Scan-based immersed isogeometric analysis

Sai Chandana Divi

The work in this dissertation has been funded by the European Commission EACEA Agency, Framework Partnership Agreement 2013-0043 Erasmus Mundus Action 1b, as a part of the EM Joint Doctorate Simulation in Engineering and Entrepreneurship Development (SEED).

Cover design by Sai Chandana Divi.
Printed in The Netherlands by Gildeprint.

A catalogue record is available from the Eindhoven University of Technology Library
ISBN: 978-90-386-5469-0.

Scan-based immersed isogeometric analysis

PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Technische Universiteit Eindhoven,
op gezag van de rector magnificus prof.dr.ir. F.P.T. Baaijens,
voor een commissie aangewezen door het College voor Promoties,
in het openbaar te verdedigen op vrijdag 18 maart 2022 om 11:00 uur

door

Sai Chandana Divi

geboren te Tirupati, India

Dit proefschrift is goedgekeurd door de promotoren en de samenstelling van de promotiecommissie is als volgt:

voorzitter:	prof.dr.ir.	D.M.J. Smeulders
1e promotor:	dr.ir.	C.V. Verhoosel
2e promotor:	prof.dr.	A. Reali (Università degli Studi di Pavia)
copromotor:	prof.dr.ir.	E.H. van Brummelen
leden:	prof.dr.	F. Auricchio (Università degli Studi di Pavia)
	prof.dr.-ing.habil	A. Düster (Technische Universität Hamburg)
	prof.dr.ir.	F.N. van de Vosse
	dr.rer.nat.	M. Möller (Technische Universiteit Delft)

Het onderzoek of ontwerp dat in dit proefschrift wordt beschreven is uitgevoerd in overeenstemming met de TU/e Gedragscode Wetenschapsbeoefening.

Preface

Computational mechanics deals with the development of computational models for simulating the physical and mechanical behavior of materials, structures, fluids and fluid-structure interactions. Efficient algorithms, robust solvers and advanced discretization techniques play a vital role in these simulations. In order to strengthen the theoretical understanding of these aspects, the European Commission under the European Education and Culture Executive Agency (EACEA) initiated the Erasmus Mundus joint doctorate program, called Simulation in Engineering and Entrepreneurship Development (SEED), coordinated by the International Centre for Numerical Methods in Engineering (CIMNE).

The SEED program brings universities, research institutes and companies together to collaborate and increase the horizon of research in the field of computational mechanics. This dissertation is part of this program and records the research that I have conducted between 2017 and 2021, as a joint doctoral candidate of the Energy Technology and Fluid Dynamics group at Eindhoven University of Technology (TU/e) and the Computational Mechanics and Advanced Materials group at the University of Pavia (UniPv). This work is supervised by Clemens Verhoosel, Harald van Brummelen, Alessandro Reali and Ferdinando Auricchio.

I was first introduced to computational mechanics at the Indian Institute of Technology Madras (IITM) in 2013, when I had to choose a topic for my summer internship under the supervision of Ratna Kumar Annabattula. Initially, I was attracted by the prediction of mechanical behaviour using colorful simulation results produced by commercial software (*e.g.*, Ansys, Abaqus). This interest made me continue the summer internship project as my bachelor's thesis project. During this phase, I gained experience in performing simulations and predicting mechanical behavior of materials. However, I did not have a theoretical understanding of the methodology at the back-end of the commercial packages I employed.

Due to my desire for learning about the numerical aspects behind simulations, I was enthused to pursue a joint Master's program in computational mechanics

at Swansea University and Universitat Politècnica de Catalunya under the Erasmus Mundus scholarship program. During the Master program, I realized that research work on biomedical applications with an emphasis on methodology is something that keeps me curious and content. While writing my Master's Thesis, I received an Erasmus Mundus joint doctorate fellowship to pursue this dissertation at TU/e and UniPv. In 2017, after obtaining my Master's degree, I started a PhD position on addressing certain challenges of immersed isogeometric analysis in the context of scan-based simulations.

Over the last four years, the potential of immersed isogeometric analysis has become obvious to me. Through this dissertation, I have been lucky enough to be able to cover a few aspects of the field with Clemens and Harald. Collaboration is an inherent part of the SEED program giving me the chance to have discussions with experts at Pavia, *i.e.*, Ale and Auricchio, and to understand the biomechanical aspect of this project with the help of biomedical experts in Pavia.

Pursuing a PhD has been a great step to improve myself professionally and personally. My PhD ends with this dissertation, however, my research work in the field of computational mechanics continues beyond this.

Sai Chandana Divi
08-12-2021

Voor mijn ouders & Ellen

Contents

Chapter 1

Introduction

Over the past decades, the rapid developments in the field of scientific computing have provided the computational mechanics community with powerful means to develop and apply numerical techniques for mathematical-physical modeling. Since the 1960s, the Finite Element Method (FEM) has been a numerical technique that lies at the frontier of current industrial applications for a myriad of problems, for example in solid mechanics, fluid dynamics, and multi-physics modeling [1–8].

One of the reasons for the success of the finite element method is that its application to the discretization of (partial) differential equations (PDEs) through Galerkin's method is supported by rigorous mathematical understanding of the technique [9–11]. As a result, approximation inaccuracies (*i.e.*, the difference between the finite element solution and the exact solution to the PDE) – resulting from, for example, geometry simplifications because of meshing, the choice of basis functions, and implementation aspects such as numerical integration – are very well understood. This, in combination with a good understanding of how such aspects influence the computational effort, makes that the FEM provides a robust workflow for high-fidelity simulations of complex problems.

1.1 Scan-based immersed isogeometric analysis

In design-based computational analyses, the geometric model is most frequently constructed by a Computer Aided Design (CAD) tool (*e.g.*, AutoCAD, Solidworks, CATIA). There are also many applications – for example in biomechanics, geomechanics, material science, microstructural analysis, and many more – where the analysis pertains to the non-engineered physical geometry, which can be modeled through scan data obtained using advanced scanning technologies

(*e.g.*, tomography or photogrammetry). For such scan-based simulations, the data sets from which the geometric model is to be constructed are typically very large, and the obtained models can be very complex in terms of both geometry and topology (see Fig. 1.1). In the context of standard finite element analyses, scan-based simulations require an image segmentation technique to produce high-quality analysis-suitable meshes that fit to the boundaries of the domain of interest. The construction of a FEM-suitable computational domain can be an error-prone and laborious process, involving geometry clean-up and mesh repairing and optimization operations. Such operations can account for the majority of the total computational analysis time and form a bottleneck in the automation of the scan-based simulation workflow [12–14].

The challenges associated with the simulation workflow for complex problems sparked the development of the IsoGeometric Analysis (IGA) paradigm by Hughes and co-workers in 2005 [16]. The pivotal idea of IGA is to directly employ the geometry interpolation functions used in computer-aided design for the discretization of partial differential equations, thereby circumventing the problems associated with the construction of (high-quality) meshes. In terms of basis function technology, IGA builds on fundamental contributions in the field of CAD, such as B-splines [17], Non-uniform Rational B-splines (NURBS) [18], and subdivision surfaces [19, 20]. In IGA, the traditional FEM basis functions (C^0-continuous Lagrange functions) are replaced by higher-order (continuous) splines. Besides the advantage of eliminating mesh-approximation errors, for many (smooth) problems, the use of higher-order continuous splines for the approximation of the solution has been demonstrated to yield efficient isogeometric discretizations, in the sense that the same accuracy as for traditional finite elements is obtained using fewer degrees of freedom [21, 22]. Prominent successful applications of IGA can be found in, for example, Refs. [22–32].

While isogeometric analysis has been successfully applied to complex three-dimensional problems based on multi-patch CAD objects [33–41], the direct application of spline discretizations to scan-based simulations is hindered by the absence of a boundary-fitting volumetric spline object. Although spline preprocessors have been developed over the years for a range of applications [42–46], the robust generation of analysis-suitable mesh-fitting volumetric splines for scan-based analyses is beyond the scope of the current tools on account of the geometric and topological complexity typically inherent to scan data. To still be able to leverage the advantageous approximation properties of splines in scan-based simulations, IGA has been used in combination with immersed methods. In immersed methods – introduced by Peskin in 1972 [47] – a non-boundary-fitting mesh is considered, in which the computational domain is submersed. Since the immersed domain does not align with the computational grid, some of the elements in the grid are cut by the immersed boundary and require a special

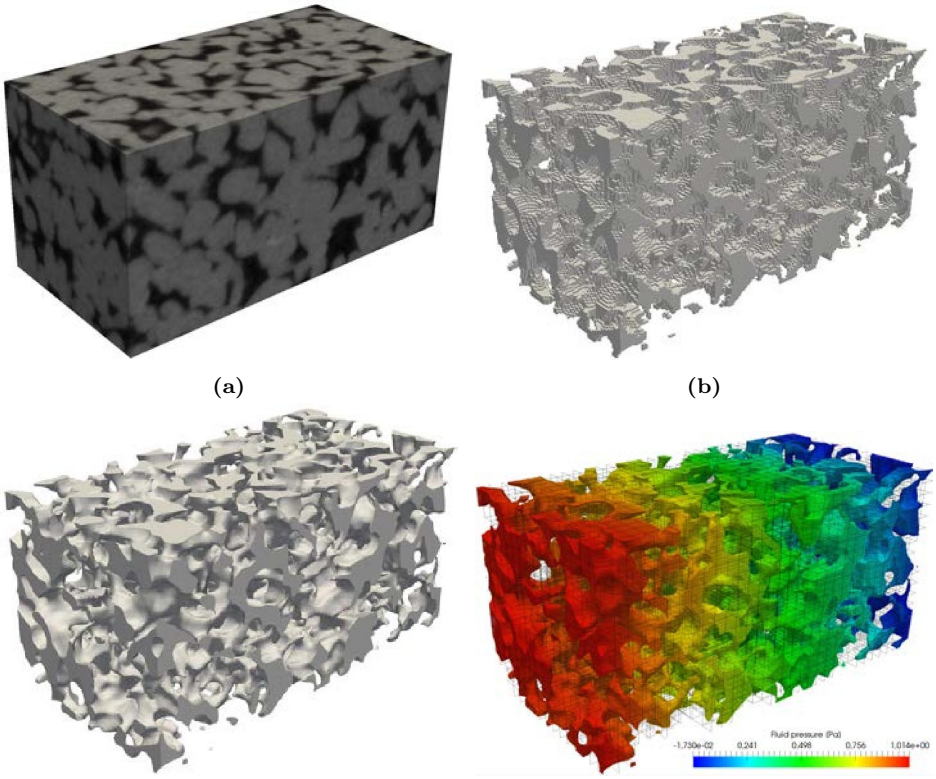


Figure 1.1: Illustration of a scan-based simulation workflow for a sintered glass specimen. The grayscale scan data for a specimen of $200 \times 100 \times 100$ voxels is shown in panel (a). The directly segmented voxel image (*i.e.*, all voxels above a certain grayscale threshold) representing the void space is shown in panel (b). A smooth reconstruction of the geometry based on a B-spline segmentation is shown in panel (c). This immersed geometry is constructed by the application of a recursive bisectioning strategy on a $32 \times 16 \times 16$ mesh, with the octree depth set equal to one. The pressure field from an immersed isogeometric Stokes flow simulation is shown in panel (d). See Ref. [15] for details.

treatment. The immersed approach has been considered in the setting of the finite element method by, for example, the Finite Cell Method (FCM) [14, 48, 49] and CutFEM [50, 51]. The immersed concept has also been used in combination with IGA [52–55], a strategy which is sometimes referred to as immersogeometric analysis [56, 57]. The versatility of immersed isogeometric analysis techniques with respect to the geometry representation – in the sense that the analysis procedure is not strongly affected by the complexity of the physical domain – makes it particularly attractive in the scan-based analysis setting. Applications can be found in, for example, the modeling of trabecular bone [58–60], porous media [15], coated metal foams [61], metal castings [62] and in additive manufacturing [63].

1.2 Challenges and recent advancements in immersed isogeometric analysis

Although immersed (isogeometric) analysis fits well in the general framework of Galerkin methods, compared to the traditional finite element method or boundary-fitting isogeometric analysis, various aspects require special treatment. Below we discuss the three most prominent challenges with respect to immersed (isogeometric) analysis, *viz.*: *(i)* the integration of cut elements; *(ii)* the imposition of Dirichlet boundary conditions on the immersed boundary; and *(iii)* the stability and conditioning of the discrete system of equations.

1.2.1 Intergration of cut elements

A well known computational challenge that is inherent to non-boundary-fitting finite element methods is the integration over cut elements, which is also a prominent problem in enriched finite element methods such as XFEM and GFEM due to the presence of discontinuous integrands (see Ref. [64] for a review). In immersed methods, the integrands over the computational domain are computed by the integration procedure on a mesh replicating the geometry rather than on the ambient mesh in which the domain is immersed.

Various integration procedures addressing the integration aspect have been developed over the years in the context of immersed FEM (see Ref. [14] for a review). The most prominent solution for the integration of cut elements is to employ an octree subdivision strategy. The concept of octree (or quadtree in two dimensions) integration is to recursively bisect a cut element, resulting in a set of sub-cells that covers the computational domain and its immersed boundary. The number of recursions is a user input parameter which influences the geometric approximation. This method was proposed in Ref. [49] in the context of the finite cell method. Due to its simplicity and robustness, it has been widely used

in immersed methods, *e.g.*, Refs. [14,15,56,58,65]. Despite various enhancements to the procedure (*e.g.*, the consideration of tetrahedral cells [66, 67], tessellation of the lowest level of bisectioning [58], adaptive integration schemes for subcells [68]), the main downside of the octree strategy is the computational effort associated with the large number of integration points that are typically required, especially in three-dimensional problems.

A common alternative integration strategy for cut elements is to perform (higher-order accurate) integration by projecting a reference element onto the (possibly curved) immersed boundary. This strategy has been exploited in the NURBS-enhanced finite element method (NEFEM) [69], in the context of implicitly defined geometries through a level set function [70], and in the finite cell method as part of a smart octree integration technique [71]. It is also possible to construct efficient integration rules for polynomial functions on cut elements. The moment fitting method [72] and the equivalent polynomial technique [73] are prominent examples of integration solutions based on this idea. Finally, in some cases it is possible to convert volumetric integrals into boundary integrals, which can reduce the required number of integration points. This technique has been exploited in the immersed setting in Refs. [74, 75]. All these alternative integration strategies can outperform the octree strategy in terms of efficiency (accuracy versus computational complexity), but, in general, they are less versatile and robust than the octree strategy, and their implementation is more involved.

1.2.2 Imposition of Dirichlet boundary conditions

In the boundary-fitting FEM setting, Dirichlet (or essential) boundary conditions are typically strongly imposed, *i.e.*, the degrees of freedom at the Dirichlet boundary are constrained to match the boundary data and the corresponding test functions are eliminated from the system of equations. Due to the disparity between the background grid and the physical domain in the immersed setting, the degrees of freedom are not associated with basis functions that are interpolatory on the immersed boundary. This prevents the strong imposition of Dirichlet conditions as used for boundary-fitting discretizations. Provided that a parametrization of the immersed boundary is available, Neumann (or natural) conditions can be imposed in the standard way, *i.e.*, through boundary integrals in the weak formulation.

In immersed finite element methods, Dirichlet boundary conditions are typically enforced weakly. One of the prominent approaches for this is the penalty method, in which one penalizes the difference between the evaluated discrete solution and the prescribed boundary data by adding a term to the weak form operator. This method has been used extensively to impose Dirichlet conditions

in the finite cell method (*e.g.*, Ref. [49]). In its standard form, the penalty method results in an inconsistent formulation, in the sense that the approximate solution generally does not converge to the exact solution. However, with an appropriate selection of the penalty parameter, the modeling error induced by this inconsistency can often be kept acceptable. The selection of an appropriate penalty parameter can be challenging, however, as a too small value does not adequately enforce the prescribed boundary condition, while a too large value can lead to ill-conditioning and non-physical gradients on cut elements [76].

An alternative approach for the weak imposition of Dirichlet boundary conditions is Nitsche's method [77], which can be interpreted as a consistent enhancement of the penalty method. Nitsche's method is typically the method of choice in CutFEM [50], for which it is shown that a stable and optimally convergent formulation is obtained, provided that the stabilization parameter is selected and scaled (with the mesh size) appropriately (see Ref. [78]). When combined with ghost-penalty stabilization [79], Nitsche's method is capable of obtaining high-fidelity solutions in the immersed (isogeometric) analysis setting.

1.2.3 Stability and conditioning

In the immersed setting, small (*i.e.*, with a small volume fraction) or unfavorably cut (*e.g.*, sliver-like cut) elements affect the stability and conditioning of the discrete immersed formulation. One reason for this is that, without special treatment, the stabilization parameter for Nitsche's method scales inversely with the size of a cut element. This results in very high (element-wise) Nitsche parameters, deteriorating the accuracy and conditioning of the discrete problem [76]. Moreover, basis functions that are only supported on small cut elements become nearly linearly dependent when restricted to the computational domain. This leads to small eigenvalues in the system matrix, resulting in a scaling of the condition number with the smallest cut volume fraction, with the scaling deteriorating as the order of the basis increases [80].

Dedicated preconditioning techniques have been developed for immersed methods, *e.g.*, Refs. [80–82], enabling the application of iterative solvers to immersed finite element problems. In this regard, the multi-grid preconditioner developed in Ref. [60] is particularly noteworthy, as it does not only make the condition number (and the number of solver iterations) independent of the cut element configurations, but also insensitive to the element size of the background mesh. This is an important step in unlocking the potential of high-performance computing for immersed finite element methods [62].

Although the above-mentioned preconditioning techniques enable the treatment of ill-conditioned systems of equations resulting from immersed formulations, they do not resolve the underlying stability problems. The most promi-

nent strategy for resolving the stability problems emerging from the scaling of the Nitsche parameter is to extend the solution obtained on the physical domain to all the elements in the background mesh containing the boundary of the physical domain. This extension can be constructed weakly by the consideration of additional stabilization terms, of which ghost-penalty stabilization is a prominent example [79, 83]. This method requires the selection of an additional (ghost-penalty) stabilization parameter [84]. Alternatively, the extension of the solution to the background grid can be accomplished in a strong manner, by altering the function spaces through the aggregation of (problematic) basis functions [85]. Both the weak and strong extensions enable scaling of the Nitsche stabilization parameter by the size of the background elements, rather than by the cut element size, thereby resolving the stability issues related to unfavorably cut elements. In addition, upon appropriate selection of the additional parameters, the condition number of the system of equations can be bound independent of the cut element volume fraction [79, 83].

1.3 Aims and scope

Immersed isogeometric analysis provides a natural framework for high-fidelity scan-based simulations on account of its ability to: *(i)* accurately represent geometrically and topologically complex volumetric domains; *(ii)* provide a smooth representation of immersed boundaries to optimally benefit from the approximation power of splines; and *(iii)* create an analysis workflow for scan-based geometries with a high degree of automation. To exploit the full potential of immersed isogeometric analysis for scan-based simulations, various innovations with respect to the computational workflow are required:

- Scan-based immersed isogeometric analyses typically employ a robust image segmentation procedure based on a recursive octree subdivision strategy. When used with a sufficiently large octree subdivision depth, this procedure is capable of accurately reconstructing complex geometries. However, with the increase in subdivision depth, the number of sub-cells used to represent the geometry of a trimmed element is typically very large, especially in the three-dimensional setting inherent to practical scan-based analyses. This results in a high number of quadrature points, making integration operations one of the critical components of the analysis workflow from an efficiency point of view. *An efficient integration quadrature scheme for trimmed elements is required to make immersed isogeometric analysis a competitive simulation technique in terms of computational performance.*

- The octree subdivision recursion can be closed with a tessellation procedure to construct an explicit parametrization of the immersed boundary.

For a sufficiently large subdivision depth, a relatively smooth (continuous piecewise-linear) boundary representation can be obtained. This enables the immersed analysis strategy to exploit the advantageous approximation properties of higher-order regular spline discretizations, especially for flow problems. To obtain a smooth boundary representation, the grayscale voxel data, that generally forms the primitive output of imaging devices, must be smoothed, for example by a B-spline-based convolution operation. A consequence of such a smoothing operation is that for features with a size similar to the voxels topological changes can occur, which can be detrimental to the simulation results. *A segmentation algorithm that preserves the topology of the original voxel data is required to enhance the robustness of the scan-based analysis workflow.*

- The immersed simulation concept enables the construction of an approximation basis independent of the geometry representation, which is essential to the automation of the scan-based analysis workflow. *In order to leverage this advantageous characteristic of immersed methods, especially for problems with multi-scale geometries or non-smooth solutions, an adaptive local mesh-refinement strategy is required to obtain an accurate solution with a minimal number of degrees of freedom.*

The overarching goal of the research work presented in this dissertation is to progress the state of the art in scan-based simulations in the context of immersed isogeometric analysis.

Each chapter in this (paper-based) dissertation addresses one of the challenges mentioned above:

In **Chapter 2** (Ref. [86]) the development of an error-estimate-based adaptive integration procedure for immersed (isogeometric) analysis is presented. This chapter includes a rigorous investigation of the accuracy and computational effort of the octree subdivision strategy, with a quantification of the integration error based on Strang's first lemma. A detailed numerical analysis of the developed adaptive integration algorithm is presented, and its application to scan-based problems is demonstrated.

In **Chapter 3** of this dissertation (Ref. [87]) a topology-preserving truncated hierarchical B-spline-based image-segmentation strategy is proposed. This computational strategy rigorously checks for the occurrence of topological alterations due to the B-spline-based voxel-data smoothing procedure. A detailed analysis of the B-spline smoothing procedure provides a basis for the development of a THB-spline-based local level set refinement strategy, which recovers features that are topologically altered by the smoothing procedure. The locations at which to

apply the level set refinement procedure are detected by analyzing local topology characteristics (specifically the Euler characteristic) using a moving-window technique. The proposed algorithm is applied to a series of scan-based test cases, both in two- and three-dimensions, and is applied in the context of stabilized immersed isogeometric analysis.

In **Chapter 4** (Ref. [88]) a dedicated THB-spline-based mesh refinement strategy using a residual-based error estimator is developed for immersed isogeometric analysis. This adaptive simulation procedure is applied to Laplace and Stokes problems. The presented element-wise error-estimator is derived for the immersed setting with a vigilant treatment of the required stabilization terms. The developed error estimation and adaptivity procedure is applied to a set of benchmark problems for steady heat conduction and viscous flows. The automated scan-based analysis workflow is demonstrated using a patient-specific carotid artery flow problem.

Finally, in **Chapter 5** conclusions are drawn regarding the overarching objective of this dissertation, and recommendations for further research are discussed.

Chapter 2

Error-estimate-based adaptive integration for immersed isogeometric analysis

In this chapter we conduct a thorough investigation of the accuracy and computational effort of the octree integration scheme in the context of immersed isogeometric analysis. We quantify the contribution of the integration error using the theoretical basis provided by Strang's first lemma. Based on this study we propose an error-estimate-based adaptive integration procedure for immersed isogeometric analysis. Additionally, we present a detailed numerical investigation of the proposed optimal integration algorithm and its application to immersed isogeometric analysis using two- and three-dimensional linear elasticity problems.

This chapter is reproduced from [86]: S.C. Divi, C.V. Verhoosel, F. Auricchio, A. Reali, and E.H. van Brummelen. Error-estimate-based adaptive integration for immersed isogeometric analysis. *Computers & Mathematics with Applications*, 2020. The (co-)promotors confirm that S.C. Divi is the primary author of this publication, i.e., she was responsible primarily for the planning, execution and preparation of the work.

2.1 Introduction

Immersed finite element methods – such as, *e.g.*, the finite cell method (FCM) [49], CutFEM [50] and immersogeometric analysis [52,55,89] – have been demonstrated to be suitable for computational problems for which the performance of mesh-fitting finite element methods is impeded by complications in the meshing procedure. In recent years, immersed finite element methods have been successfully combined with isogeometric analysis (IGA) [16], a spline-based higher-order finite element framework targeting the integration of finite element analysis (FEA) and computer aided design (CAD). On the one hand, immersed methods provide the possibility to conduct IGA on volumetric domains based on a CAD boundary surface representations [89] or voxelized geometry data [58]. On the other hand, immersed methods provide a natural framework for the consideration of trimming curves in IGA [37–39, 52–54, 81, 90, 91]. Immersed IGA has been applied successfully to various problems in, amongst others, structural and solid mechanics [14, 65], fluid-structure interactions [56, 57, 92] and scan-based analysis [58, 59].

In comparison to mesh-fitting finite element methods, immersed methods require special treatment of various computational aspects. A prominent computational challenge that is inherent to immersed finite element methods is the integration over cut-cells, a problem that closely relates to the special treatment of discontinuous integrands in enriched finite element methods such as XFEM and GFEM. Since the geometry of the computational problem is captured by the integration procedure rather than by the ambient mesh in which the domain is immersed, cut-cell integration techniques must be capable of adequately capturing a wide range of cut-cell configurations. A myriad of dedicated integration procedures with this capability has been developed over the years in the context of immersed FEM (see [14] for a review) and enriched FEM (see [64]), which can be categorized as:

- *Octree subdivision:* The general idea of octree (or quadtree in 2D) integration is to capture the geometry of a cut-cell by recursively bisecting sub-cells that intersect with the boundary of the domain. At every level of this recursion, sub-cells that are completely inside the domain are preserved, while sub-cells that are completely outside of the domain are discarded. This cut-cell subdivision strategy was initially proposed in the context of the finite cell method in Ref. [49] and is generally appraised for its simplicity and robustness with respect to cut-cell configurations. Octree integration has been widely adopted in immersed FEM, see, *e.g.*, Refs. [14, 56, 58, 65]. Various generalizations and improvements to the original octree procedure have been proposed, of which the consideration of tetrahedral cells [66, 67], the reconstruction of the immersed boundary by tessellation of the low-

est level of bisectioning [58], and the consideration of variable integration schemes for the sub-cells [68], are particularly noteworthy. Despite the various improvements to the original octree strategy, a downside of the technique remains the number of integration sub-cells that results from the procedure, especially in three-dimensional cases.

- *Cut-cell reparametrization:* Accurate cut-cell integration schemes can be obtained by modifying the geometry parametrization of cut-cells in such a way that the immersed boundary is fitted. This strategy was original developed in the context of XFEM by decomposing cut elements into various sub-cells containing cut-cells with only one curved side and then to alter the geometry mapping related to the curved sub-cell to obtain a higher-order accurate integration scheme [93]. This concept has also been considered in the context of implicitly defined geometries (level sets) [70], the NURBS-enhanced finite element method (NEFEM) [69, 94] and the Cartesian grid finite element method (cgFEM) [95]. In the context of the finite cell method the idea of cut element reparametrization has been adopted as part of the *smart octree integration* strategy [71, 96], where a boundary fitting procedure is employed at the lowest level of octree bisectioning in order to obtain higher-order integration schemes for cut-cells with curved boundaries. Reparametrization procedures have the potential to yield accurate integration schemes at a significantly lower computational cost than octree procedures, but generally compromises in terms of robustness with respect to cut-cell configurations.

- *Polynomial integration*: Provided that one can accurately evaluate integrals over cut-cells (for example using octree integration), it is possible to construct computationally efficient integration rules for specific classes of integrands. In the context of immersed finite element methods it is of particular interest to derive efficient cut-cell integration rules for polynomial functions. The two most prominent methods to integrate polynomial functions over cut-cells are *moment fitting techniques* [72, 96–99], in which integration point weights and (possibly) positions are determined in order to yield exact quadrature rules, and *equivalent polynomial methods* [73,100], in which a non-polynomial (*e.g.*, discontinuous) integrand is represented by an equivalent polynomial which can then be treated using standard integration procedures. Such methods have been demonstrated to yield efficient quadrature rules for a range of scenarios. A downside of such techniques is the need for the evaluation of the exact integrals (using an adequate cut-cell integration procedure) in order to determine the optimized integration rules. This can make the construction of such quadrature rules computationally expensive, which makes that they are of particular interest mainly

in the context of time-dependent and non-linear problems, for which the construction of the integration rule is only considered as a preprocessing operation (for each cut-cell) and the optimized integration rule can then be used throughout the simulation.

- *Boundary integral representation*: Depending on the problem under consideration, it can be possible to reformulate volumetric integrals over cut-cells by equivalent boundary integrals. This reformulation, which has been proposed in the context of XFEM in Ref. [101] and in the immersed FEM setting in Ref. [74], is advantageous from a computational effort point of view, as the equivalent boundary integrals are generally less costly to evaluate. Moreover, provided that an adequate description of the boundary surface is available, higher-order accurate integration evaluations are obtained. A downside of integration techniques of this kind is that they require reformulation of the volume integrals, which makes them less general than standard quadrature rules.

In the selection of an appropriate cut-cell integration scheme one balances integration between robustness, accuracy and expense with respect to cut-cell configurations. If one requires a method that automatically treats a wide range of cut-cell configurations and is willing to pay the price in terms of (higher-order) accuracy and computational effort, octree integration is the compelling option. If constraints are imposed from an accuracy and computational expense point of view and one has some control over the range of configurations to be considered, alternative techniques such as cut-cell reparametrization are attractive. The need to balance between robustness, accuracy and expense has driven the development of hybrid integration schemes, such as *smart octree integration* [71] and *adaptive moment-fitting* [72], which allow one to attain an integration procedure with the desired properties.

Balancing accuracy with computational effort does not necessarily require the consideration of hybrid integration procedures, but can also be achieved by controlling the parameters of the integration procedures (listed above). This is particularly the case for octree integration, the accuracy of which can be controlled by the bisectioning depth and the integration orders used on the sub-cells. Optimization of these parameters to reduce the computational expense of octree integration without compromising its robustness with respect to configurations was proposed in Ref. [68], where an algorithm is proposed to select the integration order on the different levels of sub-cells. It is demonstrated that reducing the integration order with increasing bisection depth significantly reduces the number of integration points, without unacceptably compromising the accuracy.

The idea of Ref. [68] to reduce integration orders on certain levels of the octree subdivision is in agreement with the theory of finite elements. Strang's first

lemma [9, 11, 102] provides a framework to incorporate integration effects in the error analysis of finite element methods. This lemma indicates that integration does not need to be exact in order to attain (optimal) convergence [9][1]. This lemma has been considered for the analysis of CutFEM, see, *e.g.,* Refs. [103,104]. In fact, the idea of reduced integration in finite element methods has been studied extensively over the last decades, with applications in the analysis of plates and shells [105] and mixed finite element methods [106, 107] being prominent examples.

In this manuscript we propose an error-estimation-based adaptive algorithm to obtain optimal octree integration rules for cut-cells. A sub-cell evaluation of the integration error in accordance with Strang's first lemma [9, 11, 102] is combined with a computational-cost estimate based on the number of integration cells to obtain a refinement indicator that optimally balances computational effort and accuracy. The proposed algorithm deviates from the one proposed in Ref. [68] in that it is directly based on integration error evaluations. The effectiveness of the adaptive integration procedure is demonstrated in the context of the isogeometric finite-cell framework. Besides the ability to obtain optimal integration rules for cut-cells, the developed adaptive integration procedure provides a tool to postulate rules of thumb for the selection of integration orders at different levels of bisectioning, and to assess the quality of integration rules derived using alternative algorithms such as that proposed in Ref. [68, 108].

The paper outline is as follows. In Section 2.2 we introduce the finite cell method, with a focus on the application of Strang's lemma to estimate integration errors. In Section 2.3 we study the influence of the octree integration scheme on the computational cost, and we define and evaluate the integration error of a sub-cell in a cut element. Based on the evaluated integration error, we present an optimization algorithm. The optimal integration procedure is investigated for two- and three-dimensional numerical test cases with various cut element configurations in Section 2.4. The developed algorithm is then applied to a two- and three-dimensional immersed isogeometric analysis problem in Section 2.5. Finally, concluding remarks are presented in Section 2.6.

2.2 The finite cell method

Because the integration procedure proposed in this work applies to a wide range of finite-cell simulations, in Section 2.2.1 we first introduce the finite cell method in abstract form. Based on this abstract problem setting, Section 2.2.2 presents

[1] As formulated by Strang and Fix in 1973 [9]: *"What degree of accuracy in the integration formula is required for convergence? It is not required that every polynomial which appears be integrated exactly."*

an error analysis that incorporates integration errors. In Section 2.2.3 the Poisson problem is considered to exemplify the abstract derivations.

2.2.1 Formulation

We consider a domain $\Omega \subset \mathbb{R}^d$ ($d \in \{2, 3\}$) with boundary $\partial\Omega$ as shown in Figure 2.1. The boundary is split into a part on which Dirichlet conditions are imposed, denoted by $\partial\Omega_D$, and a complementary part to which Neumann conditions apply, denoted by $\partial\Omega_N$. We consider a problem described by a field variable u. Let $W \ni u$ denote a suitable ambient space for the solution, equipped with a Hilbert structure corresponding to the inner product $(\cdot, \cdot)_W$ and the association norm $\|\cdot\|_W$. Similarly, V is a Hilbert space with inner product $(\cdot, \cdot)_V$ and norm $\|\cdot\|_V$, which encompasses the test space for the weak formulation of the problem under consideration. To accommodate the Dirichlet boundary conditions, let the spaces $W_0 \subset W$ and $V_0 \subset V$ be composed of functions that vanish at the Dirichlet boundary in a suitable manner. Moreover, let $\ell_g \in W$ denote a lift of the Dirichlet data. We consider a weak formulation of the generic form:

$$\begin{cases} \text{Find } u \in \ell_g + W_0 \text{ such that:} \\ a(u, v) = b(v) \quad \forall v \in V_0, \end{cases} \tag{2.1}$$

We assume that the bilinear form $a : W \times V \to \mathbb{R}$ is continuous on $W \times V$ and weakly coercive on $W_0 \times V_0$, and the linear form $b : V \to \mathbb{R}$ is continuous. The weak formulation (2.1) is then well-posed; see, e.g., [11].

The finite cell method provides a general framework for constructing finite dimensional subspaces $W^h \subset W$ and $V^h \subset V$, where the superscript h refers to a mesh parameter. The subspaces W^h, V^h are subordinate to a regular mesh $\mathcal{T}^h_{\mathcal{A}}$, covering an ambient (embedding) domain $\mathcal{A} \supset \Omega$. The collection of elements $K \in \mathcal{T}^h_{\mathcal{A}}$ that intersect with the physical domain comprise the background mesh:

$$\mathcal{T}^h := \{K \in \mathcal{T}^h_{\mathcal{A}} : K \cap \Omega \neq \emptyset\} \tag{2.2}$$

By trimming the elements in the background mesh, a mesh for the interior of the domain Ω is obtained:

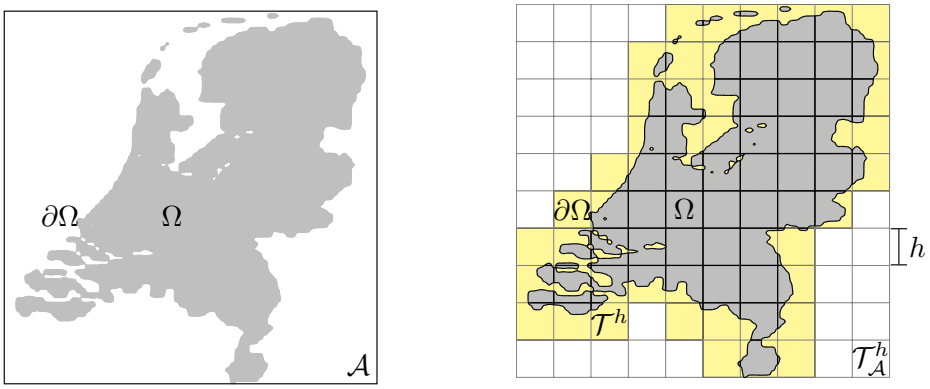
$$\mathcal{T}^h_\Omega := \{K \cap \Omega : K \in \mathcal{T}^h\} \tag{2.3}$$

Similarly, meshes for the Dirichlet and Neumann boundaries are defined as:

$$\mathcal{T}^h_{\partial\Omega_D} := \{K \cap \partial\Omega_D : K \in \mathcal{T}^h\} \quad \text{and} \quad \mathcal{T}^h_{\partial\Omega_N} := \{K \cap \partial\Omega_N : K \in \mathcal{T}^h\} \tag{2.4}$$

Note that elements in $\mathcal{T}^h_{\partial\Omega_D}$ and $\mathcal{T}^h_{\partial\Omega_N}$ are manifolds of co-dimension 1.

We consider a B-spline basis of degree k and regularity α constructed over the ambient domain using the Cox-De Boor recursion formula [18]. The span of



(a) **(b)**

Figure 2.1: Schematic representation of (a) the physical domain Ω (gray) with boundary $\partial\Omega$ which is embedded in the ambient domain $\mathcal{A}$, and (b) the ambient domain mesh $\mathcal{T}^h_{\mathcal{A}}$ and background mesh $\mathcal{T}^h$ (yellow), with mesh size parameter h.

this B-spline basis is denoted as

$$\mathcal{S}^k_\alpha(\mathcal{A}) = \{N \in C^\alpha(\mathcal{A}) : N|_K \in P^k(K), \forall K \in \mathcal{T}^h_{\mathcal{A}}\} \tag{2.5}$$

with $P^k(K)$ the collection of d-variate polynomials on K. The approximation spaces in the finite cell method are obtained by restricting the B-splines in $\mathcal{S}^k_\alpha(\mathcal{A})$ to the domain Ω:

$$W^h = V^h = \{N|_\Omega : N \in \mathcal{S}^k_\alpha(\mathcal{A})\} \tag{2.6}$$

A basis for W^h, V^h follows immediately from the restriction of the B-spline basis for $\mathcal{S}^k_\alpha(\mathcal{A})$.

It is generally infeasible to impose restrictions on W^h, V^h according to (2.6) to form subspaces of W_0, V_0 that retain suitable approximation properties. Hence, the finite cell method generally relies on weak imposition of the Dirichlet boundary conditions, typically by means of Nitsche's method [77]. Accordingly, the bilinear and linear forms for the approximation problem are adapted to weakly incorporate the Dirichlet boundary conditions, giving rise to a Galerkin formulation of the form:

$$\begin{cases} \text{Find } u^h \in W^h \text{ such that:} \\ a^h(u^h, v^h) = b^h(v^h) \quad \forall v^h \in V^h, \end{cases} \tag{2.7}$$

where the superscript h on the bilinear form $a^h : W^h \times V^h \to \mathbb{R}$ and linear form $b^h : V^h \to \mathbb{R}$ indicate an explicit dependence on the mesh parameter. In

general, the bilinear and linear forms in the finite cell method comprise contributions from both the interior and the boundaries. Corresponding to this typical setting, in the remainder of this work we assume the operators to be of the form

$$a^h(u^h, v^h) = \int_\Omega A^h_\Omega(u^h, v^h)(\boldsymbol{x}) \,\mathrm{d}V + \int_{\partial\Omega_D} A^h_{\partial\Omega_D}(u^h, v^h)(\boldsymbol{x}) \,\mathrm{d}S, \tag{2.8a}$$

$$b^h(v^h) = \int_\Omega B^h_\Omega(v^h)(\boldsymbol{x}) \,\mathrm{d}V + \int_{\partial\Omega_D} B^h_{\partial\Omega_D}(v^h)(\boldsymbol{x}) \,\mathrm{d}S + \int_{\partial\Omega_N} B^h_{\partial\Omega_N}(v^h)(\boldsymbol{x}) \,\mathrm{d}S, \tag{2.8b}$$

where the integrands A^h_Ω and $A^h_{\partial\Omega_D}$ map $W^h \times V^h$ into integrable functions on the domain Ω and the Dirichlet boundary section, $\partial\Omega_D$, respectively. Similarly, B^h_Ω, $B^h_{\partial\Omega_D}$ and $B^h_{\partial\Omega_N}$ map V^h into integrable functions on the domain Ω and its boundary sections, $\partial\Omega_D$ and $\partial\Omega_N$. In the setting of (2.7), the spaces W^h, V^h are equipped with (generally mesh-dependent) norms $\|\cdot\|_{W^h}, \|\cdot\|_{V^h}$. Assuming that the bilinear form a^h is continuous and weakly coercive on $W^h \times V^h$ and the linear form b^h is continuous on V^h, the approximation problem (2.7) is well posed [11, ch. 2].

In practice, the integrals in equation (2.8) are approximated by means of quadrature rules, leading to the definition of the quadrature-dependent bilinear and linear forms

$$a^h_{\mathcal{Q}}(u^h, v^h) = \sum_{K \in \mathcal{T}^h_\Omega} \sum_{l=1}^{l_K} \omega^l_K A^h_\Omega(u^h, v^h)(\boldsymbol{x}^l_K) + \sum_{K \in \mathcal{T}^h_{\partial\Omega_D}} \sum_{l=1}^{l_K} \omega^l_K A^h_{\partial\Omega_D}(u^h, v^h)(\boldsymbol{x}^l_K), \tag{2.9a}$$

$$b^h_{\mathcal{Q}}(v^h) = \sum_{K \in \mathcal{T}^h_\Omega} \sum_{l=1}^{l_K} \omega^l_K B^h_\Omega(v^h)(\boldsymbol{x}^l_K) + \sum_{K \in \mathcal{T}^h_{\partial\Omega_D}} \sum_{l=1}^{l_K} \omega^l_K B^h_{\partial\Omega_D}(v^h)(\boldsymbol{x}^l_K) + \sum_{K \in \mathcal{T}^h_{\partial\Omega_N}} \sum_{l=1}^{l_K} \omega^l_K B^h_{\partial\Omega_N}(v^h)(\boldsymbol{x}^l_K), \tag{2.9b}$$

where for each (interior or boundary) element K, the set $\{(\boldsymbol{x}^l_K, \omega^l_K)\}_{l=1}^{l_K}$ represents a quadrature rule, *i.e.*, a suitable collection of pairs of integration points in K and corresponding weights. We denote by $\mathcal{Q}$ the complete integration scheme, *i.e.*, the collection of all the (interior and boundary) element-wise quadrature rules. The Galerkin problem corresponding to the approximate (bi-)linear

forms $a^h_{\mathcal{Q}}$ and $b^h_{\mathcal{Q}}$ then writes:

$$\begin{cases} \text{Find } u^h_{\mathcal{Q}} \in W^h \text{ such that} \\ a^h_{\mathcal{Q}}(u^h_{\mathcal{Q}}, v^h) = b^h_{\mathcal{Q}}(v^h) \quad \forall v^h \in V^h. \end{cases} \tag{2.10}$$

We assume that the integration scheme $\mathcal{Q}$ transfers the coercivity and boundedness properties of a^h, b^h to $a^h_{\mathcal{Q}}, b^h_{\mathcal{Q}}$, so that (2.10) is well posed. We note that the solution $u^h_{\mathcal{Q}}$ generally deviates from that of (2.7) on account of the inexactness of the integration rules.

2.2.2 Finite cell error analysis

The error of the approximated finite cell solution, $u^h_{\mathcal{Q}}$, computed using the Galerkin problem (2.10) is composed of two parts, *viz.*: *i)* the discretization error, defined as the difference between the exact solution to (2.1), u, and the approximate solution to (2.7) in the absence of integration errors, u^h; and *ii)* the inconsistency error related to the integration procedure, which is defined as the difference between the approximate solution in the absence of integration errors, u^h, and the approximate solution to (2.10) with integration errors, $u^h_{\mathcal{Q}}$. In this section we present the error analysis in an abstract setting. A concrete example is provided in Section 2.2.3. To provide a setting for the error analysis, we denote by $W(h) = \operatorname{span}\{u\} \oplus W^h$ the linear space containing the actual solution to (2.1) and the approximation space W^h. We equip $W(h)$ with a norm $\|\cdot\|_{W(h)}$ such that $\|w^h\|_{W(h)} = \|w^h\|_{W^h}$ for all $w^h \in W^h$ and $\|u\|_{W(h)} \leq c\,\|u\|_W$, *i.e.*, the solution of (2.1) is continuously embedded in $W(h)$; cf. [11, Ch. 2]. We assume that the solution u of (2.1) is suitably regular, so that the bilinear form a^h admits a continuous extension to $W(h)$, *i.e.*, there exists a constant $\mathscr{C} > 0$ such that:

$$\|a^h\|_{W(h),V^h} := \sup_{(w,v^h)\in W(h)\times V^h} \frac{|a^h(w,v^h)|}{\|w\|_{W(h)}\|v\|_{V^h}} \leq \mathscr{C}. \tag{2.11}$$

In this setting, an upper bound for the error $u - u^h_{\mathcal{Q}}$ is provided by Strang's first lemma [102]:

$$\begin{aligned} \left\|u - u^h_{\mathcal{Q}}\right\|_{W(h)} \leq & \left(1 + \frac{\left\|a^h\right\|_{W(h),V^h}}{\alpha^h}\right) \left\|u - \mathcal{I}^h u\right\|_{W(h)} \\ & + \frac{1}{\alpha^h}\left(\sup_{v^h\in V^h} \frac{\left|b^h(v^h) - b^h_{\mathcal{Q}}(v^h)\right|}{\|v^h\|_{V^h}} \right. \\ & \left. + \sup_{v^h\in V^h} \frac{\left|a^h(\mathcal{I}^h u, v^h) - a^h_{\mathcal{Q}}(\mathcal{I}^h u, v^h)\right|}{\|v^h\|_{V^h}} \right), \end{aligned} \tag{2.12}$$

where $\mathcal{I}^h u$ represents the best approximation of u in W^h, *i.e.*,

$$\mathcal{I}^h u = \underset{w^h \in W^h}{\arg\min} \|u - w^h\| \tag{2.13}$$

and α^h denotes the inf-sup constant of the bilinear form $a^h : W^h \times V^h \to \mathbb{R}$:

$$\alpha^h = \inf_{w^h \in W^h} \sup_{v^h \in V^h} \frac{a^h(w^h, v^h)}{\|w^h\|_{W^h} \|v^h\|_{V^h}}. \tag{2.14}$$

Proof of (2.12) is standard; see, for instance, [11, Lemma 2.27]. The first term in (2.12) represents an upper bound to the discretization error. The second term in (2.12) bounds the consistency error, *i.e.*, the integration error.

Equation (2.8) conveys that, in principle, integration errors emerge for both the volume integrals and the surface integrals. Since the focus of this work is on the optimization of the integration orders over octree sub-cells, in the remainder of this work we will restrict our considerations to the integration errors associated with the volume integrals, thereby implicitly assuming that the boundary integrals are evaluated exactly. When neglecting the integration errors associated with the boundary integrals, the error term in (2.12) associated with the inexact integration of the linear form is bounded by

$$\begin{aligned}
&\sup_{v^h \in V^h} \frac{\left| b^h(v^h) - b^h_{\mathcal{Q}}(v^h) \right|}{\|v^h\|_{V^h}} \\
&\quad = \sup_{v^h \in V^h} \frac{\left| \sum_{K \in \mathcal{T}^h_\Omega} \left(\int_K B^h_\Omega(v^h)(\boldsymbol{x}) \, \mathrm{d}V - \sum_{l=1}^{l_K} \omega^l_K B^h_\Omega(v^h)(\boldsymbol{x}^l_K) \right) \right|}{\|v^h\|_{V^h}} \\
&\quad \leq \sum_{K \in \mathcal{T}^h_\Omega} \sup_{v^h \in V^h} \frac{\left| \int_K B^h_\Omega(v^h)(\boldsymbol{x}) \, \mathrm{d}V - \sum_{l=1}^{l_K} \omega^l_K B^h_\Omega(v^h)(\boldsymbol{x}^l_K) \right|}{\|v^h\|_{V^h}}.
\end{aligned} \tag{2.15}$$

Note that the supremum is always considered over a function space with the zero function excluded, *e.g.*, $v^h \in V^h \setminus \{0\}$. For notational brevity we omit the zero exclusion in the remainder of this manuscript. Under the (non-restrictive) assumption that B^h_Ω is a local operator in the sense that

$$\left(B^h_\Omega(v^h|_K)\right)\Big|_K \text{ is well defined and } \left(B^h_\Omega(v^h)\right)\Big|_K = \left(B^h_\Omega(v^h|_K)\right)\Big|_K, \tag{2.16}$$

e.g., if B^h_Ω corresponds to a differential operator, the summands in the ultimate

expression in (2.15) are bounded by the element-integration-error indicators

$$e_K^b = \sup_{v_K^h \in V_K^h} \frac{\left| \int_K B_\Omega^h(v_K^h)(\boldsymbol{x}) \, \mathrm{d}V - \sum_{l=1}^{l_K} \omega_K^l B_\Omega^h(v_K^h)(\boldsymbol{x}_K^l) \right|}{\left\| v_K^h \right\|_{V_K^h}}, \tag{2.17}$$

where $V_K^h = \{v^h|_K, \, v^h \in V^h\}$ is the restriction of V^h to the element K, equipped with a norm $\|\cdot\|_{V_K^h}$ such that $\|v^h|_K\|_{V_K^h} \leq \|v^h\|_{V^h}$ for all $v^h \in V^h$. The element-integration-error indicators e_K^b can be computed element-wise and provide a bound for the integration error in the linear form:

$$\sup_{v^h \in V^h} \frac{\left| b^h(v^h) - b_\mathcal{Q}^h(v^h) \right|}{\|v^h\|_{V^h}} \leq \sum_{K \in \mathcal{T}_\Omega^h} e_K^b \tag{2.18}$$

Similarly, the integration error for the bilinear form in (2.12) is bounded as

$$\sup_{v^h \in V^h} \frac{\left| a^h(\mathcal{I}^h u, v^h) - a_\mathcal{Q}^h(\mathcal{I}^h u, v^h) \right|}{\|v^h\|_{V^h}} \leq \sum_{K \in \mathcal{T}_\Omega^h} e_K^a \tag{2.19}$$

with the element-integration-error indicators defined as

$$e_K^a = \sup_{v_K^h \in V_K^h} \frac{\left| \int_K A_\Omega^h(\mathcal{I}^h u, v^h)(\boldsymbol{x}_K) \, \mathrm{d}V - \sum_{l=1}^{l_K} \omega_K^l A_\Omega^h(\mathcal{I}^h u, v^h)(\boldsymbol{x}_K^l) \right|}{\left\| v_K^h \right\|_{V_K^h}}. \tag{2.20}$$

Substitution of the bounds (2.18) and (2.19) into the error estimate (2.12) then yields

$$\begin{aligned} \left\| u - u_\mathcal{Q}^h \right\|_{W(h)} \leq & \left(1 + \frac{\left\| a^h \right\|_{W(h),V^h}}{\alpha^h} \right) \left\| u - \mathcal{I}^h u \right\|_{W(h)} \\ & + \frac{1}{\alpha^h} \sum_{K \in \mathcal{T}_\Omega^h} \left(e_K^b + e_K^a \right), \end{aligned} \tag{2.21}$$

which conveys that the element-integration-error indicators (2.17) and (2.20) yield control over the inconsistency error. This motivates the development of an algorithm to minimize the element-integration-error indicators; see Section 2.3.2.

