeX Template → monographs or contributed books).

For categories ii) and iii) we strongly recommend that all contributions in a volume be written in the same LaTeX version, preferably LaTeX2e. Electronic material can be included if appropriate. Please contact the publisher.

Careful preparation of the manuscripts will help keep production time short besides ensuring satisfactory appearance of the finished book in print and online.

5. The following terms and conditions hold. Categories i), ii) and iii):

Authors receive 50 free copies of their book. No royalty is paid.
Volume editors receive a total of 50 free copies of their volume to be shared with authors, but no royalties.

Authors and volume editors are entitled to a discount of 33.3 % on the price of Springer books purchased for their personal use, if ordering directly from Springer.

6. Springer secures the copyright for each volume.

Addresses:

Timothy J. Barth
NASA Ames Research Center
NAS Division
Moffett Field, CA 94035, USA
barth@nas.nasa.gov

Michael Griebel
Institut für Numerische Simulation
der Universität Bonn
Wegelerstr. 6
53115 Bonn, Germany
griebel@ins.uni-bonn.de

David E. Keyes
Mathematical and Computer Sciences
and Engineering
King Abdullah University of Science
and Technology
P.O. Box 55455
Jeddah 21534, Saudi Arabia
david.keyes@kaust.edu.sa

and

Department of Applied Physics
and Applied Mathematics
Columbia University
500 W. 120 th Street
New York, NY 10027, USA
kd2112@columbia.edu

Risto M. Nieminen
Department of Applied Physics
Aalto University School of Science
and Technology
00076 Aalto, Finland
risto.nieminen@aalto.fi

Dirk Roose
Department of Computer Science
Katholieke Universiteit Leuven
Celestijnenlaan 200A
3001 Leuven-Heverlee, Belgium
dirk.roose@cs.kuleuven.be

Tamar Schlick
Department of Chemistry
and Courant Institute
of Mathematical Sciences
New York University
251 Mercer Street
New York, NY 10012, USA
schlick@nyu.edu

Editor for Computational Science
and Engineering at Springer:
Martin Peters
Springer-Verlag
Mathematics Editorial IV
Tiergartenstrasse 17
69121 Heidelberg, Germany
martin.peters@springer.com

Lecture Notes in Computational Science and Engineering

1. D. Funaro, *Spectral Elements for Transport-Dominated Equations.*
2. H.P. Langtangen, *Computational Partial Differential Equations.* Numerical Methods and Diffpack Programming.
3. W. Hackbusch, G. Wittum (eds.), *Multigrid Methods V.*
4. P. Deuflhard, J. Hermans, B. Leimkuhler, A.E. Mark, S. Reich, R.D. Skeel (eds.), *Computational Molecular Dynamics: Challenges, Methods, Ideas.*
5. D. Kröner, M. Ohlberger, C. Rohde (eds.), *An Introduction to Recent Developments in Theory and Numerics for Conservation Laws.*
6. S. Turek, *Efficient Solvers for Incompressible Flow Problems.* An Algorithmic and Computational Approach.
7. R. von Schwerin, ***M**ulti **B**ody **S**ystem **SIM**ulation.* Numerical Methods, Algorithms, and Software.
8. H.-J. Bungartz, F. Durst, C. Zenger (eds.), *High Performance Scientific and Engineering Computing.*
9. T.J. Barth, H. Deconinck (eds.), *High-Order Methods for Computational Physics.*
10. H.P. Langtangen, A.M. Bruaset, E. Quak (eds.), *Advances in Software Tools for Scientific Computing*.
11. B. Cockburn, G.E. Karniadakis, C.-W. Shu (eds.), *Discontinuous Galerkin Methods.* Theory, Computation and Applications.
12. U. van Rienen, *Numerical Methods in Computational Electrodynamics.* Linear Systems in Practical Applications.
13. B. Engquist, L. Johnsson, M. Hammill, F. Short (eds.), *Simulation and Visualization on the Grid.*
14. E. Dick, K. Riemslagh, J. Vierendeels (eds.), *Multigrid Methods VI.*
15. A. Frommer, T. Lippert, B. Medeke, K. Schilling (eds.), *Numerical Challenges in Lattice Quantum Chromodynamics.*
16. J. Lang, *Adaptive Multilevel Solution of Nonlinear Parabolic PDE Systems.* Theory, Algorithm, and Applications.
17. B.I. Wohlmuth, *Discretization Methods and Iterative Solvers Based on Domain Decomposition.*
18. U. van Rienen, M. Günther, D. Hecht (eds.), *Scientific Computing in Electrical Engineering.*
19. I. Babuška, P.G. Ciarlet, T. Miyoshi (eds.), *Mathematical Modeling and Numerical Simulation in Continuum Mechanics.*
20. T.J. Barth, T. Chan, R. Haimes (eds.), *Multiscale and Multiresolution Methods.* Theory and Applications.
21. M. Breuer, F. Durst, C. Zenger (eds.), *High Performance Scientific and Engineering Computing.*
22. K. Urban, *Wavelets in Numerical Simulation.* Problem Adapted Construction and Applications.
23. L.F. Pavarino, A. Toselli (eds.), *Recent Developments in Domain Decomposition Methods.*