2.2.3 Application to the Poisson problem

We apply the general theory presented in Sections 2.2.1–2.2.2 to the particular case of the Dirichlet–Poisson problem:

$$\begin{cases} -\Delta u = f & \text{in } \Omega, \\ u = g & \text{on } \partial\Omega, \end{cases} \tag{2.22}$$

with $f : \Omega \to \mathbb{R}$ and $g : \partial\Omega \to \mathbb{R}$ exogenous data. A suitable ambient space for the weak formulation of (2.22) is provided by $W = H^1(\Omega)$, equipped with the usual H^1-norm and inner product. Denoting by $\ell_g \in H^1(\Omega)$ a lift of the Dirichlet data such that $\ell_g|_{\partial\Omega} = g$ and by $W_0 = H^1_0(\Omega)$ the subspace of functions in $H^1(\Omega)$ that vanish at the boundary in the trace sense, the weak formulation of (2.22) assumes the form (2.1) with

$$a(u, v) = \int_\Omega \nabla u \cdot \nabla v \,\mathrm{d}V, \qquad b(v) = \int_\Omega f\, v \,\mathrm{d}V. \tag{2.23}$$

We denote by V^h a finite-dimensional subspace of $H^1(\Omega)$ according to the construction in Section 2.2.1. We equip V^h with the mesh-dependent norm

$$\|v^h\|^2_{V^h} = \int_\Omega |\nabla v^h|^2 \,\mathrm{d}V + \frac{c_0}{h} \int_{\partial\Omega} (v^h)^2 \,\mathrm{d}S. \tag{2.24}$$

The symmetric Galerkin formulation in V^h with weak enforcement of the Dirichlet conditions by means of Nitsche's method conforms to (2.7) with $a^h : V^h \times V^h \to \mathbb{R}$ and $b^h : V^h \to \mathbb{R}$:

$$\begin{aligned} a^h(u^h, v^h) = &\int_\Omega \nabla u^h \cdot \nabla v^h \,\mathrm{d}V - \int_{\partial\Omega} v^h \partial_n u^h \,\mathrm{d}S \\ &- \int_{\partial\Omega} u^h \partial_n v^h \,\mathrm{d}S + \frac{c_1}{h} \int_{\partial\Omega} u^h v^h \,\mathrm{d}S, \end{aligned} \tag{2.25a}$$

$$b^h(v^h) = \int_\Omega f\, v^h \,\mathrm{d}V - \int_{\partial\Omega} g\, \partial_n v^h \,\mathrm{d}S + \frac{c_1}{h} \int_{\partial\Omega} g\, v^h \,\mathrm{d}S. \tag{2.25b}$$

For suitably chosen constants c_0 and c_1, the bilinear form is bounded and coercive on $V^h \times V^h$ and the linear form b^h is bounded on V^h. The bilinear form a^h does not admit a continuous extension to all $H^1(\Omega)$, owing to the fact that the normal derivatives $\partial_n(\cdot)$ that appear in a^h are not properly defined for all functions in $H^1(\Omega)$. We therefore assume that the solution of the Dirichlet–Poisson problem (2.22) is suitably regular, *e.g.*, such that $\partial_n u \in L^2(\partial\Omega)$. We introduce the composite space $W(h) = \mathrm{span}\{u\} \oplus V^h$ and equip $W(h)$ with the norm $\|\cdot\|_{W(h)}$, corresponding to the extension of the norm in (2.24) to $W(h)$. Trace theory

conveys that $\|u\|_{W(h)} \leq c_h \|u\|_{H^1(\Omega)}$ for some (h-dependent) constant $c_h > 0$ and, hence, the embedding of $\text{span}\{u\}$ into $W(h)$ is continuous. Under the aforementioned regularity assumption on u, the bilinear form a^h admits a continuous extension to $W(h) \times V^h$, and (2.11)–(2.14) apply. To elucidate the notation for the quadrature-dependent bilinear and linear forms for the Dirichlet–Poisson problem, we note that in this case:

$$A^h_\Omega(u^h, v^h)(\boldsymbol{x}) = \nabla u^h(\boldsymbol{x}) \cdot \nabla v^h(\boldsymbol{x}), \qquad B^h_\Omega(v^h)(\boldsymbol{x}) = f(\boldsymbol{x})\, v^h(\boldsymbol{x}). \tag{2.26}$$

One easily verifies that both A^h_Ω and B^h_Ω satisfy the locality condition (2.16).

2.3 Optimized octree cut-cell integration

Based on the error analysis discussed in Section 2.2.2, in this section we develop an algorithm to optimize the distribution of integration points over octree-subdivided cut-cells. In Section 2.3.1 we first introduce the considered octree procedure and study its complexity with respect to the most relevant parameters. In Section 2.3.2 we then express the integration-error estimate in an operator-independent form, making it applicable to a class of operators. The per-cut-element evaluation of the operator-independent integration error is also discussed in this section. Finally, the developed optimization procedure in the form of Algorithm 2.1 is presented in Section 2.3.3.

2.3.1 Octree partitioning

To construct an explicit parametrization of the geometry (including its boundary) we herein consider the octree procedure proposed in Ref. [58], which is illustrated in Figure 2.2 for a two-dimensional cut-cell. In this procedure an element in the background mesh that intersects the boundary of the domain is bisected into 2^d sub-cells. If a sub-cell is completely inside the domain, it is preserved in the partitioning of the cut-cell, whereas a sub-cell is discarded if it is completely outside of the domain. This bisectioning procedure is recursively applied to the sub-cells that intersect the boundary, until $\varrho_{\max}$-times bisected sub-cells are obtained. At the lowest level of bisectioning, *i.e.*, for the $\varrho_{\max}$-times bisected sub-cells, a tessellation procedure is applied to construct a partitioning of the sub-cells that intersect with the domain boundary, resulting in an additional level, *i.e.*, $\varrho_{\max} + 1$. On the one hand, this tessellation procedure provides an $\mathcal{O}(h^2/2^{2\varrho_{\max}})$ accurate parametrization of the interior volume [58], while, on the other hand, it provides a parametrization for the trimmed surface. See Appendix A for details regarding the employed tessellation procedure.

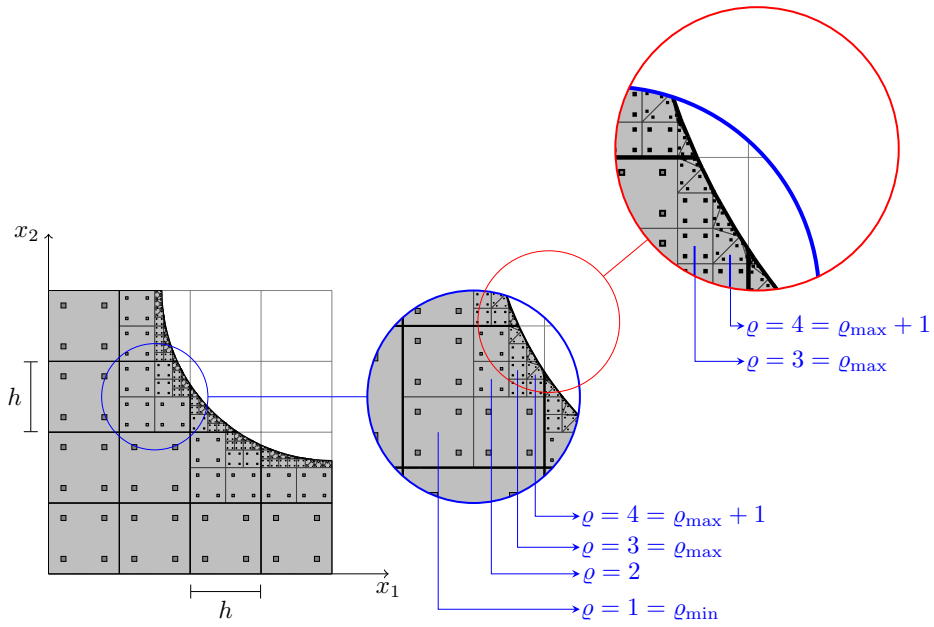


Figure 2.2: Schematic representation of the octree integration procedure with tessellation at the lowest level of bisectioning (see Appendix A).

Since the octree procedure provides a parametrization of the cut-cell and integration rules are generally available for all sub-cells [68, 108], cut-cell integration can be performed by agglomeration of all sub-cell quadrature points. The accuracy of the cut-cell integration scheme can then be controlled through the selection of the quadrature rules on the sub-cells. In particular, polynomials can be integrated exactly over the cut-cell when Gauss quadrature of the appropriate order is used on all sub-cells. An example of a cut-cell integration scheme based on Gauss quadrature for third order polynomials is illustrated in Figure 2.2, where the gray squares represent the Gauss points, and the relative size of the squares is representative for the integration weights.

Selection of Gauss points of optimal order on all sub-cells is evidently very attractive, in the sense that an adequate cut-cell integration scheme can be constructed by specification of the bisectioning depth $\varrho_{\max}$ and the order of polynomials to be integrated exactly. However, as observed from Figure 2.2, the obtained cut-cell integration rule is generally not computationally efficient, in the sense that the majority of integration points is formed on small sub-cells, as a

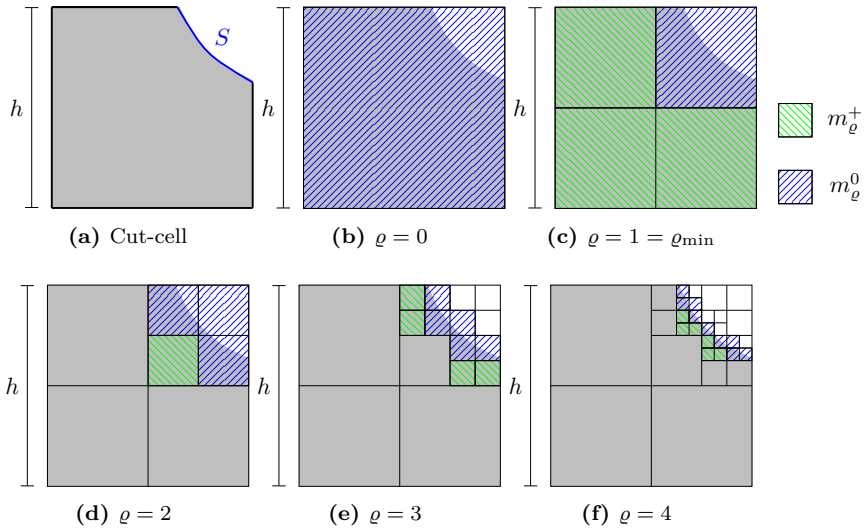


Figure 2.3: Schematic illustration of the sub-cell formation using the octree partitioning for a single cut-element. The green color represents the sub-cells at level ϱ that are completely inside the domain, m_ϱ^+, and the blue cells represent the intersected sub-cells of the level ϱ, m_ϱ^0.

consequence of which their relative contribution to the overall integral (observed from the size of the squares) is generally limited. As studied in Ref. [68], the computational efficiency of the cut-cell integration scheme can be improved by reducing the number of integration points on lower bisectioning levels. Although this reduction decreases the accuracy of the integration scheme, the obtained improvement in computational effort associated with the lower number of integration points outweighs this disadvantage. The algorithm proposed in Ref. [68] targets the optimization of this balance between accuracy and computational effort.

Since the balance between accuracy and computational effort also forms the basis of the error-estimation-based optimization procedure in this work it is important to understand the distribution of the number of octree sub-cells over the levels of bisectioning. To establish a scaling relation for the number of sub-cells we consider a random cut-cell of size h with trimmed surface area S, as illustrated in Figure 2.3a. The octree subdivision approximates the surface area at the maximum octree depth as

$$S \approx m^0_{\varrho_{\max}} \bar{s} \left(\frac{h}{2^{\varrho_{\max}}} \right)^{d-1} \qquad \varrho_{\max} \geq \varrho_{\min} \geq 1 \tag{2.27}$$

where $m^0_{\varrho_{\max}}$ denotes the average number of sub-cells at octree depth $\varrho_{\max}$ that is intersected by the trimmed boundary, and where $\bar{s}$ is the average surface area of a randomly cut unit cube in d dimensions. This surface approximation is valid under the condition that

$$\varrho_{\max} \geq \varrho_{\min} = \text{ceil}\left(\frac{\log_2(\varsigma)}{1-d}\right) \geq 1, \tag{2.28}$$

with surface fraction $\varsigma = \frac{S}{\bar{s}h^{d-1}}$. For example, for the cut-cell illustrated in Figure 2.3, $\varrho_{\max} \geq \varrho_{\min} = 1$.

From (2.27) the number of intersected elements at level $\varrho_{\max}$ follows to be:

$$m^0_{\varrho_{\max}} \approx \begin{cases} 1 & 0 \leq \varrho_{\max} < \varrho_{\min} \\ \varsigma 2^{\varrho_{\max}(d-1)} & \varrho_{\max} \geq \varrho_{\min} \geq 1 \end{cases} \tag{2.29}$$

Denoting the number of sub-cells at level ϱ that are completely inside of the domain by m^+_ϱ (see Figure 2.3), and noting that the average volume of a randomly cut unit cube is $\bar{v} = \frac{1}{2}$, the cut-cell volume can be approximated as

$$V \approx m^0_{\varrho_{\max}} \frac{1}{2}\left(\frac{h}{2^{\varrho_{\max}}}\right)^d + \sum_{\varrho=1}^{\varrho_{\max}} m^+_\varrho \left(\frac{h}{2^\varrho}\right)^d \qquad \varrho_{\max} \geq 1, \tag{2.30}$$

which is divided by the volume of the background cell to obtain the cut-cell volume fraction: insubequations

$$\eta \approx \eta_{\varrho_{\max}} = m^0_{\varrho_{\max}} 2^{-(\varrho_{\max} d+1)} + \sum_{\varrho=1}^{\varrho_{\max}} m^+_\varrho \left(\frac{1}{2^\varrho}\right)^d \qquad \varrho_{\max} \geq 1, \tag{2.31}$$

$$= \varsigma 2^{-(\varrho_{\max}+1)} + \sum_{\varrho=1}^{\varrho_{\max}} m^+_\varrho \left(\frac{1}{2^\varrho}\right)^d \qquad \varrho_{\max} \geq \varrho_{\min} \geq 1. \tag{2.32}$$

subequations Assuming that $\varrho_{\max}$ adequately resolves the volume fraction of the cut-cell, it holds that $\eta_{\varrho_{\max}} \approx \eta_{\varrho_{\max}-1}$ with $\varrho_{\max} \geq \varrho_{\min} + 1$ and

$$\eta_{\varrho_{\max}} - \eta_{\varrho_{\max}-1} = -\varsigma 2^{-(\varrho_{\max}+1)} + m^+_{\varrho_{\max}} 2^{-\varrho_{\max} d} \approx 0. \tag{2.33}$$

For the number of sub-cells at bisectioning level $\varrho_{\max}$, we then obtain

$$m^+_{\varrho_{\max}} \approx \varsigma 2^{\varrho_{\max}(d-1)-1} \qquad \varrho_{\max} > \varrho_{\min}. \tag{2.34}$$

Since it follows from (2.29) that the interface localizes in a single sub-cell if $\varrho \leq \varrho_{\min}$ ($m^0_\varrho = 1\ \varrho < \varrho_{\min}, m^0_{\varrho_{\min}} = \varsigma 2^{\varrho_{\min}(d-1)} \approx 1$), the number of preserved

sub-cells across all levels is given by

$$m_\varrho^+ \approx \begin{cases} 0 & \varrho \leq \varrho_{\min} \ \wedge \ \eta \leq \frac{1}{2} \\ 2^d - 1 & \varrho \leq \varrho_{\min} \ \wedge \ \eta > \frac{1}{2} \\ \varsigma 2^{\varrho(d-1)-1} & \varrho > \varrho_{\min} \end{cases} \tag{2.35}$$

where the volume fraction η dictates which side of the interface corresponds to the interior domain. For example, for the cut element displayed in Figure 2.3 the volume fraction is larger than a half, which implies that all but one of the sub-cells at level $\varrho = 1$ are preserved. If the complement of the element would be considered, none of the level $\varrho = 1$ sub-cells would be included in the partitioning.

If we denote the average number of integration points per octree sub-cell by $\bar{q}_\varrho$, the total number of integration points per level follows as

$$n_\varrho \approx \begin{cases} 0 & \varrho \leq \varrho_{\min} \ \wedge \ \eta \leq \frac{1}{2} \\ \bar{q}_\varrho (2^d - 1) & \varrho \leq \varrho_{\min} \ \wedge \ \eta > \frac{1}{2} \\ \bar{q}_\varrho \varsigma 2^{\varrho(d-1)-1} & \varrho_{\min} < \varrho \leq \varrho_{\max} \\ \bar{q}_\varrho \bar{t}_d \varsigma 2^{\varrho_{\max}(d-1)} & \varrho = \varrho_{\max} + 1 \end{cases} \tag{2.36}$$

where $\bar{t}_d$ is the number of sub-cells in which the lowest bisectioning level $\varrho = \varrho_{\max} + 1$ is tessellated (see Appendix A). The total number of integration points for a level $\varrho_{\max}$ octree partitioning then follows as

$$N_{\varrho_{\max}} \approx \begin{cases} \varsigma \left(\bar{q}_{\varrho_{\max}+1} \bar{t}_d 2^{\varrho_{\max}(d-1)} + \displaystyle\sum_{\varrho=\varrho_{\min}+1}^{\varrho_{\max}} \bar{q}_\varrho 2^{\varrho(d-1)-1} \right) & \eta \leq \frac{1}{2}, \\ \varsigma \left(\bar{q}_{\varrho_{\max}+1} \bar{t}_d 2^{\varrho_{\max}(d-1)} + \displaystyle\sum_{\varrho=\varrho_{\min}+1}^{\varrho_{\max}} \bar{q}_\varrho 2^{\varrho(d-1)-1} \right) & \\ \quad + \displaystyle\sum_{\varrho=1}^{\varrho_{\min}} \bar{q}_\varrho \left(2^d - 1\right) & \eta > \frac{1}{2}. \end{cases} \tag{2.37}$$

This approximation reflects the strong dependence of the number of integration points on the surface fraction ς. When ς is very small, $\varrho_{\min}$ is very large, as a result of which a significant number of integration points for non-intersected sub-cells can be present (depending on the volume fraction). When ς is large, or, more generally when $\varrho_{\max} \gg \varrho_{\min}$, the number of non-intersected sub-cells is negligible. For the case that the Gauss order is selected equal on all levels, *i.e.*, $\bar{q}_\varrho = q_{\text{line}}^d$ for $\varrho \leq \varrho_{\max}$ (with q_{line} the number of integration points along a one-dimensional line segment) and $\bar{q}_{\varrho_{\max}+1} = q_{\text{tes},d}$, the number of integration points (2.37) reduces to:

$$N_{\varrho_{\max}} \approx \varsigma \left(q_{\text{tes},d} \bar{t}_d + \frac{q_{\text{line}}^d}{2 - 2^{2-d}} \right) 2^{\varrho_{\max}(d-1)} \qquad \varrho_{\max} \gg \varrho_{\min}. \tag{2.38}$$

This estimate conveys that the number of integration points scales linearly with the surface fraction, and exponentially with the octree depth. This exponential scaling depends on the number of spatial dimensions. In two dimensions the number of integration points doubles while increasing the octree depth by one, while in three dimensions it quadruples. Hence, in comparison to the two-dimensional setting, the number of integration points in three dimensions scales much more dramatically with the octree depth. A numerical study of the number of integration points in relation to the scaling relations presented in this section is provided in Section 2.4.

2.3.2 Cut-cell integration errors

In Section 2.2.2 the application of Strang's first lemma to the quantification of integration errors in the finite-cell method was discussed. Equations (2.18) and (2.19) convey that the integration error can be estimated by the summation of the operator-dependent element-integration-error indicators e_K^b and e_K^a. This section discusses the evaluation of these errors and their localization to the integration sub-cells. In order to evaluate the cell-wise integration error, in Section 2.3.2 the integration errors are expressed as the product of an operator-independent integration error and a scaling term depending on the operator. Subsequently, in Section 2.3.2 the numerical evaluation of the operator-independent cut-cell integration error is discussed.

Integration error definition

The element-integration-error indicators e_K^b and e_K^a, expressed by equations (2.17) and (2.20) respectively, depend on the operations A_Ω^h and B_Ω^h. It is generally desirable to apply a single integration scheme for all terms and, hence, to have uniform control over e_K^a and e_K^b. To enable a uniform treatment of both e_K^a and e_K^b, we first note that the integrals in the numerators of (2.17) and (2.20) constitute linear functionals on V_K^h. By the Riesz-representation theorem, there exist elements $T^a, T^b \in V_K^h$ such that

$$\int_K T^a v_K^h \,\mathrm{d}V = \int_K A_\Omega^h \left(\mathcal{I}^h u, v_K^h\right)(\boldsymbol{x}_K)\,\mathrm{d}V, \tag{2.39a}$$

$$\int_K T^b v_K^h \,\mathrm{d}V = \int_K B_\Omega^h \left(v_K^h\right)(\boldsymbol{x}_K)\,\mathrm{d}V, \tag{2.39b}$$

for all $v_K^h \in V_K^h$. We now proceed under the assumption that the error introduced in the numerical integration is equivalent for the original functionals and their Riesz-representation, *i.e.*, the difference in applying the integral rate to the left-

and right-hand-side members of (2.39a) and (2.39b) is negligible. It then holds that

$$e_K^a \leq \sup_{v_K^h \in V_K^h} \frac{\left| \int_K T^a v_K^h \, \mathrm{d}V - \sum_{l=1}^{l_K} \omega_K^l \left(T^a v_K^h \right) (\boldsymbol{x}_K^l) \right|}{\| v_K^h \|_{V_K^h}}$$

$$\leq \| T^a \|_{L^2(K)} \sup_{T^a, v_K^h \in V_K^h} \frac{\left| \int_K T^a v_K^h \, \mathrm{d}V - \sum_{l=1}^{l_K} \omega_K^l \left(T^a v_K^h \right) (\boldsymbol{x}_K^l) \right|}{\| T^a \|_{L^2(K)} \| v_K^h \|_{V_K^h}}, \tag{2.40}$$

for (2.20) and the following similar expression applies to (2.17)

$$e_K^b \leq \| T^b \|_{L^2(K)} \sup_{T^b, v_K^h \in V_K^h} \frac{\left| \int_K T^b v_K^h \, \mathrm{d}V - \sum_{l=1}^{l_K} \omega_K^l \left(T^b v_K^h \right) (\boldsymbol{x}_K^l) \right|}{\| T^b \|_{L^2(K)} \| v_K^h \|_{V_K^h}}. \tag{2.41}$$

For T^a, v_K^h in the polynomial space V_K^h, the product $T^a v_K^h$ resides in the double-degree polynomial space P_K. By virtue of the equivalence of norms in finite dimensional spaces, P_K can be equipped with a norm $\| \cdot \|_{P_K}$ such that

$$\| T^a \|_{L^2(K)} \| v_K^h \|_{V_K^h} \geq \frac{1}{\mathcal{C}_K} \| T^a v_K^h \|_{P_K}, \tag{2.42}$$

for a constant $\mathcal{C}_K \geq 1$. A similar inequality holds for T^b. Defining $\mathcal{C}_K^a = \mathcal{C}_K \| T^a \|_{L^2(K)}$ and $\mathcal{C}_K^b = \mathcal{C}_K \| T^b \|_{L^2(K)}$, it follows from (2.40) and (2.41) that $e_K^a \leq \mathcal{C}_K^a e_K^p$ and $e_K^b \leq \mathcal{C}_K^b e_K^p$ with

$$e_K^p = \sup_{p_K \in P_K} \frac{\left| \int_K p_K(\boldsymbol{x}_K) \, \mathrm{d}V - \sum_{l=1}^{l_K} \omega_K^l p_K(\boldsymbol{x}_K^l) \right|}{\| p_K \|_{P_K}}. \tag{2.43}$$

Substitution of the operation-independent element integration errors (2.43) in the global error bound (2.21) yields

$$\left\| u - u_{\mathcal{Q}}^h \right\|_{W(h)} \leq \left(1 + \frac{\left\| a^h \right\|_{W(h), V^h}}{\alpha^h} \right) \left\| u - \mathcal{I}^h u \right\|_{W(h)} + \frac{1}{\alpha^h} \sum_{K \in \mathcal{T}_\Omega^h} \left(\mathcal{C}_K^b + \mathcal{C}_K^a \right) e_K^p. \tag{2.44}$$

The error bound (2.44) conveys that the integration errors can be controlled through the operator-independent elemental integration errors e_K^p.

Integration error evaluation

The element integration error (2.43) can be approximated by discretization of the space P_K. We select the finite dimensional space P_K as the tensor-product space of univariate polynomials of order k over the element K, *i.e.*,

$$p_K(\boldsymbol{x}_K) = \prod_{i=1}^{d} \sum_{j=0}^{k} \alpha_{i,j} x_{K,i}^{j} = \boldsymbol{\Phi}(\boldsymbol{x}_K)^T \mathbf{a}, \tag{2.45}$$

with $\boldsymbol{\Phi}(\boldsymbol{x}_K) \in \mathbb{R}^{n_p}$ the basis of monomials of size $n_p = (k+1)^d$ and $\mathbf{a} \in \mathbb{R}^{n_p}$ the corresponding coefficients. The space P_K is equipped with the Sobolev norm $\|\cdot\|_{H^r}$ with $r \geq 0$. Using the polynomial basis (2.45), the element integration error (2.43) can be expressed as

$$\begin{aligned} e_K^p &= \sup_{p_K \in \mathbb{P}_K} \frac{\left| \int_K p_K(\boldsymbol{x}_K) \, \mathrm{d}V - \sum_{l=1}^{l_K} \omega_K^l p_K(\boldsymbol{x}_K^l) \right|}{\|p_K\|_{H^r(K)}} \\ &= \sqrt{\sup_{p_K \in \mathbb{P}_K} \frac{\left| \int_K p_K(\boldsymbol{x}_K) \, \mathrm{d}V - \sum_{l=1}^{l_K} \omega_K^l p_K(\boldsymbol{x}_K^l) \right|^2}{\|p_K\|_{H^r(K)}^2}} \\ &= \sqrt{\sup_{\mathbf{a} \in \mathbb{R}^{n_p}} \frac{\mathbf{a}^T (\boldsymbol{\xi} - \bar{\boldsymbol{\xi}})(\boldsymbol{\xi} - \bar{\boldsymbol{\xi}})^T \mathbf{a}}{\mathbf{a}^T \mathbf{G} \mathbf{a}}} \\ &= \sqrt{\lambda_{\max}}. \end{aligned} \tag{2.46}$$

where $\lambda_{\max}$ is the largest eigenvalue of the generalized eigenvalue problem

$$\left[\left(\boldsymbol{\xi} - \bar{\boldsymbol{\xi}} \right) \left(\boldsymbol{\xi} - \bar{\boldsymbol{\xi}} \right)^T \right] \mathbf{v}_i = \lambda_i \mathbf{G} \mathbf{v}_i \qquad i = 1, \ldots, n_p, \tag{2.47}$$

with eigenvalues λ_i and eigenvectors $\mathbf{v}_i$, and with the vectors $\boldsymbol{\xi}$ and $\bar{\boldsymbol{\xi}}$ defined as

$$\boldsymbol{\xi} = \int_K \boldsymbol{\Phi} \, \mathrm{d}V \tag{2.48a}$$

$$\bar{\boldsymbol{\xi}} = \sum_{l=1}^{l_K} \omega_K^l \boldsymbol{\Phi}(\boldsymbol{x}_K^l). \tag{2.48b}$$

The matrix $\mathbf{G}$ in (2.47) is defined as the Gramian matrix for the basis functions $\boldsymbol{\Phi}$ associated with the $H^r(K)$ Sobolev space.

Since the left-hand-side matrix in the eigenvalue problem (2.47) is the dyadic product of two vectors its rank is equal to one. As a consequence, the eigenvalue problem (2.47) only has one non-zero eigenvalue, with the corresponding eigenvector being in the direction $\mathbf{G}^{-1}(\boldsymbol{\xi} - \bar{\boldsymbol{\xi}})$. Since the Gramian matrix is positive definite, this single non-zero eigenvalue is positive and therefore equal to the maximum eigenvalue. This maximum eigenvalue and its corresponding eigenvector are equal to

$$\lambda_{\max} = \|\boldsymbol{\xi} - \bar{\boldsymbol{\xi}}\|^2_{\mathbf{G}^{-1}}, \tag{2.49a}$$

$$\mathbf{v}_{\max} = \frac{\mathbf{G}^{-1}(\boldsymbol{\xi} - \bar{\boldsymbol{\xi}})}{\|\boldsymbol{\xi} - \bar{\boldsymbol{\xi}}\|_{\mathbf{G}^{-1}}}, \tag{2.49b}$$

with

$$\|\boldsymbol{\xi} - \bar{\boldsymbol{\xi}}\|_{\mathbf{G}^{-1}} = \sqrt{(\boldsymbol{\xi} - \bar{\boldsymbol{\xi}})^T \mathbf{G}^{-1} (\boldsymbol{\xi} - \bar{\boldsymbol{\xi}})}. \tag{2.50}$$

Substitution of $\lambda_{\max}$ in equation (2.46) expresses the element integration error in terms of the vector $\boldsymbol{\xi} - \bar{\boldsymbol{\xi}}$ as:

$$e^p_K = \|\boldsymbol{\xi} - \bar{\boldsymbol{\xi}}\|_{\mathbf{G}^{-1}}. \tag{2.51}$$

The polynomial corresponding to this error, *i.e.*, the function in P_K yielding the largest integration error follows directly from the maximum eigenvector $\mathbf{v}_{\max}$ as

$$\begin{aligned} p_{K,\max} &= \boldsymbol{\Phi}^T \mathbf{v}_{\max} \\ &= \frac{\boldsymbol{\Phi}^T \mathbf{G}^{-1}(\boldsymbol{\xi} - \bar{\boldsymbol{\xi}})}{\|\boldsymbol{\xi} - \bar{\boldsymbol{\xi}}\|_{\mathbf{G}^{-1}}}. \end{aligned} \tag{2.52}$$

Note that the scaling of this function with respect to the norm (2.50) is chosen such that

$$\begin{aligned} \|p_{K,\max}\|^2_{H^r(K)} &= \mathbf{v}^T_{\max} \mathbf{G} \mathbf{v}_{\max} \\ &= \frac{(\boldsymbol{\xi} - \bar{\boldsymbol{\xi}})^T \mathbf{G}^{-1} (\boldsymbol{\xi} - \bar{\boldsymbol{\xi}})}{\|\boldsymbol{\xi} - \bar{\boldsymbol{\xi}}\|^2_{\mathbf{G}^{-1}}} = 1, \end{aligned} \tag{2.53}$$

and hence

$$e^p_K = \left| \int_K p_{K,\max}(\boldsymbol{x}_K) \, \mathrm{d}V - \sum_{l=1}^{l_K} \omega^l_K p_{K,\max}(\boldsymbol{x}^l_K) \right|. \tag{2.54}$$

2.3.3 The cut element quadrature optimization algorithm

The ability to evaluate the maximal cut-cell integration error and the corresponding integrand that leads to this error serves as the basis for the quadrature-optimization procedure developed in this work. The error (2.54) is, in general, controlled by the octree depth, $\varrho_{\max}$, and the quadrature rules selected on all sub-cells. In the remainder of this work we assume the octree depth to be fixed at a level where the partitioning error is negligible.

For the space of polynomials considered above, *i.e.*, those of equation (2.45), the vector $\boldsymbol{\xi}$ of basis function integrals can be evaluated exactly on the exact or approximated geometry and by appropriate selection of the Gauss integration order. As discussed in Section 2.3.1, the usage of exact Gauss integration on all sub-cells is impractical from a computational effort point of view in large scale finite-cell simulations. The conceptual idea behind the quadrature optimization procedure developed here is to determine integration rules with a significantly smaller number of integration points, and to use the evaluable integration error (2.54) to find the optimal distribution of these points over the sub-cells.

The developed optimization procedure is intended as a per-element preprocessing operation, which results in optimized quadrature rules for all cut elements in a finite cell simulation. In order to conduct this preprocessing operation, for each cut element the exact shape function integrals $\boldsymbol{\xi}$ and the Gramian $\mathbf{G}$ must be determined. These computations are relatively expensive, but especially in time-dependent or non-linear problems – where the optimised quadrature scheme will be re-used many times – the computational gain in the simulations outweighs the effort in this preprocessing operation. The computational effort will be studied in Section 2.4.3.

In Section 2.3.3 we first formalize the integration error minimization problem that serves as the basis for the optimization algorithm. In Section 2.3.3 the algorithm is then introduced and various algorithmic aspects are discussed.

The minimization problem

We consider an octree partitioning of the element K as described in Section 2.3.1 and denote this partitioning by $\mathcal{P} = \bigcup_{\varrho=0}^{\varrho_{\max}+1} \mathcal{P}_{K,\varrho}^{\varrho_{\max}}$, with $\mathcal{P}_{K,\varrho}^{\varrho_{\max}}$ the sub-cells corresponding to the octree level ϱ. On each sub-cell $\wp \in \mathcal{P}$ a sequence of quadrature rules is defined:

$$\left\{ \mathcal{Q}_{\wp}^{\imath} \mid \imath = 0, 1, 2, \dots \right\}. \tag{2.55}$$

We assume that the quadrature rules $\mathcal{Q}_{\wp}^{\imath}$ are nested, in the sense that the set of polynomials on $\wp$ that are integrated exactly by $\mathcal{Q}_{\wp}^{\imath+1}$ includes all polynomials that are integrated exactly by $\mathcal{Q}_{\wp}^{\imath}$. The quadrature rule for the complete

partitioning is defined as

$$\mathcal{Q}_{\mathcal{P}}^{\boldsymbol{\imath}} = \bigcup_{\wp \in \mathcal{P}} \mathcal{Q}_{\wp}^{\imath_\wp}, \tag{2.56}$$

with $\boldsymbol{\imath} = \{\imath_\wp \mid \wp \in \mathcal{P}\}$ a list of length $m = \#\mathcal{P}$ with sub-cell quadrature rule indices. For a given type of integration rule, *e.g.*, Gauss integration, the index list $\boldsymbol{\imath} \in \mathbb{N}^m$ fully determines the quadrature rule for the partitioned cut-cell K.

Both the total number of integration points in the cut element, defined by $\#\mathcal{Q}_{\wp}^{\imath}$, and the integration error (2.54) can be expressed as functions of the quadrature index list $\boldsymbol{\imath}$:

$$e_K^{p,\boldsymbol{\imath}} = \left| \int_K p_{K,\max}(\boldsymbol{x}_K)\,\mathrm{d}V - \sum_{\wp \in \mathcal{P}} \sum_{(\omega_K, \boldsymbol{x}_K) \in \mathcal{Q}_{\wp}^{\imath_\wp}} \omega_K p_{K,\max}(\boldsymbol{x}_K) \right|, \tag{2.57a}$$

with $p_{K,\max}$ according to (2.52) for the integration rule $\mathcal{Q}_{\mathcal{P}}^{\boldsymbol{\imath}}$. Using these functions the intended quadrature optimization procedure can be formulated as the constrained minimization problem

$$\underset{\boldsymbol{\imath} \in \mathbb{N}^m}{\text{minimize}} \quad e_K^{p,\boldsymbol{\imath}} \quad \text{subject to} \quad \#\mathcal{Q}_{\mathcal{P}}^{\boldsymbol{\imath}} = q^\star, \tag{2.58}$$

with $q^\star$ the specified number of integration points. Conversely, we can express the optimization problem in terms of the minimization of the number of integration points for a fixed error

$$\underset{\boldsymbol{\imath} \in \mathbb{N}^m}{\text{minimize}} \quad \#\mathcal{Q}_{\mathcal{P}}^{\boldsymbol{\imath}} \quad \text{subject to} \quad e_K^{p,\boldsymbol{\imath}} = e_K^\star, \tag{2.59}$$

with $e_K^\star$ the intended error level.

Remark 2.1 (Quadrature refinement strategy)**.** *In this work, the integration quadrature per sub-cell is refined by increasing the order of the integration scheme. Alternatively, refinement would be possible by subdivision of a sub-cell and then constructing quadrature rules over the refined sub-cells. Since we consider the integration of smooth functions over the cut-cells, improving the quadrature by increasing its order is more efficient than spatial refinement of the sub-cells.*

The optimization algorithm

The developed algorithm to obtain the optimal distribution of integration points over all sub-cells is presented in Algorithm 2.1. This algorithm approximates the minimization problem (2.58) through the generation of a sequence of refinement schemes, $\{\boldsymbol{\imath}^0, \boldsymbol{\imath}^1, \cdots, \boldsymbol{\imath}^\star\}$, where for the initial quadrature rule the lowest order of

integration on each sub-cell is considered, *i.e.*, $\boldsymbol{\imath}^0 = \mathbf{0}$, and where the optimized quadrature rule (approximately) satisfying the constraint condition is denoted by $\boldsymbol{\imath}^\star$.

Given the r-th iterate in the optimization procedure, $\boldsymbol{\imath}^r$, the next integration rule in the sequence, $\boldsymbol{\imath}^{r+1}$, is determined in three steps:

1. The function corresponding to the maximum integration error, $p_{K,\max}$, is determined for an integration rule $\boldsymbol{\imath}^r$ using equation (2.52). The corresponding integration error can be found using (2.57a). This step corresponds to the lines (3) – (11) in Algorithm 2.1.

2. The integration error (2.57a) is localized to the sub-cells in order to form indicator functions representing the sub-cell-wise error reduction per added integration point. The definition of the sub-cell indicators, which is discussed in detail below, ensures that the algorithm approximates the minimization problem (2.58). This step corresponds to the lines (11) – (12) in Algorithm 2.1.

3. The integration rule index of the sub-cells with the largest indicators, *i.e.*, with the largest reduction in error per added integration point, is increased by one as to reduce the integration error in these sub-cells. The employed marking strategies are discussed below. This step corresponds to the line (14) in Algorithm 2.1.

These steps are repeated until the specified number of integration points is reached, *i.e.*, when $\#\mathcal{Q}^{\boldsymbol{\imath}}_{\wp} \geq q^\star$. The algorithm is stopped prematurely when the integration rule sequence (2.55) is depleted. Specifically, in the considered implementation [109] the maximum quadrature orders for the lowest level tessellation is equal to 6 (12 points) for triangles and 7 (31 points) for tetrahedrons. The algorithm termination conditions correspond to line (17) in Algorithm 2.1.

Sub-cell indicators To form the sub-cell indicators we consider a uniform refinement of the integration indices, *i.e.*, $\imath^{r+1} = \imath^r + 1$ for each sub-cell. The error reduction per added integration point in the case of this uniform refinement can be expressed as

$$-\frac{e_K^{p,\boldsymbol{\imath}^{r+1}} - e_K^{p,\boldsymbol{\imath}^r}}{\#\mathcal{Q}_{\mathcal{P}}^{\boldsymbol{\imath}^{r+1}} - \#\mathcal{Q}_{\mathcal{P}}^{\boldsymbol{\imath}^r}} \leq \frac{e_K^{p,\boldsymbol{\imath}^r}}{\#\mathcal{Q}_{\mathcal{P}}^{\boldsymbol{\imath}^{r+1}} - \#\mathcal{Q}_{\mathcal{P}}^{\boldsymbol{\imath}^r}}. \tag{2.60}$$

Using the sub-cell integration error according to

$$e_{\wp}^{p,\imath} = \left| \int_{\wp} p_{K,\max}(\boldsymbol{x}_K)\,\mathrm{d}V - \sum_{(\omega_K,\boldsymbol{x}_K)\in\mathcal{Q}_{\wp}^{\imath}} \omega_K p_{K,\max}(\boldsymbol{x}_K) \right|, \tag{2.61}$$

and the reduction in the number of integration points[1], the error reduction (2.60) is bounded as:

$$\begin{aligned}
\frac{e_K^{p,\boldsymbol{\imath}^r}}{\#\mathcal{Q}_{\mathcal{P}}^{\boldsymbol{\imath}^{r+1}} - \#\mathcal{Q}_{\mathcal{P}}^{\boldsymbol{\imath}^r}} &\leq \sum_{(\imath,\wp)\in(\boldsymbol{\imath},\mathcal{P})} \frac{e_\wp^{p,\imath^r}}{\#\mathcal{Q}_{\mathcal{P}}^{\boldsymbol{\imath}^{\boldsymbol{r+1}}} - \#\mathcal{Q}_{\mathcal{P}}^{\boldsymbol{\imath}^{\boldsymbol{r}}}} \\
&\leq \frac{1}{m} \frac{\max\limits_{(\imath,\wp)\in(\boldsymbol{\imath},\mathcal{P})} \left(\#\mathcal{Q}_\wp^{\imath^{r+1}} - \#\mathcal{Q}_\wp^{\imath^r}\right)}{\min\limits_{(\imath,\wp)\in(\boldsymbol{\imath},\mathcal{P})} \left(\#\mathcal{Q}_\wp^{\imath^{r+1}} - \#\mathcal{Q}_\wp^{\imath^r}\right)} \\
&\qquad \sum_{(\imath,\wp)\in(\boldsymbol{\imath},\mathcal{P})} \frac{e_\wp^{p,\imath^r}}{\left(\#\mathcal{Q}_\wp^{\imath^{r+1}} - \#\mathcal{Q}_\wp^{\imath^r}\right)} \\
&\leq \frac{c}{m} \sum_{(\imath,\wp)\in(\boldsymbol{\imath},\mathcal{P})} \mathcal{R}_\wp^{\imath^r},
\end{aligned} \tag{2.62}$$

with the sub-cell indicator function defined as

$$\mathcal{R}_\wp^\imath = \frac{e_\wp^{p,\imath}}{\#\mathcal{Q}_\wp^{\imath+1} - \#\mathcal{Q}_\wp^\imath}. \tag{2.63}$$

The sub-cell indicator (2.63) weighs the local integration error by the cost of increasing the integration order. Under the assumption that the sub-cell integration error reduces significantly when increasing the order of the integration scheme by one, the numerator in (2.63) can be interpreted as the error reduction rather than the error itself. This assumption of local error reduction is, however, in general, not satisfied. As a consequence, in the proposed algorithm sub-cells whose integration error does not decrease significantly (or even increases) by raising the order of integration by one would be underrated in the indicator function. Considering the error rather than the error reduction in the indicator provides a more robust measure of the potential to decrease the error in a sub-cell.

Sub-cell marking strategy Based on the upper bound (2.62) of the error reduction per added integration point, sub-cells are marked for refinement. We

[1] $$\begin{aligned}
\#\mathcal{Q}_{\mathcal{P}}^{\boldsymbol{\imath}^{r+1}} - \#\mathcal{Q}_{\mathcal{P}}^{\boldsymbol{\imath}^r} &\geq m \min\left(\#\mathcal{Q}_\wp^{\imath^{r+1}} - \#\mathcal{Q}_\wp^{\imath^r}\right) \\
&\geq m \frac{\min\left(\#\mathcal{Q}_\wp^{\imath^{r+1}} - \#\mathcal{Q}_\wp^{\imath^r}\right)}{\max\left(\#\mathcal{Q}_\wp^{\imath^{r+1}} - \#\mathcal{Q}_\wp^{\imath^r}\right)} \max\left(\#\mathcal{Q}_\wp^{\imath^{r+1}} - \#\mathcal{Q}_\wp^{\imath^r}\right) \\
&\geq m \frac{\min\left(\#\mathcal{Q}_\wp^{\imath^{r+1}} - \#\mathcal{Q}_\wp^{\imath^r}\right)}{\max\left(\#\mathcal{Q}_\wp^{\imath^{r+1}} - \#\mathcal{Q}_\wp^{\imath^r}\right)} \left(\#\mathcal{Q}_\wp^{\imath^{r+1}} - \#\mathcal{Q}_\wp^{\imath^r}\right).
\end{aligned}$$

herein consider two marking strategies, with $\mathcal{M}$ denoting the set of sub-cells marked for increasing the integration order:

- *Sub-cell marking:* In this marking strategy we increase the integration order of the single sub-cell with the largest indicator:

$$\mathcal{M} = \underset{(\imath,\wp)\in(\boldsymbol{\imath},\mathcal{P})}{\arg\max} \left(\mathcal{R}^{\imath}_{\wp}\right). \tag{2.64}$$

- *Octree level marking:* Sub-cell marking involves refinement of individual sub-cell in each refinement step. This generally leads to a large number of optimization steps, which is undesirable from a computational effort point of view. In order to expedite the optimization procedure, it is possible to mark multiple sub-cells simultaneously. A natural way of marking multiple sub-cells is to agglomerate all sub-cells in a particular level. This leads to a marking strategy in which we increase the integration order for the level (in the octree partitioning) with the largest indicator:

$$\mathcal{M} = \underset{\{\mathcal{P}^{\varrho}_{K}\}_{\varrho=0}^{\varrho_{\max}+1}}{\arg\max} \left(\sum_{(\imath,\wp)\in\left(\boldsymbol{\imath},\mathcal{P}^{\varrho}_{K}\right)} \mathcal{R}^{\imath}_{\wp} \right). \tag{2.65}$$

Remark 2.2 (Global adaptive integration)**.** *The algorithm presented above is framed in an element-by-element setting,* i.e.*, each cut element is considered separately in the preprocessing operation to determine the integration schemes. Since the evaluation of the integration error and its indicators involves the repeated computation of solutions to a linear system with the size equal to the number of supported basis functions,* i.e.*, equation* (2.54)*, global application of the algorithm is impractical from a computational effort point of view.*

It should be noted that the element-by-element setting considered here does not account for the fact that the operator-dependent constants in the multiplicative decomposition (2.43) *in principle vary per element. For example, a source term might be negligibly small on an element that is challenging to integrate, and hence improving quadrature on that element would be inefficient from the error approximation point of view. Additional integration points are, however, assigned to such a cut element in the element-by-element strategy employed here, as the operator is not considered in the marking strategy.*

*Global adaptive integration is feasible, however, by computing the indicators on all elements individually (*i.e.*, solving multiple relatively small local linear systems, rather than a large global system), and then to apply a global marking strategy. Both the sub-cell marking and the level marking strategy can then be applied globally. The potential benefit of such a global marking strategy is*

that it automatically accounts for the fact that not all cut elements are equally hard to integrate. However, in principle, the operator-dependent constants should then be incorporated in the error indicators. Moreover, one should be aware that parallelization of the preprocessing operation, which is trivial in the per-element setting, is more challenging using the global marking strategy, as communication between the elements is required. The development of a rigorous global integration optimization routine, which would consider the above-mentioned complications, is considered beyond the scope of the current work.

2.4 Numerical study of the adaptive integration procedure

To assess the performance of the developed adaptive integration technique, in this section we study its behavior in terms of integration accuracy versus the number of integration points. We here consider the case of a single cut-cell. The effect of the integration accuracy on actual finite-cell simulations will be studied subsequently in Section 2.5.

We consider a d-dimensional unit cube $[0,1]^d$ in two and three dimensions as a single element of the background mesh. Throughout this section we exclude an ellipsoid with semi-major axis r_1 and semi-minor axes r_2 (with the sphere as the special case $r_1 = r_2$) that is centered at the origin of the background element and with its major axis residing in the $x_1 - x_2$ plane at an inclination of φ with respect to the x_1 axis. The resulting cut-element corresponds to the subdomain for which the level set function

$$l(\boldsymbol{x}) = \left(\frac{\bar{x}_1}{r_1}\right)^2 + \sum_{\delta=2}^{d} \left(\frac{\bar{x}_\delta}{r_2}\right)^2 \tag{2.66}$$

is positive, where $\bar{x}_1 = x_1 \cos\varphi - x_2 \sin\varphi$, $\bar{x}_2 = x_1 \sin\varphi + x_2 \cos\varphi$, and, if $d = 3$, $\bar{x}_3 = x_3$. Schematics of various cut-elements generated using the octree procedure, discussed in Section 2.3.1, with a maximum octree depth of $\varrho_{\max} = 3$ for a spherical exclusion and $\varrho_{\max} = 4$ for an elliptical exclusion are displayed in Figure 2.4. Here, the maximum octree depth is chosen such that the geometric error is negligible.

2.4.1 Equal-order degree integration

Before we proceed with the presentation of the results obtained using the adaptive integration procedure, we first consider the case for which the order of the integration scheme is chosen to be same over all integration sub-cells. We consider

Algorithm 2.1: Adaptive integration algorithm.

Input: $\mathcal{P} = \cup_0^{\varrho_{\max}+1} \mathcal{P}_K^{\varrho} = \cup\wp$, k, *marking_strategy*
Output: $\mathcal{Q}_{\mathcal{P}}^{\imath^\star}$ # optimized quadrature rule

Initialization
$\boldsymbol{\imath} = \mathbf{0}$ # quadrature index
$\boldsymbol{\Phi}$ =get_monomial_basis(k) # monomial basis

Reference integrals
$\xi_i = \int_{\mathcal{P}} \Phi_i \, \mathrm{d}V$ # basis function integral vector
$G_{ij} = (\Phi_i, \Phi_j)_{H^1}$ # Gramian matrix

while not *terminate* **:**

Quadrature rule
$\mathcal{Q}_{\mathcal{P}}^{\boldsymbol{\imath}}$ = get_quadrature($\mathcal{P}$) # quadrature rule
$\bar{\boldsymbol{\xi}} = \sum_{(\omega_{\mathcal{P}}, \boldsymbol{x}_{\mathcal{P}}) \in \mathcal{Q}_{\mathcal{P}}^{\boldsymbol{\imath}}} \omega_{\mathcal{P}} \boldsymbol{\Phi}(\boldsymbol{x}_{\mathcal{P}})$ # approximate basis function integral vector

Worst possible function to integrate
$\mathbf{v}_{\max} = \dfrac{\mathbf{G}^{-1}(\boldsymbol{\xi} - \bar{\boldsymbol{\xi}})}{\|\boldsymbol{\xi} - \bar{\boldsymbol{\xi}}\|_{\mathbf{G}^{-1}}}$ # eigenvector
$p_{\max} = \boldsymbol{\Phi}^T \mathbf{v}_{\max}$ # eigenfunction

Sub-cell errors and indicators
for $\wp$ **in** $\mathcal{P}$**:**
$e_{\wp}^{p,\imath_\wp} = \left| \int_{\wp} p_{\max}(\boldsymbol{x}_{\mathcal{P}}) \, \mathrm{d}V - \sum_{(\omega_\wp, \boldsymbol{x}_\wp) \in \mathcal{Q}_{\wp}^{\imath_\wp}} \omega_\wp p_{\max}(\boldsymbol{x}_\wp) \right|$ # error
$\mathcal{R}_{\wp}^{\imath_\wp} = \frac{e_{\wp}^{p,\imath_\wp}}{\#\mathcal{Q}_{\wp}^{\imath_\wp+1} - \#\mathcal{Q}_{\wp}^{\imath_\wp}}$ # indicator
end

Sub-cells marked for refinement
$\mathcal{M}$ = mark_sub_cells($\{\mathcal{R}_{\wp}^{\imath_\wp}\}_{\wp \in \mathcal{P}}$, *marking_strategy*)
for $\wp$ **in** $\mathcal{M}$**:**
$\imath_\wp \to \imath_\wp + 1$
stopping criterion
if not $\mathcal{Q}_{\wp}^{\imath}$ **then** *terminate* = **True**
end
end
return $\mathcal{Q}_{\mathcal{P}}^{\imath^\star}$

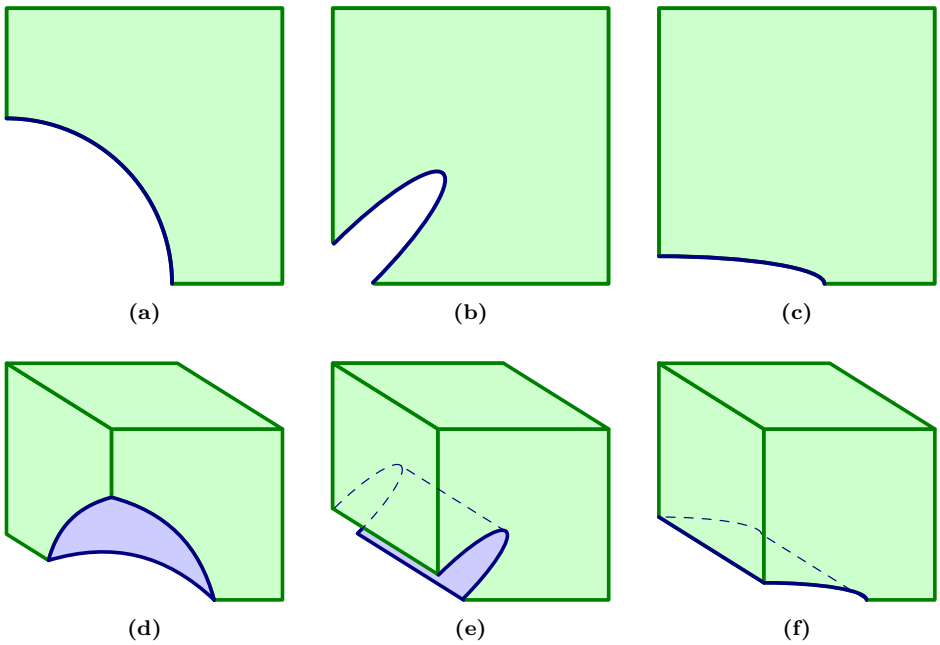


Figure 2.4: Schematic representation of the two-dimensional (top row) and three-dimensional (bottom row) cut-element corresponding to the level set function (2.66) with $r_1 = r_2 = 0.6$ for panel (a), $r_1 = 0.6$, $r_2 = 0.1$, $\varphi = 45^\circ$ for panel (b), $r_1 = 0.6$, $r_2 = 0.1$, $\varphi = 0^\circ$ for panel (c), $r_1 = r_2 = r_3 = 0.6$ for panel (d), $r_1 = 0.6$, $r_2 = 0.1$, $\varphi = 45^\circ$ for panel (e) and $r_1 = 0.6$, $r_2 = 0.1$, $\varphi = 0^\circ$ for panel (f).

the case of a spherical exclusion with $r_1 = r_2 = 0.6$. In Figure 2.5 the integration error (2.54) is plotted for the cases of $\varrho_{\max} = 3, 4, 5$ in two dimensions and $\varrho_{\max} = 2, 3, 4$ in three dimensions. In each of the sub plots the integration index $\boldsymbol{i}$ is increased uniformly over all sub-cells. For all presented results the polynomial order $k = 8$ is considered for two dimensions and $k = 5$ is considered for three dimensions. The error (2.54) is defined with respect to the H^1 norm. The integration error is in all cases computed with respect to the exact integral over the considered partitioning, and hence the geometric error corresponding to the octree partitioning is not represented in the results.

Figure 2.5 displays integration errors based on both equal-order Gauss quadrature and equal-order uniform quadrature on all sub-cells. It is observed that for all $\varrho_{\max}$ the observed errors for a particular integration order are similar, which is explained by the fact that the integration error is dominated by contributions from the sub-cells that are already present at the lowest $\varrho_{\max}$ and that the er-

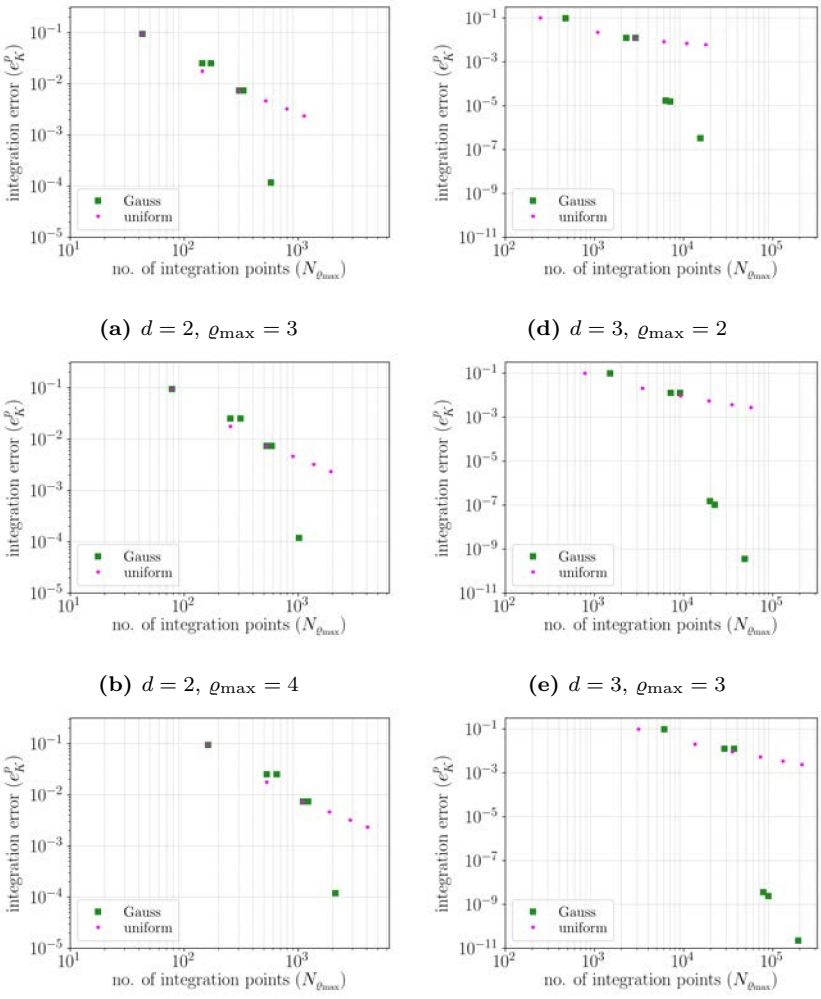
rors are computed with respect to the considered partitioning. In terms of the number of integration points, the octree depth does, however, have a substantial effect. The observed increase in number of integration points is studied in detail in Figure 2.6 for the case of a uniform scheme with 4^2 points per sub-cell in two dimensions and 4^3 points per sub-cell in three dimensions and for a Gauss scheme of order 4 on each sub-cell. This figure conveys that the total number of integration points scales in agreement with the relation (2.37). In particular the doubling of the number of points in two dimensions and the quadrupling in three dimensions while increasing the octree by a single level is clearly observed.

From Figure 2.5 it is observed that for Gauss orders that are significantly below the order required for exact integration there is not a substantial difference with uniform integration. Once the Gauss order reaches that needed for exact integration, the error observed for Gauss integration is substantially smaller than that using uniform integration. When considering the same integration order on all sub-cells, this advantage is, however, only attained at the expense of introducing a large number of integration points, in particular in cases of high octree depths.

2.4.2 Adaptive integration

We now consider the adaptive-integration procedure for the circular exclusion setting considered in Section 2.4.1. In Figure 2.7 various steps in the quadrature-optimization procedure are shows, displaying for each step the number of integration points per sub-cell (left column), error and the function leading to that error (middle column), and the sub-cell error indicators (2.54) (right column). The presented results are generated using the per sub-cell marking strategy. The sequence of steps demonstrates that in each step the worst possible function in terms of integration is determined. In the first step, this corresponds to a function that is large in magnitude at the largest sub-cell. This function leads to an indicator that is highest in that sub-cell, and hence the order of integration on that cell is increased. As a result, in the second step a function that is large in magnitude on the sub-cells surrounding the largest sub-cell is found to be the worst possible in terms of integration error. The larger volume of the $\varrho = 1$ integration cell makes, however, that the largest indicator is still found for that cell. After another increase in integration order on that sub-cell, in the subsequent steps the largest indicators are found on the $\varrho = 2$ sub-cells, which are therefore gradually increased in order.

Figure 2.8 displays the integration error versus the total number of integration points as evolving during the optimization procedure. The displayed results pertain to $\varrho_{\max} = 3$ in both two and three dimensions, displaying the equal-order results discussed in Section 2.4.1 for reference. As can be seen, the error per inte-



(a) $d = 2$, $\varrho_{\max} = 3$

(d) $d = 3$, $\varrho_{\max} = 2$

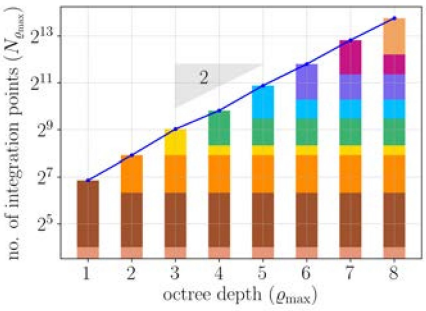
(b) $d = 2$, $\varrho_{\max} = 4$

(e) $d = 3$, $\varrho_{\max} = 3$

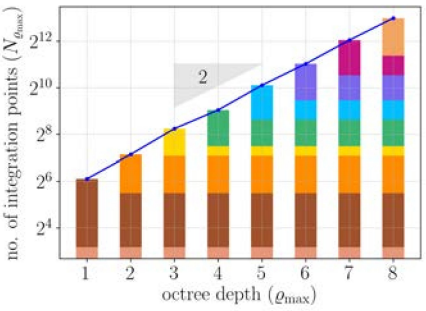
(c) $d = 2$, $\varrho_{\max} = 5$

(f) $d = 3$, $\varrho_{\max} = 4$

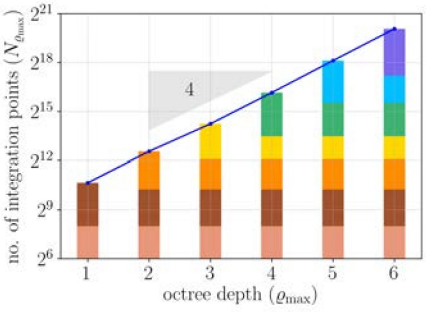
Figure 2.5: Integration errors for a spherical exclusion with radius 0.6 using the same integration scheme on all sub-cells.



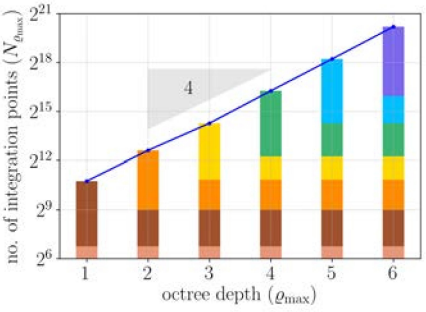
(a) Uniform, $d = 2$



(b) Gauss, $d = 2$



(c) Uniform, $d = 3$



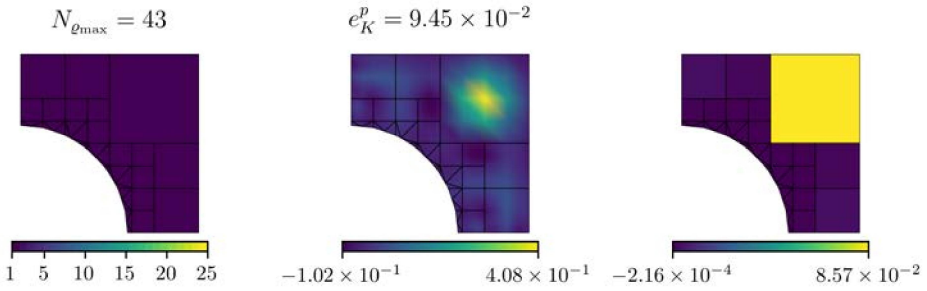
(d) Gauss, $d = 3$

Figure 2.6: Dependence of the number of integration points on the octree depth for the case of uniform integration (with 4 point per dimension) on all sub-cells (top row) and Gauss quadrature of order 4 for all sub-cells (bottom row).

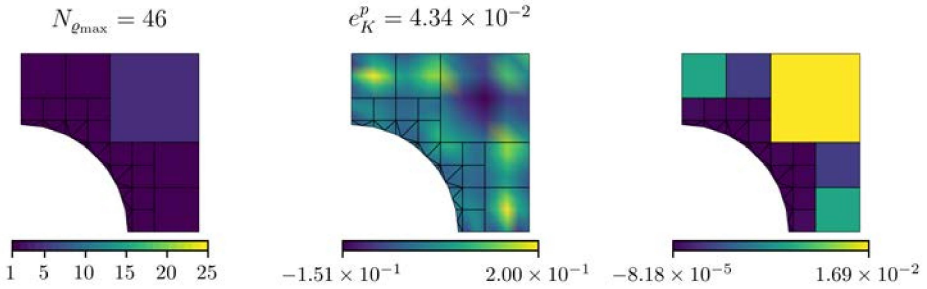
gration point is substantially lower using the adaptive integration procedure. For example, in two dimensions, the error corresponding to the equal second order Gauss scheme is equal to 2.52×10^{-2}, while that particular integration scheme involves 144 points. For the same number of points, the error corresponding to the optimized quadrature is equal to 1.00×10^{-3}, *i.e.*, a factor 25 reduction in error. Figure 2.9 displays the distribution of the integration points over the sub-cells for the equal-order Gauss scheme and the optimized case, which clearly demonstrates that the significant reduction in error is achieved by assigning more integration points to the larger sub-cells before introducing additional points in the smaller sub-cells.

In general, for a fixed integration error, the number of integration points is substantially lower using the adaptive integration procedure than when using the equal-order schemes. For example, in two dimensions, the error corresponding to the equal fourth order Gauss scheme is equal to 7.35×10^{-3}, which involves 303 points. For a similar error, the number of points for the optimized quadrature is equal to 83, *i.e.*, a reduction by a factor of approximately 4. Figure 2.10 displays the distribution of integration points over the sub-cells for the equal-order Gauss scheme and the optimized quadrature, which clearly shows the reduction in number of points for a fixed error. This figure also conveys that the observed reduction factor of 4 is significantly influenced by the initial integration order in the adaptive procedure, which in this case has been set to 1 (for this case the minimum number of points is equal to 43).

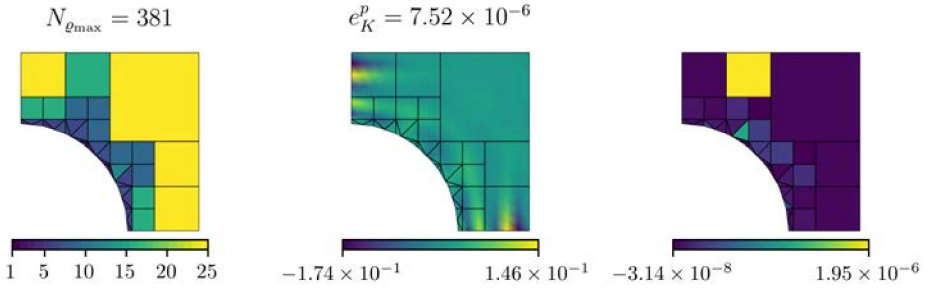
Figure 2.8b displays the error versus the number of integration points for the three-dimensional spherical exclusion with $\varrho_{\max} = 3$. The integration error corresponding to the equal second order Gauss schemes is equal to approximately 1.14×10^{-2} and pertains to 7168 integration points. For the same number of points, the adaptive procedure reduces the error to 2.67×10^{-5}, which constitutes a reduction factor of approximately 450. This reduction factor is significantly higher than that observed in the two-dimensional case, despite the similarity in asymptotic scaling rate between the integration error and the number of integration points in the two- and three-dimensional cases. The larger reduction factor in three dimensions is attributed to the significant reduction in error that is achieved during the first integration point optimization steps, in which the integration order on the $\varrho = 1$ sub-cells is increased. The difference between the two-dimensional and three-dimensional setting in this regard is the fact that in three dimensions there is a significant number of such sub-cells, whereas in two dimensions there is only one sub-cell of level $\varrho = 1$. For an integration error 1.139×10^{-2}, the adaptive procedure requires only 1646 points, which, as in the two-dimensional case, is a reduction by a factor of approximately 4. As for the two-dimensional case, this reduction factor is significantly influenced by the integration order with which the adaptive procedure is initiated.



(a) Step 1

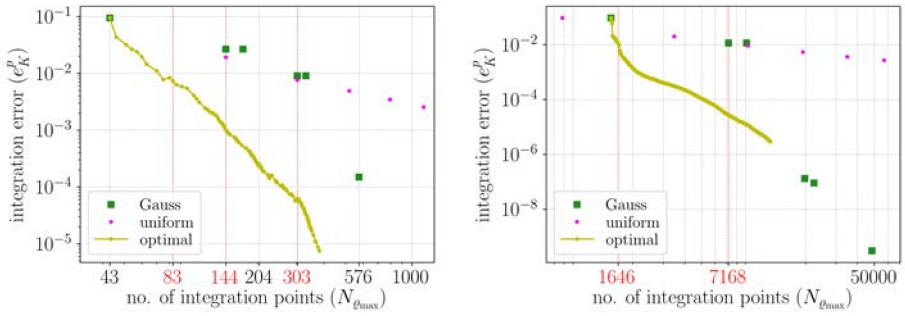


(b) Step 2



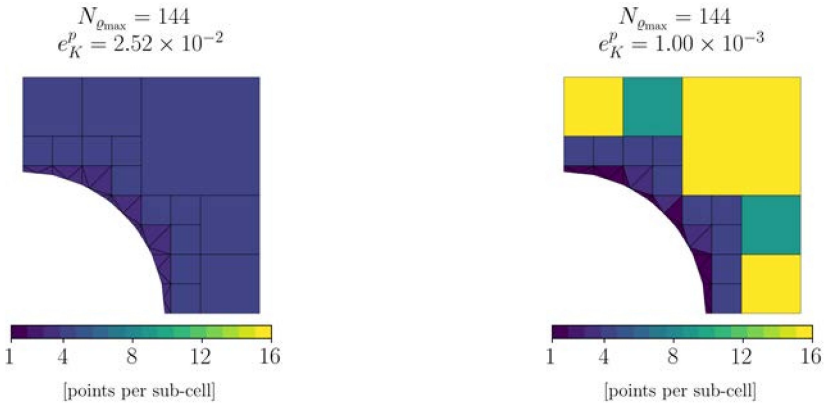
(c) Step 115

Figure 2.7: Selection of steps in the quadrature optimization procedure for a circular exclusion. (left column) The number of integration points per sub-cell and the total number of points on the cut-element. (middle column) The integration error and the function leading to that error. (right column) The sub-cell integration error indicators.



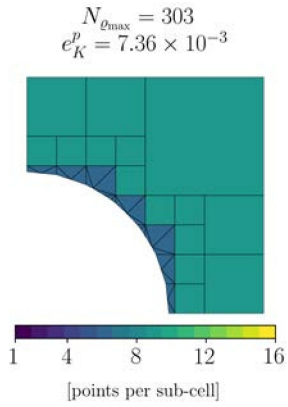
(a) $d = 2$ **(b)** $d = 3$

Figure 2.8: Evolution of the integration error and number of integration points using the per sub-cell marking strategy. The results for equal-order quadrature are shown for reference.

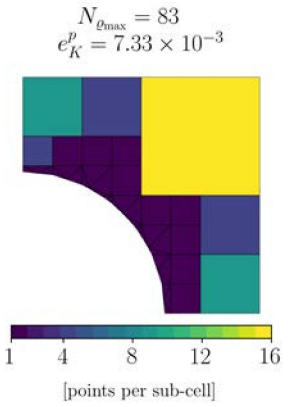


(a) Equal-order Gauss **(b)** Optimal integration

Figure 2.9: Distribution of integration points over the cut-element for 144 points in two dimensions, which corresponds to the number of points attained using a second order Gauss scheme on all sub-cells. Note that the error is reduced by a factor of 25 in two dimensions by using the per sub-cell marking strategy.



(a) Equal-order Gauss



(b) Optimal integration

Figure 2.10: Distribution of integration points over the cut-element for an error of approximately 7.3×10^{-3}. Note that the number of integration points is reduced by a factor of 4 in two dimensions by using the per sub-cell marking strategy.

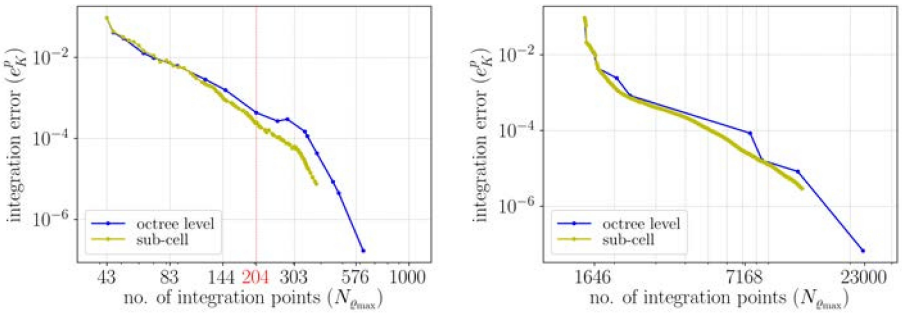
2.4.3 Influence of algorithmic settings and cut-cell configurations

In the previous section the performance of the adaptive integration routine has been studied using representative settings for the cut-cell geometry and algorithmic parameters. In this section we will study the influence of the most prominent parameters on the performance of the adaptive integration technique.

The marking strategy

From the refinement patterns that emerge from the per sub-cell refinement strategies – such as the ones discussed in the previous section – it is observed that, as a general trend, integration orders are increased on a per-level basis, *i.e.*, the number of integration points is increased from level ϱ to $\varrho_{\max} + 1$. This is explained by the fact that the indicators scale with the volume of the sub-cells. Based on this observation it is anticipated that the level-based marking strategy discussed in Section 2.3.3 can be very efficient, in the sense that it yields a similar refinement pattern as the per-cell marking, but that it needs fewer iterations by virtue of marking a larger number of sub-cells per step.

In Figure 2.11 the per-level and per-cell marking strategies are compared for the test case introduced above. In both two and three dimensions it is observed that the per-level marking strategy closely follows the per-cell marking.



(a) $d = 2$ **(b)** $d = 3$

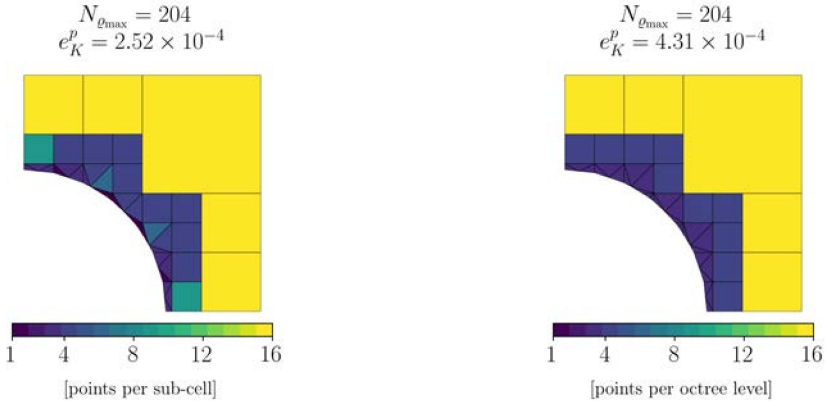
Figure 2.11: Comparison of the evolution of the integration error and number of integration points using the per-sub-cell and per-octree-level marking strategy.

In particular in the initial steps of the optimization a very close agreement is observed between the marking strategies. In Figure 2.12 we inspect the distribution of points corresponding to the two marking strategies with 204 points (in the two-dimensional setting). This figure conveys that, indeed, the distribution of integration points between the two markings is very similar.

Although the observed point distributions in Figure 2.12 are similar, the number of iterations required to attain these distributions is significantly different. The per-cell marking strategy requires 54 iterations, whereas the corresponding result using the per-level marking is achieved in 9 iterations. Evidently, this reduction in number of iterations translates into a computational-effort advantage for the per-level marking strategy.

In Table 2.1 we study the computational effort for both marking strategies corresponding to the optimization procedure to attain the results in Figure 2.11a. The reported CPU times are based on a Python implementation using the open source finite element toolbox Nutils [109], which is executed on a four core CPU with a 2.60 GHz 7th generation Intel Core i5 processor. Although the reported CPU times are highly dependent on a myriad of aspects, it is noted that the computation time of the Gramian matrix (0.4–0.45 s) is representative for the computational effort involved in the construction of the element contribution to the system matrix. By relating the reported CPU times to this element system matrix construction time, a more meaningful notion of the computational effort is obtained, although, of course, also this relative notion of performance is subject to implementation and architecture considerations.

Table 2.1 conveys that the optimization algorithm has a similar initialization time for both marking strategies. This initialization time pertains to the com-



(a) Sub-cell marking **(b)** Level marking

Figure 2.12: Distribution of integration points over a cut-element with 204 integration points in two dimensions using both sub-cell and level marking.

Table 2.1: Analysis of the CPU time to attain the results displayed in Figures 2.11a.

Marking strategy	sub-cell		octree level	
	n_{call}	t/call[s]	n_{call}	t/call[s]
Initialization	1	0.5	1	0.55
Reference basis function integral ($\boldsymbol{\xi}$)	1	0.1	1	0.1
Gramian matrix ($\mathbf{G}$)	1	0.4	1	0.45
Iterations	115	0.33	17	0.35
Approximate basis function integral ($\bar{\boldsymbol{\xi}}$)	115	0.12	17	0.1
Maximum eigenvalue ($\mathbf{v}_{\max}$)	115	0.004	17	0.005
Localized integration errors ($e_{\wp}^{p^k}$)	115	0.15	17	0.15
Integration error indicator ($\mathcal{R}$)	115	0.01	17	0.01
Integration error marker ($\mathcal{M}$)	115	0.0005	17	0.005
Post-process	115	0.05	17	0.05
Total CPU time [s]	**39**		**7**	

putation of the reference basis function integrals (2.48) and the Gramian matrix ($\mathbf{G}$). Moreover, it is observed that the CPU time per step is also very similar between the two marking strategies. The majority of the computational effort per step resides in the computation of the basis function integrals, $\bar{\boldsymbol{\xi}}$, and in the evaluation of the integration error localized to the sub-cells. In total, the per-level marking strategy is, however, computationally more efficient on account of the significant reduction in number of iterations.

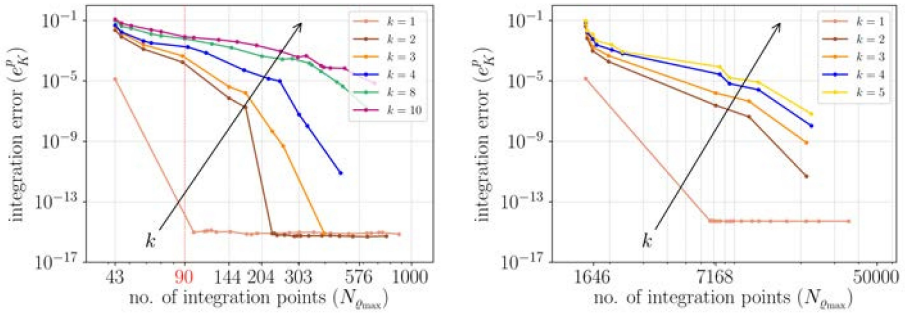
The computation time for the determination of the optimized integration scheme depends primarily on the number of sub-cells and on the dimension of the considered polynomial space. In principle, the computation time of the integral evaluations in the algorithm – *i.e.*, the computation of the basis function integrals, the Gramian, and the localized errors – scales linearly with the number of sub-cells. It should be noted that the number of sub-cells is highly dependent on the bisectioning level and on the number of dimensions; see equation (2.35). For example, for the two-dimensional case reported in Figure 2.11a and Table 2.1 the number of sub-cells is equal to 43, whereas the three-dimensional case in Figure 2.11b has 341 sub-cells. Since the integral evaluations dominate the overall CPU time of the initialization phase and that of an optimization step, these operations become proportionally more expensive. It is important to note, however, that the number of iterations scales with the number of sub-cells in the case of the per sub-cell marking, whereas it scales with the number of octree levels in the case of the per-level marking. This implies that, with a growing number of sub-cells, the per-level marking strategy becomes more favorable from a computational effort point of view.

In terms of computational effort, the dimension of the polynomial space primarily has an effect on the computation time for the basis function integrals and that for the Gramian. In particular the Gramian computation becomes more expensive with an increase in basis size, as the number of terms to be integrated scales quadratically with this size. The linear system solving step involved in the computation of the worst possible function also increases with an increase in system size, but for all considered systems the computational effort involved in the employed direct solver remains negligible compared to the integral evaluations.

Influence of the functional setting

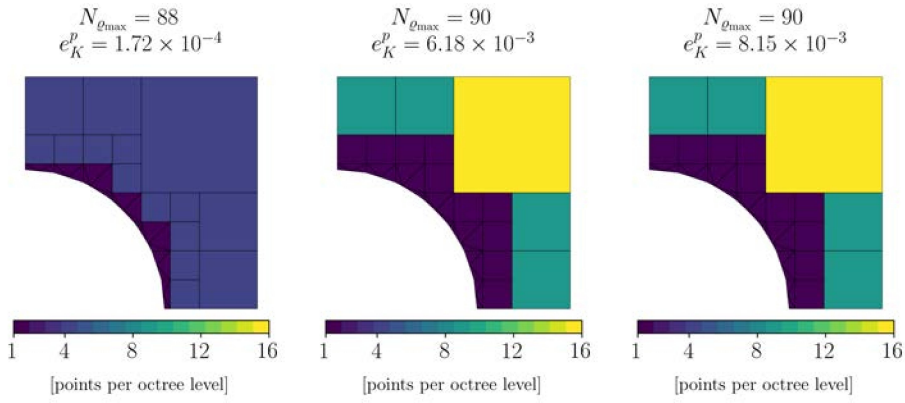
From the functional-setting point of view, the distribution of integration points is influenced by both the approximation order of the polynomial space (2.45) and by the norm (2.52) in which the integrands are considered, encoded by the Gramian $\mathbf{G}$.

In Figure 2.13 we study the influence of the integration error for various orders of the integrands. Note that the lower bound on the error at approximately 10^{-16} corresponds to machine precision errors, and hence these integrands can



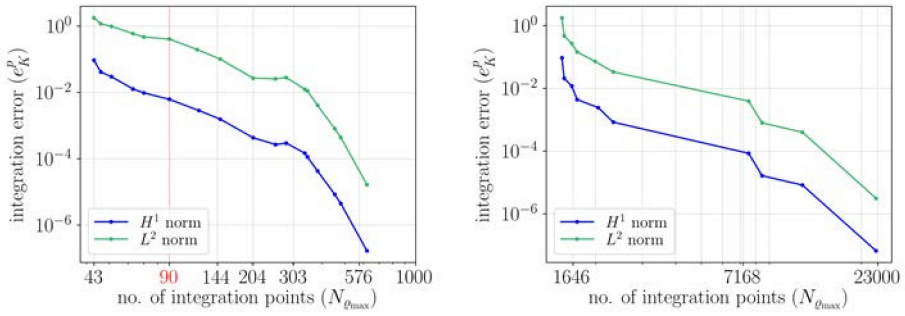
(a) $d = 2$ **(b)** $d = 3$

Figure 2.13: Comparison of the evolution of the integration error versus the number of integration points using the octree level marking strategy for various polynomial functions of order k as an integrand.



(a) $k = 2$ **(b)** $k = 8$ **(c)** $k = 10$

Figure 2.14: Distribution of approximately 90 integration points over the cut-element in two dimensions using the octree level marking strategy for various polynomial orders k.



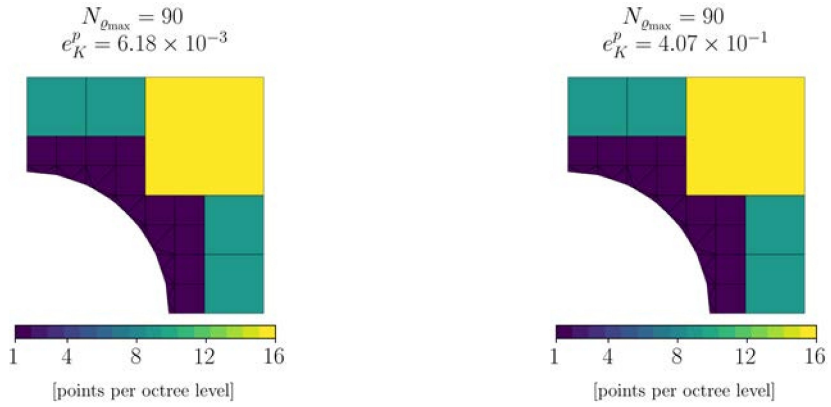
(a) $d = 2$ **(b)** $d = 3$

Figure 2.15: Comparison of the evolution of the integration error versus the number of integration points using the octree level marking strategy and different settings of the integrand space norms.

be interpreted to be exact. As expected, with an increase of the integrand order, a larger number of points is required to attain a certain accuracy. From the relatively low order settings, in particular the case of linear functions, it is observed that the algorithm quickly reaches machine precision, as all sub-cells are marked up until the degree required for exact polynomial integration. Note that in the initial setting, *i.e.*, one point per sub-cell, the $k = 1$ integrands are not resolved exactly, because the cross term $x_1 x_2$ (in two dimensions) is not integrated exactly on the triangulated sub-cells. The same applies to the three-dimensional setting.

In Figure 2.14 the point distributions for the two dimensional case corresponding to approximately 90 points are shown for various orders. It is observed that the distribution of the points over the levels is initially rather uniform for the $k = 2$ case, but becomes more dispersed as the order increases. The reason for this is that the higher-order integrands encompass functions that are more localized to the bigger sub-cells than the lower-order integrands. For a fixed number of integration points, the distribution of integration points over the elements is observed to converge with increasing integrand orders.

In Figure 2.15 the influence of the function space norm (2.52) is studied. Although the normalization does affect the overall magnitude of the error, the difference in norm is observed to have a minor effect on the distribution of the integration points. This is confirmed in Figure 2.16 for the two-dimensional setting, where it is observed that for a total number of 90 points, the distribution using either the L^2-norm or the H^1-norm is virtually identical.



(a) H^1-normalized **(b)** L^2-normalized

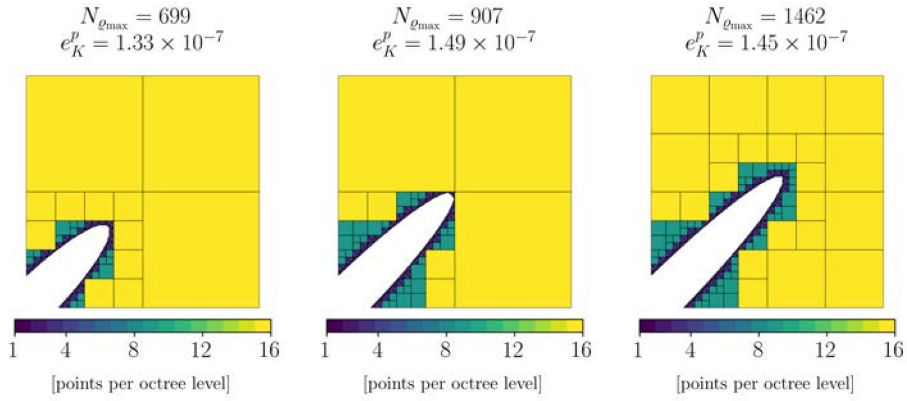
Figure 2.16: Distribution of 90 integration points over a two-dimensional cut-element using the octree level marking strategy and integration errors normalized by the H^1 and L^2 norms.

Influence of the cut-element geometry

The studies presented above were all conducted for cut-elements with spherical exclusions. To study the influence of the geometry of the cut-elements, we here consider the effect of variations in the ellipticity of the exclusions, and in the orientation of the excluded ellipsoids.

In Figure 2.17 we consider a range of exclusions in a two-dimensional cut-element with differing ellipticity. The corresponding error per integration point using the per-level adaptive integration procedure is displayed in Figure 2.18. It is noted that the rate with which the error decreases is similar for all geometries, but that the number of points to attain a particular error is geometry dependent. This behavior is explained by the fact that although the distribution of the orders over the levels is very similar for each of the geometries, the number of sub-cells in each level is geometry dependent. As elaborated in Section 2.3.1 the number of cells on each level scales with the surface fraction. This scaling is reflected in the results in Figure 2.17. For example, the surface-to-volume ratio of Figure 2.17c is approximately two times that of Figure 2.17a. An increase in number of integration points by a factor of approximately 2 is also observed from the offset of the corresponding curves in Figure 2.18.

Figure 2.19 shows the integration error versus the number of integration points for a cut-cell with $r_1 = 0.6$ and $r_2 = 0.1$ for various inclination angles φ. This



(a) $r_1 = 0.5$ **(b)** $r_1 = 0.7$ **(c)** $r_1 = 0.8$

Figure 2.17: Distribution of integration points over cut-elements with various surface to volume ratios ς. The inclination angle for all cases is equal to $\varphi = 45°$.

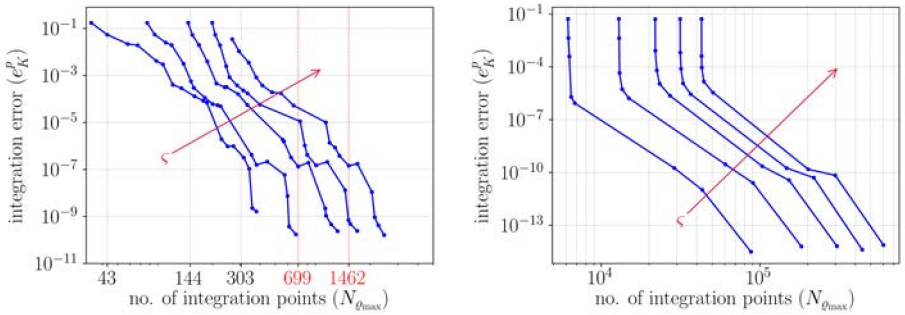
study confirms the scaling relation with the surface to volume ratio as observed for the ellipticity variations considered above.

2.4.4 Manually selected quadrature rules

Although the computational effort involved in the construction of the optimized quadrature rules is in general acceptable when one wants to re-use a quadrature rule multiple times, some computational effort is involved in this construction. In addition, one has to set up a suitable code to determine the optimal distributions for arbitrary cut-cells. Considering this, one may not be interested in obtaining the optimized distributions of the points, but may instead want a simple rule of thumb to select the quadrature on a cut-cell; see, *e.g.*, Refs. [68, 73].

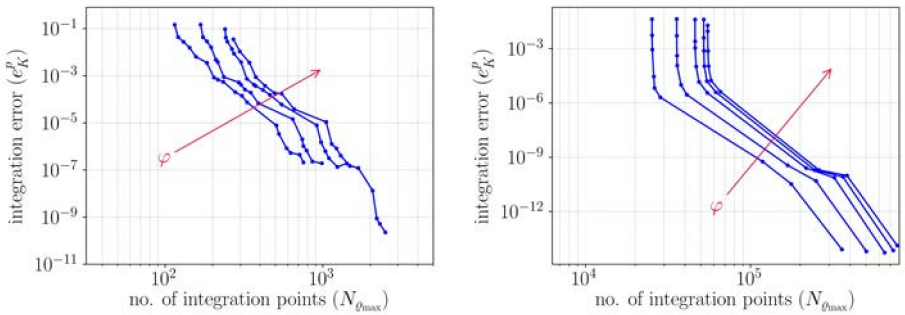
The per-level selection of the integration order makes it practical to manually select integration rules that outperform full order integration on all octree levels. We here consider two manual selection strategies based on Gauss integration of the sub-cells:

A. *Minimal degree lowering:* In this strategy we set the order of the Gauss scheme on the level $\varrho = 1$ sub-cells to $k_{\max}$. We then reduce the number of Gauss points per direction by one for each level, which implies that for $\varrho \leq \varrho_{\max}$ the Gauss degree is decreased by two between two octree levels. The integration order of the tessellated sub-cells at level $\varrho_{\max} + 1$ is set to be two orders lower than that at level $\varrho_{\max}$. Once a Gauss degree of zero is



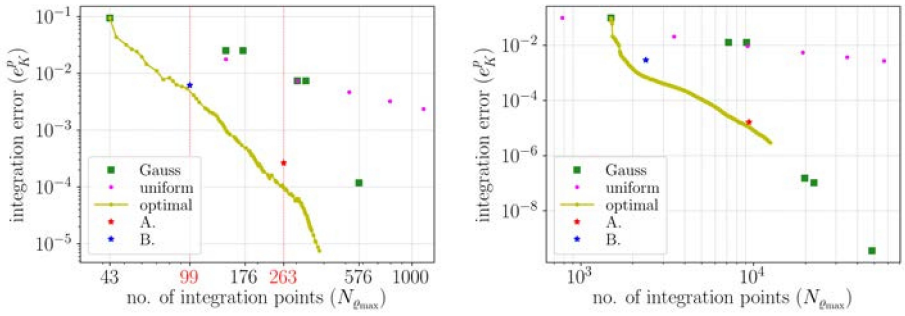
(a) $d = 2$ **(b)** $d = 3$

Figure 2.18: Comparison of the evolution of the integration error versus the number of integration points using the octree level marking strategy for various surface to volume ratios ς.



(a) $d = 2$ **(b)** $d = 3$

Figure 2.19: Comparison of the evolution of the integration error versus the number of integration points using the octree level marking strategy for various surface to volume ratios ς related to an exclusion with $r_1 = 0.6$, $r_2 = 0.1$ and varying inclination angle φ.



(a) $d = 2$ **(b)** $d = 3$

Figure 2.20: Comparison of the integration error and the number of integration points using various thumb rules.

reached this value is maintained for the underlying levels. For example, for the case of $k = 8$ and $\varrho_{\max} = 3$, the Gauss orders over the levels are set to 8 for $\varrho = 1$, 6 for $\varrho = 2$, 4 for the $\varrho = \varrho_{\max}$, and 2 for the $\varrho = \varrho_{\max} + 1$ (see Figure 2.21a). That is, for an octree-depth of $\varrho_{\max} = 3$, the list of levels $[1, 2, 3, 4]$ contains $[8, 6, 4, 2]$ as the corresponding list of the Gauss orders.

B. *Uniform degree lowering:* This strategy also starts with selecting the $\varrho = 1$ integration degree as $k_{\max}$, but then decreases the degree between two levels in such a way that the single point (degree is zero) is reached at the levels $\varrho_{\max}$, and $\varrho_{\max}+1$. For the case considered above, the orders are set to 8 for $\varrho = 1$, 4 for $\varrho = 2$, and 0 for $\varrho = \varrho_{\max}$ and $\varrho = \varrho_{\max} + 1$ (see Figure 2.21c). That is, for an octree-depth of $\varrho_{\max} = 3$, the list of levels $[1, 2, 3, 4]$ contains $[8, 4, 0, 0]$ as the corresponding list of the Gauss orders.

Evidently, alternative quadrature rules can be formulated, but a detailed study of such rules is beyond the scope of the current work.

The adaptive algorithm developed in this work allows us to asses the suitability of the rules of thumb defined above. As a preliminary study on the effectiveness of these rules of thumb, we again consider the case of a two-dimensional circular exclusion with $\varrho_{\max} = 3$ and a polynomial function of order $k = 8$. In Figure 2.20a the manually selected schemes are compared to the full order integration schemes and to the per sub-cell optimized integration schemes. This plot conveys that although the manually selected schemes are, as expected, outperformed by the optimized schemes, they generally do provide a dramatic improvement in accuracy per integration point for a fixed number of points. This observed behavior is explained by consideration of the distributions of orders over the sub-cells as

shown in Figure 2.21, from which it is observed that the per-level decrease of the order of both rules of thumb indeed qualitatively matches the results of the optimization procedure. Evidently, a difference between the two rules of thumb is that the minimal degree lowering strategy (A.) results in a larger number of points and a lower error compared to the uniform degree lowering (B.). Both schemes do, however, reasonably well resemble the optimized distribution patterns. Similar observations for the three-dimensional setting are obtained, as illustrated in Figure 2.20b. The displayed three-dimensional results are based on $\varrho_{\max} = 3$ and $k = 5$.