24. T. Schlick, H.H. Gan (eds.), *Computational Methods for Macromolecules: Challenges and Applications.*

25. T.J. Barth, H. Deconinck (eds.), *Error Estimation and Adaptive Discretization Methods in Computational Fluid Dynamics*.

26. M. Griebel, M.A. Schweitzer (eds.), *Meshfree Methods for Partial Differential Equations*.

27. S. Müller, *Adaptive Multiscale Schemes for Conservation Laws*.

28. C. Carstensen, S. Funken, W. Hackbusch, R.H.W. Hoppe, P. Monk (eds.), *Computational Electromagnetics*.

29. M.A. Schweitzer, *A Parallel Multilevel Partition of Unity Method for Elliptic Partial Differential Equations*.

30. T. Biegler, O. Ghattas, M. Heinkenschloss, B. van Bloemen Waanders (eds.), *Large-Scale PDE-Constrained Optimization*.

31. M. Ainsworth, P. Davies, D. Duncan, P. Martin, B. Rynne (eds.), *Topics in Computational Wave Propagation*. Direct and Inverse Problems.

32. H. Emmerich, B. Nestler, M. Schreckenberg (eds.), *Interface and Transport Dynamics.* Computational Modelling.

33. H.P. Langtangen, A. Tveito (eds.), *Advanced Topics in Computational Partial Differential Equations.* Numerical Methods and Diffpack Programming.

34. V. John, *Large Eddy Simulation of Turbulent Incompressible Flows.* Analytical and Numerical Results for a Class of LES Models.

35. E. Bänsch (ed.), *Challenges in Scientific Computing - CISC 2002.*

36. B.N. Khoromskij, G. Wittum, *Numerical Solution of Elliptic Differential Equations by Reduction to the Interface.*

37. A. Iske, *Multiresolution Methods in Scattered Data Modelling.*

38. S.-I. Niculescu, K. Gu (eds.), *Advances in Time-Delay Systems.*

39. S. Attinger, P. Koumoutsakos (eds.), *Multiscale Modelling and Simulation.*

40. R. Kornhuber, R. Hoppe, J. Périaux, O. Pironneau, O. Wildlund, J. Xu (eds.), *Domain Decomposition Methods in Science and Engineering.*

41. T. Plewa, T. Linde, V.G. Weirs (eds.), *Adaptive Mesh Refinement – Theory and Applications.*

42. A. Schmidt, K.G. Siebert, *Design of Adaptive Finite Element Software.* The Finite Element Toolbox ALBERTA.

43. M. Griebel, M.A. Schweitzer (eds.), *Meshfree Methods for Partial Differential Equations II.*

44. B. Engquist, P. Lötstedt, O. Runborg (eds.), *Multiscale Methods in Science and Engineering.*

45. P. Benner, V. Mehrmann, D.C. Sorensen (eds.), *Dimension Reduction of Large-Scale Systems.*

46. D. Kressner, *Numerical Methods for General and Structured Eigenvalue Problems.*

47. A. Boriçi, A. Frommer, B. Joó, A. Kennedy, B. Pendleton (eds.), *QCD and Numerical Analysis III*.

48. F. Graziani (ed.), *Computational Methods in Transport.*

49. B. Leimkuhler, C. Chipot, R. Elber, A. Laaksonen, A. Mark, T. Schlick, C. Schütte, R. Skeel (eds.), *New Algorithms for Macromolecular Simulation.*

50. M. Bücker, G. Corliss, P. Hovland, U. Naumann, B. Norris (eds.), *Automatic Differentiation: Applications, Theory, and Implementations.*

51. A.M. Bruaset, A. Tveito (eds.), *Numerical Solution of Partial Differential Equations on Parallel Computers.*

52. K.H. Hoffmann, A. Meyer (eds.), *Parallel Algorithms and Cluster Computing.*

53. H.-J. Bungartz, M. Schäfer (eds.), *Fluid-Structure Interaction.*

54. J. Behrens, *Adaptive Atmospheric Modeling.*

55. O. Widlund, D. Keyes (eds.), *Domain Decomposition Methods in Science and Engineering XVI.*

56. S. Kassinos, C. Langer, G. Iaccarino, P. Moin (eds.), *Complex Effects in Large Eddy Simulations.*

57. M. Griebel, M.A Schweitzer (eds.), *Meshfree Methods for Partial Differential Equations III.*

58. A.N. Gorban, B. Kégl, D.C. Wunsch, A. Zinovyev (eds.), *Principal Manifolds for Data Visualization and Dimension Reduction.*