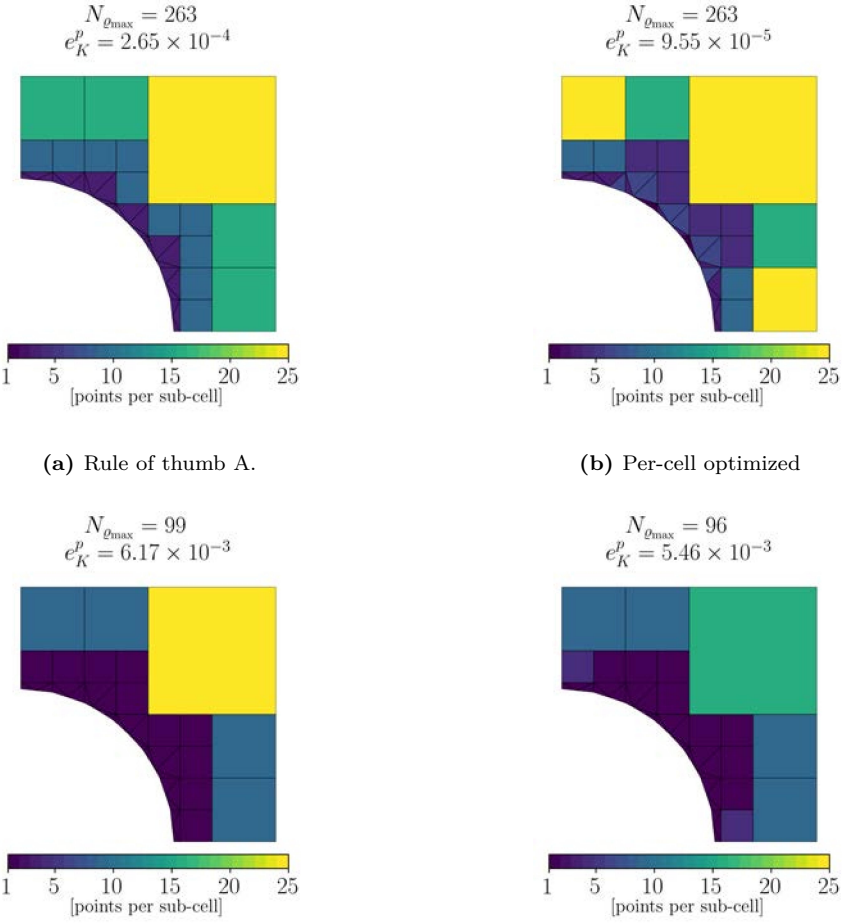
2.5 Application to immersed IGA

In this section we assess the suitability of the optimized integration schemes in the context of immersogeometric analysis. We consider an elasticity problem in two and three dimensions, which we solve using globally defined values of the integration degrees for each octree level. The integration degrees are obtained by application of the optimization procedure to the entire mesh, starting with a single integration point in all sub-cells. The integration errors computed per octree level are summed over all elements, after which the level with the highest global error is refined in terms of integration degree. It is noted that this global optimization strategy ignores operator-dependent variations between the elements, as discussed in Remark 2.2.

We consider a linear elasticity problem on a computational domain that is immersed into a background mesh of size L^d; see Figure 2.22 for $d = 2$. Displacements are prescribed on the exterior boundary, while the interior boundary is traction free. In the absence of inertia effects and body forces, the boundary value problem reads as

$$\begin{cases} \text{Find } \boldsymbol{u} \text{ such that:} & \\ \mathbf{div}(\,\boldsymbol{\sigma}(\boldsymbol{u})\,) = \mathbf{0} & \text{in } \Omega \\ \boldsymbol{u} = \mathbf{0} & \text{on } \partial\mathcal{A}_0 \\ \boldsymbol{u} = \bar{u}\boldsymbol{n} & \text{on } \partial\mathcal{A}_{\bar{u}} \\ \boldsymbol{u} \cdot \boldsymbol{n} = 0 & \text{on } \partial\mathcal{A} \setminus (\mathcal{A}_0 \cup \mathcal{A}_{\bar{u}}) \\ [\boldsymbol{I} - \boldsymbol{n} \otimes \boldsymbol{n}]\,\boldsymbol{\sigma}\boldsymbol{n} = \mathbf{0} & \text{on } \partial\mathcal{A} \setminus (\mathcal{A}_0 \cup \mathcal{A}_{\bar{u}}) \\ \boldsymbol{\sigma}\boldsymbol{n} = \mathbf{0} & \text{on } \partial\Omega \setminus \partial\mathcal{A} \end{cases} \tag{2.67}$$

with stress tensor $\boldsymbol{\sigma}(\boldsymbol{u}) = \lambda \mathbf{div}(\boldsymbol{u})\boldsymbol{I} + 2\mu\nabla^s\boldsymbol{u}$, where ∇^s denotes the symmetric gradient operator. Throughout this section, the Lamé parameters are set to $\lambda = \frac{1}{2}$ and $\mu = \frac{1}{2}$.



(a) Rule of thumb A.

(b) Per-cell optimized

(c) Rule of thumb B.

(d) Per-cell optimized

Figure 2.21: Distribution of integration points over the cut-element in two dimensions using the manual selection of quadrature rules per-level for a polynomial order $k = 8$.

The Dirichlet condition on the external boundary is applied strongly, *i.e.*, by constraining the degrees of freedom related to the boundary displacements. The Galerkin problem corresponding to (2.67) then follows as

$$\begin{cases} \text{Find } \boldsymbol{u}^h \in W^h(\Omega) \text{ such that for all } \boldsymbol{v}^h \in W_0^h(\Omega): \\ \displaystyle\int_\Omega \nabla^s \boldsymbol{v}^h : \boldsymbol{\sigma}(\boldsymbol{u}^h)\,\mathrm{d}V = 0 \end{cases} \tag{2.68}$$

with the discrete trial space being a subset of $H^1(\Omega)$ and satisfying the Dirichlet boundary conditions. The discrete test space is identical to the trial space, modulo inhomogeneous boundary conditions. All results in this section pertain to full regularity, C^{p-1}, B-splines of degree $p = 2$. The polynomial order used to compute the optimized integration schemes is taken equal to $k = 4$.

As a quantity of interest we consider the constrained modulus

$$M^\star = \frac{\sigma_{11}}{\varepsilon_{11}} = \frac{L}{A}\frac{F}{\bar{u}}, \tag{2.69}$$

where F is the resultant reaction force in normal direction along the displaced boundary, and where A is the area (length in two dimensions) at which this resultant force acts. We normalize the computed constrained modulus by that of a homogeneous square with the same material properties as the considered model, *i.e.*, by $M = \lambda + 2\mu$.

For both the two and three dimensional test cases we consider three levels of bisectioning, *i.e.*, $\varrho_{\max} = 3$. We represent the selection of the per level integration orders as a list of length $\varrho_{\max} + 2$, where the first $\varrho_{\max} + 1$ entries refer to the bisectioning levels 0 to $\varrho_{\max}$. Note that the level $\varrho = 0$ in fact pertains to cells that are not intersected by the boundary. The final entry in the list of orders pertains to the integration order on the tessellated sub-cells at level $\varrho_{\max} + 1$.

2.5.1 Two-dimensional test case

We consider the two-dimensional test case introduced in Ref. [82], which is illustrated in Figure 2.22. The size of the bounding square is set to $L = 1$, and the radii of the exclusions are takes as $R_1 = 0.3$ and $R_2 = 0.2$. The right boundary is displaced by $\bar{u} = \frac{1}{2}$. Figure 2.23 displays the solution to the elasticity problem (2.68) in terms of the displacement magnitude and the shear stress. The displayed result is based on a 10×10 ambient domain mesh with uniformly distributed fourth order Gauss integration scheme.

Figure 2.24a displays the computed constrained modulus for different steps in the integration optimization procedure. It is observed that by selecting a second order Gauss scheme on the untrimmed elements, and a single point in each

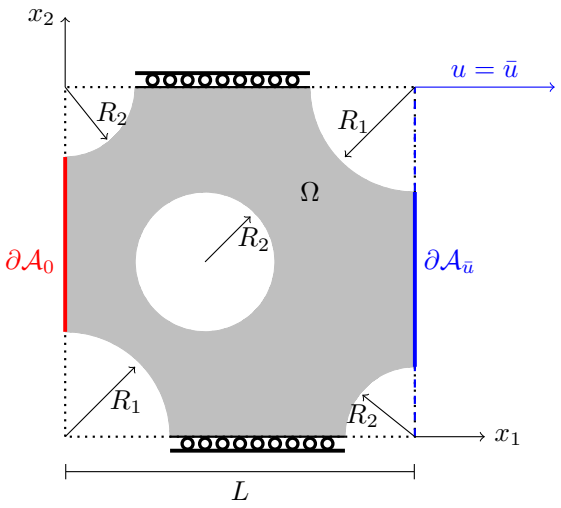


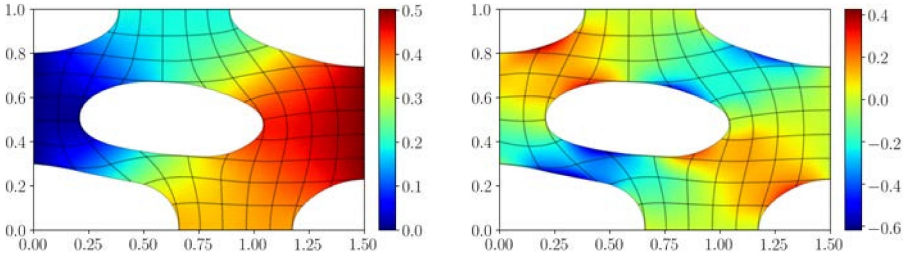
Figure 2.22: Schematic representation of the considered domain and the mechanical loading conditions.

sub-cell of the trimmed elements, yields a modulus of approximately $0.3484M$, which is already within 0.1% of the fully resolved integral result. This computation, however, uses a total of only 1208 points, in comparison to the 6810 points used for the full integral computation. Upon application of the integration optimization procedure the constrained modulus evidently converges to that using full integration. A virtually identical result as the full integration result is obtained with 3654 points. For this result fourth order Gauss quadrature is used on the untrimmed cells and on the first bisection level, and second order Gauss quadrature on all lower levels.

2.5.2 Three-dimensional test case

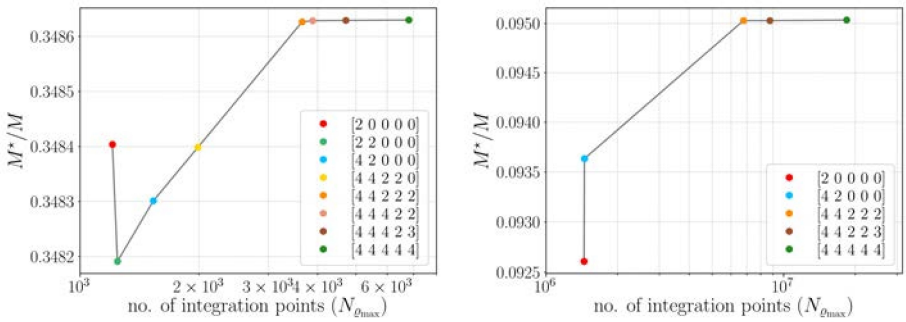
We consider the elastic deformation of a sintered glass specimen, which is extracted from μCT-scan data as discussed in Ref. [15][1]. The size of the bounding box is set to $L = 1.5$, and the right boundary is displaced by $\bar{u} = \frac{1}{2}$. Figure 2.25 displays the solution to the elasticity problem (2.68) in terms of the displacement magnitude and the σ_{xy} shear stress. The displayed result is based on a $10\times10\times10$ ambient domain mesh and uniformly distributed fourth order Gauss integration scheme. As reported in Ref. [58], for coarse meshes the basis functions artificially

[1]The scan data used for this simulation can be downloaded from: www.gitlab.tue.nl/20175645/sintered_scan_data.



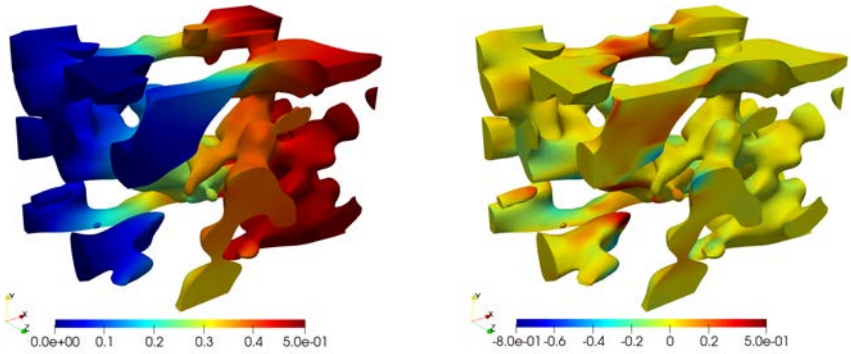
(a) Displacement magnitude **(b)** Shear stress σ_{12}

Figure 2.23: Solution of the two-dimensional linear elasticity problem, displayed in the deformed configuration.



(a) $d = 2$ **(b)** $d = 3$

Figure 2.24: Comparison of the constrained modulus for different quadrature rules that have evolved from optimization algorithm for both the two-dimensional and three-dimensional linear elasticity problem.



(a) Displacement magnitude **(b)** Shear stress σ_{xy}

Figure 2.25: Solution of the three-dimensional linear elasticity problem, displayed in the deformed configuration.

connect disjoint parts of the geometry. This effect is also observed in the shear stress displayed in Figure 2.25b. For the purpose of illustrating the performance of the integration optimization procedure considered in this work, this effect is, however, not prohibitive.

Figure 2.24b displays the constrained modulus versus the number of integration points. By using second order quadrature on the untrimmed cells and one point per sub-cell in the trimmed elements, provides a modulus of approximately $0.0925M$, which is within 2.5% of the fully resolved integral result. However, this result is obtained at the expense of 1.4×10^6 integration points. This is a factor thirteen lower than the 18.4×10^6 points used to compute the fully resolved integral result for a reference. A virtually identical result to the reference result is obtained at 6.8×10^6 points, which is still approximately a factor of three lower than the full integration scheme.

2.6 Concluding remarks

We have developed an algorithm to construct quadrature rules for cut-elements in which the integration points are distributed over the (cut) element in such a way that the integration error is minimized. Strang's first lemma – which provides an error estimate for the approximate solution including integration errors – provides a theoretical underpinning of the developed integration procedure. In the setting of this approximation theory, it is found that the integration error

is bound by the supremum of integration errors over the considered test space. When the integration error is localized to the (cut) elements, this supremum translates to the consideration of a worst possible function to be integrated over a particular element. By discretizing the space of functions over an element by means of polynomials, the worst possible function and its associated integration error become computable.

The ability to evaluate the integration error forms the basis of the proposed integration optimization procedure for octree subdivision. The pivotal idea behind the proposed algorithm is to gradually increase the number of integration points by adding integration points to the sub-cells for which the error reduction per added integration point is highest. As the number of sub-cells increases dramatically with an increase in number of dimensions and with the octree depth – as illustrated by the derived scaling relation for the number of sub-cells – such an optimization procedure has the potential to significantly reduce the computational effort involved in the integration procedures for immersed methods. The presented numerical simulations demonstrate that, for a given number of integration points, the integration error resulting from the optimization algorithm is, in general, significantly lower than that of the integration scheme considering the same integration orders on all sub-cells. Conversely, when fixing the error, the number of integration points required for the optimized quadrature rule is significantly lower than that of the equal order schemes. The considered simulations demonstrate that the developed optimization algorithm efficiently distributes integration points over cut-elements for a wide range of cut-cell configurations.

The developed algorithm provides insight into the way in which integration points should be distributed over the cut-elements in order to minimize integration errors. Based on these insights, it is evident that integration rules for which the number of points per sub-cell is decreased with increasing octree depth form a good approximation to the optimized distribution of integration points. The integration point distribution algorithm presented in Ref. [68] therefore can also be expected to yield quadrature rules that are a good approximation to the rules that follow from the theoretical framework considered in this contribution.

Although the costs involved in the quadrature optimization algorithm are limited on account of the fact that the employed polynomial basis should only be integrated once using the expensive full order integration scheme, the determined optimized integration rules are particularly attractive when they can be re-used for multiple simulations. This is, for example, the case in time-dependent or non-linear problems. The attained reduction in number of integration points is also highly advantageous when data is to be stored in integration points, such as is the case for example with history data in many non-linear constitutive models.

The developed algorithm is highly generic in the sense that one can use the algorithm to optimize other integration schemes. For example, nested Gauss

schemes can be attractive on account of the fact that the points (or locations) are nested in these schemes, which facilitates the re-using of configurations between integration schemes.

When employing the quadrature optimization algorithm one must choose various parameters, most importantly the order of the polynomial integrand to be considered and the norm used for the integrand normalization. In particular the order of polynomials is important with respect to the obtained distribution of integration points, in the sense that selecting this order too low will lead to sub-optimal integration results. In general it is possible to determine the order of the integrands based on the problem under consideration and the finite element basis functions being considered.

In this contribution we have restricted ourselves to optimizing the distribution of integration points over the volumes of the cut-elements. The underlying theoretical framework is, however, more general in the sense that other parameters controlling the integration error can be incorporated as well. In the context of the considered octree integration procedure it would be natural to include the octree depth in the optimization procedure. Since both the error reduction associated with the increase in octree depth and the associated computational cost are quantifiable, including this depth as an additional parameter in the optimization algorithm is anticipated to be feasible. Inclusion of this parameter is, however, considered beyond the scope of the current work. The same holds for the incorporation of errors associated with the boundary integrals, in particular the Nitsche terms typically considered in finite cell simulations.

The focus in this work has been on the optimization of the integration error contribution in Strang's first lemma, not taking into account the approximation error. It is noted, however, that the need to optimize the integration error depends on the approximation error. If the approximation error is small compared to the overall error, then there is a need to optimize the integration error. However, if the approximation error is the dominating contribution to the overall error, then there is not a strong need to optimize the integration error. Evaluation of the balance between the approximation error and the integration error is an important topic of further study.

Chapter 3

Topology-preserving scan-based immersed isogeometric analysis

To exploit the advantageous properties of isogeometric analysis (IGA) in a scan-based setting, it is important to extract a smooth geometric domain from the scan data (*e.g.*, voxel data). IGA-suitable domains can be constructed by convoluting the grayscale data using B-splines. A negative side-effect of this convolution technique is, however, that it can induce topological changes in the process of smoothing when features with a size similar to that of the voxels are encountered. This chapter presents an enhanced B-spline-based segmentation procedure using a refinement strategy based on truncated hierarchical (TH)B-splines. A Fourier analysis is presented to explain the effectiveness of local grayscale function refinement in repairing topological anomalies. A moving-window-based topological anomaly detection algorithm is proposed to identify regions in which the grayscale function refinements must be performed. The criterion to identify topological anomalies is based on the Euler characteristic, giving it the capability to distinguish between topological and shape changes. The proposed topology-preserving THB-spline image segmentation strategy is studied using a range of test cases. These tests pertain to both the segmentation procedure itself, and its application in an immersed IGA setting.

This chapter is reproduced from [87]: S.C. Divi, C.V. Verhoosel, F. Auricchio, A. Reali, and E.H. van Brummelen. Topology-preserving scan-based immersed isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 2022. The (co-)promotors confirm that S.C. Divi is the primary author of this publication, i.e., she was responsible primarily for the planning, execution and preparation of the work.

3.1 Introduction

Computational analyses based on volumetric scan data are of interest in many fields of research, such as biomechanics, geomechanics, material science, microstructural analysis, and many more. Scan-based simulations are inherently three-dimensional and, frequently, the computational domains are complex, both in terms of geometry and in terms of topology. In addition, the data sets obtained from, *e.g.*, tomography or photogrammetry techniques are large in size and represented in data formats which are not directly suitable for analysis (*e.g.*, DICOM[1], NIfTI[2]). For these reasons, performing high-fidelity simulations at practical computational costs is still very challenging.

Over the past decade, the use of Isogeometric Analysis (IGA) [110] to perform accurate scan-based simulations at acceptable computational costs has been explored. Originally, IGA was proposed as an analysis paradigm to better integrate analysis and design by employing the spline basis functions from Computer Aided Design (CAD) directly for the analysis, without intermediate geometry clean-up and meshing operations [16], as required in traditional finite element analyses. The advantageous properties of spline basis functions, in particular their higher-order regularity, have made IGA also attractive for simulations where the geometry is not represented by a CAD model [29, 30, 33, 111]. In the context of scan-based isogeometric analysis, the usage of standard discretization techniques such as the voxel method [112] or (unstructured) conforming mesh approaches (*e.g.*, generated using a marching cube algorithm [113, 114]) is not possible, as this deteriorates the favorable properties of IGA. Therefore, various enhanced isogeometric techniques for scan-based analysis have been developed. These can be summarized as follows:

- **Template-fitting techniques** construct a spline-based template geometry (typically consisting of multiple patches) that captures the essential features of the scanned object, and subsequently apply fitting procedures to reposition the control points of the template to match the scan data. An advantage of such techniques is that an explicit parametrization of the scan object is retrieved, which is beneficial from an analysis point of view. The downside of template fitting techniques is that they are not useful in cases where the topology of the object is not known *a priori*. For cases where the topology of the object is known but complex, template fitting techniques are typically laborsome. Template fitting techniques using IGA have, for example, been applied to fluid-structure interaction analyses of arterial blood flow [24, 25], patient-specific blood-flow analyses in arteries [33, 40] (see [46] for a review), the mechanical behavior of a femur (consisting of both hard

[1] Digital Imaging and Communications in Medicine
[2] Neuroimaging Informatics Technology Initiative

outer cortical bone and inner trabecular bone) [42], the mechanical behavior of a carotid artery stent [22], damage in composite laminates [115], the human heart [41], and fluid-structure interactions of a heart valve [31].

- **Immersed methods** construct a structured mesh for the entire scan domain, typically a box, and then represent the scanned object by an inside/outside relation to indicate whether a specific point in the box is part of the object. In the context of scan-based analysis, the inside/outside relation can directly be retrieved from the scan data (intensity) in the form of a level set function [58], but the immersed concept can also be applied to obtain volumetric representations of objects identified by boundary representations (*e.g.*, BREP [14, 52, 53, 59, 89, 116], STL[1] [49, 117–120]) and to represent trimming operations in CAD models (*e.g.*, [37, 121, 122]). The advantage of immersed techniques is their versatility with respect to the geometry and topology of the scanned objects, in the sense that the analysis procedure is not essentially affected by increasing the complexity of the objects. From an analysis perspective, the immersed approach poses additional challenges compared to mesh-fitting analyses. Most notably, numerical integration of trimmed elements is challenging from an efficiency point of view, application of boundary conditions can be non-standard, and the system of equations is generally ill-conditioned without dedicated treatment. Immersed scan-based IGA has, for example, been applied for the analysis of trabecular bone [58–60], coated metal foams [61], porous media [15], metal castings [62] and in additive manufacturing [63].

The choice for either a template-fitting technique or an immersed technique is to a large extent dictated by the topological complexity of the scanned object. If a template with a reasonably low number of control points can be constructed for the object of interest, template-fitting is generally favorable. If the creation of a template is impractical, immersed methods are preferred. It should be noted that it can be favorable to combine the two techniques for particular scan-based analyses, to exploit the advantages of both of them. A noteworthy example in this regard is the heart-valve problem by Kamensky *et al.* [56, 123, 124], where the heart chambers are modeled through a template-fitting approach, and the moving valves through an immersed approach.

In this work we build on the immersed scan-based analysis framework originally proposed in Ref. [58]. In recent years, significant progress has been made to tackle the above-mentioned challenges associated with immersed methods. A myriad of advanced integration techniques has been developed to reduce the computational burden associated with the integration of trimmed elements; see, *e.g.*, Refs. [71, 86, 98, 125]. Nitsche's method [77] has been demonstrated to be

[1] Standard Triangle Language or Standard Tessellation Language

a reliable technique to impose essential boundary conditions along immersed boundaries, *e.g.*, [50, 53, 78, 126], and various techniques have been developed to construct a parametrization of immersed boundaries to impose boundary conditions [51,76,126,127]. With respect to the stability and conditioning, fundamental understanding was obtained in Refs. [53, 80], and various types of remedies have been demonstrated to be effective, see, *e.g.*, [15, 30, 51, 62, 79, 82, 84, 85, 127, 128]. The various advancements of immersed techniques have made it into a versatile technique for scan-based analyses, being able to model problems in different physical domains (solid mechanics, fluid dynamics), considering high-performance computing problems [15,49,54,58,60,62,65,125], and being applicable in the context of highly nonlinear problems [55, 129–131].

Although immersed isogeometric analysis is very flexible with respect to capturing topologically complex volumetric objects, topological anomalies associated with the image segmentation procedure can occur [132–136]. This is specifically the case when the resolution of the scan data is only nearly sufficient to capture the smallest features in the scanned object. Image smoothing operations, in the case of the procedure of Ref. [58] associated with the order of the spline level set construction, can trigger undesirable topology alterations (*e.g.*, closure of a channel, (dis)connection of structures). The occurrence of topological anomalies associated with smoothing operations is well understood in the field of image segmentation (*e.g.*, [45, 137–139]). Various enhanced image-segmentation techniques have been proposed in order to ameliorate such problems, like homology-based preservation techniques [140–146], topology-derivative-based techniques [147–151], and adaptive refinement techniques [45, 152–157].

In this contribution we propose an enhancement of the immersed isogeometric analysis framework of Ref. [58] that rigorously avoids the occurrence of topological anomalies associated with the B-spline-based image segmentation procedure. The filtering analysis in Ref. [58] is extended to include the effect of the level set mesh size. Based on this analysis, it is proposed to employ truncated hierarchical B-splines (THB-splines) [158–160] to discretize the grayscale intensity of the scan data. Inspired by topology characteristics (*e.g.*, Betti numbers, Euler characteristic) to detect local topology anomalies [143,144,161–163], this local refinement capability for the level set representation is complemented with an Euler characteristic evaluation and moving-window technique to detect local topology anomalies. The resulting topology-preserving immersed isogeometric analysis is demonstrated using prototypical test cases in two and three dimensions, considering both the image-segmentation problem and the complete (stabilized) immersed isogeometric analysis problem [15]. A scan-based analysis on a representative data set is considered to demonstrate the practical applicability of the proposed technique. In this contribution, we restrict ourselves to user-specified locally-refined meshes, keeping in mind the extension to an adaptive analysis framework

as a possible further development.

This paper is organized as follows. In Section 3.2 we first extend the filtering analysis of Ref. [58]. Based on this analysis, in Section 3.3 we introduce the topology-preserving extension of the spline-based image segmentation technique. This technique comprises the moving-window technique to detect topological anomalies (Section 3.3.1) and a THB-spline-based discretization of the level set function (Section 3.3.2). The adopted immersed isogeometric analysis framework, including stabilization techniques, is introduced in Section 3.4. Numerical examples are then presented in Section 3.5 to test the proposed technique. Finally, conclusions and recommendations are drawn in Section 3.6.

3.2 Spline-based image segmentation

In this section we review the spline-based image segmentation procedure of Ref. [58]. The occurrence of topological anomalies on relatively coarse voxel grids is illustrated and explained using a Fourier analysis. Based on this analysis, a solution strategy to avoid topological anomalies is proposed.

3.2.1 B-spline level set construction

Consider an n_d-dimensional image domain, $\Omega_{\text{img}} = [0, L_1] \times \ldots \times [0, L_{n_d}]$, which is partitioned by a set of voxels, as illustrated in Figure 3.1. We denote the voxel mesh by

$$\mathcal{V}_{\text{vox}} = \{\Omega_{\text{vox}} \subset \Omega_{\text{img}} \mid \exists \mathbf{i} \in \mathbb{Z}^{n_d}_{\geq 0},\ \text{s.t. } \Omega_{\text{vox}} = T_{\mathbf{i}} \circ [0,1]^{n_d}\}, \tag{3.1}$$

where the transformation $T_{\mathbf{i}} : [0,1]^{n_d} \to \Omega_{\text{vox}}$ is defined as

$$T_{\mathbf{i}}(\boldsymbol{\xi}) = \text{diag}(\boldsymbol{\Delta})(\boldsymbol{\xi} + \mathbf{i}), \tag{3.2}$$

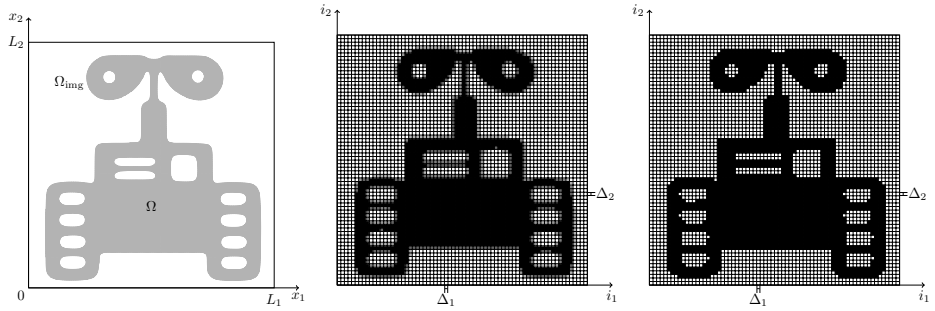
with $\boldsymbol{\Delta}$ the voxel size in each direction and $\mathbf{i} = (i_1, \ldots, i_{n_d})$ the n_d-dimensional voxel index. We denote the number of voxels in the voxel image by $m_{\text{vox}} = \#\mathcal{V}_{\text{vox}}$. The grayscale intensity function, illustrated in Figure 3.1b, is then defined as

$$g : \mathcal{V}_{\text{vox}} \to \mathscr{G}, \tag{3.3}$$

with $\mathscr{G}$ the range of the grayscale data (*e.g.*, from 0 to 255 for 8 bit unsigned integers). An approximation of the object Ω can be obtained by thresholding the gray scale data,

$$\Omega \approx \{\boldsymbol{x} \in \Omega_{\text{img}} | g(\boldsymbol{x}) > g_{\text{crit}}\} \subset \Omega_{\text{img}}, \tag{3.4}$$

where g_{crit} is the threshold value (see Figure 3.1c). As a consequence of the piecewise definition of the grayscale data in equation (3.3), the boundary of the



(a) **(b)** **(c)**

Figure 3.1: Representative two-dimensional geometry to illustrate the spline-based segmentation procedure. (a) The assumed geometry, which in a real-life scan-based analysis setting is evidently not available. (b) The grayscale data on a 75×75 voxel grid, which we derived from the exact geometry by computing the volume fraction of each voxel. In the scan-based analysis setting this voxel image is obtained directly from the scan data. (c) The segmented geometry for a grayscale threshold corresponding to 50% of the volume fraction.

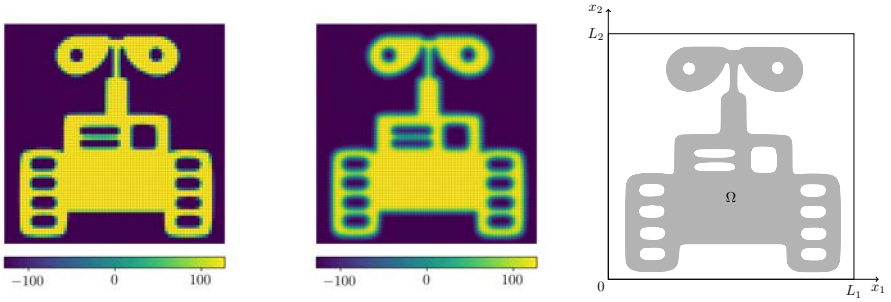
segmented object is non-smooth when the grayscale data is segmented directly. In the context of analysis, the non-smoothness of the boundary can be problematic, as irregularities in the surface may lead to non-physical singularities in the solution to the problem.

The B-spline segmentation procedure in Ref. [58] – the behavior of which in the case of linear basis functions closely corresponds with that of marching volume algorithms [113] – enables the construction of a smooth boundary approximation based on voxel data. The key idea of this B-spline segmentation technique is to smoothen the grayscale function (3.3) by convoluting it using a B-spline basis of size n, $\{N_{i,k}(\mathcal{V}^h)\}_{i=1}^n$, defined over a regular mesh, $\mathcal{V}^h$, with fixed element size, h_d, per direction. Note that this mesh size can be different from the voxel size. The B-spline basis can be constructed using the Cox-de Boor algorithm [164]. We consider full-regularity (C^{k-1}-continuous) B-splines of order k, with the order assumed to be constant and isotropic. By convoluting the grayscale function (3.3), a smooth level set approximation is obtained as

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} N_{i,k}(\boldsymbol{x}) a_i, \qquad a_i = \frac{\int_{\Omega_{\text{img}}} N_{i,k}(\boldsymbol{x}) g(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}}{\int_{\Omega_{\text{img}}} N_{i,k}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}}, \tag{3.5}$$

where the coefficients $\{a_i\}_{i=1}^n$ are the control point level set values.

The B-spline level set function corresponding to the voxel data in Figure 3.1b is illustrated in Figure 3.2 for the case where the regular mesh, $\mathcal{V}^h$, coincides



(a) Grayscale data **(b)** B-spline level set function **(c)** Segmented geometry

Figure 3.2: Illustration of the B-spline based segmentation procedure on a 70×70 voxel grid using second-order B-splines constructed on a regular mesh that coincides with the voxels. (a) Grayscale data, $g(\boldsymbol{x})$, of Figure 3.1b shown on the 8 bit signed integer range. (b) Level set function, $f(\boldsymbol{x})$, computed by equation (3.5). (c) Segmented domain extracted using the midpoint tessellation procedure outlined in Refs. [58, 86].

with the voxel grid (*i.e.*, $\Delta_d = h_d$ for $d = 1, \ldots, n_d$) and second order ($k = 2$) B-splines. As can be seen, the object retrieved from the convoluted level set function more closely resembles the original geometry in Figure 3.1a compared to the voxel segmentation in Figure 3.1c. Also, the boundaries of the domain are smooth as a consequence of the higher-order continuity of the B-spline basis (see Figure 3.2c). The segmented geometry in Figure 3.2c is constructed using the midpoint tessellation procedure proposed in Refs. [58, 86], which constructs a partitioning of the elements that intersect the domain boundary and results in an accurate parametrization of the interior volume. The reader is referred to Ref. [58] for a detailed discussion of the properties of the employed B-spline segmentation technique, and, additionally, to Ref. [86] for details regarding the midpoint tessellation procedure.

3.2.2 The occurrence of topological anomalies

The spline-based segmentation technique reviewed above has been demonstrated to yield computational domains that are very well suited for isogeometric analysis (see, *e.g.*, [15, 45, 82]). However, although the smoothing characteristic of the technique is frequently beneficial, it may lead to the occurrence of topological anomalies when the features of the object to be described are not significantly larger in size than the voxels (*i.e.*, the Nyquist criterion is not satisfied [58]).

For the object in Figure 3.1a this occurs when a voxel grid of 35×35 is considered, as illustrated in Figure 3.3 for a second-order ($k = 2$) B-spline basis defined on the 35×35 voxels mesh. As can be seen, small and slender features, such

as the regions highlighted in green in Figure 3.3c, are detectable in the original grayscale data, albeit with a very coarse representation. The corresponding level set function is smoother, but introduces topological anomalies in the form of the disappearance of some of the small and slender features. When we consider the same voxel data, but now define the B-spline basis for the convolution of the level set function on a twice as fine mesh (see Figure 3.3d), *i.e.*, $h_d = \Delta_d/2$ for $d = 1, \ldots, n_d$, a topologically correct segmented domain is again obtained, but still with smoothed boundaries (see Figure 3.3e) compared to the direct segmentation.

To elucidate the smoothing behavior of the B-spline segmentation technique, we generalize the filtering analysis for a univariate B-spline ($n_d = 1$) presented in Ref. [58] to the case of non-coinciding voxel and B-spline grid sizes, *i.e.*, $h \neq \Delta$. Note that, in the univariate setting considered here, we drop the index for the geometric direction to simplify our notation. The goal of our analysis is to provide insight into the filtering properties by considering the level set approximation in the frequency domain. Additionally, we will obtain an analytical expression for the smoothed level set function in the spatial domain with a parametrization that provides insight into the smoothing properties of the level set construction.

We commence our analysis with rewriting the operation (3.5) as an integral transform

$$f(x) = \int_{\Omega_{\text{img}}} \mathcal{K}(x,y) g(y) \, \mathrm{d}y, \qquad \mathcal{K}(x,y) = \sum_{i=1}^{n} \frac{N_{i,k}(x) N_{i,k}(y)}{V_i}, \tag{3.6}$$

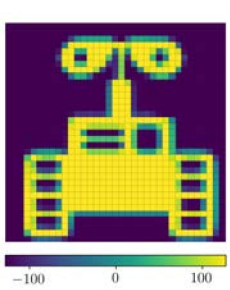
where $\mathcal{K}(x,y)$ is the kernel of the transformation and where $V_i = \int_{\Omega_{\text{img}}} N_{i,k}(x) \, \mathrm{d}x$ is the integral of the basis function $N_{i,k}$. Following the derivation in Ref. [58] – in which the essential step is to approximate the B-spline basis functions by rescaled Gaussians [165] – the integration kernel (3.6) can be approximated by

$$\mathcal{K}(x,y) \approx \kappa(x-y) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-y)^2}{2\sigma^2}\right), \tag{3.7}$$

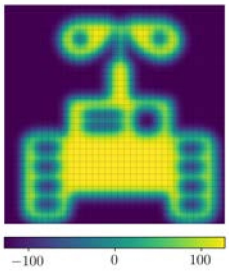
where the width of the smoothing kernel is given by

$$\sigma = h\sqrt{\frac{k+1}{6}}. \tag{3.8}$$

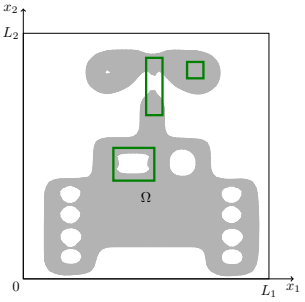
This result is very similar to that obtained in Ref. [58], with the important difference that the width of the kernel depends on the mesh size, h, on which the B-spline is defined, and not on the voxel size, Δ. Note that the approximate kernel κ depends on the difference between the coordinates x and y only, this in contrast to the kernel $\mathcal{K}$. Consequently, when the approximate kernel κ is considered, the integral transform (3.6) becomes a convolution operation.



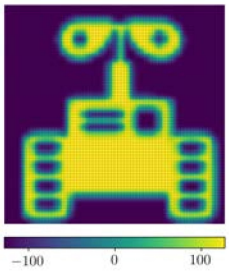
(a) 35×35 grayscale data



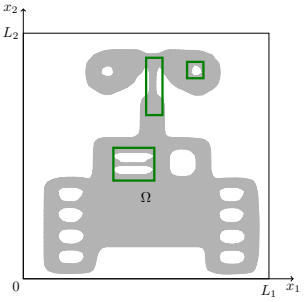
(b) Level set for $h_d = \Delta_d$



(c) Geometry for $h_d = \Delta_d$

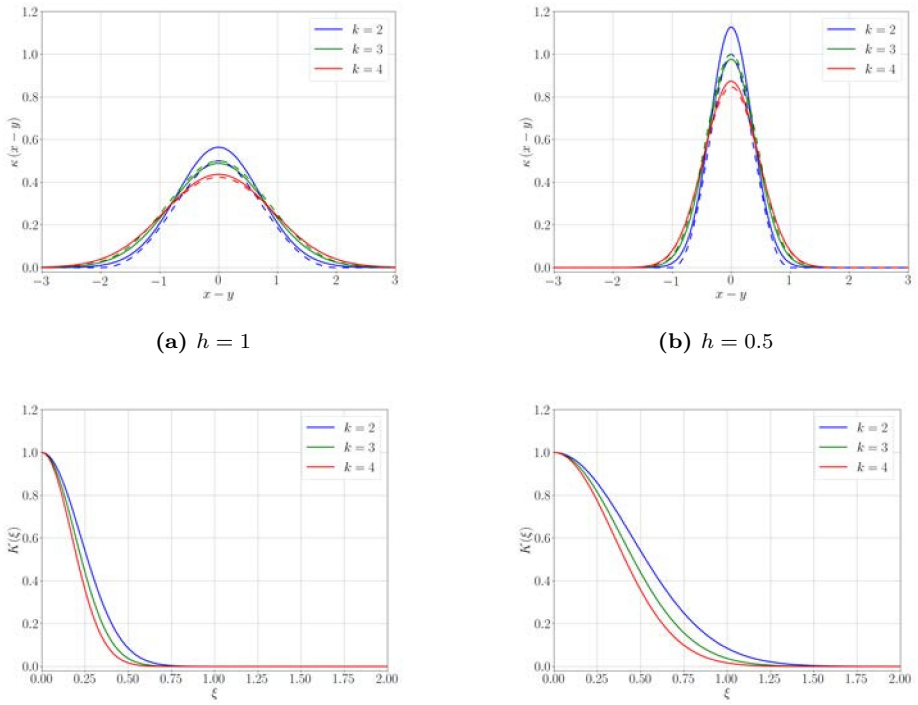


(d) Level set for $h_d = \Delta_d/2$



(e) Geometry for $h_d = \Delta_d/2$

Figure 3.3: Illustration of the B-spline-based segmentation procedure for 35×35 voxel grayscale data (panel (a)). Using second-order B-splines constructed on a regular mesh that coincides with the voxels (panel (b)) leads to topological changes in the segmented domain (panel (c)) compared to the 70×70 voxels case in Figure 3.2. When the B-spline basis is constructed on a mesh that is twice as fine as the voxel grid, a smooth level set function is obtained (panel (d)) which, after segmentation, correctly represents the topology of the object (panel (e)).



(a) $h = 1$ **(b)** $h = 0.5$

(c) $h = 1$ **(d)** $h = 0.5$

Figure 3.4: The convolution kernel (3.7) in the spatial domain (panels (a) and (b)), and in the frequency domain (panels (c) and (d)) for $k = 2, 3, 4$ using different mesh sizes. The integration kernel (3.6) is plotted in the spatial domain for reference (dashed lines).

The dependence of the convolution kernel on the mesh size and on the order of the B-spline interpolation is illustrated in Figures 3.4a and 3.4b. The B-spline integral transform around $x = 0$ computed on a domain of size 10 with a voxel size 1 is shown for reference, indicating that the Gaussian approximation improves with an increasing B-spline order. Figure 3.4 conveys that, following equation (3.8), increasing the mesh size and the B-spline order both increase the zone over which the grayscale data is averaged. The smoothing width scales linearly with the mesh size, and, for sufficiently large B-spline orders, with the square root of k.

To provide detailed insights into how the filter properties lead to topological anomalies, we express the convolution operation (3.6) with the approximate kernel (3.7) in the frequency domain as

$$F(\xi) = K(\xi)G(\xi), \tag{3.9}$$

where $F(\xi)$ and $G(\xi)$ are the Fourier transforms of the original grayscale data and the convoluted level set function, respectively, and where the Fourier transform of the convolution kernel (3.7) is given by

$$K(\xi) = \exp\left(-2\pi^2\xi^2\sigma^2\right). \tag{3.10}$$

This frequency-domain form of the kernel is shown in Figures 3.4c and 3.4d. The Fourier-form of the convolution operation in equation (3.9) conveys that features corresponding to frequencies for which $K(\xi)$ is close to unity are preserved in the smoothing operations, whereas features corresponding to frequencies for which $0 < K(\xi) \ll 1$ are filtered.

To further clarify the preservation of features, we consider a one-dimensional object of size ℓ, represented by the grayscale function (in both the spatial and in the frequency domain)

$$g(x) = \begin{cases} 1 & |x| < \ell/2 \\ 0 & \text{otherwise} \end{cases}, \qquad G(\xi) = \ell\,\mathrm{sinc}(\ell\xi) = \frac{\sin(\pi\ell\xi)}{\pi\xi}. \tag{3.11}$$

The smooth approximation of this feature in the frequency domain follows from equation (3.9) as

$$F(\omega) = \ell\,\mathrm{sinc}(\ell\xi)\exp\left(-2\pi^2\xi^2\sigma^2\right). \tag{3.12}$$

The corresponding function in the spatial domain can be determined by expressing the sinc function as [166]

$$\mathrm{sinc}(\ell\xi) = \lim_{m\to\infty}\frac{1}{m}\sum_{n=1}^{m}\cos\left(2\mu(m,n)\pi\xi\right), \tag{3.13}$$

with $\mu(m,n) = \frac{(2n-1)\ell}{4m}$, such that the final form of the approximated convolution operation (3.6) can be written as

$$f(x) = \frac{\ell}{\sigma\sqrt{2\pi}} \lim_{m\to\infty} \frac{1}{2m} \sum_{n=1}^{m} \left[\exp\left(-\frac{(x-\mu(m,n))^2}{2\sigma^2} \right) + \exp\left(-\frac{(x+\mu(m,n))^2}{2\sigma^2} \right) \right]. \tag{3.14}$$

The approximate level set function (3.14) is illustrated in Figure 3.5 for various feature-size-to-mesh ratios, $\hat{\ell} = \ell/h = 2, 1, \frac{1}{2}$, and B-spline degrees, $k = 2, 3, 4$. For the considered range of feature-size-to-mesh ratios, the limit in equation (3.14) is already approximated well with $m = 1$, such that

$$f(h\hat{x}) \approx \hat{f}_1(\hat{x}) = \sqrt{\frac{3\hat{\ell}^2}{4\pi(k+1)}} \left[\exp\left(-\frac{3\left(4\hat{x}-\hat{\ell}\right)^2}{16(k+1)} \right) + \exp\left(-\frac{3\left(4\hat{x}+\hat{\ell}\right)^2}{16(k+1)} \right) \right], \tag{3.15}$$

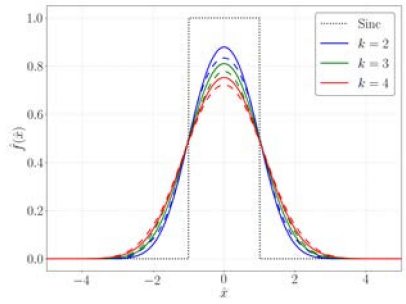
where $\hat{x} = x/h$. The value of the smoothed level set function at $\hat{x} = 0$ follows as

$$\hat{f}_1(0) = \sqrt{\frac{3\hat{\ell}^2}{\pi(k+1)}} \exp\left(-\frac{3\hat{\ell}^2}{16(k+1)} \right) \approx \hat{\ell}\sqrt{\frac{3}{\pi(k+1)}}, \tag{3.16}$$

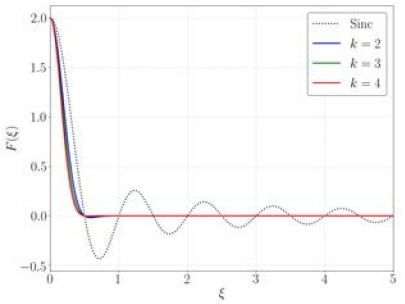
which conveys that the maximum value of the smoothed level set depends linearly on the relative feature size $\hat{\ell}$, and decreases with increasing B-spline order.

The top row in Figure 3.5 shows the case for which the considered feature is twice as large as the mesh size, *i.e.*, $\hat{\ell} = 2$. Figure 3.5a illustrates that the sharp boundaries of the original grayscale function are significantly smoothed, which is also apparent from the frequency domain plot in Figure 3.5b, which shows that the high frequency modes required to represent the sharp boundary are filtered out by the smoothing operation. In Figure 3.5a, the decrease in the maximum level set value as given by equation (3.16) is observed. When the level set function is segmented by a threshold of $g_{\text{crit}} = 0.5$, a geometric feature that closely resembles the original one is recovered.

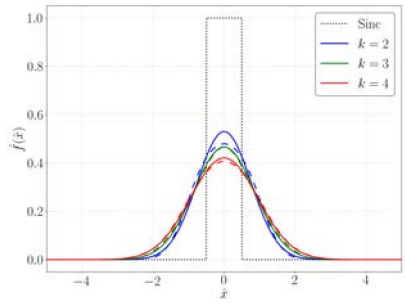
The middle and bottom rows of Figure 3.5 illustrate cases where the feature width is not significantly larger than the mesh size. For the case where the feature size is equal to the size of the mesh, the maximum of the level set function



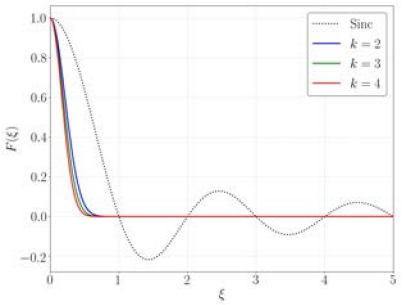
(a) $\hat{\ell} = \frac{\ell}{h} = 2$



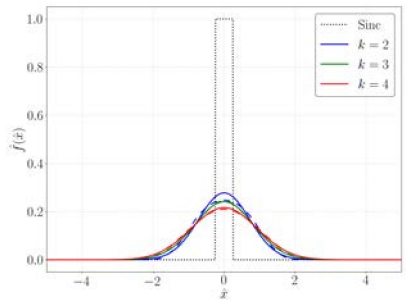
(b) $\hat{\ell} = \frac{\ell}{h} = 2$



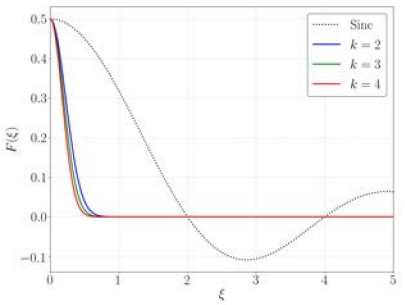
(c) $\hat{\ell} = \frac{\ell}{h} = 1$



(d) $\hat{\ell} = \frac{\ell}{h} = 1$



(e) $\hat{\ell} = \frac{\ell}{h} = \frac{1}{2}$



(f) $\hat{\ell} = \frac{\ell}{h} = \frac{1}{2}$

Figure 3.5: Smoothed level set approximation (3.14) of a geometric feature in the spatial domain (left) and in the frequency domain (right) for various feature-size-to-mesh ratios, $\hat{\ell} = \ell/h$, and B-spline degrees. The B-spline level set function (3.6) is shown in the spatial domain for reference (dashed lines).

drops significantly compared to the case of $\hat{\ell} = 2$. When considering second-order B-splines, the maximum is still marginally above $g_{\text{crit}} = 0.5$. Although the recovered feature is considerably smaller than the original one, it is still detected in the segmentation procedure. When increasing the B-spline order, the maximum value of the level set drops below the segmentation threshold, however, indicating that the feature will no longer be detected. As a consequence, the B-spline segmentation procedure then induces a topological alteration. When decreasing the feature length further, as illustrated in the bottom row of Figure 3.5, topological changes are encountered at lower segmentation thresholds. For the case of $g_{\text{crit}} = 0.5$, also the quadratic B-splines would lead to a loss of the feature.

In summary, the above analysis shows that topological anomalies occur when the relative feature length scale, $\hat{\ell} = \ell/h$, becomes too small. The smallest features in a voxel data file are of the size of a single voxel, *i.e.*, $\ell = \Delta$. Hence, topological features are lost when the mesh size on which the B-spline level set is constructed is relatively large compared to the voxel size. For moderate B-spline orders, this practically means that for features with the size of a single voxel, the mesh of the B-spline level set should be a uniform refinement of the voxel mesh, such that the relative feature size $\hat{\ell}$ is sufficiently large. With this mesh refinement, the resulting level set function will still be C^{k-1}-continuous, but higher-frequent modes are present in the refined level set function. In places where the geometric features are sufficiently large, this is in principle not desirable. Ideally, one only should refine the B-spline level set function in places where this is needed to preserve the topology, *i.e.*, around small features. In the next section, we propose a fully-automated topology-preserving B-spline segmentation strategy that refines the B-spline level set function in such a manner that topological anomalies are avoided.

3.3 Topology-preserving image segmentation using THB-splines

In this section we present a topology-preserving B-spline-based image segmentation strategy relying on the technique proposed in Ref. [58]. The proposed strategy consists of two steps (schematically illustrated in Figure 3.6). In the first step, which we will discuss in detail in Section 3.3.1, a moving-window strategy is applied to locally compare the topology of the original voxel data and its smoothly segmented counterpart. The result of this first step is a function that indicates regions where topological anomalies occur. In the second step, which is discussed in Section 3.3.2, truncated hierarchical (TH)B-splines are employed to locally repair the topology by mesh refinement, following the analysis presented in Section 3.2. To demonstrate the proposed image segmentation technique, var-

ious test cases, including the problematic scenario considered in Section 3.2, will be discussed in Section 3.3.3.

3.3.1 Moving-window topological anomaly detection

In this section we detail the moving-window strategy to detect topological anomalies. In Section 3.3.1 we commence with the definition of the moving window concept, as is commonly used in image processing techniques (*e.g.*, [167–169]). Subsequently, in Section 3.3.1 we introduce the window-comparison operator used to identify topological changes. A masking operation is finally introduced in Section 3.3.1 to distinguish between boundary changes and topological changes.

The moving window

To detect local topological changes, local views on the original voxel data and its smoothed counterpart will be compared. The windows are created by considering the r-neighborhood of each voxel in the original image, *i.e.*,

$$\Omega_{\text{win}} = \mathcal{N}_r(\Omega_{\text{vox}}) \qquad \forall \Omega_{\text{vox}} \in \mathcal{V}_{\text{vox}}. \tag{3.17}$$

The 0-neighborhood of a voxel is defined as the voxel itself, *i.e.*, $\mathcal{N}_0(\Omega_{\text{vox}}) := \Omega_{\text{vox}}$, and the r-neighborhood for $r \geq 1$ is defined recursively as the union of all voxels that share a vertex with the $(r-1)$-neighborhood. This window definition is illustrated in Figure 3.7 for a 10×10 voxel grid.

As discussed in Section 3.2, it is desirable to keep the mesh refinement to repair topological anomalies as local as possible. This means that the moving window should be as small as possible, but large enough to detect topological anomalies. Hence, the window should be larger than the size of the geometric features that are subject to topological changes. From the analysis in Section 3.2.2 it follows that features with a size similar to that of the voxels, *i.e.*, $\ell \approx \Delta$, can be subject to topological changes if the smoothing is performed on a mesh with a size similar to that of the voxels, *i.e.*, $h \approx \Delta$. Practically this means that a window with only a few rings, *i.e.*, $\ell \approx \Delta < (2r+1)\Delta \ll L$, is expected to be an adequate choice. The influence of the window size will be studied for the example discussed in Section 3.3.3.

Window image comparison

The moving-window technique indicates whether, for the window focused at every voxel, a topological difference is observed between the directly segmented image, V, and the geometry related to the smoothed image, S (see Figure 3.6). We herein employ the midpoint tessellation strategy proposed in Ref. [86] to construct a geometry parametrization, as illustrated in Figure 3.8a. Details of this

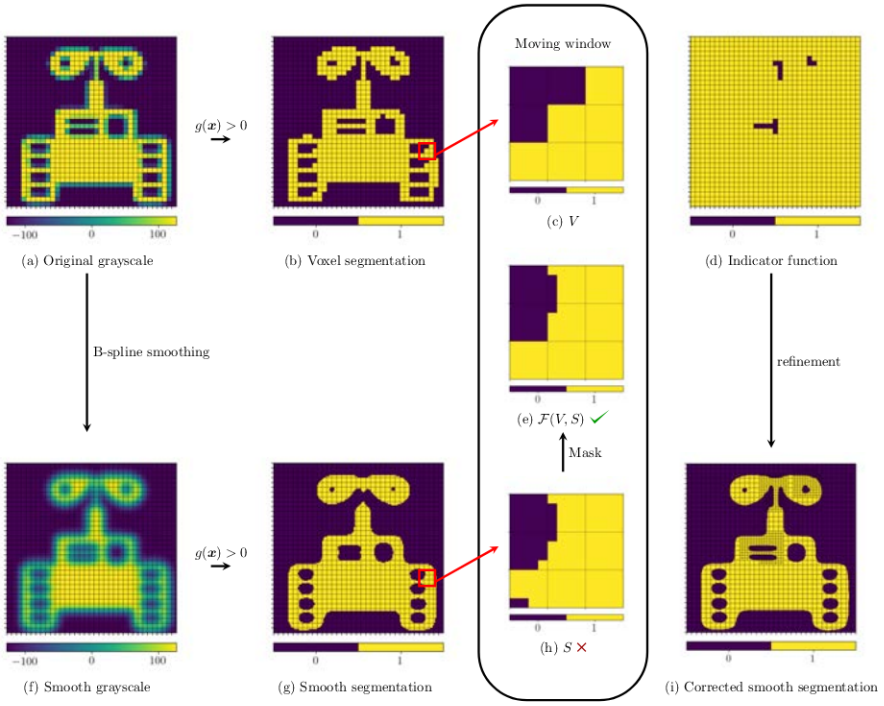


Figure 3.6: Illustration of the topology-preserving image segmentation procedure. The original grayscale image (panel (a)) is segmented in two ways, *viz.*, directly by thresholding the voxel data (panel (b)), and through the B-spline smoothing strategy (panels (f) and (g)). A moving-window technique then locally compares the topology between the two segmented images, which results in an indicator function (panel (d)) to mark topological differences. THB-spline-based refinements are then introduced to locally increase the resolution of the smooth level set function (panel (i)), thereby preserving the topology of the original image.

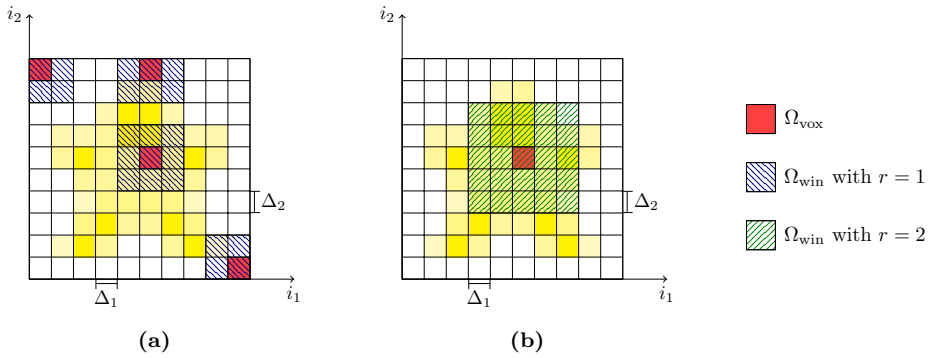


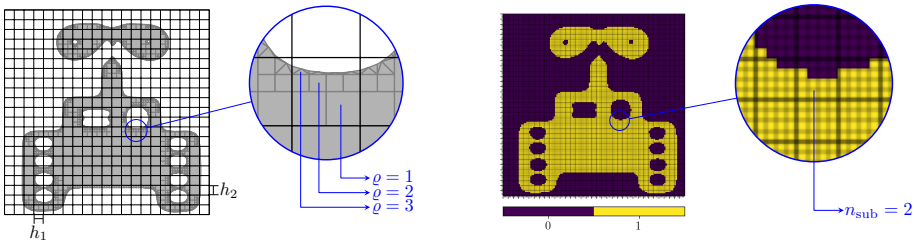
Figure 3.7: Illustration of the moving window Ω_{win} centered at a voxel Ω_{vox}, with (a) a one-neighborhood ($r = 1$) and (b) a two-neighborhood ($r = 2$).

procedure will be discussed in the context of the immersed isogeometric analysis framework discussed in Section 3.4. To enable the window topology comparison, it is convenient to employ the same representation concept for both images, V and S. Therefore, the tessellated geometry is first voxelized by segmentation of the B-spline level set function on a grid that is n_{sub}-times uniformly refined with respect to the original voxel grid, as shown in Figure 3.8b. This refinement level should be chosen such that the voxelization of the smoothed image does not induce topological changes compared to the geometry on which the simulation is performed. Since the employed midpoint tessellation strategy also relies on a recursive subdivision approach (of the computational cells), the number of voxel subdivisions, n_{sub}, can be related directly to the number of recursive refinements used for the midpoint tessellation (see Figure 3.8).

To characterize topological similarity, we define the window topology-comparison operator:

$$\mathscr{C}(V,S) = \begin{cases} \text{true} & \text{if } V \text{ and } S \text{ are topologically equivalent,} \\ \text{false} & \text{if } V \text{ and } S \text{ are topologically different.} \end{cases} \tag{3.18}$$

To compute this boolean operator, the filled voxels in the images V and S are divided into connected regions. For the image V, for example, the regions are denoted by $R \in \mathcal{R}_V$, where $\mathcal{R}_V$ is the set of all connected regions. We herein employ vertex-connectivity, meaning that voxels sharing a vertex are considered to be connected, but the same procedure can be applied using face-connectivity. For each connected region, R, we determine the Euler characteristic, $\chi(R)$, which is defined as one minus the number of connected holes, $\#\mathcal{V}_R$, in that region. Following the definition in Ref. [170], the Euler characteristic of the window



(a) **(b)**

Figure 3.8: Schematic of (a) the midpoint tessellation procedure (with a recursion depth of $\varrho_{\max} = 2$) presented in Refs. [58, 86] and (b) the voxelization on a grid (with $n_{\text{sub}} = 2$) that is refined to match the recursion depth of the tessellation procedure.

image is given by

$$\chi(V) = \sum_{R \in \mathcal{R}_V} \chi(R) = \sum_{R \in \mathcal{R}_V} (1 - \#\mathcal{V}_R) = \#\mathcal{R}_V - \sum_{R \in \mathcal{R}_V} \#\mathcal{V}_R. \tag{3.19}$$

We note that various methods of evaluating the Euler characteristic have been rigorously studied in the field of homology, see, *e.g.*, Refs. [171–174]. In our work we consider the method of Ref. [170] which has been implemented in scikit-image [175], an open-source image processing library for Python.

To construct the boolean operator (3.18), we define sets of regions with a specific Euler characteristic, $\bar{\chi} \in \boldsymbol{\chi}_V = \{\chi(R) \mid R \in \mathcal{R}_V\} \subset \mathbb{Z}$, in an image as

$$\mathcal{R}_V^{\bar{\chi}} = \{R \in \mathcal{R}_V \mid \chi(R) = \bar{\chi}\}. \tag{3.20}$$

Figure 3.9 shows illustrative examples of these region sets. Using the region sets (3.20), the following indicator function for topological anomalies is proposed:

$$\begin{aligned} \mathscr{C}(V,S) = &\left[\bigwedge_{\bar{\chi} \in \boldsymbol{\chi}_V \cup \boldsymbol{\chi}_S} (\#\mathcal{R}_V^{\bar{\chi}} \equiv \#\mathcal{R}_S^{\bar{\chi}})\right] \\ &\bigwedge \left[\bigwedge_{\bar{\chi} \in \boldsymbol{\chi}_{V'} \cup \boldsymbol{\chi}_{S'}} (\#\mathcal{R}_{V'}^{\bar{\chi}} \equiv \#\mathcal{R}_{S'}^{\bar{\chi}})\right] \end{aligned} \tag{3.21}$$

This indicator states that the number of regions with a particular Euler characteristic is equal in the images V and S, as well as in their complements V' and S'. Note that, by construction, the operator (3.21) is invariant to the complement operation, *i.e.*, $\mathscr{C}(V,S) = \mathscr{C}(V',S')$, which makes the comparison operator

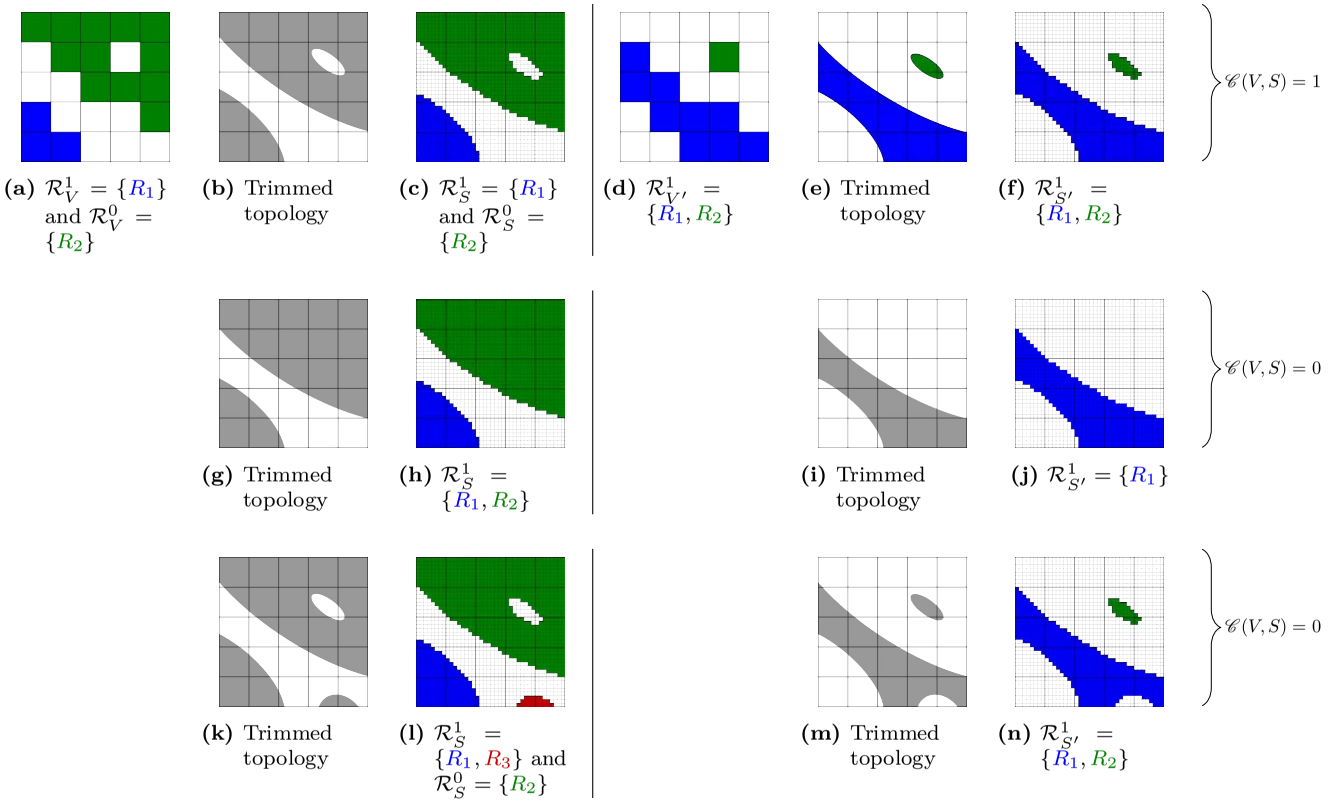


Figure 3.9: Illustration of the comparison operator (3.21) for the images V and V' (original gray scale data) in the first and fourth column and the image S and S' (B-spline-based segmented domain extracted using the midpoint tessellation procedure with $\varrho_{\max} = 3$) in the second and fifth column. The third and sixth column are the $n_{\text{sub}} = 3$ voxelizations of the segmented domain. The images in the first row result in $\mathscr{C}(V,S) = \mathscr{C}(V',S') = 1$ as V and S are topologically equivalent. The images in the second row results $\mathscr{C}(V,S) = \mathscr{C}(V',S') = 0$ as V and S are topologically different. The images in the third row results in $\mathscr{C}(V,S) = \mathscr{C}(V',S') = 0$ as V and S are topologically different due to a boundary spill over.

objective with respect to the definition of the solid and void regions. One may note that if $\chi_V \neq \chi_S$ or $\chi_{V'} \neq \chi_{S'}$, then $\mathscr{C}(V,S)$ automatically reverts to false.

Figure 3.9 illustrates the comparison operator (3.21), where use has been made of a $n_{\text{sub}} = 3$ sampling of the B-spline-based segmented domain extracted using the midpoint tessellation procedure with $\varrho_{\max} = 3$. The first row in Figure 3.9 shows a case where the topology matches, despite the substantial changes in geometry. The second row in Figure 3.9 shows a case where the elliptical region is missing, which is a typical case of a topological anomaly. The third row in Figure 3.9 shows a case in which the comparison operator returns false, which is caused by the appearance of a boundary spillover due to the smoothing procedure at the right bottom border. From the perspective of the window, the observed change indeed classifies as a topological change. However, when considering the change from the perspective of the complete image, the observed difference comes from a boundary that moves into the view of the window under consideration. This boundary movement classifies as a shape (geometry) change, and not as a topological change. To correctly account for this type of changes, in the next section we propose an image masking operation.

Window image masking

To mask the window changes associated with moving boundaries, as a preprocessing step to the comparison operation (3.21), the smoothed image S is masked using

$$F = \mathcal{F}(V,S) = (M \cap V) \cup (M' \cap S), \tag{3.22}$$

where F is the masked image and the mask M depends on arguments V and S, *i.e.,* $M = \mathcal{M}(V,S)$. This mask corresponds to a set-indication function which is one in places where a boundary change is detected and zero everywhere else. The mask is illustrated in Figure 3.10 for the case considered in the third row of Figure 3.9. The key idea behind this masking operation is that in regions where shape changes occur, the masked image F is replaced by the original voxel image V so that the changes associated with boundary movement are effectively reverted.

To identify the locations of the changes between the images V and S, we consider the mask to be a subset of the symmetric difference between the original and the smoothed image, *i.e.*, $M \subseteq V \Delta S = (V \cap S') \cup (S \cap V')$ (see Figure 3.10). We now only mask the regions in the symmetric difference which reside completely in the outer ring of the window *and* have an Euler characteristic of one (there are no voids), that is,

$$\mathcal{M}(V,S) = \left\{ R \in \mathcal{R}^1_{V \Delta S} \mid R \cap W_{r-1} = \emptyset \right\}, \tag{3.23}$$

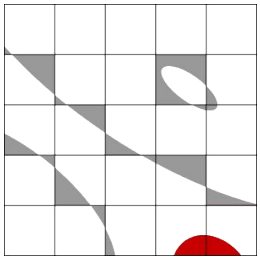


Figure 3.10: The symmetric difference (3.24) between the images V and S in Figure 3.9, with the boundary mask $M = \mathcal{M}(V, S)$ shown in red.

where W_{r-1} is equal to zero in the outer voxel ring of the window and one everywhere else. Note that since the symmetric difference is unchanged when the complement images are considered, *i.e.*,

$$(V')\Delta(S') = (V' \cap S) \cup (S' \cap V) = V\Delta S, \tag{3.24}$$

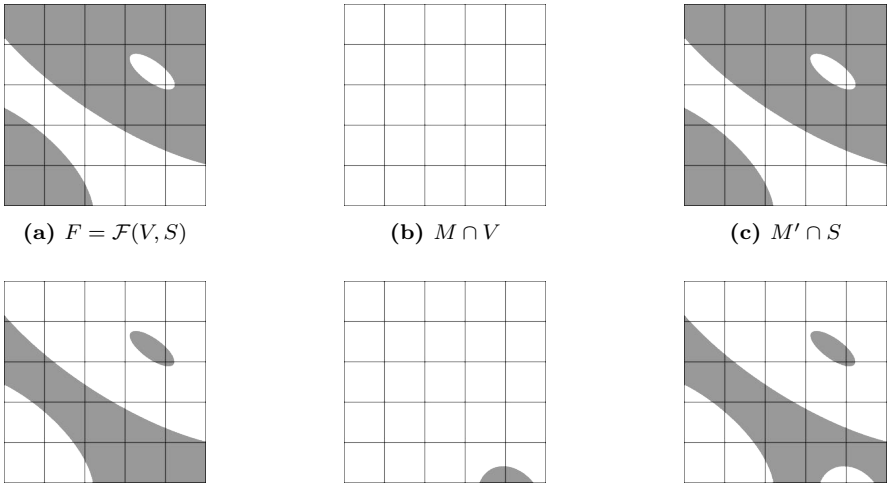
for the mask (3.23) it holds that

$$\mathcal{M}(V', S') = \mathcal{M}(V, S) = M. \tag{3.25}$$

From this property of the mask it then follows that the complement of the masked image F is equal to the masked complement images V' and S':

$$\begin{aligned} F' = \mathcal{F}(V,S)' &= (M \cap V)' \cap (M' \cap S)' = (M' \cup V') \cap (M \cup S') \\ &= [M' \cap (M \cup S')] \cup [V' \cap (M \cup S')] \\ &= (M' \cap S') \cup (V' \cap M) \cup (V' \cap S') \\ &= (M' \cap S') \cup (V' \cap M) \cup (V' \cap S' \cap M) \cup (V' \cap S' \cap M') \\ &= (M \cap V') \cup (M' \cap S') \\ &= \mathcal{F}(V', S') \end{aligned} \tag{3.26}$$

The choice to identify shape changes associated with moving boundaries by definition (3.23) is motivated by the idea that if one considers a boundary in the global image with small curvature, the image smoothing operation discussed in Section 3.2 will typically retain the movement of the boundary within one voxel spacing. When the curvature of the boundary is locally high, the boundary movement can be larger than a single voxel, which would lead to falsely identifying a shape change as a topological change. In our algorithm this would lead to a refinement of the level set function which would not be strictly required to preserve topology. Avoiding such auxiliary refinements would, however, severely



(a) $F = \mathcal{F}(V, S)$ **(b)** $M \cap V$ **(c)** $M' \cap S$

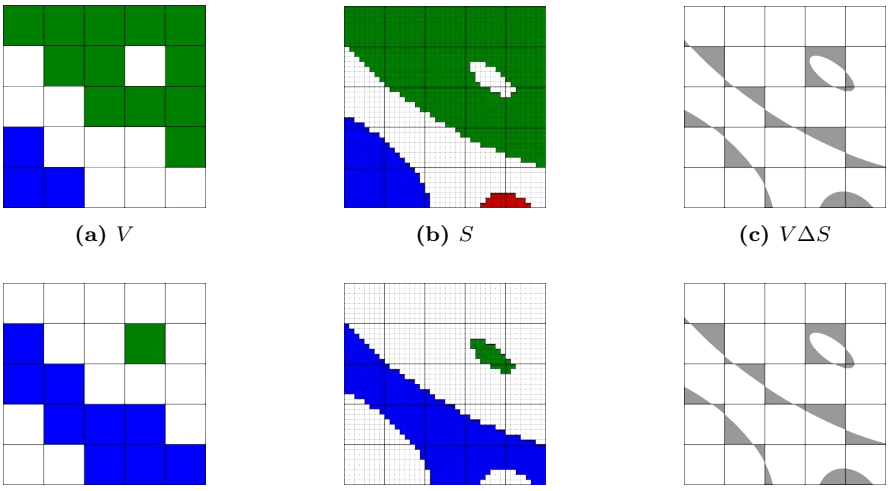
(d) $F' = \mathcal{F}(V', S')$ **(e)** $M \cap V'$ **(f)** $M' \cap S'$

Figure 3.11: An illustration of the filtering operation (3.22) for an image V and the corresponding smoothed image S (see Figure 3.12) and for their complements.

complicate the algorithm. Moreover, since they generally result in an improved geometry representation at high-curvature boundaries, it is considered desirable not to avoid them.

The masking operation (3.22) is illustrated in Figure 3.11 for an exemplifying image V and its smoothed version S (as defined in Figure 3.12), as well as for the complements of these images. The symmetric difference between the images V and S is shown in Figure 3.12c, which, in agreement with equation (3.24), is identical to the symmetric difference between V' and S' in Figure 3.12f. The masked regions are color coded in red, indicating that only completely solid regions in the outer ring are considered in the mask. The masked images $F = \mathcal{F}(V, S)$ and $F' = \mathcal{F}(V', S')$ are shown in Figures 3.11a and 3.11d, from which it is observed that, in accordance with equation (3.26), the complement of the masked image F is equal to the masked complement image $\mathcal{F}(V', S')$.

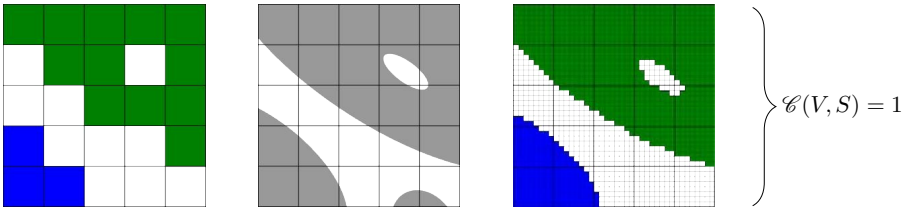
In Figure 3.13 we consider the case of the third row of Figure 3.9. As discussed above, when the images V and S are compared directly, the comparison operator (3.21) marks these images to be topologically different on account of the shape change associated with the boundary movement. When comparing the image V with the masked image $F = \mathcal{F}(V, S)$, as shown in Figure 3.13, the images are considered to be topologically equivalent, which, in this case, is considered as the correct indicator result.



(a) V **(b)** S **(c)** $V \Delta S$

(d) V' **(e)** S' **(f)** $V' \Delta S'$

Figure 3.12: An example of (a) the original segmented image V, (b) the spline-based segmented image S, and (c) the symmetric difference. Panels (d) and (e) are the complements of V and S, respectively. Panel (f) is the symmetric difference of the complements, which is identical to that in panel (c).



(a) V **(b)** Trimmed topology **(c)** $F = \mathcal{F}(V,S)$

Figure 3.13: An illustration of the boundary masking procedure applied to the case in the third row of Figure 3.9. The image V in panel (a) is compared with the masked image F in panel (c) to obtain the indicator function $\mathscr{C}(V,S)$. In contrast to the result in Figure 3.9, the masking operation results in a $\mathscr{C}(V,S) = 1$, indicating that the images V and S are topologically equivalent.

3.3.2 THB-spline-based refinement strategy

After identification of the windows in which topological changes occur, the mesh on which the level set function is constructed is locally refined to resolve these anomalies (see Figure 3.6). More specifically, a support-based refinement procedure is employed, in which all basis functions that are supported on a voxel that displays a topological anomaly are replaced by basis functions from the next higher hierarchical level.

We herein employ truncated hierarchical B-splines [158–160] to construct a spline basis over locally refined meshes. The THB-spline construction is illustrated in Figure 3.14, which, for the sake of generality, considers the case of multiple refinement levels, with the level $\ell = 0$ corresponding to the coarsest elements, and the level $\ell = \ell_{\max}$ to the most refined elements. We denote the region covered by elements that are at least ℓ times refined by Ω^ℓ (note that the refinement regions are nested, *i.e.*, $\Omega^\ell \subseteq \Omega^{\ell-1} \subseteq \ldots \subseteq \Omega^0 = \Omega_{\text{vox}}$). The locally refined mesh corresponding to these refinement regions is denoted by $\mathcal{V}_{\text{vox}}$.

To construct the truncated hierarchical B-spline basis, we consider $\ell_{\max}$ uniform refinements, $\mathcal{V}^\ell_{\text{vox}}$, of the original voxel mesh $\mathcal{V}^0_{\text{vox}}$. The mesh size of the original voxel mesh is denoted by h and that of its refinements by $2^{-\ell}h$. With each level we associate a mesh

$$\mathcal{V}^\ell = \left\{ v \in \mathcal{V}^\ell_{\text{vox}} \mid v \cap \Omega^\ell \neq \emptyset \right\}. \tag{3.27}$$

We define a B-spline basis of degree k and regularity α over each of these meshes as

$$\mathcal{B}(\mathcal{V}^\ell) = \{ N \in \mathcal{B}(\mathcal{V}^\ell_{\text{vox}}) \mid \text{supp}\,(N) \cap \Omega^\ell \neq \emptyset \}, \tag{3.28}$$

where $\mathcal{B}(\mathcal{V}^\ell_{\text{vox}})$ is the B-spline basis on $\mathcal{V}^\ell_{\text{vox}}$.

To construct the truncated hierarchical B-spline basis, splines from the bases over the uniform meshes (3.28) are selected and truncated. On the most refined level, *i.e.*, at $\ell = \ell_{\max}$, all basis functions that are completely inside $\Omega^{\ell_{\max}}_{\text{vox}}$ are selected:

$$\mathcal{H}(\mathcal{V}^{\ell_{\max}}) = \{ N \in \mathcal{B}(\mathcal{V}^{\ell_{\max}}) \mid \text{supp}\,(N) \subseteq \Omega^{\ell_{\max}} \}. \tag{3.29}$$

At coarser levels, *i.e.*, $0 \leq \ell < \ell_{\max}$, the functions that are completely inside the domain Ω^ℓ, but not completely inside the refined domain $\Omega^{\ell+1}$, are selected and truncated:

$$\mathcal{H}(\mathcal{V}^\ell) = \{ \text{trunc}(N) \mid N \in \mathcal{B}(\mathcal{V}^\ell), \text{supp}\,(N) \subseteq \Omega^\ell, \text{supp}\,(N) \nsubseteq \Omega^{\ell+1} \} \tag{3.30}$$

The truncation operation reduces the support of the B-spline functions by projecting away basis functions retained from the refined levels (see Ref. [158, 176] for details). The THB-spline basis then follows as

$$\mathcal{H}(\mathcal{V}) = \bigcup_{\ell=0}^{\ell_{\max}} \mathcal{H}(\mathcal{V}^\ell). \tag{3.31}$$

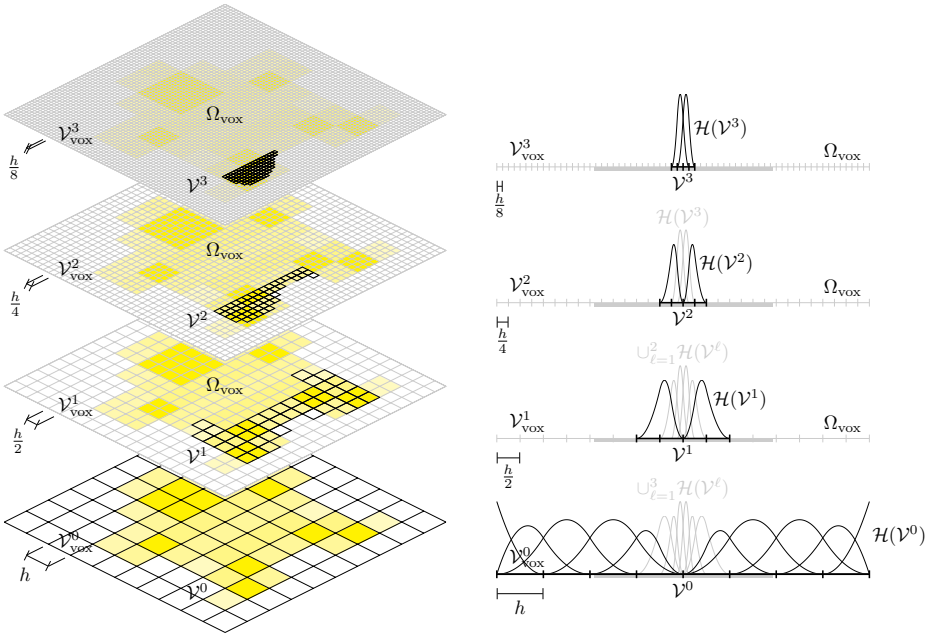
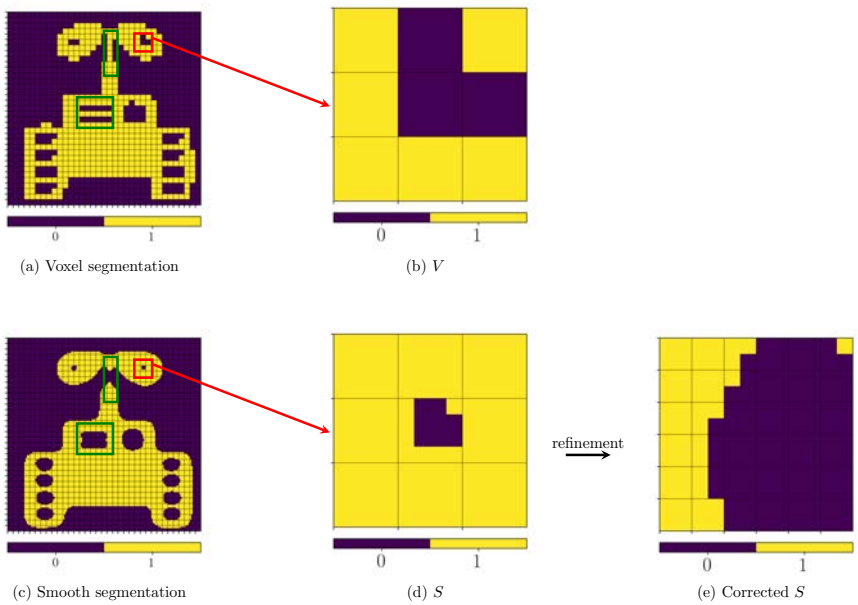


Figure 3.14: An illustration of truncated hierarchical B-splines. The left column shows the hierarchical levels of a mesh $\mathcal{V}_{\text{vox}}$, while the right column illustrates the concept for a one-dimensional voxel domain Ω_{vox}.

This selection and truncation procedure is illustrated in Figure 3.14. It is noted that THB-splines satisfy the partition of unity property (in contrast to non-truncated hierarchical B-splines), which is an important property from the perspective of the image smoothing procedure (3.5) as it guarantees that the average grayscale intensity is preserved [58]. In this work we employ the THB-spline implementation in the Python-based open source numerical library Nutils [109], which is based on the element-wise construction discussed in Ref. [177].

3.3.3 Examples

To illustrate the topology preserving segmentation strategy presented above we consider the example presented in the workflow in Figure 3.6. The voxelized smooth image with $n_{sub} = 2$ is shown in Figure 3.6g, which closely resembles the tessellated image with a bi-sectioning depth of $\varrho_{\max} = 3$ in Figure 3.3c. By comparison with the voxel segmentation in Figure 3.6b, it is clearly observed that topological changes occur at two locations. These locations are highlighted



(a) Voxel segmentation (b) V

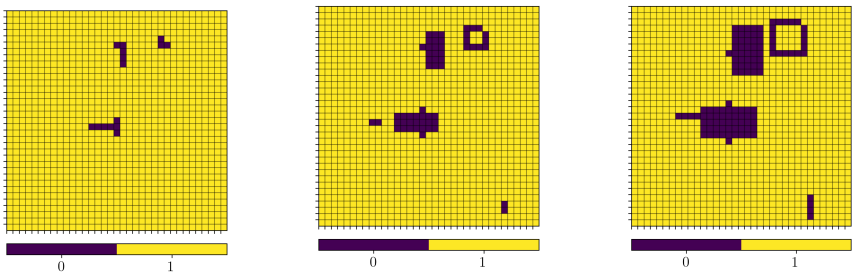
(c) Smooth segmentation (d) S (e) Corrected S

Figure 3.15: Example of the moving window strategy with a 3×3 window. Topological changes are correctly detected in the green boxes. In the red window, an L-shaped exclusion which extends beyond the outer ring of the window is falsely detected as a topological change.

in green in Figure 3.15c.

Figure 3.6d presents the comparison indicator for a window size of 3×3 ($r = 1$). It is observed that the topological changes are indeed detected. Note that for this example, the mask operation is also active. As an example of the mask operation, Figure 3.6h shows a region with a boundary change before the mask operation. It is observed that the indicator function for topological anomalies (3.21) results in zero (false) when compared to the voxel region in Figure 3.6c. Figure 3.6e shows the region after the mask operation. After application of the mask, the comparison operator results in one (true).

In Figure 3.6d it is seen that an additional region, highlighted in red in Figure 3.15c, is also marked as a topological change. Strictly speaking, this would not be necessary, as in both images a hole is present in that region. Figure 3.15b shows the window causing this behavior. In the voxel image, the L-shaped exclusion is not detected as a hole in a region, but as a void region splitting two solid regions. On the level of the window, this is a topologically ambiguous case



(a) **(b)** **(c)**

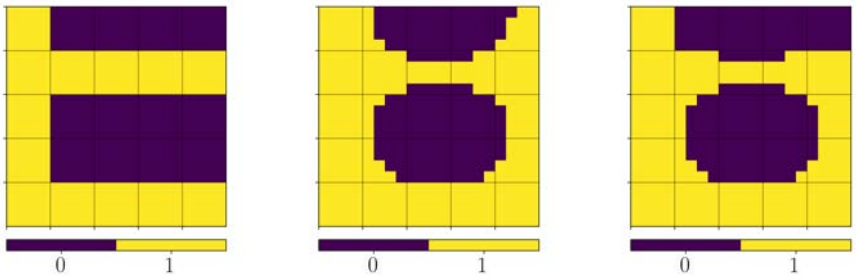
Figure 3.16: The comparison indicator function for the case considered in Figure 3.6 with a window size of (a) 3×3 ($r = 1$), (b) 5×5 ($r = 2$), and (c) 7×7 ($r = 3$).

in the sense that one needs to look outside of the window to see whether the exclusion extends beyond the window. As shown in Figure 3.6i and in the zoom in Figure 3.15e, the effect of refining the level set in this region is that the voxel geometry is better captured.

In Figure 3.16 the influence of the window size is examined. As can be seen, the refinement regions increase in size with increasing window size. Since the 3×3 window already adequately corrects the topological anomalies, the growth of the refinement regions is unnecessary. As can be seen, the effect of a larger window size on the corrected images is minimal, as elaborated in Section 3.2.2. In Figure 3.16b and 3.16c we also observe a case where a shape change is marked as a topological change, which, as discussed in Section 3.3.1, is caused by the high curvature of the boundary. A zoom of a typical window in which this occurs is shown in Figure 3.17. It is also observed in Figure 3.16c that for the incorrectly identified topology change discussed above, increasing the window size results in an indicator function in the form of a ring. This situation is not further explored in this work, but would require tailoring of the refinement marking strategy to ensure that the interior of the ring is refined.

3.4 The isogeometric finite cell method

To provide a basis for the boundary value problems considered in Section 3.5, in this section the abstract formulation for the isogeometric finite cell method [55] is introduced. We consider a physical domain, Ω, formed by the topology-preserving segmentation procedure outlined above. The domain and its boundary, $\partial\Omega$, are immersed into an ambient domain $\mathcal{A} \supset \Omega$ as shown in Figure 3.18. In the



(a) V **(b)** S **(c)** $\mathcal{F}(V,S)$

Figure 3.17: A 5×5 window showing an example of a shape change associated with a high curvature region. Since this shape change extends beyond the outer ring of the window, it is not masked and hence it is (incorrectly) detected as a topological change.

remainder we consider the ambient domain to coincide with the image (scan) domain, *i.e.*, $\mathcal{A} = \Omega_{\text{img}}$. We suppose that the problem under consideration is described by a field variable u – which can be scalar-valued or vector-valued – and the weak formulation

$$\begin{cases} \text{Find } u \in W \text{ such that:} \\ a(u,v) = f(v) \quad \forall v \in V, \end{cases} \tag{3.32}$$

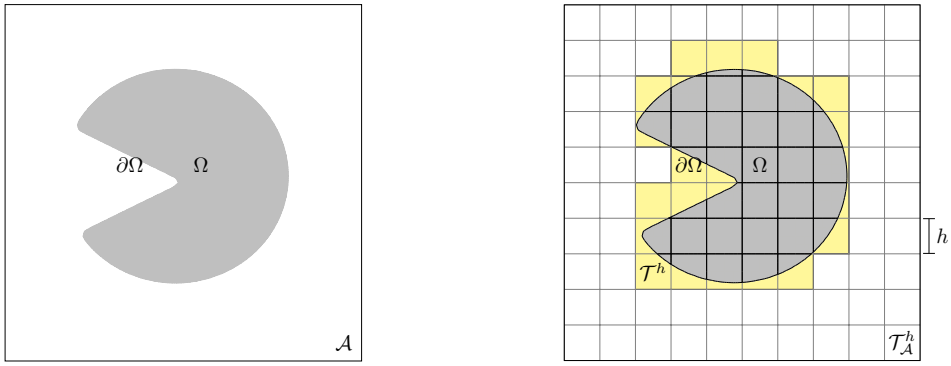
where W is the trial (solution) space, V is the test space, $a : W \times V \to \mathbb{R}$ is a continuous bilinear form and $b : V \to \mathbb{R}$ is a continuous linear form. The (isogeometric) finite cell method provides a general framework for constructing the finite dimensional subspaces $W^h \subset W$ and $V^h \subset V$, where the superscript h refers to the mesh parameter associated with the ambient domain mesh, $\mathcal{T}_{\mathcal{A}}^h$, on which the approximation to the field variable u is computed. Note that the mesh $\mathcal{T}_{\mathcal{A}}^h$ can be different from the level set mesh $\mathcal{V}^h$ discussed in Section 3.2, as the mesh resolution requirements following from the approximation of the field variable u generally differ from those for the level set function.

We define the background mesh as all elements in the ambient domain that touch the physical domain, *i.e.*,

$$\mathcal{T}^h := \{K \in \mathcal{T}_{\mathcal{A}}^h : K \cap \Omega \neq \emptyset\}, \tag{3.33}$$

and the interior mesh of the domain Ω as

$$\mathcal{T}_{\Omega}^h := \{K \cap \Omega : K \in \mathcal{T}_{\mathcal{A}}^h\}, \tag{3.34}$$



(a) **(b)**

Figure 3.18: Schematic representation of (a) the physical domain Ω (gray) with boundary $\partial\Omega$ which is embedded in the ambient domain $\mathcal{A}$, and (b) the ambient domain mesh $\mathcal{T}^h_{\mathcal{A}}$ and the background mesh $\mathcal{T}^h$ (yellow), with mesh size parameter h.

where the elements in the background mesh are trimmed to the physical domain (see Ref. [86] for details). The physical domain boundary mesh is defined as

$$\mathcal{T}^h_{\partial\Omega} := \{K \cap \partial\Omega : K \in \mathcal{T}^h_{\mathcal{A}}\}. \tag{3.35}$$

The finite dimensional subspaces W^h and V^h are then constructed using THB-splines (as elaborated in Section 3.3.2). We denote the THB-spline space of degree k and regularity α, constructed over the locally-refined ambient domain $\mathcal{T}^h_{\mathcal{A}}$, by

$$S^k_\alpha(\mathcal{A}) = \{N \in C^\alpha(\mathcal{A}) \,:\, N|_K \in P^k(K), \forall K \in \mathcal{T}^h_{\mathcal{A}}\}, \tag{3.36}$$

where $P^k(K)$ is the collection of n_d-variate polynomials on the element $K \subset \mathbb{R}^{n_d}$. The approximation spaces are then obtained by restricting the THB-splines in $S^k_\alpha(\mathcal{A})$ to the physical domain Ω:

$$W^h = V^h = \{N|_\Omega : N \in \mathcal{S}^k_\alpha(\mathcal{A})\}. \tag{3.37}$$

Due to the non-mesh-conforming character of the (isogeometric) finite cell method, it is infeasible to impose Dirichlet boundary conditions by (strongly) constraining functions in the spaces (3.37). Instead, Dirichlet conditions are imposed weakly through Nitsche's method [77, 78]. By employing a mesh-dependent consistent stabilization term, a well-posed Galerkin problem is obtained:

$$\begin{cases} \text{Find } u^h \in W^h \text{ such that:} \\ a^h(u^h, v^h) = b^h(v^h) \quad \forall v^h \in V^h \end{cases} \tag{3.38}$$

In this problem the bilinear form $a^h : W^h \times V^h \to \mathbb{R}$ and linear form $b^h : V^h \to \mathbb{R}$ are the finite dimensional versions of the operators in (3.32), augmented with the above-mentioned Nitsche terms (which will be specified in Section 3.5).

Although the Galerkin problem, in (3.38), closely resembles that of mesh-conforming finite element formulations, the immersed setting requires a dedicated consideration of various computational aspects. In the context of this work, the following aspects are particularly noteworthy:

Ghost-penalty and skeleton-penalty stabilization To avoid ill-conditioning associated with small volume-fraction trimmed cells, we apply ghost-penalty stabilization [79]. The idea behind this stabilization technique is to penalize the jump in the (higher-order) normal gradients of the solution along all edges in the ghost mesh

$$\mathcal{F}^h_{\text{ghost}} = \{\partial K \cap \partial K' | K, K' \in \mathcal{T}^h, K \cap \partial\Omega \neq \emptyset, K \neq K'\}, \tag{3.39}$$

by augmenting the bilinear form with an additional ghost-penalty term. This ghost-penalty term, which will be detailed for the specific problems considered in Section 3.5, also enables scaling of the Nitsche penalty term by the reciprocal mesh size parameter of the background mesh (independent of the trimmed-element configurations) [51].

For the flow problems considered in this work, we employ equal-order discretizations. To make the considered mixed velocity/pressure-discretizations inf-sup stable, we apply skeleton-stabilization [15] to the pressure space along all edges in the skeleton mesh

$$\mathcal{F}^h_{\text{skeleton}} = \{\partial K \cap \partial K' | K, K' \in \mathcal{T}^h, K \neq K'\}. \tag{3.40}$$

The skeleton-penalty term with which the bilinear form is augmented will be specified in Section 3.5.

Numerical integration on trimmed elements To evaluate integrals over the trimmed elements, we consider a recursive octree bisectioning strategy (see, *e.g.*, Ref. [52,58]), with the maximum number of bisections equal to $\varrho_{\max}$. On the lowest level of bisectioning, *i.e.*, $\varrho = \varrho_{\max}$, the midpoint tessellation procedure detailed in Ref. [86] is employed to construct an explicit parametrization of the trimmed boundary. An illustration of the octree bisectioning procedure with the midpoint tessellation is shown in Figure 3.8a.

Considering equal-order (Gauss) integration schemes on all sub-cells in the tessellated trimmed elements leads to high computational costs, in particular when three-dimensional simulations are considered [86]. Various methods have been proposed to reduce the computational cost, *e.g.*, smart octree methods [71],

moment fitting techniques [98], and error-estimate-based adaptive integration [86]. We herein consider the error-informed manual selection strategy proposed in Ref. [86]. At the coarsest level ($\varrho = 1$) in the octree tessellation we set the integration order to $k_{\max}$. We then decrease the order between two levels in such a way that the degree is zero (a single integration point) at the finest octree levels $\varrho \geq \varrho_{\max}$.

3.5 Immersed isogeometric analysis simulations

In this section we consider three applications of the topology-preserving image-based immersed isogeometric analysis technique presented above. In Section 3.5.1 we start with the case of a single-field problem in two dimensions by considering an elasticity problem. Subsequently, in Section 3.5.2 we consider a multi-field problem in the form of a Stokes flow through a carotid artery geometry. For this flow case we first consider a representative two-dimensional test case. The third application pertains to the extension of the Stokes flow case to a three-dimensional patient-specific geometry based on scan data. Let us note in advance that the carotid artery test case realistically pertains to moderate Reynolds number flows [178], but that a Stokes flow setting is here considered to focus on the topology-preserving analysis scheme developed in this work.