59. H. Ammari (ed.), *Modeling and Computations in Electromagnetics: A Volume Dedicated to Jean-Claude Nédélec.*

60. U. Langer, M. Discacciati, D. Keyes, O. Widlund, W. Zulehner (eds.), *Domain Decomposition Methods in Science and Engineering XVII.*

61. T. Mathew, *Domain Decomposition Methods for the Numerical Solution of Partial Differential Equations.*

62. F. Graziani (ed.), *Computational Methods in Transport: Verification and Validation.*

63. M. Bebendorf, *Hierarchical Matrices.* A Means to Efficiently Solve Elliptic Boundary Value Problems.

64. C.H. Bischof, H.M. Bücker, P. Hovland, U. Naumann, J. Utke (eds.), *Advances in Automatic Differentiation.*

65. M. Griebel, M.A. Schweitzer (eds.), *Meshfree Methods for Partial Differential Equations IV*.

66. B. Engquist, P. Lötstedt, O. Runborg (eds.), *Multiscale Modeling and Simulation in Science.*

67. I.H. Tuncer, Ü. Gülcat, D.R. Emerson, K. Matsuno (eds.), *Parallel Computational Fluid Dynamics 2007.*

68. S. Yip, T. Diaz de la Rubia (eds.), *Scientific Modeling and Simulations*.

69. A. Hegarty, N. Kopteva, E. O'Riordan, M. Stynes (eds.), *BAIL* 2008 – *Boundary and Interior Layers.*

70. M. Bercovier, M.J. Gander, R. Kornhuber, O. Widlund (eds.), *Domain Decomposition Methods in Science and Engineering XVIII.*

71. B. Koren, C. Vuik (eds.), *Advanced Computational Methods in Science and Engineering.*

72. M. Peters (ed.), *Computational Fluid Dynamics for Sport Simulation.*

73. H.-J. Bungartz, M. Mehl, M. Schäfer (eds.), *Fluid Structure Interaction II - Modelling, Simulation, Optimization.*

74. D. Tromeur-Dervout, G. Brenner, D.R. Emerson, J. Erhel (eds.), *Parallel Computational Fluid Dynamics 2008.*

75. A.N. Gorban, D. Roose (eds.), *Coping with Complexity: Model Reduction and Data Analysis.*

76. J.S. Hesthaven, E.M. Rønquist (eds.), *Spectral and High Order Methods for Partial Differential Equations.*

77. M. Holtz, *Sparse Grid Quadrature in High Dimensions with Applications in Finance and Insurance*.

78. Y. Huang, R. Kornhuber, O.Widlund, J. Xu (eds.), *Domain Decomposition Methods in Science and Engineering XIX*.

79. M. Griebel, M.A. Schweitzer (eds.), *Meshfree Methods for Partial Differential Equations V*.

80. P.H. Lauritzen, C. Jablonowski, M.A. Taylor, R.D. Nair (eds.), *Numerical Techniques for Global Atmospheric Models*.

81. C. Clavero, J.L. Gracia, F.J. Lisbona (eds.), *BAIL 2010 – Boundary and Interior Layers, Computational and Asymptotic Methods*.

82. B. Engquist, O. Runborg, Y.R. Tsai (eds.), *Numerical Analysis and Multiscale Computations*.

83. I.G. Graham, T.Y. Hou, O. Lakkis, R. Scheichl (eds.), *Numerical Analysis of Multiscale Problems*.

84. A. Logg, K.-A. Mardal, G. Wells (eds.), *Automated Solution of Differential Equations by the Finite Element Method*.

85. J. Blowey, M. Jensen (eds.), *Frontiers in Numerical Analysis - Durham 2010*.

86. O. Kolditz, U.-J. Gorke, H. Shao, W. Wang (eds.), *Thermo-Hydro-Mechanical-Chemical Processes in Fractured Porous Media - Benchmarks and Examples*.