3.5.1 Uniaxial extension of a two-dimensional structure: a linear elasticity problem

We consider the two-dimensional specimen shown in Figure 3.19, which is represented by 32×32 grayscale voxels. The physical domain, Ω, with boundary $\partial\Omega$, is immersed into an ambient domain $\mathcal{A}$ of (dimensionless) size $L \times L$ (with $L = 1$), with boundary $\partial\mathcal{A}$; see Figure 3.19c (with $\varrho_{\max} = 3$). We consider a linear elasticity problem for which the displacement field, $\boldsymbol{u}$, is prescribed on the exterior (top and bottom) boundary, while the interior (immersed) boundary is traction free. In the absence of inertia effects and body forces, the boundary value problem reads as:

$$
\begin{cases}
\text{Find } \boldsymbol{u} \text{ such that:} & \\
\quad \operatorname{div}(\boldsymbol{\sigma}(\boldsymbol{u})) = \boldsymbol{0} & \text{in } \Omega \\
\quad \boldsymbol{u} = \boldsymbol{0} & \text{on } \partial\mathcal{A}_0 \\
\quad \boldsymbol{u} = \bar{u}\boldsymbol{n} & \text{on } \partial\mathcal{A}_{\bar{u}} \\
\quad \boldsymbol{u} \cdot \boldsymbol{n} = 0 & \text{on } \partial\mathcal{A} \setminus (\partial\mathcal{A}_0 \cup \partial\mathcal{A}_{\bar{u}}) \\
\quad [\boldsymbol{I} - \boldsymbol{n} \otimes \boldsymbol{n}]\,\boldsymbol{\sigma}\boldsymbol{n} = \boldsymbol{0} & \text{on } \partial\mathcal{A} \setminus (\partial\mathcal{A}_0 \cup \partial\mathcal{A}_{\bar{u}}) \\
\quad \boldsymbol{\sigma}\boldsymbol{n} = \boldsymbol{0} & \text{on } \partial\Omega \setminus \partial\mathcal{A}
\end{cases}
\tag{3.41}
$$

The various boundaries are specified in Figure 3.19b. The top boundary is displaced in normal direction by $\bar{u} = 0.2$ (20%). In the above problem definition, the stress is related to the strain by Hooke's law, *i.e.*, $\boldsymbol{\sigma}(\boldsymbol{u}) = \lambda \mathrm{div}(\boldsymbol{u})\boldsymbol{I} + 2\mu\nabla^s\boldsymbol{u}$, where ∇^s denotes the symmetric gradient operator. Throughout this section, the non-dimensionalized Lamé parameters are set to $\lambda = \frac{1}{2}$ and $\mu = \frac{1}{2}$. In our analyses, as quantities of interest we consider the stress state in the specimen and the effective elastic modulus

$$\mathcal{Q} = \frac{L}{\bar{u}} \frac{1}{V_{\mathrm{img}}} \int_{\Omega} \sigma_{22} \, \mathrm{d}V. \tag{3.42}$$

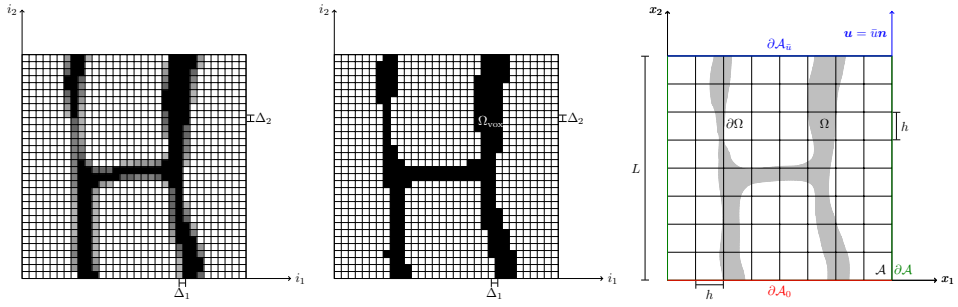
In our simulations, the Dirichlet conditions on the external boundary are applied strongly, *i.e.*, by constraining the degrees of freedom related to the boundary displacements. This is enabled by the fact that the top and bottom boundaries are mesh conforming. The Galerkin problem corresponding to (3.41) then follows as

$$\begin{cases} \text{Find } \boldsymbol{u}^h \in W^h(\Omega) \text{ such that for all } \boldsymbol{v}^h \in W_0^h(\Omega): \\ \displaystyle\int_{\Omega} \nabla^s \boldsymbol{v}^h : \boldsymbol{\sigma}(\boldsymbol{u}^h) \, \mathrm{d}V = 0 \end{cases} \tag{3.43}$$

with the discrete spaces being subsets of $H^1(\Omega)$ satisfying the Dirichlet boundary conditions. The spaces are discretized using second-order ($k = 2$) B-spline basis functions defined on a background mesh with uniform element size h. For this setting, immersed analysis results can be obtained without additional stabilization terms.

In Figure 3.20 we present the results using a mesh size of $h = L/32$, which is equal to the voxel size. As can be seen in Figure 3.20, without application of the topology-correction algorithm (first column), the left connection (marked in green) is not reconstructed by the image segmentation procedure. As a result, the left side of the specimen carries only a small portion of the load, in the sense that (the vertical component of) the stress is equal to zero in the left part connected to the top boundary, and relatively small in the left section that is connected to the bottom boundary. When the topology-preservation algorithm is applied (second column in Figure 3.20), the left part of the structure remains connected, and the left side of the specimen is appropriately loaded.

From Figure 3.20 it is observed that the topological anomaly drastically affects the simulation results. When considering the effective elastic modulus (3.42), as shown in Figure 3.21 for various mesh sizes, it is observed that this quantity of interest shows fundamentally different behavior between the two considered cases. Since the topological anomaly occurs independently of the background-mesh element size, mesh refinement for the determination of the approximate solution does not repair this problem. Both solutions with and without the



(a) **(b)** **(c)**

Figure 3.19: Illustration of (a) the original grayscale image, $g(\boldsymbol{x})$, (b) the voxel-segmentation of the image, i.e, $g(\boldsymbol{x}) > 0$, and (c) the computational domain (using $\varrho_{\max} = 3$) with the boundary conditions.

topological anomaly converge under mesh refinement, but the problem with the anomaly converges to an erroneous result on account of the incorrect geometry representation.

3.5.2 Flow through a carotid artery: a Stokes flow problem

We now consider Stokes flow through a domain $\Omega \subset \mathbb{R}^{n_d}$ $(n_d = 2, 3)$, representative of a carotid artery. This domain is constructed using the topology-preserving segmentation procedure presented in Section 3.3, and is immersed in an ambient domain $\mathcal{A}$. We consider a presssure-driven incompressible flow of a Newtonian fluid, with viscosity μ, through the carotid artery. The fluid velocity, $\boldsymbol{u}$, and pressure, p, satisfy the strong formulation

$$\begin{cases} \text{Find } \boldsymbol{u} \text{ and } p \text{ such that:} & \\ -\nabla \cdot (2\mu \nabla^s \boldsymbol{u}) + \nabla p = \boldsymbol{0} & \text{in } \Omega \\ \nabla \cdot \boldsymbol{u} = 0 & \text{in } \Omega \\ \boldsymbol{u} = \boldsymbol{0} & \text{on } \Gamma_D = \partial\Omega \setminus \partial\mathcal{A} \\ (2\mu \nabla^s \boldsymbol{u} - p\boldsymbol{I})\,\boldsymbol{n} = \boldsymbol{0} & \text{on } \partial\mathcal{A}_0 \\ (2\mu \nabla^s \boldsymbol{u} - p\boldsymbol{I})\,\boldsymbol{n} = -\bar{p}\boldsymbol{n} & \text{on } \partial\mathcal{A}_{\bar{p}} \end{cases} \tag{3.44}$$

where $\bar{p}$ denotes the pressure applied at the inflow (bottom) boundary.

The no-slip boundary condition on the immersed boundary Γ_D is imposed weakly through Nitsche's method. Ghost- and skeleton-stabilizations are used to avoid oscillations in the velocity and pressure fields (see Section 3.4). The mixed

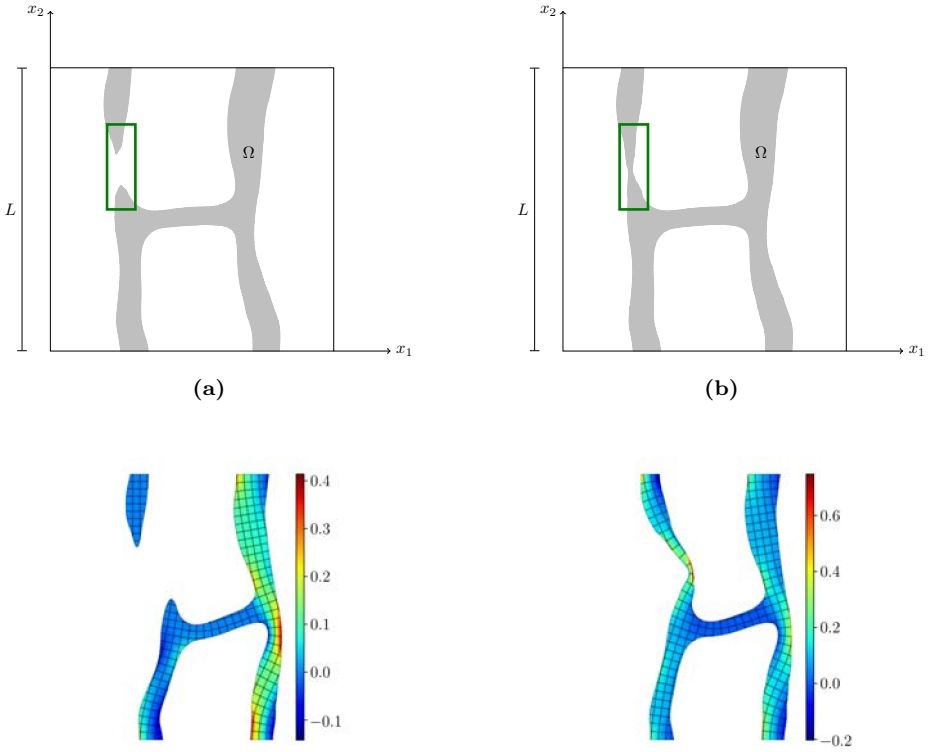


Figure 3.20: Comparison of the segmented geometry obtained from the grayscale image without (a) and with (b) topology preservation. The vertical (dimensionless) stress component for the two cases, computed using $h = L/32$, is shown in the panels (c) and (d).

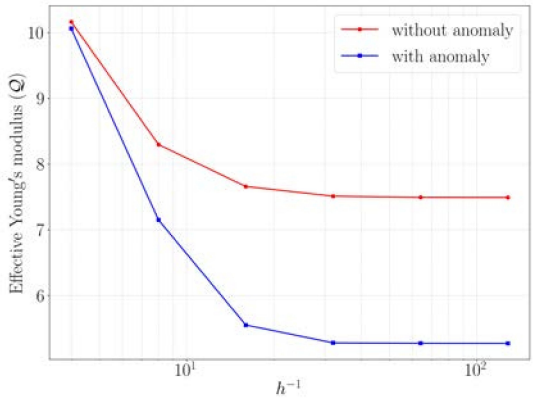


Figure 3.21: The effective elasticity modulus (3.42) computed with and without application of the topology-preservation technique for different mesh sizes h.

Galerkin form is given by

$$\begin{cases} \text{Find } \boldsymbol{u}^h \in \boldsymbol{V}^h \text{ and } p^h \in Q^h \text{ such that:} \\ a(\boldsymbol{u}^h, \boldsymbol{v}^h) + b(p^h, \boldsymbol{v}^h) + s_{\text{ghost}}(\boldsymbol{u}^h, \boldsymbol{v}^h) = l(\boldsymbol{v}^h) \quad \forall \boldsymbol{v}^h \in \boldsymbol{V}^h \\ b(q^h, \boldsymbol{u}^h) - s_{\text{skeleton}}(p^h, q^h) = 0 \qquad \forall q^h \in Q^h, \end{cases} \tag{3.45}$$

where the bilinear and linear operators are defined as [15]

$$\begin{aligned} a(\boldsymbol{u}^h, \boldsymbol{v}^h) &:= 2\mu(\nabla^s \boldsymbol{u}^h, \nabla^s \boldsymbol{v}^h) - 2\mu \left[\langle \nabla^s \boldsymbol{u}^h \cdot \boldsymbol{n}, \boldsymbol{v}^h \rangle_{\Gamma_D} \right. \\ &\quad \left. + \langle \nabla^s \boldsymbol{v}^h \cdot \boldsymbol{n}, \boldsymbol{u}^h \rangle_{\Gamma_D} \right] + \langle \mu \beta h^{-1} \boldsymbol{u}^h, \boldsymbol{v}^h \rangle_{\Gamma_D} && \text{(3.46a)} \\ b(p^h, \boldsymbol{v}^h) &:= -(p^h, \operatorname{div} \boldsymbol{v}^h) && \text{(3.46b)} \\ l(\boldsymbol{v}^h) &:= -\langle \bar{p}, \boldsymbol{v}^h \cdot \boldsymbol{n} \rangle_{\partial \mathcal{A}_{\bar{p}}} && \text{(3.46c)} \\ s_{\text{skeleton}}(p^h, q^h) &:= \sum_{F \in \mathcal{F}_{\text{skeleton}}} \int_F \gamma \mu^{-1} h^{2k+1} [\![\partial_n^k p^h]\!] [\![\partial_n^k q^h]\!] \, \mathrm{d}S && \text{(3.46d)} \\ s_{\text{ghost}}(\boldsymbol{u}^h, \boldsymbol{v}^h) &:= \sum_{F \in \mathcal{F}_{\text{ghost}}} \int_F \tilde{\gamma} \mu h^{2k-1} [\![\partial_n^k \boldsymbol{u}^h]\!] \cdot [\![\partial_n^k \boldsymbol{v}^h]\!] \, \mathrm{d}S, && \text{(3.46e)} \end{aligned}$$

and $(\cdot, \cdot)$ denotes the inner product in $L^2(\Omega)$, $\langle \cdot, \cdot \rangle_{\Gamma_D}$ denotes the inner product in $L^2(\Gamma_D)$, and $[\![\cdot]\!]$ is the jump operator. The parameters β, γ, and $\tilde{\gamma}$ denote the penalty constants for the Nitsche term, the Skeleton-stabilization term, and the Ghost-stabilization term, respectively. We consider second-order ($k = 2$)

B-splines constructed on a variety of uniform background meshes with element size h. Let us note that we use equal-order approximations for the velocity and pressure fields, which is admissible by virtue of the skeleton-penalization acting on the pressure field.

Two-dimensional test case

To demonstrate the developed methodology in the Stokes flow case, we first consider the idealized two-dimensional geometry shown in Figure 3.22 (constructed from 32×32 grayscale voxels). In this two-dimensional case, we consider the ambient domain to be a unit square (with $L = 1$), and we set the non-dimensionalized parameters to $\mu = 1$ and $\bar{p} = 1$; see Figure 3.22c (constructed with $\varrho_{\max} = 3$). For the penalty parameters we take, $\beta = 100$, $\gamma = 0.05$, and $\tilde{\gamma} = 0.0005$, which have been determined empirically.

Figure 3.23 shows the pressure and velocity contour plots for the case where a topological anomaly occurs in the form of a pinched-off channel (top row), and in the case where the topology-preservation algorithm is applied (bottom row). The topological anomaly evidently obstructs fluid from flowing through the left branch, resulting in a zero pressure and zero fluid velocity solution in the top left disconnected domain. The topology-correction strategy proposed in this work avoids the left channel from being closed and results in a different flow profile. It is noteworthy that, if Dirichlet conditions were imposed on the top boundary, the pathological case without the topology-correction would have become singular.

The influence of the mesh size is studied in Figure 3.24, which, similar to the elasticity problem considered above, conveys that both simulation cases converge under mesh refinement. However, without application of the topology-correction strategy, the solution converges to an erroneous result.

Three-dimensional scan-based simulation

To demonstrate the developed methodology in a real scan-based setting, we consider a Stokes flow through a carotid artery. The geometry of the carotid artery is obtained from a CT-scan (see Figure 3.25). The scan data consists of 80 slices of 85×70 voxels. The size of the voxels is $300\times300\,\mu\text{m}^2$ and the distance between the slices is $400\,\mu$m. Hence, the total size of the scan domain is $25.6\times21.1\times32.0\,\text{mm}^3$. The original grayscale data, represented in DICOM format [178], is preprocessed using ITK-SNAP (open-source medical image processing tool [179]) from which binary voxel data is exported and read into our Python-based implementation of the developed topology-correction strategy.

The direct, non-smooth, segmentation of the image data is visualized in Figure 3.25a. When applying the spline-based segmentation procedure using second-order B-splines, the stenotic part of the artery in the original voxel image is lost

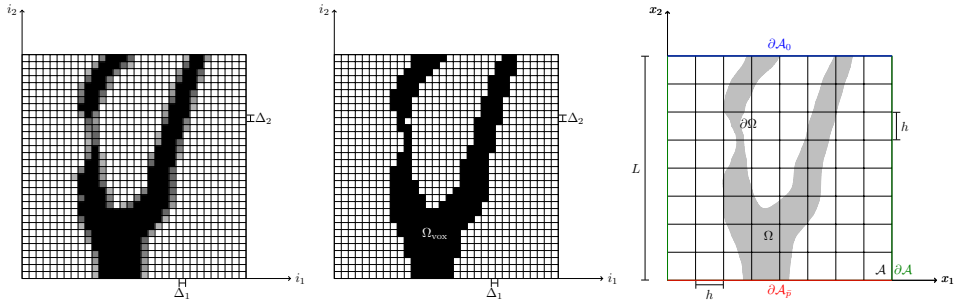


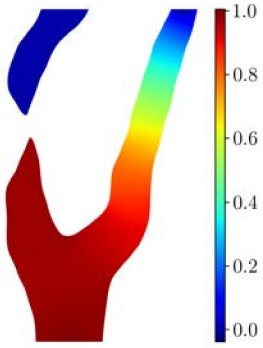
Figure 3.22: Illustration of (a) the original grayscale image, $g(\boldsymbol{x})$, (b) the voxel-segmentation of the image, *i.e*, $g(\boldsymbol{x}) > 0$, and (c) the computational domain (using $\varrho_{\max} = 3$) with the boundary conditions.

(Figure 3.25b). As discussed in Section 3.2.2, this topological anomaly resulting from the spline-based segmentation procedure is expected on account of the feature-to-mesh-size ratio in that particular area.

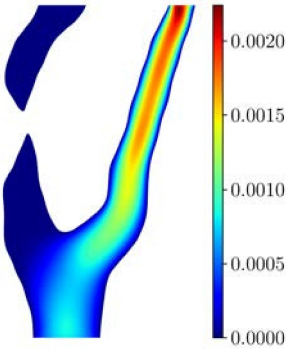
In Figure 3.26 a zoom of the stenotic part of the artery is shown. Figure 3.26c shows the indicator function (3.21) as determined by the topology-correction strategy. This image conveys that the topological anomaly in the form of the missing stenotic part of the artery is appropriately detected. Figure 3.26d shows the smoothly segmented geometry after THB-spline refinement. From this figure it is observed that the topological anomaly is corrected by the proposed strategy. In the case of the complete topology (see Figure 3.25c), it is observed that additional boundary regions are tagged for refinement on account of the high-curvature of the boundary surface in these regions.

To simulate the flow through the stenotic artery we consider a viscosity of $\mu = 4\,\text{mPa}\,\text{s}$ and a pressure drop of 17.3 kPa (130 mm of Hg). These parameters are selected based on Ref. [178]. The penalty parameters associated with the weak formulation (3.45) are set to $\beta = 100$, $\gamma = 0.05$, and $\tilde{\gamma} = 0.0005$, which have been determined empirically to yield a stable formulation without adversely affecting the accuracy of the approximation.

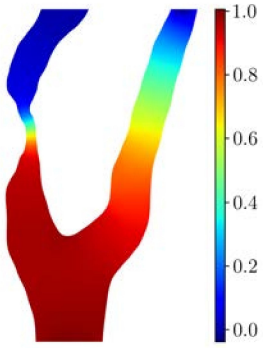
The results for the velocity and pressure fields computed on a uniform mesh with $h = 1.75\,\text{mm}$ in the directions perpendicular to the pressure gradient and $h = 2\,\text{mm}$ in the direction of the pressure gradient are shown in Figure 3.27. Similar to the two-dimensional case, the topological anomaly evidently obstructs fluid from flowing through the stenotic part of the artery, resulting in a zero pressure and zero fluid velocity solution in the right artery (see Figure 3.27). The topology-correction strategy avoids the stenotic part from being closed and results in a



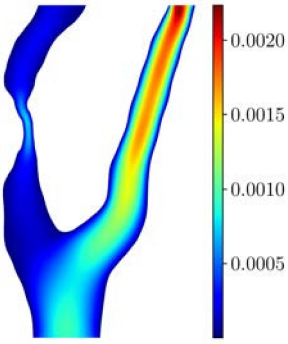
(a) Pressure



(b) Velocity magnitude



(c) Pressure



(d) Velocity magnitude

Figure 3.23: Comparison of the pressure, p, and velocity magnitude, $|\boldsymbol{u}|$, for the segmented domain constructed without (top row) and with (bottom row) topology-preservation using $h = L/64$.

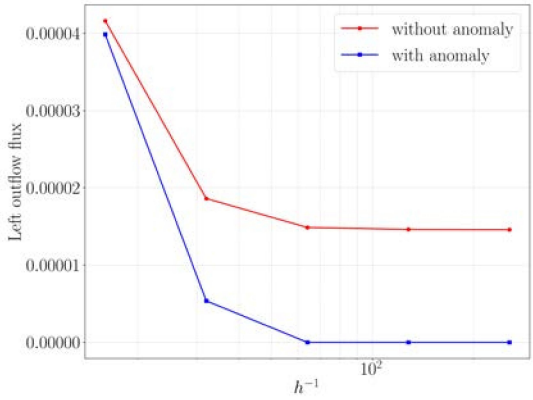
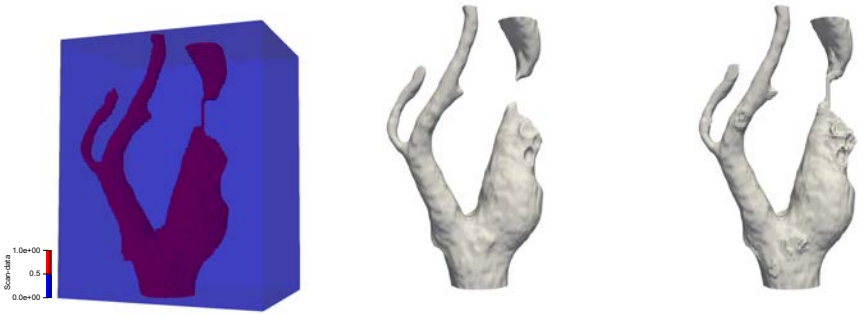
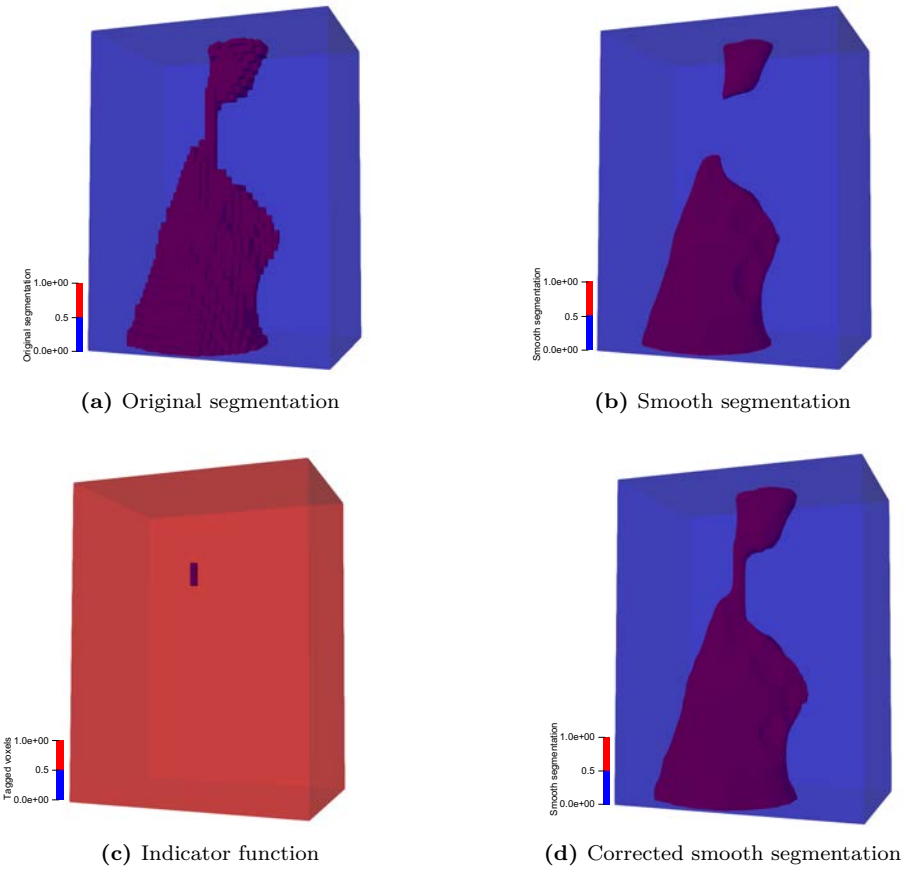


Figure 3.24: Total outflow from the left branch of the carotid artery, computed using different mesh sizes h, with and without the use of the topology-correction strategy.



(a) **(b)** **(c)**

Figure 3.25: Illustration of (a) the directly segmented scan-data, $g(\boldsymbol{x})$. The computational domain of the stenotic carotid artery extracted by the B-spline-based segmentation procedure (b) before and (c) after application of the topology preservation algorithm. The smooth level set function, $f(\boldsymbol{x})$, is segmented using the midpoint tessellation procedure with a subdivision level of $\varrho_{\max} = 2$.



(a) Original segmentation

(b) Smooth segmentation

(c) Indicator function

(d) Corrected smooth segmentation

Figure 3.26: Illustration of the topology-preserving procedure focused on the stenotic part of the carotid artery. Panel (a) shows the original segmentation obtained directly by thresholding the voxel data, and panel (b) shows the segmentation through the B-spline-based smoothing strategy. A moving-window technique then locally compares the topology between the two segmentations, which results in the indicator function (c) that marks topological differences. THB-spline-based refinements are then introduced to locally increase the resolution of the smooth level set function, thereby preserving the topology of the original scan-data (d).

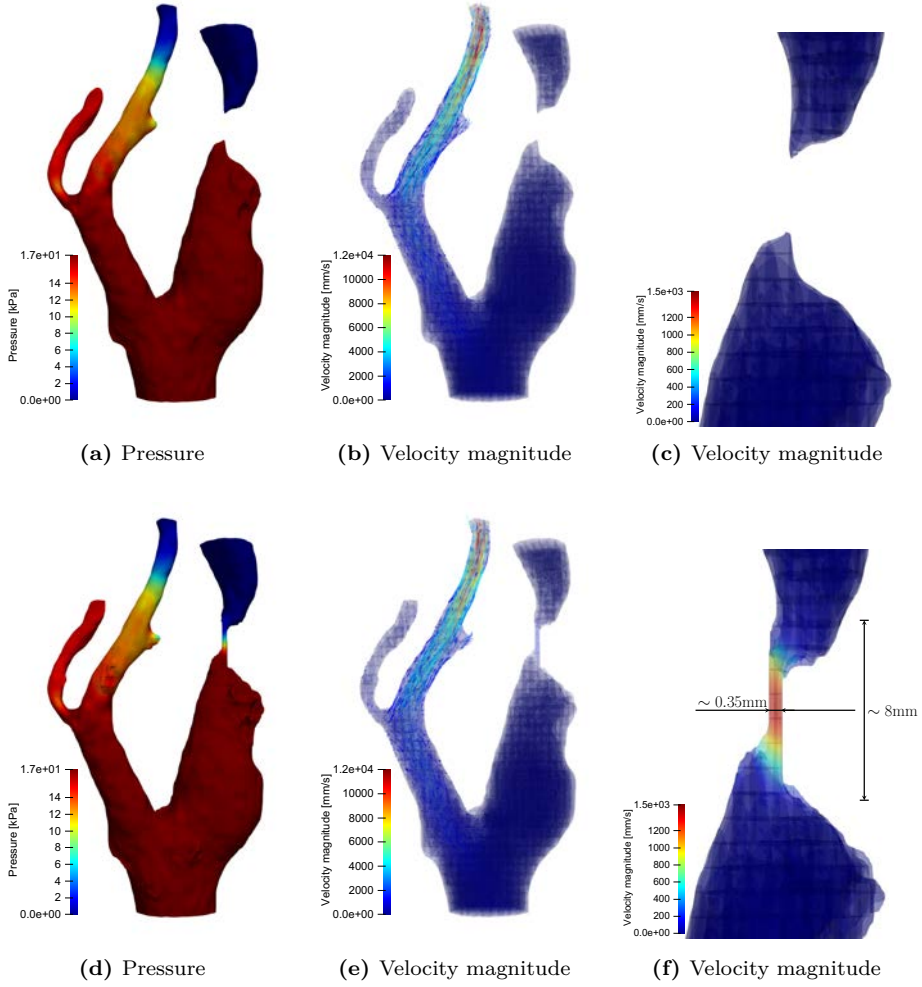
meaningful flow profile. Although the mesh considered here is relatively coarse, the flux through the stenotic region of approximately $200\,\mathrm{mm}^3/\mathrm{s}$ (corresponding to an average velocity of approximately $1.3\,\mathrm{m/s}$, see Figure 3.27f) corresponds reasonably well with the flow through a circular tube (Hagen–Poiseuille flow) of length 8 mm and diameter 0.35 mm subject to the above-mentioned pressure drop, which corroborates that the computed speed is meaningful.

In Figure 3.28 the outflow through the stenotic branch of the artery is depicted for various uniform meshes, ranging from a very coarse mesh with 8,760 (active) degrees of freedom to a refined mesh with 96,196 degrees of freedom. Similar to the problems studied above, this mesh verification study conveys that both simulation cases converge under mesh refinement, but that an erroneous result is obtained in the case that the topology is not corrected. Note that, due to the employed uniform meshes, the number of degrees of freedom increases rapidly under mesh refinement, which will limit the size of the domain that can be considered by this type of analysis in practice. It is important to note, however, that mesh refinements throughout most of the domain do not substantially improve the accuracy of the simulation (in particular for the considered quantity of interest). Therefore, to properly leverage the property of immersed techniques that the mesh resolution can be controlled independently of the geometry (and topology) representation, significant improvements in computational efficiency can be obtained by means of adaptive local mesh refinement. The combination of the proposed technique with an error-estimation-and-adaptivity strategy is an important topic of further study.

3.6 Concluding remarks

To leverage the advantageous properties of isogeometric analysis in a scan-based setting, a smooth representation of the computational domain must be obtained. This can be achieved by applying a smoothing operation on the voxel-based gray-scale data and subsequently applying an octree-based tessellation procedure. A negative side-effect of this smoothing procedure is that it can induce topological changes when the scan data contains features with a characteristic length scale similar to the voxel size.

Based on a Fourier analysis of the B-spline-based smoothing operation, it is proposed to repair smoothing-induced topological anomalies by locally refining the smoothed gray-scale function using THB-splines. In combination with a moving-window strategy to detect topological changes, the local refinement technique is used to develop a topology-preserving image segmentation technique. Based on a comparison of the Euler characteristic between the window view on the original voxel data and that on the smoothed representation, the proposed technique systematically distinguishes shape changes from topological changes.



(a) Pressure **(b)** Velocity magnitude **(c)** Velocity magnitude

(d) Pressure **(e)** Velocity magnitude **(f)** Velocity magnitude

Figure 3.27: Comparison of the pressure, p, and velocity magnitude, $|\boldsymbol{u}|$, for the segmented domain constructed without (top row) and with (bottom row) topology-preservation using a mesh size of $h = 2$ mm in the vertical direction, and $h = 1.75\,\text{mm}$ in the other directions.

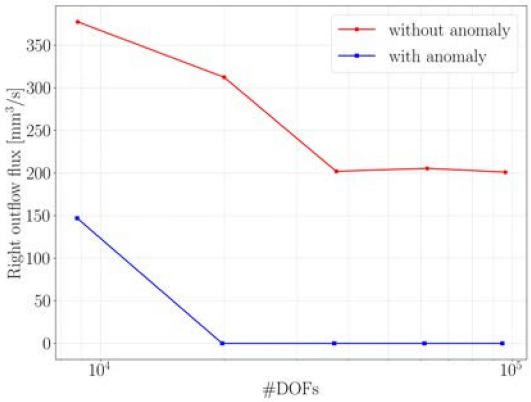


Figure 3.28: Total outflow from the right branch of the carotid artery, computed using different degrees of freedom (DOFs), with and without the use of the topology-correction strategy.

The algorithm is fail-safe in that it detects and repairs topological changes, and does not essentially change the geometry in the (rare) case that a shape change is accidentally marked for refinement. The proposed location-based masking strategy to detect shape changes is very effective for the considered test cases, but it is envisioned that further robustness improvements can be made by the development of a more advanced masking procedure.

The developed algorithm works for two- and three-dimensional scan data. Numerical simulations demonstrate the effectivity of the algorithm in both settings. For all considered test cases, a topologically consistent smoothed image is obtained after a single topology-correction step. Based on the presented Fourier analysis this is to be expected, as refining the mesh for the B-spline grayscale function has a strong effect on the filtering properties. It can, in principle, occur that topologial changes are not repaired after a single correction step. Although not considered in this work, the presented algorithm has the potential to be extended so that it can be applied recursively in such scenarios.

In this work we have restricted ourselves to immersed isogeometric analyses based on uniform meshes. In order to optimally benefit from the fact that the computaional mesh is decoupled from the segmented geometry in the immersed setting, use should be made of (adaptive) local refinements for the analysis mesh. Combination of the proposed topology-preservation technique with an adaptive meshing strategy is therefore an important topic of further study.

Chapter 4

Residual-based error estimation and adaptivity for stabilized immersed isogeometric analysis using truncated hierarchical B-splines

In this chapter, we propose an adaptive mesh refinement strategy for immersed isogeometric analysis, with application to steady heat conduction and viscous flow problems. The proposed strategy is based on residual-based error estimation, which has been tailored to the immersed setting by the incorporation of appropriately scaled stabilization and boundary terms. Element-wise error indicators are elaborated for the Laplace and Stokes problems, and a THB-spline-based local mesh refinement strategy is proposed. The error estimation and adaptivity procedure is applied to a series of benchmark problems, demonstrating the suitability of the technique for a range of smooth and non-smooth problems. The

This chapter is reproduced from [88]: S.C. Divi, P.H. van Zuijlen, T. Hoang, F. de Prenter, F. Auricchio, A. Reali, E.H. van Brummelen and C.V. Verhoosel, Residual-based error estimation and adaptivity for stabilized immersed isogeometric analysis using truncated hierarchical B-splines. *arXiv preprint*, submitted, 2022. The (co-)promotors confirm that S.C. Divi is the primary author of this publication, i.e., she was responsible primarily for the planning, execution and preparation of the work.

adaptivity strategy is also integrated in a scan-based analysis workflow, capable of generating reliable, error-controlled, results from scan data, without the need for extensive user interactions or interventions.

4.1 Introduction

Immersed finite element methods, such as the finite cell method [14, 48, 49] and CutFEM [50, 51, 83], are a natural companion to isogeometric analysis [16, 110]. The combination of immersed methods with the spline-based discretization strategy provided by the isogeometric analysis paradigm is recognized as a valuable extension of isogeometric analysis, because the immersed analysis concept provides a cogent framework for the consideration of trimmed CAD objects [37, 52–54, 81, 89]. Moreover, immersed isogeometric analysis enables the construction of spline-based discretization spaces for geometrically and topologically complex volumetric domains [52, 55, 89], a simulation strategy referred to as immersogeometric analysis [56, 57].

In comparison to boundary-fitting isogeometric analysis, the immersed isogeometric analysis strategy requires consideration of three (categories of) non-standard computational aspects. First, the geometry of elements that intersect with the boundary of the computational domain must be resolved by a dedicated integration procedure; see *e.g.*, [71, 73, 75, 86, 98, 125]. Second, Dirichlet boundary conditions on immersed boundaries can generally not be imposed through basis function constraints. Instead, such boundary conditions are frequently imposed weakly; see, *e.g.*, [50, 77, 78, 180]. Third, unfavorably trimmed elements are notorious for causing ill-conditioning problems and, along Dirichlet boundaries, large unphysical gradients [14, 53, 76, 79, 80, 84, 181]. This problem is amplified in the higher-order discretization setting of isogeometric analysis [80]. Prominent computational remedies to overcome these problems are to supplement the weak formulation with stabilization terms, see, *e.g.*, [51, 79, 83], or to constrain, extend, or aggregate of basis functions, see *e.g.*, [81, 85, 182–187], or to apply dedicated preconditioning techniques, *e.g.*, [62, 80, 82].

For mixed formulations, such as standard weak forms of the Stokes and Navier-Stokes equations, the immersed isogeometric analysis setting imposes an additional challenge. In order to satisfy the inf-sup condition [188,189] in boundary-fitting (isogeometric) analyses, generally use is made of stable pairs of basis functions (*e.g.*, Taylor-Hood [24, 190–192] or Raviart-Thomas [191, 193–195]). Alternatively, stabilization techniques such as GLS [196–198], VMS [199–201] or projection methods [202, 203] can be used. Direct utilization of these elements or stabilization techniques in the immersed setting can lead to non-physical spurious oscillations in the solution, even with relatively large and regular cut element configurations [15,204]. One remedy for tackling this issue is to employ a skeleton-

stabilized immersed isogeometric technique [15]. The fundamental idea of this stabilization technique is to penalize (high-order) pressure derivative jumps over the edges/faces of the background mesh, resulting in stable discretizations using equal-order spline spaces. The technique proposed in Ref. [15] is inspired by the (continuous) interior penalty ((C)IP) and the ghost penalty (GP) methods [79], extending these techniques to the case of high-regularity isogeometric analysis.

An appraised property of immersed methods in general, and immersed isogeometric analysis in particular, is that the discretization resolution can be controlled independently of the geometry parametrization. The immersed analysis concept avoids the need for geometry-induced mesh refinements in the vicinity of geometric details that are irrelevant in relation to the objective of an analysis. This decoupling of the discretization resolution from the geometry makes it natural to consider immersed finite elements in combination with adaptive discretization strategies. In fact, adaptivity in the form of local p- and $hp-$refinements has always been an integral part of the finite cell method [65, 118, 205, 206].

A posteriori error estimation and adaptivity techniques are well-established in the context of finite element methods; see, *e.g.*, the reviews [207–209]. A variety of error estimation and adaptivity techniques has been studied in isogeometric analysis, such as residual-based error estimators for T-splines [210] and hierarchical splines [158, 211, 212], and goal-oriented techniques [213]. The contemporary overview [214] is also noteworthy, as is the advanced industrial application considered in Ref. [215]. In the context of Nitsche-based finite element methods (see Refs. [216, 217] for an overview), studies on a posteriori error estimators have been conducted [50, 218–221]. Local refinement strategies in immersed method are predominantly feature based, *i.e.*, either based on geometric features such as boundaries, or based on solution features such as sharp gradients in the solution fields; see, *e.g.*, [89, 222, 223] for examples of local refinement capabilities in finite cell simulations. Goal-oriented error estimation and adaptivity for immersed methods has also been studied [58, 213, 224, 225]. In the context of stabilized immersed finite elements, Ref. [226] considered a posteriori element-wise error estimation and adaptivity to improve boundary approximations.

Although the computational setting of immersed isogeometric analysis enables the use of volumetric spline patches, the standard h, p and k-type refinement strategies in patch-based isogeometric analysis [16] are not suitable because of the non-local propagation of refinements. Various alternative refinement strategies have been proposed over the last decade to construct local spline refinements, the most prominent of which are T-splines [57, 89, 227–231], LRB-splines [232, 233], U-splines [234], and (Truncated) Hierarchical B-splines [176]. In the context of immersed isogeometric analysis on volumetric domains, hierarchical splines are particularly suitable, as they optimally leverage the advantages offered by the geometrically simple background mesh.

In this contribution we propose a computational strategy for the application of residual-based a-posteriori error estimation and mesh adaptivity to stabilized immersed isogeometric analyses. We study various computational aspects of the framework that are non-standard in comparison to error estimation and adaptivity for boundary-fitting analyses, *viz.*: *(i)* In immersed analyses, the discretization basis is constructed over a mesh comprised of all elements in an ambient mesh that intersect with the computational domain. As a direct consequence of this setting, the support of the computational basis in general changes under refinement operations. The same holds for the mesh skeleton, which is a key ingredient of the considered stabilization methods. The considered computational strategy preserves the geometry of the computational domain under local mesh refinements, despite the change of the background mesh; *(ii)* Weak formulations in stabilized immersed isogeometric analysis generally involve operators with an explicit dependence on the mesh size. While this mesh size is unambiguously defined in the case of a uniform background mesh, the local mesh refinements considered in the adaptive setting warrant careful consideration of the scaling of the stabilization terms. We herein propose and study a scaling of the stabilization terms based on the local element sizes.

We demonstrate the performance of the proposed computational strategy using a series of test cases for steady heat conduction problems (Poisson problem) and steady viscous flow problems (Stokes problem). We consider the application of the proposed adaptivity technique in a scan-based isogeometric analysis setting, and demonstrate that a robust automatic simulation workflow is realized when the methodology presented herein is combined with the topology-preserving image segmentation algorithm presented in Ref. [86].

This paper is outlined as follows. Section 4.2 introduces the immersed isogeometric analysis framework, along with a detailed stability analysis for the considered model problems. This analysis focuses particularly on the scaling relations for the stabilization terms. In Section 4.3 the residual-based error estimator is introduced, and a mesh-refinement strategy is proposed. Benchmark simulation results are then presented in Section 4.4 for both the steady heat conduction problem and the viscous flow problem, after which the developed framework is applied in a scan-based setting in Section 4.5. Conclusions are finally drawn in Section 4.6.

4.2 Stabilized immersogeometric analysis with local mesh refinements

In this section we introduce the stabilized immersed isogeometric analysis formulations for the steady heat conduction (Laplace) problem and steady viscous

flow (Stokes) problem. We commence with presenting the general setting of the problems in Section 4.2.1, after which the stabilized formulations are presented in Section 4.2.2. In preparation of the *a posteriori* error estimation concept discussed in Section 4.3, in Section 4.2.3 we study the stability of the considered formulations.

4.2.1 The finite cell setting

We consider a physical domain $\Omega \in \mathbb{R}^d$ (with $d \in \{2, 3\}$) with boundary $\partial\Omega$, as illustrated in Figure 4.1. The boundary is composed of a Neumann part, $\partial\Omega_N$, and a Dirichlet part, $\partial\Omega_D$, such that $\overline{\partial\Omega_N} \cup \overline{\partial\Omega_D} = \partial\Omega$ and $\partial\Omega_N \cap \partial\Omega_D = \emptyset$. The outward-pointing unit normal vector to the boundary is denoted by $\mathbf{n}$.

The physical domain is immersed in a geometrically simple ambient domain, *i.e.*, $\mathcal{A} \supset \Omega$, on which a locally refined ambient mesh $\mathcal{T}_\mathcal{A}$ with elements K is defined. In this work, the ambient domain is chosen to be rectangular or cuboid, to facilitate simple, tensor-product, spline discretizations. The locally-refined meshes are constructed by sequential bisectioning of (selections of) elements in the mesh, starting from a Cartesian mesh. Truncated hierarchical B-splines can be formed on such meshes, as will be elaborated in Section 4.2.2.

Elements that do not intersect with the physical domain can be omitted from the ambient mesh, resulting in the locally refined (active) background mesh

$$\mathcal{T} := \{K \,|\, K \in \mathcal{T}_\mathcal{A}, K \cap \Omega \neq \emptyset\}. \tag{4.1}$$

In the remainder, with the abuse of notation, we will use $\mathcal{T}$ (and other meshes) to denote both the set of elements in the mesh and the geometry obtained from the union of these elements. The local mesh size of the locally refined background mesh is denoted by

$$h_\mathcal{T} : K \to h_K = \sqrt[d]{\text{meas}(K)}. \tag{4.2}$$

By cutting the elements that are intersected by the immersed boundary $\partial\Omega$, a mesh that conforms to the physical domain Ω is obtained:

$$\mathcal{T}_\Omega := \{K \cap \Omega \,|\, K \in \mathcal{T}\} \tag{4.3}$$

The collection of elements in the background mesh that are crossed by the immersed boundary $\partial\Omega$ is defined as

$$\mathcal{G} := \{K \in \mathcal{T} \mid K \cap \partial\Omega \neq \emptyset\}. \tag{4.4}$$

In immersed methods, the geometry of the physical domain is captured by the integration procedure on the cut elements, *i.e.*, elements that are intersected by

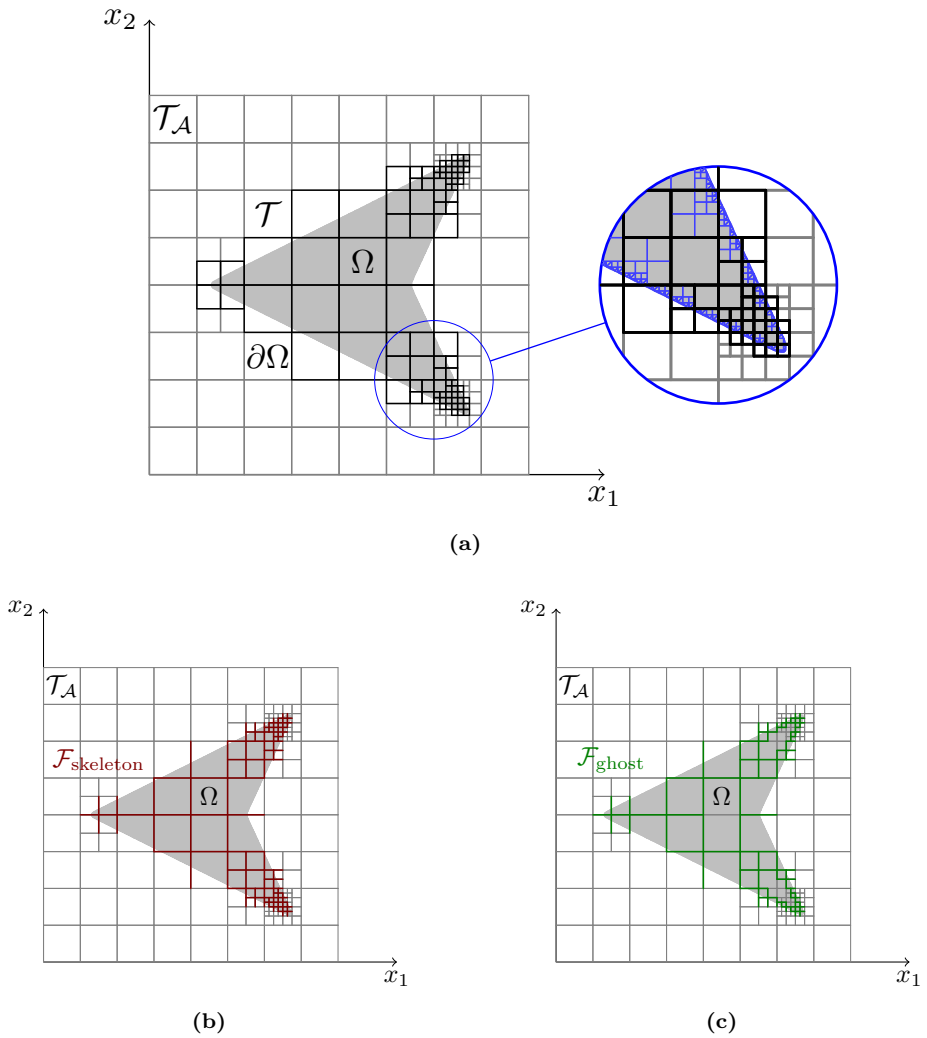


Figure 4.1: (a) A physical domain Ω, with boundary $\partial\Omega$, is embedded in an ambient domain $\mathcal{A}$. The background mesh $\mathcal{T}$, which consists of all elements that intersect the physical domain, is constructed by locally refining the ambient domain mesh $\mathcal{T}_\mathcal{A}$. The zoom illustrates the employed bisectioning procedure to capture the immersed boundaries. The integration subcells are marked in blue, whereas the background cells are marked in black. The skeleton mesh, $\mathcal{F}_{\text{skeleton}}$, and ghost mesh, $\mathcal{F}_{\text{ghost}}$, are shown in panels (b) and (c), respectively.

the immersed boundary $\partial\Omega$. We herein employ an octree integration procedure [58, 86], which we close at the lowest level of bisectioning with a tessellation procedure. The considered integration procedure is illustrated in Figure 4.1 (in blue) for a typical cut element; see Ref. [86] for further details. The employed tessellation provides an explicit parametrization of a polygonal approximation of the immersed boundary $\partial\Omega$ through the set of boundary faces

$$\mathcal{T}_{\partial\Omega} := \{E \subset \partial\Omega \,|\, E = \partial K \cap \partial\Omega,\, K \in \mathcal{T}_\Omega\}. \tag{4.5}$$

All faces $E \subset \partial\Omega_N$ (respectively $E \subset \partial\Omega_D$) are assigned to a set of Neumann faces $\mathcal{T}_{\partial\Omega_N}$ (respectively Dirichlet faces $\mathcal{T}_{\partial\Omega_D}$). In general, a single polygon face can overlap with both the Neumann and the Dirichlet boundary. Let us note that in an adaptive refinement procedure, the refinements can serve to provide an increasingly accurate approximation of the transition between the Neumann and Dirichlet boundary.

The formulations considered in the remainder of this work incorporate stabilization terms formulated on the edges of the background mesh (see Section 4.2.2), which we refer to as the skeleton mesh

$$\mathcal{F}_{\text{skeleton}} = \{\partial K \cap \partial K' \mid K, K' \in \mathcal{T}, K \neq K'\}. \tag{4.6}$$

Note that the boundary of the background mesh is not part of the skeleton mesh. In addition to the skeleton mesh, we define the ghost mesh as the subset of the skeleton mesh that contain a face of an element intersected by the domain boundary

$$\mathcal{F}_{\text{ghost}} = \{\partial K \cap \partial K' \,|\, K \in \mathcal{G}, K' \in \mathcal{T}, K \neq K'\}. \tag{4.7}$$

As will be detailed in Section 4.2.3, the stabilization terms formed on the skeleton and ghost mesh account for stability and ill-conditioning effects related to unfavorably cut elements, as well as for preventing pressure oscillations in equal-order discretizations of the Stokes problem.

4.2.2 Immersogeometric analysis

We consider the immersogeometric analysis of a single-field steady heat-conduction problem and of a two-field viscous flow problem. Both problems are represented by the abstract Galerkin problem

$$\begin{cases} \text{Find } u^h \in U^h \text{ such that:} \\ a^h(u^h, v^h) = b^h(v^h) \qquad\qquad \forall v^h \in V^h, \end{cases} \tag{4.8}$$

with mesh-dependent bilinear and linear forms, a^h and b^h, respectively. Note that the superscript h is used to indicate mesh-dependence. The finite dimensional

trial and test spaces, U^h and V^h, are spanned by truncated hierarchical B-spline (THB-splines) [158,176] basis functions of degree k and regularity α constructed over the locally-refined background mesh, *viz.*

$$\mathcal{S}_\alpha^k(\mathcal{T}) = \{N \in C^\alpha(\mathcal{T}) : N|_K \in P^k(K), \forall K \in \mathcal{T}\}, \tag{4.9}$$

with $P^k(K)$ the set of d-variate polynomials on the element K constructed by the tensor-product of univariate polynomials of order k. Truncated hierarchical B-splines, which are illustrated in Figure 4.2, form a partition of unity and have a reduced support compared to their non-truncated counterpart, which is advantageous from the perspective of system matrix sparsity. Our implementation is based on the open source finite element library Nutils [109], which provides support for THB-splines.

Since the imposition of strong Dirichlet boundary conditions over the immersed boundary $\partial\Omega$ is intractable in the immersogeometric analysis setting, such boundary conditions are imposed weakly through Nitsche's method; see, *e.g.*, Ref. [78]. A mesh-dependent consistent stabilization term is introduced in order to ensure the well-posedness of the Galerkin problem (4.8).

Steady heat conduction

Steady heat conduction is governed by the Poisson problem, which, in dimensionless form, can be formulated as

$$\begin{cases} \text{Find } u \text{ such that:} \\ -\Delta u = f \text{ in } \Omega, \\ \quad u = g \text{ on } \partial\Omega_D, \\ \partial_n u = q \text{ on } \partial\Omega_N, \end{cases} \tag{4.10}$$

where u is the scalar temperature field, f is a heat source term, q represents the prescribed heat flux on the Neumann boundary, and g is the prescribed temperature on the Dirichlet boundary. The normal gradient is defined as $\partial_n u = \nabla u \cdot \mathbf{n}$.

The discretized solution to the strong formulation (4.10) with the Dirichlet conditions enforced by Nitsche's method is denoted by

$$u^h \in U^h = \mathcal{S}_\alpha^k(\mathcal{T}) \subset H^1(\mathcal{T}),$$

with the corresponding test functions given by $v^h \in V^h = U^h$. We herein consider maximum regularity B-splines, *i.e.*, $\alpha = k - 1$. The bilinear and linear forms in

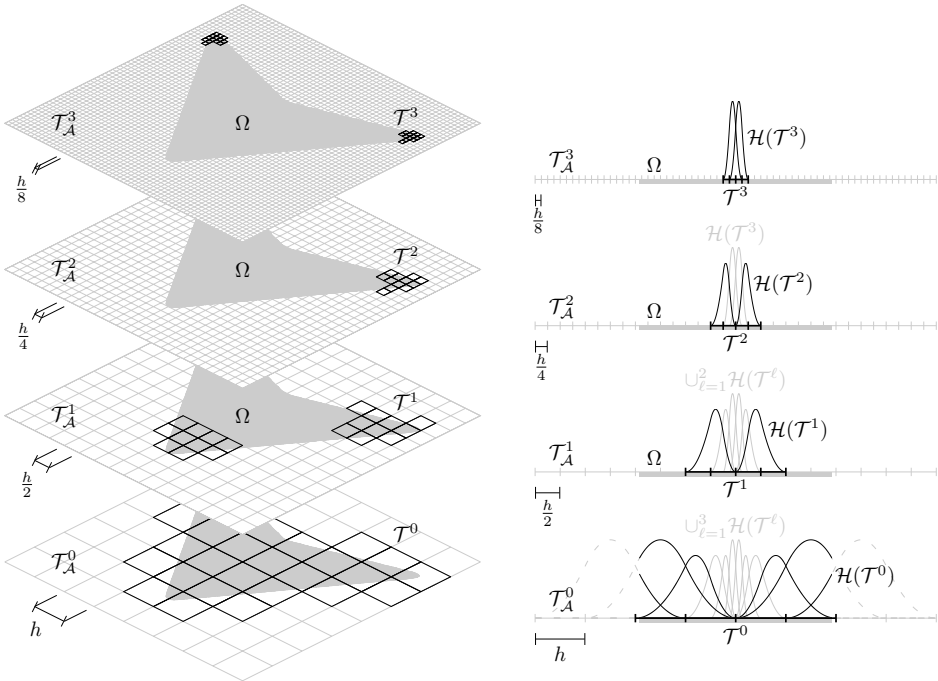


Figure 4.2: Illustration of truncated hierarchical B-splines [158,176] in the immersogeometric analysis setting. The left column shows the hierarchical levels of the mesh $\mathcal{T}$ in Figure 4.1, while the right column illustrates the concept for a one-dimensional immersed domain Ω. The background mesh at the level $\ell = 0, \cdots, \ell_{\max}$ (with $\ell_{\max} = 3$ in this illustration) is defined as $\mathcal{T}^\ell = \{K \in \mathcal{T}^\ell_{\mathcal{A}} \mid K \cap \Omega \neq \emptyset\}$ where $\mathcal{T}^\ell_{\mathcal{A}}$ is a regular mesh with mesh size parameter $2^{-\ell}h$. Note that the meshes are nested, in the sense that the domain covered by the physical mesh at level ℓ, $\mathcal{T}^\ell$, is completely inside that of level $\ell-1$, $\mathcal{T}^{\ell-1}$, *i.e.*, $\mathcal{T}^\ell \subseteq \mathcal{T}^{\ell-1}$. The THB-spline basis, $\mathcal{H}(\mathcal{T})$, is constructed by selection and truncation of the basis functions in the B-spline basis $\mathcal{B}(\mathcal{T}^\ell) = \{N \in \mathcal{B}(\mathcal{T}^\ell_{\mathcal{A}}) \mid \mathrm{supp}\,(N) \subseteq \mathcal{T}^\ell\}$ defined at each level. At the most refined level, *i.e.*, at $\ell = \ell_{\max}$, all basis functions that are completely inside $\mathcal{T}^{\ell_{\max}}$ are selected: $\mathcal{H}(\mathcal{T}^{\ell_{\max}}) = \{N \in \mathcal{B}(\mathcal{T}^{\ell_{\max}}) \mid \mathrm{supp}\,(N) \subseteq \mathcal{T}^{\ell_{\max}}\}$. At the coarser levels, *i.e.*, $\ell < \ell_{\max}$, the functions that are completely inside the domain $\mathcal{T}^\ell$ but not completely inside the refined domain $\mathcal{T}^{\ell+1}$ are selected and truncated: $\mathcal{H}(\mathcal{T}^\ell) = \{\mathrm{trunc}(N) \mid N \in \mathcal{B}(\mathcal{T}^\ell), \mathrm{supp}\,(N) \not\subseteq \mathcal{T}^{\ell+1}\}$. The truncation operation reduces the support of the B-spline functions by projecting away basis functions retained from the refined levels. The THB-spline basis then follows as $\mathcal{H}(\mathcal{T}) = \cup_{\ell=0}^{\ell_{\max}}\mathcal{H}(\mathcal{T}^\ell)$. The reader is referred to Ref. [158] for details of THB-spline basis and Ref. [176] for THB-spline basis construction.

equation (4.8) are

$$a^h(u^h, v^h) = \int_\Omega \nabla u^h \cdot \nabla v^h \,\mathrm{d}V - \int_{\partial\Omega_D} \left((\partial_n u^h) v^h + u^h (\partial_n v)\right) \mathrm{d}S + \sum_{E \in \mathcal{T}_{\partial\Omega_D}} \int_E \tilde{\beta} u^h v^h \,\mathrm{d}S + \sum_{F \in \mathcal{F}_{\text{ghost}}} \int_F \tilde{\gamma}_g [\![\partial_n^k u^h]\!] [\![\partial_n^k v]\!] \,\mathrm{d}S, \tag{4.11a}$$

$$b^h(v^h) = \int_\Omega f v^h \,\mathrm{d}V + \int_{\partial\Omega_N} q v^h \,\mathrm{d}S - \int_{\partial\Omega_D} g \partial_n v^h \,\mathrm{d}S + \sum_{E \in \mathcal{T}_{\partial\Omega_D}} \int_E \tilde{\beta} g v^h \,\mathrm{d}S, \tag{4.11b}$$

where $\tilde{\beta}$ is the Nitsche stabilization parameter. This parameter should be selected and scaled (with the mesh size) appropriately, being large enough to ensure stability, while not being too large to cause a reduction in accuracy (see, *e.g.*, Ref. [76, 85]). The ghost-penalty operator in (4.11a) controls the k^{th}-order normal derivative jumps, indicated by $[\![\cdot]\!]$, over the interfaces of the elements which are intersected by the domain boundary $\partial\Omega$. Since in this contribution B-splines of degree k with C^{k-1}-continuity are considered, only the k^{th} normal derivative is non-vanishing at the ghost mesh. As will be discussed in detail in Section 4.2.3, upon approriate selection and scaling (with the mesh size) of $\tilde{\gamma}_g$, a Nitsche stabilization parameter, $\tilde{\beta}$, can be selected in such a way that stability of the formulation can be assured independent of the cut-cell configurations. To avoid loss of accuracy, the ghost-penalty parameter $\tilde{\gamma}_g$ should also not be too large [235].

Steady viscous flow

Steady viscous flow can be modeled by the Stokes equations,

$$\begin{cases} \text{Find } \boldsymbol{u} \text{ and } p \text{ such that:} \\ \quad -\nabla \cdot (2\mu \nabla^s \mathbf{u}) + \nabla p = \mathbf{f} \quad \text{in } \Omega \\ \quad \nabla \cdot \mathbf{u} = 0 \quad \text{in } \Omega \\ \quad \mathbf{u} = \mathbf{g} \quad \text{on } \partial\Omega_D \\ \quad 2\mu \left(\nabla^s \mathbf{u}\right) \mathbf{n} - p\mathbf{n} = \mathbf{t} \quad \text{on } \partial\Omega_N \end{cases} \tag{4.12}$$

with velocity $\mathbf{u}$, pressure p, constant viscosity μ, body force $\mathbf{f}$, Dirichlet data $\mathbf{g}$ and Neumann data $\mathbf{t}$. By consideration of the solution in the abstract Galerkin problem (4.8) as a velocity-pressure pair, *i.e.*, $u^h = (\mathbf{u}^h, p^h) \in U^h = U_u^h \times U_p^h = [\mathcal{S}_{k-1}^k]^d \times \mathcal{S}_{k-1}^k \subset [H^1(\Omega)]^d \times L^2(\Omega)$ and the corresponding test functions as

$v^h = (\mathbf{v}^h, q^h) \in V^h = V_u^h \times V_p^h = U^h$, the aggregate bilinear and linear forms corresponding to (4.12) follow as

$$\begin{aligned} a^h(u^h, v^h) = &\int_\Omega \left(2\mu\nabla^s\mathbf{u}^h : \nabla^s\mathbf{v}^h - p^h\nabla\cdot\mathbf{v}^h - q^h\nabla\cdot\mathbf{u}^h\right) \mathrm{d}V \\ &- \int_{\partial\Omega_D} \big[2\mu\left((\nabla^s\mathbf{u}^h)\mathbf{n}\cdot\mathbf{v}^h + (\nabla^s\mathbf{v}^h)\mathbf{n}\cdot\mathbf{u}^h\right) \\ &\quad - \left(p^h\mathbf{v}^h\cdot\mathbf{n} + q^h\mathbf{u}^h\cdot\mathbf{n}\right)\big] \,\mathrm{d}S \\ &+ \sum_{E\in\mathcal{T}_{\partial\Omega_D}} \int_E \tilde{\beta}\mu\mathbf{u}^h\cdot\mathbf{v}^h \,\mathrm{d}S \\ &+ \sum_{F\in\mathcal{F}_{\text{ghost}}} \int_F \tilde{\gamma}_g\mu[\![\partial_n^k\mathbf{u}^h]\!]\cdot[\![\partial_n^k\mathbf{v}^h]\!] \,\mathrm{d}S \\ &- \sum_{F\in\mathcal{F}_{\text{skeleton}}} \int_F \tilde{\gamma}_s\mu^{-1}[\![\partial_n^k p^h]\!][\![\partial_n^k q^h]\!] \,\mathrm{d}S, \end{aligned} \tag{4.13a}$$

$$\begin{aligned} b^h(v^h) = &\int_\Omega \mathbf{f}\cdot\mathbf{v}^h \,\mathrm{d}V + \int_{\partial\Omega_N} \mathbf{t}\cdot\mathbf{v}^h \,\mathrm{d}S \\ &- \int_{\partial\Omega_D} \left(2\mu(\nabla^s\mathbf{v}^h)\mathbf{n} - q^h\mathbf{n}\right)\cdot\mathbf{g} \,\mathrm{d}S + \sum_{E\in\mathcal{T}_{\partial\Omega_D}} \int_E \tilde{\beta}\mathbf{g}\cdot\mathbf{v}^h \,\mathrm{d}S. \end{aligned} \tag{4.13b}$$

For the selection of the Nitsche parameter, $\tilde{\beta}$, and ghost stabilization constant, $\tilde{\gamma}_g$, the same arguments apply as for the steady heat conduction problem discussed above. A discussion on the selection and scaling of these parameters for the Stokes problem is presented in Section 4.2.3.

An additional stability issue is encountered for the immersed Stokes flow problem (4.13) on account of the selected equal-order optimal regularity spline spaces of degree k. In the conforming setting, inf-sup stability is achieved by adopting a suitable velocity-pressure pair, *e.g.*, Taylor-Hood [24, 190–192] or Raviart-Thomas [191, 193–195]. In the immersed setting, such pairs can still lead to pressure oscillations in the vicinity of cut elements [204]. To resolve these pressure oscillations, the immersogeometric skeleton stabilization technique developed in Ref. [15] is applied. This stabilization technique can be regarded as the higher-order continuous version of the method proposed in Ref. [236], which has also been applied in the conforming isogeometric analysis setting [204].

From equation (4.13a) it is seen that the skeleton stabilization term penalizes jumps in higher-order pressure gradients, where the parameter $\tilde{\gamma}_s$ should be selected such that oscillations vanish, while the influence of the additional term

on the accuracy of the solution remains limited. The purpose of the skeleton stabilization method is to avoid pressure oscillations associated inf-sup stability problems, allowing for the utilization of identical spaces for the velocity components and the pressure. Since the inf-sup stability problem is not restricted to the immersed boundary, the skeleton stabilization pertains to all interfaces of the background mesh. The appropriate selection and scaling of the skeleton stability parameter is discussed in detail in Section 4.2.3.

4.2.3 Selection of the stabilization parameters: continuity and coercivity of the formulation

Before considering a-posteriori error estimation in Section 4.3, we first study the continuity and coercivity of the immersed formulations introduced above. We commence with the introduction of the following inequalities:

- Using Young's inequality, it follows that for any constant $\varepsilon > 0$ it holds that

$$2\|u^h\|_{L^2}\|\tilde{u}^h\|_{L^2} \leq \varepsilon\|u^h\|_{L^2}^2 + \frac{1}{\varepsilon}\|\tilde{u}^h\|_{L^2}^2 \quad \forall u^h \in U^h,\ \forall \tilde{u}^h \in U^h. \tag{4.14}$$

  In combination with the Cauchy-Schwarz inequality, this inequality can be applied to obtain

$$2\int_{\partial\Omega_D} (\partial_n u^h) u^h \,\mathrm{d}S \leq \varepsilon\|\partial_n u^h\|_{L^2(\partial\Omega_D)}^2 + \frac{1}{\varepsilon}\|u^h\|_{L^2(\partial\Omega_D)}^2 \quad \forall u^h \in U^h. \tag{4.15}$$

- For any background element K crossed by the boundary $\partial\Omega$, with $E = K \cap \partial\Omega$, under an assumption of shape regularity (i.e., provided with an upper bound on the length of the intersection of the boundary within one single element $\mathrm{meas}(K \cap \partial\Omega)$), it holds that (see, *e.g.*, Ref. [237, Lemma 4.2])

$$\|\phi\|_{L^2(E)}^2 \leq C_T\|{h_K}^{-1/2}\phi\|_{L^2(K)}^2 \qquad \forall \phi \in P^k, \tag{4.16}$$

  where it is noted that this inequality holds for the finite-dimensional space P^k of tensor-product polynomials of order k (not for functions in H^1 in general). The constant $C_T > 0$, referred to as the trace inequality constant, is independent of the size of the element, but dependent on the order k. Note that the right part of the inequality contains the norm over the full background element K, and not just its intersection with the physical domain.

Using inequality (4.16), the following bound for the normal gradient of u^h on the immersed boundary is obtained as

$$\begin{aligned}\|\partial_n u^h\|^2_{L^2(\partial\Omega)} &\leq \|\nabla u^h\|^2_{L^2(\partial\Omega)} = \sum_{E\in\mathcal{T}_{\partial\Omega}} \|\nabla u^h\|^2_{L^2(E)} \\ &\leq \sum_{K\in\mathcal{G}} C_T \|{h_K}^{-1/2}\nabla u^h\|^2_{L^2(K)} \\ &\leq C_T \|{h_\mathcal{T}}^{-1/2}\nabla u^h\|^2_{L^2(\mathcal{T})} \qquad \forall u^h \in U^h, \quad (4.17)\end{aligned}$$

with $h_\mathcal{T}$ defined in Eq. (4.2) and where, with abuse of notation, the constant C_T is used to both represent the local trace inequality constant (second line) and its global maximum (third line).

- Norms of functions over the entire background domain $\mathcal{T}$ can be bounded by norms over the physical domain Ω and the ghost penalty. Using the ghost-penalty, the gradients on the background mesh are bounded by those in the physical domain. To demonstrate this bound, we split the norm over the background mesh as

$$\begin{aligned}\|\nabla u^h\|^2_{L^2(\mathcal{T})} &= \|\nabla u^h\|^2_{L^2(\mathcal{T}\setminus\mathcal{G})} + \|\nabla u^h\|^2_{L^2(\mathcal{G})} \\ &\leq \|\nabla u^h\|^2_{L^2(\Omega)} + \|\nabla u^h\|^2_{L^2(\mathcal{G})} \\ &\leq \|\nabla u^h\|^2_{L^2(\Omega)} + \sum_{K\in\mathcal{G}} \|\nabla u^h\|^2_{L^2(K)} \qquad \forall u^h \in U^h. \quad (4.18)\end{aligned}$$

To show the last inequality, we consider an element $K \in \mathcal{G}$ which shares the interface F with an element $K' \notin \mathcal{G}$ that completely lies inside Ω, such that the volume integral over the background element K' is included in the norm over Ω. We will first demonstrate that the gradients on K are controlled by the ghost penalty and the norms on the physical domain. Later on, elements in $\mathcal{G}$ that do not share an interface with an element in $\mathcal{T} \setminus \mathcal{G}$ will be considered by means of recursion. To demonstrate that the gradients on K are bounded by those in the physical domain, we define the polynomial extension of $u^h\big|_{K'}$ as the global polynomial $\bar{u}^h_{K'} \in P^k$ (see Figure 4.3). Using this extension, the spline function u^h on the element K can be decomposed as

$$u^h\big|_K = \bar{u}^h_{K'} + \tilde{u}^h_{K'}. \quad (4.19)$$

Let us consider $\boldsymbol{x}_F$ as a projection of $\boldsymbol{x}$ on the straight or flat interface F, such that $\boldsymbol{x}$ can be written as $\boldsymbol{x}_F + x_n \boldsymbol{n}_F$, where $x_n = (\boldsymbol{x} - \boldsymbol{x}_F) \cdot \boldsymbol{n}_F$. Here, the interface coordinate $\boldsymbol{x}_F \in F$ is interpreted to be on the side of

the element K, and related to the coordinate $\boldsymbol{x} \in K$. The function $\tilde{u}^h_{K'}$ has no support on K' and has vanishing normal derivatives up to order k at the interface F. By Taylor-series expansion, one can infer

$$\begin{aligned}\tilde{u}^h_{K'}(\boldsymbol{x}) &= \frac{1}{k!}\partial^k_n(u^h(\boldsymbol{x}_F) - \bar{u}^h_{K'}(\boldsymbol{x}_F))x^k_n \\ &= \frac{1}{k!}\partial^k_n[\![u^h(\boldsymbol{x}_F)]\!]x^k_n \qquad\qquad \forall \boldsymbol{x} \in K, \forall u^h \in U^h. \end{aligned} \tag{4.20}$$

This splitting is very natural through the use of maximum regularity splines (i.e., $\tilde{u}^h_{K'}$ contains all degrees of freedom of K that are independent of K'). For the polynomial extension $\bar{u}^h_{K'}$ it holds that

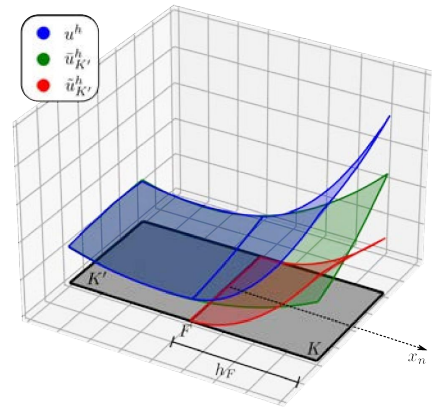
$$\|\nabla \bar{u}^h_{K'}\|^2_{L^2(K)} \leq C_Q \|\nabla \bar{u}^h_{K'}\|^2_{L^2(K')} \qquad \forall u^h \in U^h, \tag{4.21}$$

where the constant C_Q is independent of the mesh size, but dependent on the order of the approximation and the ratio of the size of the elements at either side of the interface. The order-dependence of this constant is illustrated in Figure 4.3c. The presented results have been computed by solving the generalized eigenvalue problem corresponding to Eq. (4.21).

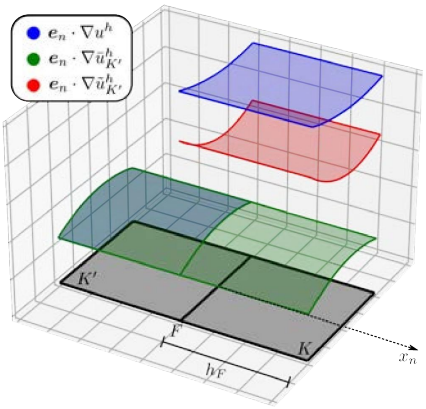
From the expansion $\tilde{u}^h_{K'}$ in equation (4.20) it follows that

$$\begin{aligned}\|\nabla \tilde{u}^h_{K'}\|^2_{L^2(K)} &= \|\frac{1}{k!}\left(\nabla_F \partial^k_n \tilde{u}^h\right) x^k_n + \frac{k}{k!}\left(\partial^k_n \tilde{u}^h\right) x^{k-1}_n\|^2_{L^2(K)} \\ &\leq \frac{2}{(k!)^2}\Big(\|\left(\nabla_F \partial^k_n \tilde{u}^h\right) x^k_n\|^2_{L^2(K)} \\ &\qquad + \|k\left(\partial^k_n \tilde{u}^h\right) x^{k-1}_n\|^2_{L^2(K)}\Big) \\ &\leq \frac{2}{(k!)^2}\Big(\|\nabla_F \partial^k_n \tilde{u}^h\|^2_{L^2(F)}(2k+1)^{-1}h_F^{2k+1} \\ &\qquad + k^2\|\partial^k_n \tilde{u}^h\|^2_{L^2(F)}(2k-1)^{-1}h_F^{2k-1}\Big) \\ &\leq \frac{C_F}{2}h_F^{2k-1}\|\partial^k_n \tilde{u}^h\|^2_{L^2(F)} \\ &\leq \frac{C_F}{2}h_F^{2k-1}\|[\![\partial^k_n u^h]\!]\|^2_{L^2(F)} \end{aligned} \tag{4.22}$$

with h_F the size of K in the direction normal to the interface and where ∇_F denotes the surface gradient in the interface F and where use has been made of the polynomial inequality $\|\nabla_F f^h\|^2_{L^2(F)} \lesssim h_F^{-2}\|f^h\|^2_{L^2(F)}$ for all $f^h \in P^k$ [238]. The dependence of the constant C_F in the inequality (4.22) on the order is illustrated in Figure 4.3d. This constant is independent of the mesh size.

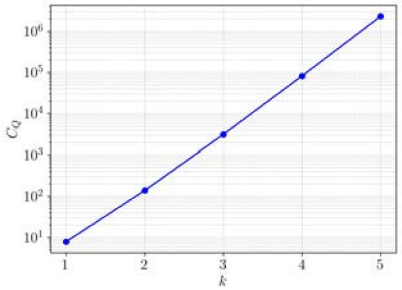


(a)



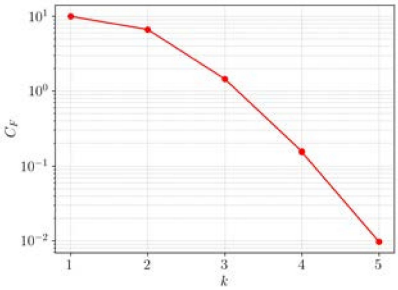
(b)

$$\|\nabla \bar{u}^h_{K'}\|^2_{L^2(K)} \leq C_Q \|\nabla \bar{u}^h_{K'}\|^2_{L^2(K')}$$



(c)

$$\|\nabla \tilde{u}^h_{K'}\|^2_{L^2(K)} \leq C_F h_F^{2k-1} \|[\![\partial_n^k u^h]\!]\|^2_{L^2(F)}$$



(d)

Figure 4.3: Panel (a) is an illustration of a second order B-spline on an element $K \in \mathcal{G}$ and its adjacent element K' with an interface F. Panel (b) is its second order gradient in the direction normal to the interface (with $\boldsymbol{e}_n$ the unit vector in the normal direction). Panels (c) and (d) show the dependence of the constants in (4.21) and (4.22) on the order k.

Substituting the decomposition (4.19) in equation (4.18) yields

$$\begin{aligned}\|\nabla u^h\|^2_{L^2(\mathcal{G})} &= \sum_{K\in\mathcal{G}} \|\nabla \bar{u}^h_{K'} + \nabla \tilde{u}^h_{K'}\|^2_{L^2(K)} \\ &\leq 2 \sum_{K\in\mathcal{G}} \left(\|\nabla \bar{u}^h_{K'}\|^2_{L^2(K)} + \|\nabla \tilde{u}^h_{K'}\|^2_{L^2(K)} \right) \quad \forall u^h \in U^h. \end{aligned} \tag{4.23}$$

Using the inequalities (4.21) and (4.22), and noting that since $K' \in \mathcal{T} \setminus \mathcal{G}$ it follows that $\|\nabla \bar{u}^h_{K'}\|^2_{L^2(K')} = \|\nabla u^h\|^2_{L^2(K')}$, then results in

$$\begin{aligned}\|\nabla u^h\|^2_{L^2(\mathcal{G})} &\leq (C_G - 1)\|\nabla u^h\|^2_{L^2(\Omega)} \\ &\quad + \sum_{F\in\mathcal{F}} C_F h_F^{2k-1} \|[\![\partial_n^k u^h]\!]\|^2_{L^2(F)} \qquad \forall u^h \in U^h, \end{aligned} \tag{4.24}$$

where $C_G = 1 + 2\max_{K\in\mathcal{G}}(C_Q)$. To obtain this result, the inequality is first applied to the layer of elements in $\mathcal{G}$ that share an interface with the interior mesh $\mathcal{T} \setminus \mathcal{G}$. With control over the gradients in this first layer, the inequality is then applied to a second layer of elements. This recursive application is repeated until all elements in $\mathcal{G}$ have been considered. As a result of this recursive application of the ghost inequality, the constant C_G depends on the number of layers, which in turn depends on the mesh size.

The boundedness of the gradients on the background mesh finally follows by substitution of (4.24) in (4.18):

$$\begin{aligned}\|\nabla u^h\|^2_{L^2(\mathcal{T})} &\leq C_G\|\nabla u^h\|^2_{L^2(\Omega)} \\ &\quad + \sum_{F\in\mathcal{F}} C_F h_F^{2k-1} \|[\![\partial_n^k u^h]\!]\|^2_{L^2(F)} \qquad \forall u^h \in U^h. \end{aligned} \tag{4.25}$$

One may note that the second term in (4.25) corresponds to the ghost-penalty term with an appropriate scaling of the stabilization parameter; see *e.g.*, (4.11a).

- Following Ref. [239, Lemma 3.11], where piecewise linear ($k = 1$) polynomials are considered, we conjecture that for any $p^h \in U_p^h$ there exists a $\mathbf{w}^h \in V_u^h$ such that, for sufficiently small mesh sizes, it holds that

$$\begin{aligned}-a_2^h(p^h, \mathbf{w}^h) &\geq C_1\|\mu^{-1/2} p^h\|^2_{L^2(\mathcal{T})} \\ &\quad - C_2 \sum_{F\in\mathcal{F}_{\text{skeleton}}} \int_F h_F^{2k+1} \mu^{-1} [\![\partial_n^k p^h]\!] [\![\partial_n^k p^h]\!] \,\mathrm{d}S, \end{aligned} \tag{4.26a}$$

$$\left|\!\left|\!\left| \mathbf{w}^h \right|\!\right|\!\right|_u \leq C_3 \|\mu^{-1/2} p^h\|_{L^2(\mathcal{T})}, \tag{4.26b}$$

with

$$a_2^h(p^h, \mathbf{w}^h) := -\int_\Omega p^h \nabla \cdot \mathbf{w}^h \, \mathrm{d}V + \int_{\partial\Omega_D} p^h \mathbf{w}^h \cdot \mathbf{n} \, \mathrm{d}S,$$

$\left|\left|\left|\mathbf{w}^h\right|\right|\right|_u$ the velocity energy norm (see Section 4.2.3), and constants $C_1, C_3 > 0$ and $C_2 \geq 0$. Since the function spaces considered herein do not coincide with those considered in Ref. [239], it is not evident that the proof presented there is also applicable to the situation considered in this work. However, numerical evidence of the formulation (4.13) being stable in the (optimal-regularity) spline setting has been presented [15, 30]. These numerical results do suggest that the inequalities (4.26) hold, but a formal proof is so far unavailable.

Steady heat conduction

Continuity of the bilinear form (4.11a) cannot be shown in the H^1-norm on account of the immersed boundary terms, and coercivity cannot be shown on the infinite-dimensional space. However, with an appropriate selection of the stabilization parameters, continuity and coercivity can be established with respect to the mesh-dependent norm

$$\begin{aligned}\left|\left|\left|u^h\right|\right|\right|^2 := \|\nabla u^h\|^2_{L^2(\mathcal{T})} + \|\tilde{\beta}^{-1/2}\partial_n u^h\|^2_{L^2(\partial\Omega_D)} \\ + \|\tilde{\beta}^{1/2} u^h\|^2_{L^2(\partial\Omega_D)} + \sum_{F\in\mathcal{F}_{\text{ghost}}} \|\tilde{\gamma}_g^{1/2}[\![\partial_n^k u^h]\!]\|^2_{L^2(F)} \quad \forall u^h \in U^h, \end{aligned} \tag{4.27}$$

which we refer to as the *energy norm*.

The bilinear form (4.11a) is continuous on $U^h \times V^h$ if there exists a constant, $\mathscr{C} > 0$, independent of the mesh size, such that

$$a^h(u^h, v^h) \leq \mathscr{C} \left|\left|\left|u^h\right|\right|\right| \left|\left|\left|v^h\right|\right|\right| \qquad \forall u^h \in U^h, \ \forall v^h \in V^h. \tag{4.28}$$

Using the Cauchy-Schwarz inequality, for all $u^h \in U^h, \ v^h \in V^h$ one obtains

$$\begin{aligned} a^h(u^h, u^h) \leq \|\nabla u^h\|_{L^2(\Omega)} \|\nabla v^h\|_{L^2(\Omega)} \\ + \|\tilde{\beta}^{-1/2}\partial_n u^h\|_{L^2(\partial\Omega_D)} \|\tilde{\beta}^{1/2} v^h\|_{L^2(\partial\Omega_D)} \\ + \|\tilde{\beta}^{1/2} u^h\|_{L^2(\partial\Omega_D)} \|\tilde{\beta}^{-1/2}\partial_n v^h\|_{L^2(\partial\Omega_D)} \\ + \sum_{E\in\mathcal{T}_{\partial\Omega_D}} \|\tilde{\beta}^{1/2} u^h\|_{L^2(E)} \|\tilde{\beta}^{1/2} v^h\|_{L^2(E)} \\ + \sum_{F\in\mathcal{F}_{\text{ghost}}} \|\tilde{\gamma}_g^{1/2}[\![\partial_n^k u^h]\!]\|_{L^2(F)} \|\tilde{\gamma}_g^{1/2}[\![\partial_n^k v^h]\!]\|_{L^2(F)}. \end{aligned}$$

Since each of the norms in this expression is bounded from above by the energy norm (4.27), it follows that $a^h(u^h,u^h) \leq 5 \left|\!\left|\!\left| u^h \right|\!\right|\!\right| \left|\!\left|\!\left| v^h \right|\!\right|\!\right|$. Hence the bilinear form is continuous.

The bilinear form (4.11a) is uniformly (*i.e.*, independent of h) coercive on U^h if there exists a constant, $\mathscr{C}^\star > 0$, such that

$$a^h(u^h,u^h) \geq \mathscr{C}^\star \left|\!\left|\!\left| u^h \right|\!\right|\!\right|^2 \qquad\qquad u^h \in U^h. \tag{4.29}$$

To demonstrate that this is indeed the case, we apply the inequalities (4.15) and (4.25) to obtain

$$\begin{aligned} a^h(u^h,u^h) \geq & \left(C_G^{-1} \|\nabla u^h\|^2_{L^2(\mathcal{T})} - \sum_{F\in\mathcal{F}_{\text{ghost}}} C_G^{-1} C_F \int_F h_F^{2k-1} [\![\partial_n^k u^h]\!]^2 \,\mathrm{d}S \right) \\ & - \left(\varepsilon \|\partial_n u^h\|^2_{L^2(\partial\Omega_D)} + \varepsilon^{-1} \|u^h\|^2_{L^2(\partial\Omega_D)} \right) \\ & + \sum_{E\in\mathcal{T}_{\partial\Omega_D}} \|\tilde{\beta}^{\frac{1}{2}} u^h\|^2_{L^2(E)} + \sum_{F\in\mathcal{F}_{\text{ghost}}} \int_F \tilde{\gamma}_g [\![\partial_n^k u^h]\!]^2 \,\mathrm{d}S. \end{aligned}$$

Application of the trace inequality (4.16) and collecting terms then yields

$$\begin{aligned} a^h(u^h,u^h) \geq & \left\| \left(C_G^{-1} - \varphi C_T h_{\mathcal{T}}^{-1} \right)^{\frac{1}{2}} \nabla u^h \right\|^2_{L^2(\mathcal{T})} + \left\| (\varphi - \varepsilon)^{\frac{1}{2}} \, \partial_n u^h \right\|^2_{L^2(\partial\Omega_D)} \\ & + \left\| \left(\tilde{\beta} - \varepsilon^{-1} \right)^{\frac{1}{2}} u^h \right\|^2_{L^2(\partial\Omega_D)} \\ & + \sum_{F\in\mathcal{F}_{\text{ghost}}} \int_F \left(\tilde{\gamma}_g - C_G^{-1} C_F h_F^{2k-1} \right) [\![\partial_n^k u^h]\!]^2 \,\mathrm{d}S, \end{aligned}$$

for arbitrary $\varphi > 0$. By selecting element-wise constants $0 < \varepsilon \leq \varphi$ and $0 < \varphi \leq \frac{C_G^{-1}}{C_T} h_{\mathcal{T}}$, one can infer that coercivity is ensured provided that

$$\tilde{\beta} = \beta h_K^{-1} \geq C_T C_G h_K^{-1} \qquad\qquad \tilde{\gamma}_g = \gamma_g h_F^{2k-1} \geq \frac{C_F}{C_G} h_F^{2k-1}. \tag{4.30}$$

for all elements K and interfaces F, where the positive constants $\beta \geq C_T C_G$ and $\gamma_g \geq C_F C_G^{-1}$ are independent of the mesh size. The interface length scale is defined as $h_F = \max(h_K, h_{K'})$ with K and K' being the elements on either side of the interface F. The rational behind this choice is that the ghost stabilization term scales with h_F^{2k-1} ($k \geq 1$) and that hence the larger element size ensures that the stability constant is sufficiently large.

Steady viscous flow

Recalling that for the Stokes problem $u^h = (\boldsymbol{u}^h, p^h)$, we define the mesh-dependent error norm as

$$\left|\!\left|\!\left| u^h \right|\!\right|\!\right|^2 = \left|\!\left|\!\left| (\boldsymbol{u}^h, p^h) \right|\!\right|\!\right|^2 = \left|\!\left|\!\left| \mathbf{u}^h \right|\!\right|\!\right|_u^2 + \left|\!\left|\!\left| p^h \right|\!\right|\!\right|_p^2, \tag{4.31}$$

with

$$\begin{aligned}\left|\!\left|\!\left| \mathbf{u}^h \right|\!\right|\!\right|_u^2 &:= \|\mu^{1/2}\nabla^s \boldsymbol{u}^h\|_{L^2(\mathcal{T})}^2 + \|\tilde{\beta}^{-1/2}\mu^{1/2}\partial_n \boldsymbol{u}^h\|_{L^2(\partial\Omega_D)}^2 \\ &\quad + \|\tilde{\beta}^{1/2}\mu^{1/2}\boldsymbol{u}^h\|_{L^2(\partial\Omega_D)}^2 + \sum_{F\in\mathcal{F}_{\text{ghost}}} \|\tilde{\gamma}_g^{1/2}\mu^{1/2}[\![\partial_n^k \boldsymbol{u}^h]\!]\|_{L^2(F)}^2, \end{aligned} \tag{4.32a}$$

$$\left|\!\left|\!\left| p^h \right|\!\right|\!\right|_p^2 := \|\mu^{-1/2} p^h\|_{L^2(\mathcal{T})}^2 + \sum_{F\in\mathcal{F}_{\text{skeleton}}} \|\tilde{\gamma}_s^{1/2}\mu^{-1/2}[\![\partial_n^k p^h]\!]\|_{L^2(F)}^2. \tag{4.32b}$$

Continuity of the bilinear form (4.13a) with respect to this energy norm in the sense of (4.28) follows directly by application of the Cauchy-Schwarz inequality to all terms in (4.13a).

With an appropriate selection of the stability parameters for the Stokes problem, it holds that the bilinear form (4.13a) is inf-sup stable in accordance with

$$\sup_{v^h\in V^h\setminus\{0\}} \frac{a^h(u^h, v^h)}{\left|\!\left|\!\left| v^h \right|\!\right|\!\right|} \geq C^\star \left|\!\left|\!\left| u^h \right|\!\right|\!\right| \qquad \forall u^h \in U^h, \tag{4.33}$$

where $C^\star > 0$ is the inf-sup stability constant. To demonstrate this stability property, we split the bilinear form as

$$a^h(u^h, v^h) = a_1^h(\mathbf{u}^h, \mathbf{v}^h) + a_2^h(q^h, \mathbf{u}^h) + a_2^h(p^h, \mathbf{v}^h) - a_3^h(p^h, q^h), \tag{4.34}$$

with

$$\begin{aligned} a_1^h(\mathbf{u}^h, \mathbf{v}^h) := &\int_\Omega 2\mu\nabla^s\mathbf{u}^h : \nabla^s\mathbf{v}^h \,\mathrm{d}V \\ &- \int_{\partial\Omega_D} \left[2\mu\left((\nabla^s\mathbf{u}^h)\mathbf{n}\cdot\mathbf{v}^h + (\nabla^s\mathbf{v}^h)\mathbf{n}\cdot\mathbf{u}^h\right)\right] \mathrm{d}S \\ &+ \sum_{E\in\mathcal{T}_{\partial\Omega_D}} \int_E \frac{\beta\mu}{h_K}\mathbf{u}^h\cdot\mathbf{v}^h \,\mathrm{d}S \\ &+ \sum_{F\in\mathcal{F}_{\text{ghost}}} \int_F \gamma\mu h_F^{2k-1}[\![\partial_n^k\mathbf{u}^h]\!]\cdot[\![\partial_n^k\mathbf{v}^h]\!], \end{aligned} \tag{4.35a}$$

$$a_2^h(q^h, \mathbf{u}^h) := -\int_\Omega q^h \nabla \cdot \mathbf{u}^h \,\mathrm{d}V + \int_{\partial\Omega_D} q^h \mathbf{u}^h \cdot \mathbf{n} \,\mathrm{d}S, \tag{4.35b}$$

$$a_3^h(p^h, q^h) := \sum_{F \in \mathcal{F}_{\text{skeleton}}} \int_F \tilde{\gamma}_s \mu^{-1} [\![\partial_n^k p^h]\!] [\![\partial_n^k q^h]\!] \,\mathrm{d}S. \tag{4.35c}$$

We now take a function

$$\varphi^h = (\mathbf{v}^h, q^h) = (\mathbf{u}^h - \alpha \mathbf{w}^h, -p^h) \in V^h,$$

where $\mathbf{w}^h$ depends on p^h in accordance with (4.26), and with some constant $\alpha > 0$, such that

$$\begin{aligned}
&\sup_{v^h \in V^h \setminus \{0\}} \frac{a^h(u^h, v^h)}{|||v^h|||} \\
&\geq \frac{a^h(u^h, \varphi^h)}{|||\varphi^h|||} \\
&\geq \frac{a_1^h(\mathbf{u}^h, \mathbf{u}^h) - \alpha a_1^h(\mathbf{u}^h, \mathbf{w}^h)}{|||\varphi^h|||} \\
&\quad + \frac{-\alpha a_2^h(p^h, \mathbf{w}^h) + a_3^h(p^h, p^h)}{|||\varphi^h|||} \qquad \forall u^h \in U^h.
\end{aligned} \tag{4.36}$$

Following Section 4.2.3, $a_1^h(\mathbf{u}^h, \mathbf{u}^h)$ is coercive and $a_1^h(\mathbf{u}^h, \mathbf{w}^h)$ is continuous with respect to the velocity energy norm (4.32a) in accordance with Eqs. (4.28) and (4.29), respectively. Hence,

$$\begin{aligned}
&\sup_{v^h \in V^h \setminus \{0\}} \frac{a^h(u^h, v^h)}{|||v^h|||} \\
&\geq \frac{C_u^\star \left|||\mathbf{u}^h|||_u^2 - \alpha \mathcal{C} \left|||\mathbf{u}^h|||_u \left|||\mathbf{w}^h|||_u \right.\right.}{\sqrt{2\left|||\mathbf{u}^h|||_u^2 + 2\alpha^2 \left|||\mathbf{w}^h|||_u^2 + |||p^h|||_p^2\right.}} \\
&\quad + \frac{-\alpha a_2^h(p^h, \mathbf{w}^h) + a_3^h(p^h, p^h)}{\sqrt{2\left|||\mathbf{u}^h|||_u^2 + 2\alpha^2 \left|||\mathbf{w}^h|||_u^2 + |||p^h|||_p^2\right.}} \qquad \forall u^h \in U^h,
\end{aligned} \tag{4.37}$$

where use has been made of

$$\begin{aligned}
|||\varphi^h||| &= \sqrt{|||\mathbf{u}^h - \alpha \mathbf{w}^h|||_u^2 + |||p^h|||_p^2} \\
&\leq \sqrt{2\left|||\mathbf{u}^h|||_u^2 + 2\alpha^2 \left|||\mathbf{w}^h|||_u^2 + |||p^h|||_p^2\right.}.
\end{aligned} \tag{4.38}$$

Using the inequalities (4.26), it follows that

$$\begin{aligned}
&\sup_{v^h \in V^h \setminus \{0\}} \frac{a^h(u^h, v^h)}{|||v^h|||} \\
&\geq \frac{C_u^\star \left|||\mathbf{u}^h|||_u^2 - \alpha \mathcal{C} C_3 \left|||\mathbf{u}^h|||_u \|\mu^{-1/2} p^h\|_{L^2(\mathcal{T})}}{\sqrt{2\,|||\mathbf{u}^h|||_u^2 + (1 + 2\alpha^2 C_3^2)\,|||p^h|||_p^2}} \\
&\quad + \frac{\alpha C_1 \|\mu^{-1/2} p^h\|_{L^2(\mathcal{T})}^2}{\sqrt{2\,|||\mathbf{u}^h|||_u^2 + (1 + 2\alpha^2 C_3^2)\,|||p^h|||_p^2}} \\
&\quad + \frac{\sum\limits_{F \in \mathcal{F}_{\text{skeleton}}} \left(1 - \alpha C_2 \tilde{\gamma}_s^{-1} h_F^{2k+1}\right) \|\tilde{\gamma}_s^{1/2} \mu^{-1/2} [\![\partial_n^k p^h]\!]\|_F^2}{\sqrt{2\,|||\mathbf{u}^h|||_u^2 + (1 + 2\alpha^2 C_3^2)\,|||p^h|||_p^2}} \qquad \forall u^h \in U^h, \quad (4.39)
\end{aligned}$$

which, using Young's inequality (4.14) with $\varepsilon = 1$, can be reformulated as

$$\begin{aligned}
&\sup_{v^h \in V^h \setminus \{0\}} \frac{a^h(u^h, v^h)}{|||v^h|||} \\
&\geq \frac{\left(C_u^\star - \dfrac{\alpha \mathcal{C} C_3}{2}\right) |||\mathbf{u}^h|||_u^2}{C_4\,|||u^h|||} \\
&\quad + \frac{\alpha \left(C_1 - \dfrac{\mathcal{C} C_3}{2}\right) \|\mu^{-1/2} p^h\|_{L^2(\mathcal{T})}^2}{C_4\,|||u^h|||} \\
&\quad + \frac{\sum\limits_{F \in \mathcal{F}_{\text{skeleton}}} \left(1 - \alpha C_2 \tilde{\gamma}_s^{-1} h_F^{2k+1}\right) \|\tilde{\gamma}_s^{1/2} \mu^{-1/2} [\![\partial_n^k p^h]\!]\|_F^2}{C_4\,|||u^h|||} \qquad \forall u^h \in U^h. \quad (4.40)
\end{aligned}$$

Inf-sup stability as in (4.33) then holds, provided that

$$\alpha < \frac{2 C_u^\star}{\mathcal{C} C_3}, \qquad \tilde{\gamma}_s = \gamma_s h_F^{2k+1} \geq \alpha C_2 h_F^{2k+1}. \quad (4.41)$$

We note that the skeleton penalty has two purposes: *i)* It extends the stability of the pressure field to the background grid $\mathcal{T}$ as in (4.26), in the same way as for the ghost penalty discussed in Section 4.2.3. Since stability is here defined with respect to the L^2-norm of the pressure field, the skeleton stability constant $\tilde{\gamma}_s$ scales with h^{2k+1}, following the same reasoning as in Eq. (4.25); *ii)* It ensures the inf-sup stability for equal-order discretizations, essentially meaning that pressure oscillations in the interior are penalized. This is the reason why this term is applied over the complete skeleton and not only the ghost interfaces.