87. S. Forth, P. Hovland, E. Phipps, J. Utke, A. Walther (eds.), *Recent Advances in Algorithmic Differentiation*.

88. J. Garcke, M. Griebel (eds.), *Sparse Grids and Applications*.

89. M. Griebel, M.A. Schweitzer (eds.), *Meshfree Methods for Partial Differential Equations VI*.

90. C. Pechstein, *Finite and Boundary Element Tearing and Interconnecting Solvers for Multiscale Problems*.

91. R. Bank, M. Holst, O. Widlund, J. Xu (eds.), *Domain Decomposition Methods in Science and Engineering XX*.

92. H. Bijl, D. Lucor, S. Mishra, C. Schwab (eds.), *Uncertainty Quantification in Computational Fluid Dynamics*.

93. M. Bader, H.-J. Bungartz, T. Weinzierl (eds.), *Advanced Computing*.

94. M. Ehrhardt, T. Koprucki (eds.), *Advanced Mathematical Models and Numerical Techniques for Multi-Band Effective Mass Approximations*.

95. M. Azaïez, H. El Fekih, J.S. Hesthaven (eds.), *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2012*.

96. F. Graziani, M.P. Desjarlais, R. Redmer, S.B. Trickey (eds.), *Frontiers and Challenges in Warm Dense Matter*.

97. J. Garcke, D. Pflüger (eds.), *Sparse Grids and Applications – Munich 2012*.

98. J. Erhel, M. Gander, L. Halpern, G. Pichot, T. Sassi, O. Widlund (eds.), *Domain Decomposition Methods in Science and Engineering XXI*.

99. R. Abgrall, H. Beaugendre, P.M. Congedo, C. Dobrzynski, V. Perrier, M. Ricchiuto (eds.), *High Order Nonlinear Numerical Methods for Evolutionary PDEs - HONOM 2013*.

100. M. Griebel, M.A. Schweitzer (eds.), *Meshfree Methods for Partial Differential Equations VII*.

101. R. Hoppe (ed.), *Optimization with PDE Constraints - OPTPDE 2014*.

102. S. Dahlke, W. Dahmen, M. Griebel, W. Hackbusch, K. Ritter, R. Schneider, C. Schwab, H. Yserentant (eds.), *Extraction of Quantifiable Information from Complex Systems*.

103. A. Abdulle, S. Deparis, D. Kressner, F. Nobile, M. Picasso (eds.), *Numerical Mathematics and Advanced Applications - ENUMATH 2013*.

104. T. Dickopf, M.J. Gander, L. Halpern, R. Krause, L.F. Pavarino (eds.), *Domain Decomposition Methods in Science and Engineering XXII*.

105. M. Mehl, M. Bischoff, M. Schäfer (eds.), *Recent Trends in Computational Engineering - CE2014*. Optimization, Uncertainty, Parallel Algorithms, Coupled and Complex Problems.

106. R.M. Kirby, M. Berzins, J.S. Hesthaven (eds.), *Spectral and High Order Methods for Partial Differential Equations - ICOSAHOM '14*.

107. B. Simeon, B. Jüttler (eds.), *Isogeometric Analysis and Applications 2014*.

For further information on these books please have a look at our mathematics catalogue at the following URL: www.springer.com/series/3527

Monographs in Computational Science and Engineering

1. J. Sundnes, G.T. Lines, X. Cai, B.F. Nielsen, K.-A. Mardal, A. Tveito, *Computing the Electrical Activity in the Heart.*

For further information on this book, please have a look at our mathematics catalogue at the following URL: www.springer.com/series/7417

Texts in Computational Science and Engineering

1. H. P. Langtangen, *Computational Partial Differential Equations.* Numerical Methods and Diffpack Programming. 2nd Edition
2. A. Quarteroni, F. Saleri, P. Gervasio, *Scientific Computing with MATLAB and Octave.* 4th Edition
3. H. P. Langtangen, *Python Scripting for Computational Science*. 3rd Edition
4. H. Gardner, G. Manduchi, *Design Patterns for e-Science*.
5. M. Griebel, S. Knapek, G. Zumbusch, *Numerical Simulation in Molecular Dynamics*.
6. H. P. Langtangen, *A Primer on Scientific Programming with Python.* 4th Edition
7. A. Tveito, H. P. Langtangen, B. F. Nielsen, X. Cai, *Elements of Scientific Computing.*
8. B. Gustafsson, *Fundamentals of Scientific Computing.*
9. M. Bader, *Space-Filling Curves.*
10. M. Larson, F. Bengzon, *The Finite Element Method: Theory, Implementation and Applications.*
11. W. Gander, M. Gander, F. Kwok, *Scientific Computing: An Introduction using Maple and MATLAB.*
12. P. Deuflhard, S. Röblitz, *A Guide to Numerical Modelling in Systems Biology*.

For further information on these books please have a look at our mathematics catalogue at the following URL: www.springer.com/series/5151