4.3 Error estimation and adaptivity

We study a posteriori error estimation and adaptivity for immersogeometric analysis. In Section 4.3.1 we first introduce a residual-based error indicator, and elaborate it for the heat conduction problem and viscous flow problem introduced in the previous section. In Section 4.3.2 the refinement strategy is discussed.

4.3.1 Residual-based error estimation

We propose an error estimator pertaining to the background mesh, $\mathcal{T}$, of the form

$$\mathcal{E} = \sqrt{\sum_{K\in\mathcal{T}} \eta_K^2}, \tag{4.42}$$

where the element-wise error indicators, η_K, will serve to guide an adaptive refinement procedure. The derivations of the indicators for the heat conduction problem and viscous flow problem will be elaborated in the following sections.

From an abstract perspective, the element-wise error indicators are defined in such a way that the estimator (4.42) bounds the residual from above as

$$\mathcal{E} \gtrsim \|r^h\|_{\widehat{V}^h}. \tag{4.43}$$

In this expression, the residual and its norm are defined as

$$\begin{aligned} r^h(\widehat{v}^h) &:= r^h(u^h)(\widehat{v}^h) := b^h(\widehat{v}^h) - a^h(u^h, \widehat{v}^h), \\ \|r^h\|_{\widehat{V}^h} &:= \sup_{\widehat{v}^h\in\widehat{V}^h\setminus\{0\}} \frac{r^h(\widehat{v}^h)}{|||\widehat{v}^h|||}, \end{aligned} \tag{4.44}$$

with the discrete function space $\widehat{V}^h$ being defined on the same mesh and with the same regularity as the space V^h, but with the order of the basis elevated in such a way that $\widehat{V}^h \supset V^h$.

The reason for considering the finite-dimensional order-elevated space $\widehat{V}^h$ in the definition of the norm (4.44) is that in the considered immersed formulations, inf-sup stability can only be shown in the discrete setting. Obtaining a stable formulation in the space $\widehat{V}^h$ requires additional stabilization terms on higher-order non-vanishing derivatives, for which appropriate stabilization constants must be selected [30, 240]. Note that, in principle this means that the operators defined above need to be augmented with these additional stabilization terms, but we assume that for the solution in the order-elevated space these terms are small. Similar assumptions, referred to as saturation assumptions, have been considered in *e.g.*, [219–221].

Assuming that the stability constants are selected such that a stable formulation is obtained both in V^h and in $\widehat{V}^h$, it follows that

$$\begin{aligned}\mathcal{E} \gtrsim \sup_{\widehat{v}^h \in \widehat{V}^h \setminus \{0\}} \frac{r^h(\widehat{v}^h)}{|||\widehat{v}^h|||} &= \sup_{\widehat{v}^h \in \widehat{V}^h \setminus \{0\}} \frac{b^h(\widehat{v}^h) - a^h(u^h, \widehat{v}^h)}{|||\widehat{v}^h|||} \\ &= \sup_{\widehat{v}^h \in \widehat{V}^h \setminus \{0\}} \frac{a^h(\widehat{u}^h - u^h, \widehat{v}^h)}{|||\widehat{v}^h|||} \gtrsim |||e|||\,, \end{aligned} \tag{4.45}$$

where the solution error is defined as the difference between the solution obtained in the order-elevated space, $\widehat{u}^h \in \widehat{V}^h$, and the Galerkin approximation (4.8), *i.e.*,

$$e = \widehat{u}^h - u^h. \tag{4.46}$$

As we will elaborate in Sections 4.3.1 and 4.3.1, the order-elevated solution is not required for the calculation of the residual-based estimator (4.42). This refined space is only used to provide a proper functional setting for the error estimator.

Steady heat conduction

To derive the error indicators for the steady heat conduction problem introduced in Section 4.2.2, it is first noted that because of Galerkin orthogonality

$$r^h(\widehat{v}^h) = r^h(\widehat{v}^h - \Pi^h \widehat{v}^h) = r^h(\tilde{v}), \tag{4.47}$$

where $\tilde{v} = \widehat{v}^h - \Pi^h \widehat{v}^h \in \widehat{V}^h$ and $\Pi^h : \widehat{V}^h \to V^h$ is an interpolation operator [241, 242]. Note that, for notational convenience, we will drop the diacritic and superscript from $\widehat{v}^h \in \widehat{V}^h$ in the remainder of this section, *i.e.*, $\widehat{v}^h = v$.

Using the definition of the residual (4.44) in combination with the definitions of the bilinear and linear forms (4.11a) and (4.11b), (reverse) integration by parts yields

$$\begin{aligned} r^h(\tilde{v}) = \sum_{K \in \mathcal{T}} \Bigg\{ & \int_{K \cap \Omega} r^h_{\text{volume}} \tilde{v} \, \mathrm{d}V + \int_{K \cap \partial\Omega_N} r^h_{\text{neumann}} \tilde{v} \, \mathrm{d}S \\ & + \int_{K \cap \partial\Omega_D} \left(-r^h_{\text{nitsche}}\right) \partial_n \tilde{v} \, \mathrm{d}S + \int_{K \cap \partial\Omega_D} \frac{\beta}{h_K} r^h_{\text{nitsche}} \tilde{v} \, \mathrm{d}S \\ & + \int_{\partial K \cap \mathcal{F}_{\text{skeleton}}} \left(-r^h_{\text{jump}}\right) \tilde{v} \, \mathrm{d}S \\ & + \int_{\partial K \cap \mathcal{F}_{\text{ghost}}} \gamma_g h_F^{2k-1} \left(-r^h_{\text{ghost}}\right) [\![\partial_n^k \tilde{v}]\!] \, \mathrm{d}S \Bigg\}, \end{aligned} \tag{4.48}$$

where

$$r^h_{\text{volume}} := f + \Delta u^h, \tag{4.49a}$$
$$r^h_{\text{neumann}} := q - \partial_n u^h, \tag{4.49b}$$
$$r^h_{\text{nitsche}} := g - u^h, \tag{4.49c}$$
$$r^h_{\text{jump}} := \tfrac{1}{2}[\![\partial_n u^h]\!], \tag{4.49d}$$
$$r^h_{\text{ghost}} := \tfrac{1}{2}[\![\partial^k_n u^h]\!]. \tag{4.49e}$$

The factor $\frac{1}{2}$ in the jump and ghost terms accounts for the presence of the associated interfaces in two elements. Using the Cauchy-Schwarz inequality it then follows that

$$\begin{aligned}
|r^h(\tilde{v})| \leq \sum_{K\in\mathcal{T}} \Bigg\{ & \|r^h_{\text{volume}}\|_{L^2(K\cap\Omega)} \|\tilde{v}\|_{L^2(K\cap\Omega)} \\
& + \|r^h_{\text{neumann}}\|_{L^2(K\cap\partial\Omega_N)} \|\tilde{v}\|_{L^2(K\cap\partial\Omega_N)} \\
& + \|r^h_{\text{nitsche}}\|_{L^2(K\cap\partial\Omega_D)} \|\partial_n\tilde{v}\|_{L^2(K\cap\partial\Omega_D)} \\
& + \beta h_K^{-1} \|r^h_{\text{nitsche}}\|_{L^2(K\cap\partial\Omega_D)} \|\tilde{v}\|_{L^2(K\cap\partial\Omega_D)} \\
& + \sum_{F\in\mathcal{F}_{\text{skeleton}}} \|r^h_{\text{jump}}\|_{L^2(\partial K\cap F)} \|\tilde{v}\|_{L^2(\partial K\cap F)} \\
& + \sum_{F\in\mathcal{F}_{\text{ghost}}} \gamma_g h_F^{2k-1} \|r^h_{\text{ghost}}\|_{L^2(\partial K\cap F)} \|[\![\partial^k_n\tilde{v}]\!]\|_{L^2(\partial K\cap F)} \Bigg\}.
\end{aligned} \tag{4.50}$$

Using standard interpolation inequalities [11,24] and the definition of the norm (4.27), and noting that we consider the functions $\tilde{v}$ and v to be polynomials, it follows that

$$\|\tilde{v}\|_{L^2(K\cap\Omega)} \lesssim h_K \|\nabla v\|_{L^2(\widetilde{K})} \lesssim h_K \, |||v|||_{\widetilde{K}}, \tag{4.51a}$$
$$\|\tilde{v}\|_{L^2(K\cap\partial\Omega)} \lesssim h_K^{-\frac{1}{2}} \|\tilde{v}\|_{L^2(K)} \lesssim h_K^{\frac{1}{2}} \, |||v|||_{\widetilde{K}}, \tag{4.51b}$$
$$\|\partial_n\tilde{v}\|_{L^2(K\cap\partial\Omega_D)} \lesssim h_K^{-\frac{1}{2}} \|\nabla\tilde{v}\|_{L^2(K)} \lesssim h_K^{-\frac{1}{2}} \, |||v|||_{\widetilde{K}}, \tag{4.51c}$$
$$\|\tilde{v}\|_{L^2(\partial K\cap F)} \lesssim h_K^{-\frac{1}{2}} \|\tilde{v}\|_{L^2(K)} \lesssim h_K^{\frac{1}{2}} \, |||v|||_{\widetilde{K}}, \tag{4.51d}$$

$$\begin{aligned}\|[\![\partial_n^k \tilde{v}]\!]\|_{L^2(\partial K \cap F)} &\lesssim \|\partial_n^k \tilde{v}_K\|_{L^2(\partial K \cap F)} + \|\partial_n^k \tilde{v}_{K'}\|_{L^2(\partial K' \cap F)} \\ &\lesssim h_F^{\frac{1}{2}-k} \, |||v|||_{\widetilde{K} \cup \widetilde{K}'}\,, \end{aligned} \tag{4.51e}$$

where $\widetilde{K}$ is the support extension [24] of the element K and K' is the element that shares the interface F with element K. The residual can then be bounded as

$$\begin{aligned}|r^h(\tilde{v})| \lesssim \sum_{K \in \mathcal{T}} \Bigg\{ & h_K \|r^h_{\text{volume}}\|_{L^2(K \cap \Omega)} + h_K^{\frac{1}{2}} \|r^h_{\text{neumann}}\|_{L^2(K \cap \partial\Omega_N)} \\ & + h_K^{-\frac{1}{2}} \|r^h_{\text{nitsche}}\|_{L^2(K \cap \partial\Omega_D)} + \beta h_K^{-\frac{1}{2}} \|r^h_{\text{nitsche}}\|_{L^2(K \cap \partial\Omega_D)} \\ & + \sum_{F \in \mathcal{F}_{\text{skeleton}}} h_F^{\frac{1}{2}} \|r^h_{\text{jump}}\|_{L^2(\partial K \cap F)} \\ & + \sum_{F \in \mathcal{F}_{\text{ghost}}} \gamma_g h_F^{k-\frac{1}{2}} \|r^h_{\text{ghost}}\|_{L^2(\partial K \cap F)} \Bigg\} \, |||v|||_{\widetilde{K} \cup \widetilde{K}'}\,, \end{aligned} \tag{4.52}$$

which, using the discrete Cauchy-Schwarz inequality can be rewritten as

$$\begin{aligned}\frac{|r^h(v)|^2}{|||v|||^2} \lesssim \sum_{K \in \mathcal{T}} \Bigg\{ & h_K^2 \|r^h_{\text{volume}}\|^2_{L^2(K \cap \Omega)} + h_K \|r^h_{\text{neumann}}\|^2_{L^2(K \cap \partial\Omega_N)} \\ & + h_K^{-1} \|r^h_{\text{nitsche}}\|^2_{L^2(K \cap \partial\Omega_D)} + \beta^2 h_K^{-1} \|r^h_{\text{nitsche}}\|^2_{L^2(K \cap \partial\Omega_D)} \\ & + \sum_{F \in \mathcal{F}_{\text{skeleton}}} h_F \|r^h_{\text{jump}}\|^2_{L^2(\partial K \cap F)} \\ & + \sum_{F \in \mathcal{F}_{\text{ghost}}} \gamma_g^2 h_F^{2k-1} \|r^h_{\text{ghost}}\|^2_{L^2(\partial K \cap F)} \Bigg\}. \end{aligned} \tag{4.53}$$

Using the definition of the residual norm (4.44) it follows that

$$\|r^h\|_{\widehat{V}^h} \lesssim \mathcal{E} = \sqrt{\sum_{K \in \mathcal{T}} \eta_K^2}, \tag{4.54}$$

with the element error indicators defined as

$$\begin{aligned}\eta_K^2 := {} & h_K^2 \|r^h_{\text{volume}}\|^2_{L^2(K\cap\Omega)} + h_K \|r^h_{\text{neumann}}\|^2_{L^2(K\cap\partial\Omega_N)} \\ & + h_K^{-1} \|r^h_{\text{nitsche}}\|^2_{L^2(K\cap\partial\Omega_D)} + \beta^2 h_K^{-1} \|r^h_{\text{nitsche}}\|^2_{L^2(K\cap\partial\Omega_D)} \\ & + \sum_{F\in\mathcal{F}_{\text{skeleton}}} h_F \|r^h_{\text{jump}}\|^2_{L^2(\partial K\cap F)} \\ & + \sum_{F\in\mathcal{F}_{\text{ghost}}} \gamma_g^2 h_F^{2k-1} \|r^h_{\text{ghost}}\|^2_{L^2(\partial K\cap F)}. \end{aligned} \tag{4.55}$$

This error indicator reflects that the total element error for all elements that do not intersect the boundary of the domain is composed of the interior residual and the residual term for the jump in the solution normal derivative across the element interfaces. It is noted that for higher-order continuous discretizations, *i.e.*, $\alpha > 0$, the jump contribution vanishes. For elements that intersect the Neumann boundary, additional error contributions are obtained from the Neumann residual and the ghost penalty residual, while additional Nitsche-related contributions appear for elements intersecting the Dirichlet boundary.

Steady viscous flow

For the Stokes problem introduced in Section 4.2.2, using (reverse) integration by parts, the error indicators in equation (4.42) are obtained by considering the residual (4.44) as

$$\begin{aligned} r^h(\tilde{v}) = \sum_{K\in\mathcal{T}} \Bigg\{ & \int_{K\cap\Omega} \mathbf{r}^h_{\text{int},\mathbf{u}} \cdot \tilde{\mathbf{v}} + r^h_{\text{int},p} \tilde{q} \,\mathrm{d}V + \int_{K\cap\partial\Omega_N} \mathbf{r}^h_{\text{neumann}} \cdot \tilde{\mathbf{v}} \,\mathrm{d}S \\ & + \int_{K\cap\partial\Omega_D} (-\mathbf{r}^h_{\text{nitsche}}) \cdot \left((2\mu\nabla^s\tilde{\mathbf{v}})\,\mathbf{n} + \tilde{q}\mathbf{n}\right) \mathrm{d}S \\ & + \sum_{F\in\mathcal{F}_{\text{skeleton}}} \int_{\partial K\cap F} (-\mathbf{r}^h_{\text{jump}}) \cdot \tilde{\mathbf{v}} \,\mathrm{d}S + \int_{K\cap\partial\Omega_D} \frac{\beta\mu}{h_K} \mathbf{r}^h_{\text{nitsche}} \cdot \tilde{\mathbf{v}} \,\mathrm{d}S \\ & + \sum_{F\in\mathcal{F}_{\text{ghost}}} \int_{\partial K\cap F} \gamma_g \mu h_F^{2k-1} (-\mathbf{r}^h_{\text{ghost}}) \cdot [\![\partial_n^k \tilde{\mathbf{v}}]\!] \,\mathrm{d}S \\ & + \sum_{F\in\mathcal{F}_{\text{skeleton}}} \int_{\partial K\cap F} \frac{\gamma_s h_F^{2k+1}}{\mu} r^h_{\text{skeleton}} [\![\partial_n^k \tilde{q}]\!] \,\mathrm{d}S \Bigg\}, \end{aligned} \tag{4.56}$$

where $\tilde{v} = v - \Pi^h v = (\tilde{\mathbf{v}}, \tilde{q})$ and

$$\mathbf{r}^h_{\text{int},\mathbf{u}} := \mathbf{f} + \nabla \cdot \left(2\mu \nabla^s \mathbf{u}^h\right) - \nabla p^h, \tag{4.57a}$$

$$r^h_{\text{int},p} := \nabla \cdot \mathbf{u}^h, \tag{4.57b}$$

$$\mathbf{r}^h_{\text{neumann}} := \mathbf{t} - \left(2\mu \nabla^s \mathbf{u}^h\right) \mathbf{n} + p^h \mathbf{n}, \tag{4.57c}$$

$$\mathbf{r}^h_{\text{nitsche}} := \mathbf{g} - \mathbf{u}^h, \tag{4.57d}$$

$$\mathbf{r}^h_{\text{jump}} := \tfrac{1}{2} [\![\left(2\mu \nabla^s \mathbf{u}^h\right) \mathbf{n}]\!], \tag{4.57e}$$

$$\mathbf{r}^h_{\text{ghost}} := \tfrac{1}{2} [\![\partial_n^k \mathbf{u}^h]\!], \tag{4.57f}$$

$$r^h_{\text{skeleton}} := \tfrac{1}{2} [\![\partial_n^k p^h]\!]. \tag{4.57g}$$

Application of the Cauchy-Schwarz inequality gives

$$\begin{aligned}
\left|r^h(\tilde{v})\right| \leq \sum_{K \in \mathcal{T}} \Bigg\{ & \|\mathbf{r}^h_{\text{int},\mathbf{u}}\|_{L^2(K\cap\Omega)} \|\tilde{\mathbf{v}}\|_{L^2(K\cap\Omega)} + \|r^h_{\text{int},p}\|_{L^2(K\cap\Omega)} \|\tilde{q}\|_{L^2(K\cap\Omega)} \\
& + \|\mathbf{r}^h_{\text{neumann}}\|_{L^2(K\cap\partial\Omega_N)} \|\tilde{\mathbf{v}}\|_{L^2(K\cap\partial\Omega_N)} \\
& + \|\mathbf{r}^h_{\text{nitsche}}\|_{L^2(K\cap\partial\Omega_D)} \\
& \qquad \left(2\mu \| \left(\nabla^s \tilde{\mathbf{v}}\right) \mathbf{n}\|_{L^2(K\cap\partial\Omega_D)} + \|\tilde{q}\|_{L^2(K\cap\partial\Omega_D)}\right) \\
& + \sum_{F \in \mathcal{F}_{\text{skeleton}}} \|\mathbf{r}^h_{\text{jump}}\|_{L^2(\partial K\cap F)} \|\tilde{\mathbf{v}}\|_{L^2(\partial K\cap F)} \\
& + \mu\beta h_K^{-1} \|\mathbf{r}^h_{\text{nitsche}}\|_{L^2(K\cap\partial\Omega_D)} \|\tilde{\mathbf{v}}\|_{L^2(K\cap\partial\Omega_D)} \\
& + \sum_{F \in \mathcal{F}^K_{\text{ghost}}} \mu\gamma_g h_F^{2k-1} \|\mathbf{r}^h_{\text{ghost}}\|_{L^2(\partial K\cap F)} \|[\![\partial_n^k \tilde{\mathbf{v}}]\!]\|_{L^2(\partial K\cap F)} \\
& + \sum_{F \in \mathcal{F}^K_{\text{skeleton}}} \mu^{-1}\gamma_s h_F^{2k+1} \|r^h_{\text{skeleton}}\|_{L^2(\partial K\cap F)} \\
& \qquad \|[\![\partial_n^k \tilde{q}]\!]\|_{L^2(\partial K\cap F)} \Bigg\},
\end{aligned} \tag{4.58}$$

which, using the inequalities (4.51) and

$$\|\tilde{q}\|_{L^2(K\cap\Omega)} \lesssim \|q\|_{L^2(K)}, \tag{4.59a}$$

$$\|\tilde{q}\|_{L^2(K\cap\partial\Omega)} \lesssim h_K^{-\frac{1}{2}}\|\tilde{q}\|_{L^2(K)} \lesssim h_K^{-\frac{1}{2}}\|q\|_{L^2(K)}, \tag{4.59b}$$

$$\begin{aligned}\|[\![\partial_n^k \tilde{q}]\!]\|_{L^2(\partial K\cap F)} &\lesssim \|\partial_n^k \tilde{q}_K\|_{L^2(\partial K\cap F)} + \|\partial_n^k \tilde{q}_{K'}\|_{L^2(\partial K'\cap F)} \\ &\lesssim h_F^{-\frac{1}{2}-k}\|q\|_{L^2(\widetilde{K}\cup\widetilde{K'})},\end{aligned} \tag{4.59c}$$

can be rewritten as

$$\begin{aligned}\left|r^h(\tilde{v})\right| \lesssim \sum_{K\in\mathcal{T}} \Bigg\{ &\mu^{-\frac{1}{2}} h_K \|\mathbf{r}^h_{\text{int},\mathbf{u}}\|_{L^2(K\cap\Omega)} + \mu^{\frac{1}{2}}\|r^h_{\text{int},p}\|_{L^2(K\cap\Omega)} \\ &+ \mu^{-\frac{1}{2}} h_K^{\frac{1}{2}}\|\mathbf{r}^h_{\text{neumann}}\|_{L^2(K\cap\partial\Omega_N)} + 3\mu^{\frac{1}{2}} h_K^{-\frac{1}{2}}\|\mathbf{r}^h_{\text{nitsche}}\|_{L^2(K\cap\partial\Omega_D)} \\ &+ \sum_{F\in\mathcal{F}_{\text{skeleton}}} \mu^{-\frac{1}{2}} h_K^{\frac{1}{2}}\|\mathbf{r}^h_{\text{jump}}\|_{L^2(\partial K\cap F)} + \mu^{\frac{1}{2}}\beta h_K^{-\frac{1}{2}}\|\mathbf{r}^h_{\text{nitsche}}\|_{L^2(K\cap\partial\Omega_D)} \\ &+ \sum_{F\in\mathcal{F}^K_{\text{ghost}}} \mu^{\frac{1}{2}}\gamma_g h_F^{k-\frac{1}{2}}\|\mathbf{r}^h_{\text{ghost}}\|_{L^2(\partial K\cap F)} \\ &+ \sum_{F\in\mathcal{F}^K_{\text{skeleton}}} \mu^{-\frac{1}{2}}\gamma_s h_F^{k+\frac{1}{2}}\|r^h_{\text{skeleton}}\|_{L^2(\partial K\cap F)} \Bigg\} |||v|||_{\widetilde{K}\cup\widetilde{K'}}\,.\end{aligned} \tag{4.60}$$

Note that the factor 3 in front of the Nitsche residual results from the fact that both terms $2\mu\|\left(\nabla^s\tilde{\mathbf{v}}\right)\mathbf{n}\|_{L^2(K\cap\partial\Omega_D)}$ and $\|\tilde{q}\|_{L^2(K\cap\partial\Omega_D)}$ are bound by the same norm. Following the same steps as for the heat conduction problem we then obtain the element error indicators as

$$\begin{aligned}\eta_K^2 = &\mu^{-1}h_K^2\|\mathbf{r}^h_{\text{int},\mathbf{u}}\|^2_{L^2(K\cap\Omega)} + \mu\|r^h_{\text{int},p}\|^2_{L^2(K\cap\Omega)} \\ &+ \mu^{-1}h_K\|\mathbf{r}^h_{\text{neumann}}\|^2_{L^2(K\cap\partial\Omega_N)} + 9\mu h_K^{-1}\|\mathbf{r}^h_{\text{nitsche}}\|^2_{L^2(K\cap\partial\Omega_D)} \\ &+ \sum_{F\in\mathcal{F}_{\text{skeleton}}} \mu^{-1}h_K\|\mathbf{r}^h_{\text{jump}}\|^2_{L^2(\partial K\cap F)} + \mu\beta^2 h_K^{-1}\|\mathbf{r}^h_{\text{nitsche}}\|^2_{L^2(K\cap\partial\Omega_D)} \\ &+ \sum_{F\in\mathcal{F}^K_{\text{ghost}}} \mu\gamma_g^2 h_F^{2k-1}\|\mathbf{r}^h_{\text{ghost}}\|^2_{L^2(\partial K\cap F)} \\ &\sum_{F\in\mathcal{F}^K_{\text{skeleton}}} \mu^{-1}\gamma_s^2 h_F^{2k+1}\|r^h_{\text{skeleton}}\|^2_{L^2(\partial K\cap F)}.\end{aligned} \tag{4.61}$$

Compared to the error indicators for the heat conduction problem, we here get one additional term to represent the error in the balance of mass, *i.e.*, $\|r^h_{\text{int},p}\|_{L^2(K\cap\Omega)}$, and one term related to the skeleton-stabilization, *i.e.*, $\|r^h_{\text{skeleton}}\|_{L^2(\partial K\cap F)}$. Moreover, note that the mass and momentum balance terms are scaled with $\mu^{-\frac{1}{2}}$ and $\mu^{\frac{1}{2}}$, respectively, in order to be dimensionally-consistent with the energy norm (4.31).

4.3.2 Adaptive solution procedure

We employ the residual-based error estimator introduced above in an iterative mesh refinement procedure. In each iteration, for the given mesh we solve the Galerkin problem (4.8) and subsequently compute the element-wise error indicators (4.42) (and the corresponding estimator). Based on the indicators, certain elements are then refined, after which the procedure is repeated on the refined mesh. These iterations are continued until a stopping criterion is satisfied.

We consider Dörfler marking [243] to select the elements to be refined. In this marking strategy, the marked set, $\mathcal{M}$, is defined as a minimal set of elements such that

$$\sqrt{\sum_{K\in\mathcal{M}} \eta_K^2} \geq \lambda \sqrt{\sum_{K\in\mathcal{T}} \eta_K^2} = \lambda\mathcal{E}, \tag{4.62}$$

with λ a selected fraction of the error estimator. For the considered (truncated) hierarchical spline meshes, refining elements does not necessarily result in a refinement of the approximation space [176, 213]. To ensure that the approximation space is refined, an additional step is required in which a refinement mask $\widetilde{\mathcal{M}} \supset \mathcal{M}$ is defined. To determine the refinement mask, for each element K in the marked set $\mathcal{M}$ we determine the support extension

$$\widetilde{\mathcal{K}} = \bigcup \{\text{supp}(N) \mid \text{supp}(N) \cap K \neq \emptyset,\ N \in \mathcal{H}(\mathcal{T})\}, \tag{4.63}$$

and then refine the elements in each support extension which are not smaller than the element K, *i.e.*,

$$\widetilde{\mathcal{M}} = \bigcup_{K\in\mathcal{M}} \left\{ K' \in \widetilde{\mathcal{K}} \mid K' \in \cup_{\ell=0}^{\ell_K} \mathcal{T}^\ell,\ K \in \mathcal{T}^{\ell_K} \right\}. \tag{4.64}$$

During the element refinement procedure the geometry approximation is not altered, as illustrated in Figure 4.4. In our implementation, the bisectioning depth used to determine the integration subcells is lowered under refinement, resulting in the preservation of the integration subcells under refinement. This ensures that the boundary of the segmented geometry is invariant under mesh

refinement. A consequence of this choice is that an element can only be refined up to the level of the integration subcells. Elements requiring refinement beyond the level of the integration subcells are discarded from the refinement list, and the adaptive refinement procedure is stopped if there are no more elements that can be refined.

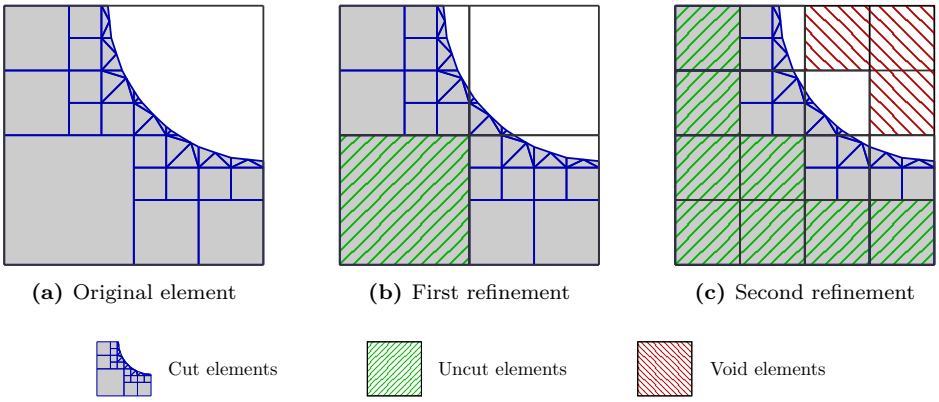
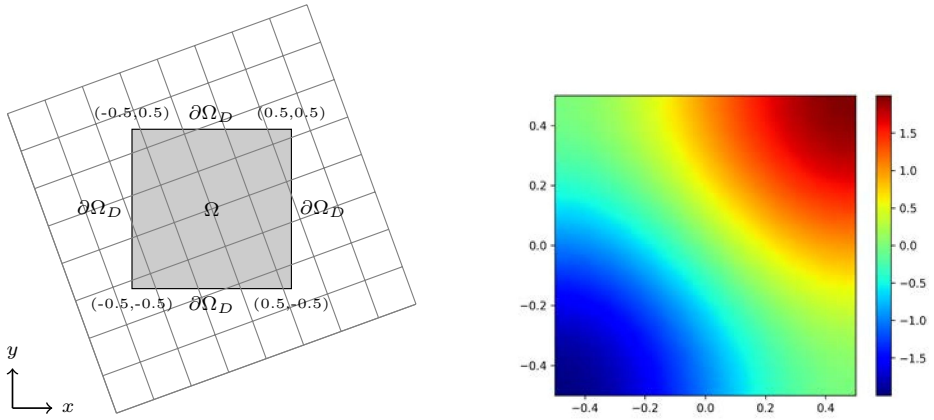


Figure 4.4: Illustration of the refinement procedure for cut elements. The original element is subdivided in integration subcells (blue borders) using the recursive bisectioning procedure detailed in Ref. [86]. At the lowest level of bisectioning, a triangulation procedure is employed. After one refinement of the original element, the original element is split into 4 elements, of which one is now an uncut element and the other three are cut elements. The bisectioning depth for the determination of the integration subcells is reduced by one level compared to the original element, so that the subcells remain identical under the element refinement operation. After one further refinement step, each of the four elements in the first refinement is further refined, resulting now also in elements that are void and are hence discarded from the background mesh.

4.4 Benchmark simulations

In this section we assess the developed residual-based adaptive refinement technique on a range of numerical experiments. For both the heat conduction problem (Section 4.4.1) and the viscous flow problem (Section 4.4.2), both singular and non-singular test cases are considered. For all simulations exact reference solutions are available, allowing for a rigorous study of the stability and accuracy of the developed adaptive immersed isogeometric analysis framework. For all simulations the octree subdivision depth is set equal to the desired maximum number of refinements (see Section 4.3.2) and the refinement threshold is set to $\lambda = 0.8$. Throughout this section, the problems are considered to be in dimensionless form.



(a) **(b)**

Figure 4.5: (a) Problem setup, and (b) the exact solution $u(x, y)$, Eq. (4.65), for the Laplace problem on the unit square domain.

4.4.1 Steady heat conduction

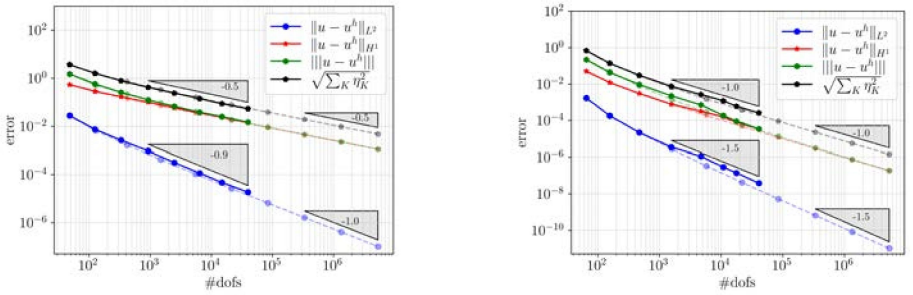
We consider the two-dimensional heat conduction problem on a unit square and on a star-shaped domain with a smooth exact solutions, and on a domain with a re-entrant corner, for which the exact solution has a reduced regularity (Section 4.4.1). The problems are discretized with linear ($k = 1$) and quadratic ($k = 2$) (TH)B-splines using both uniform and adaptive refinement. All examples consider a non-conforming ambient mesh positioned at an angle of 20 degrees (see Figure 4.5a and Figure 4.9a), unless specified otherwise. The empirically selected Nitsche and ghost penalty parameters are set to $\beta = 50$ and $\gamma_g = 10^{-(k+2)}$, respectively.

Unit square

Let $\Omega = [-\frac{1}{2}, \frac{1}{2}]^2$ be a unit square with Dirichlet boundary $\partial\Omega_D$ (see Figure 4.5a). We define the exact solution of the problem (4.10) as

$$u(x, y) = \sin(\pi x) + \sin(\pi y), \tag{4.65}$$

which is shown in Figure 4.5b. The heat source f corresponding to this exact solution is equal to zero, and the Dirichlet data is set to $g = u|_{\partial\Omega_D}$, matching the exact solution.



(a) $k = 1$ **(b)** $k = 2$

Figure 4.6: Error convergence results for the Laplace problem on the unit square domain under residual-based adaptive refinement (solid) and uniform refinement (dashed) for linear ($k = 1$) and quadratic ($k = 2$) basis functions.

Figure 4.6 shows error-analysis results using both uniform and adaptive refinements for the linear case (Figure 4.6a) and for the quadratic case (Figure 4.6b). Both refinement procedures start from an initial mesh consisting of 8×8 elements covering the ambient domain $[-1, 1]^2$. Optimal convergence rates are obtained for both the error in the L^2-norm (*i.e.*, $\mathcal{O}(n^{-\frac{1}{2}(k+1)})$) and in the H^1-norm (*i.e.*, $\mathcal{O}(n^{-\frac{1}{2}k})$), with n denoting the number of degrees of freedom. Moreover, as the number of refinement steps increases, the energy norm and H^1-norm of the error coincide, indicating that the error is dominated by the H^1-semi-norm contribution in Eq. (4.27). The estimator (4.42) is observed to converge at the same rate as the energy norm, bounding the energy norm from above, consistent with Eq. (4.45). Because of the smooth solution (4.65), the refinement pattern following from the adaptive refinement procedure closely resembles the uniform refinements, as observed from the close correspondence between the error results for the uniform and adaptive simulations in Figure 4.6.

Star-shaped domain

To study the sensitivity of the adaptive simulation framework to the cut-cell configurations, we consider the star-shaped domain shown in Figure 4.7a for various orientation angles ϑ. The star-shaped domain is constructed using the level set function

$$\psi(x, y) = R_1 + R_2 \sin(n_{\text{fold}} \arctan2(y, x)) - \sqrt{x^2 + y^2},$$

with $R_1 = 0.6$, $R_2 = 0.2$ and $n_{\text{fold}} = 5$ [60]. On the boundary of the domain, the Dirichlet data is set equal to the same exact solution (4.65) as in the previous example. For all orientations, an initial mesh of 10×10 elements covering the ambient domain $[-1, 1]^2$ is considered, after which local refinements using second-order THB-splines are performed until the smallest elements have been refined six times.

Figures 4.7b–4.7f show the error $u - u^h$ after completion of the refinement procedure. These figures convey that both the error and the refinement pattern are similar for all orientations. This is corroborated by the results in Figure 4.8, which indicates that both the number of degrees of freedom and the errors (in various norms) are insensitive to the orientation angle.

Re-entrant corner

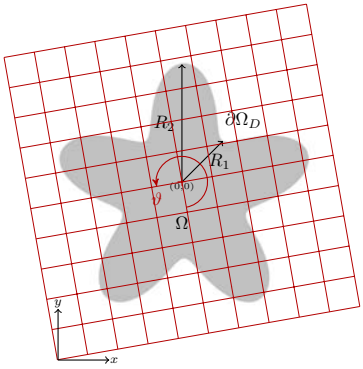
To study the behavior of the adaptive simulation strategy for problems with (weakly) singular solutions, we consider a domain with a re-entrant corner, as shown in Figure 4.9a. The data on the Dirichlet and Neumann boundaries, $u|_{\partial\Omega_D} = g = 0$ and $\partial_n u|_{\partial\Omega_N} = q$, is set to match the exact solution [206, 213]

$$u(x, y) = (x^2 + y^2)^{\frac{1}{3}} \cos\left(\frac{2}{3}\text{arctan2}(x - y, x + y)\right). \tag{4.66}$$

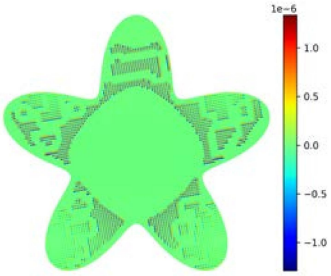
The convergence behavior of the L^2-error, H^1-error, energy norm error (4.27) and the residual-based estimator (4.55) is studied for uniform refinement and residual-based adaptive refinement. Both refinement procedures start from an initial mesh of 10×10 elements formed on the ambient domain $[-\frac{3}{2}, \frac{3}{2}]^2$. The convergence results for first and second order B-splines are shown in Figure 4.10a and Figure 4.10b, respectively.

Under uniform refinement, the convergence rates are impeded by the weak singularity at the re-entrant corner. For the L^2-error and H^1-error, suboptimal rates of $\mathcal{O}(n^{-\frac{2}{3}})$ and $\mathcal{O}(n^{-\frac{1}{3}})$ are observed, which is in agreement with the expected rates [244]. These rates are independent of the order of the approximation, as the regularity of the exact solution limits the rate already for the linear case. As for the cases considered above, the energy error and estimator follow the convergence of the H^1-error.

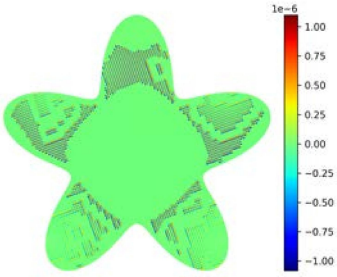
Using the adaptive refinement strategy with linear basis functions, the optimal rates of $\mathcal{O}(n^{-1})$ and $\mathcal{O}(n^{-\frac{1}{2}})$ are recovered for the L^2-error and H^1-error, respectively. For the quadratic case, rates that are substantially higher than the theoretical rates are observed. We attribute this to pre-asymptotic behavior, in which the refinement pattern as shown in Figure 4.10 is strongly focused on the re-entrant corner singularity. After the first two steps, the errors become dominated by the singularity at the re-entrant corner, which results in the further



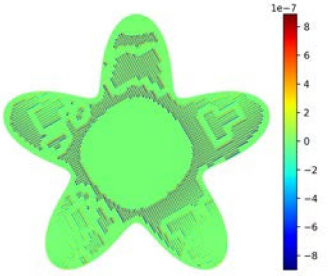
(a) Star shaped domain



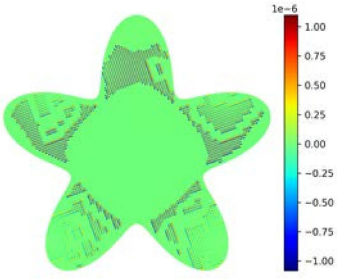
(b) $\vartheta = 10$



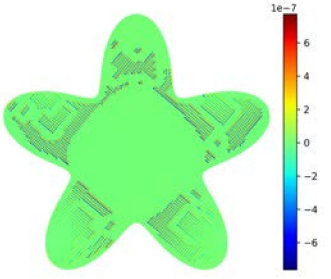
(c) $\vartheta = 20$



(d) $\vartheta = 30$

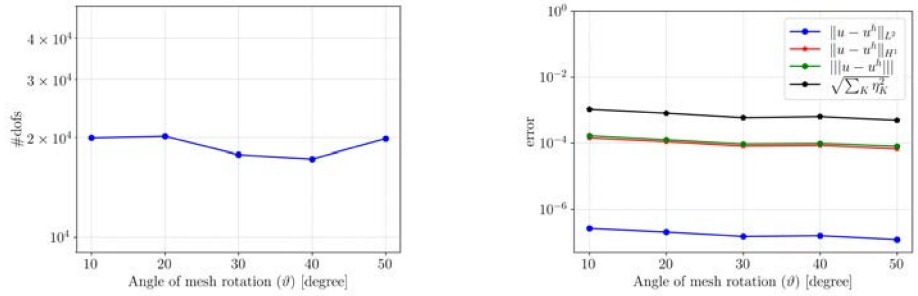


(e) $\vartheta = 40$



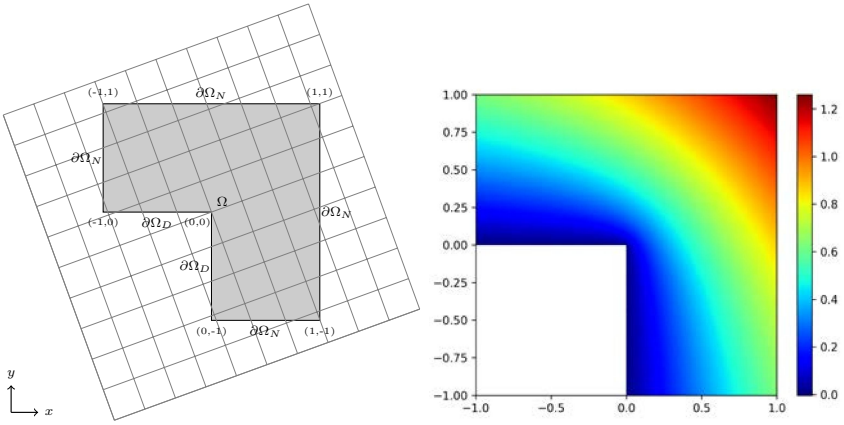
(f) $\vartheta = 50$

Figure 4.7: (a) Problem setup, and (b)-(f) contour plots of the error, $u - u^h$, for the Laplace problem on the star shaped domain at the end of 6 adaptive refinement steps for different angles of mesh rotation ϑ.



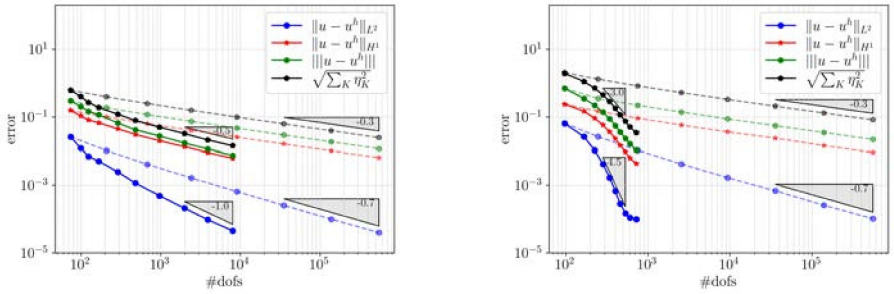
(a) **(b)**

Figure 4.8: (a) Degrees of freedom, and (b) error norms for the Laplace problem on the star shaped domain after 6 adaptive refinement steps for different angles of mesh rotation ϑ.



(a) **(b)**

Figure 4.9: (a) Problem setup, and (b) the exact solution $u(x, y)$, Eq. (4.66), for the Laplace problem on the re-entrant corner domain.



(a) $k = 1$ **(b)** $k = 2$

Figure 4.10: Error convergence results for the Laplace problem on the re-entrant corner domain under residual-based adaptive refinement (solid) and uniform refinement (dashed) for linear ($k = 1$) and quadratic ($k = 2$) basis functions.

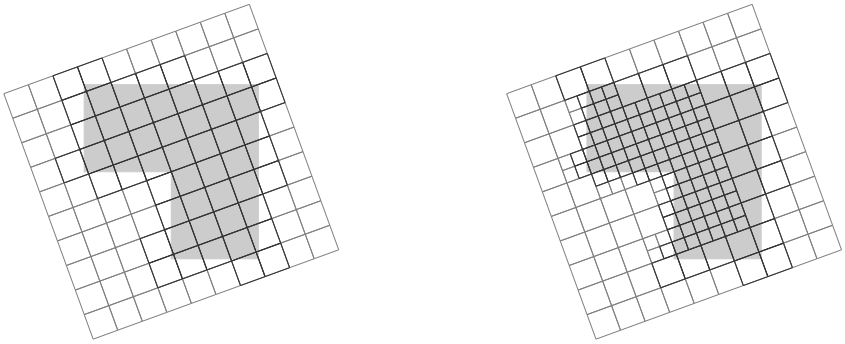
refinement of the few elements in the vicinity of the corner. These refinements do reduce the error, while they only introduce a limited number of additional degrees of freedom. The observed flattening in the rate of the L^2-error in the quadratic case is caused by the refinement reaching the maximum level in the elements in the corner, which causes the marking strategy to tag elements that do not carry the largest error contributions.

4.4.2 Steady viscous flow

We regard the two-dimensional Stokes flow problem on a quarter annulus ring domain with a smooth solution and on the above-introduced re-entrant corner domain with a singular solution. We consider equal-order discretizations for the velocity and pressure fields using optimal regularity (TH)B-splines of degree $k = 1$ and $k = 2$. For the Nitsche and ghost-penalty parameter the same settings are used as for the Laplace problem considered above, *i.e.*, $\beta = 50$ and $\gamma_g = 10^{-(k+2)}$. In addition, a skeleton-penalty parameter of $\gamma_s = 10^{-(k+1)}$ is used for all simulations.

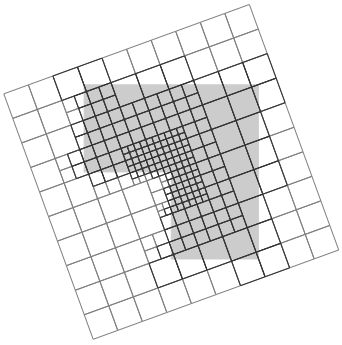
Quarter annulus ring

We consider an annulus ring domain $\Omega = \{(x, y) \in \mathbb{R}^2_{>0} : R_1^2 < x^2 + y^2 < R_2^2\}$ with inner radius $R_1 = 1$, outer radius $R_2 = 4$, Dirichlet boundary $\partial\Omega_D$ and Neumann boundary $\partial\Omega_N$, as shown in Figure 4.12a. The Dirichlet data $\boldsymbol{g}$ and Neumann data $\boldsymbol{t}$ are prescribed in accordance with the divergence-free

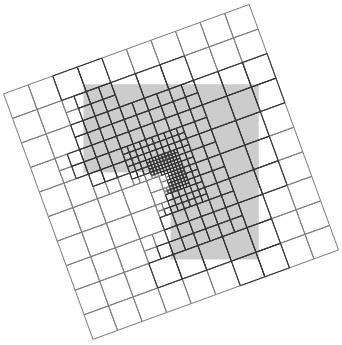


(a) Initial mesh

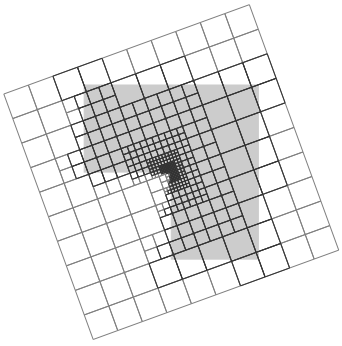
(b) Step 1



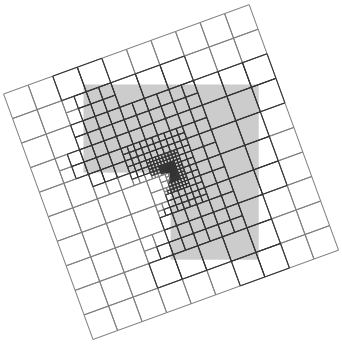
(c) Step 2



(d) Step 3



(e) Step 4



(f) Step 5

Figure 4.11: Evolution of the mesh using the adaptive refinement procedure for the Laplace problem on the re-entrant corner domain using $k = 2$.

manufactured solution [204]

$$u_1(x,y) = 10^{-6}x^2y^4(x^2+y^2-1)(x^2+y^2-16)\\ \left(5x^4+18x^2y^2-85x^2+13y^4-153y^2+80\right), \tag{4.67a}$$

$$u_2(x,y) = 10^{-6}xy^5(x^2+y^2-1)(x^2+y^2-16)\\ \left(102x^2+34y^2-10x^4-12x^2y^2-2y^4-32\right), \tag{4.67b}$$

$$p(x,y) = 10^{-7}xy(y^2-x^2)(x^2+y^2-16)^2\\ (x^2+y^2-1)^2\exp(14(x^2+y^2)^{-1/2}). \tag{4.67c}$$

The body force $\boldsymbol{f}$ in the Stokes problem (4.12) is determined based on this manufactured solution, with the viscosity set to $\mu = 1$.

Figure 4.13 displays the convergence results for the annulus ring problem. Both the uniform refinement results and the adaptive refinement results are obtained starting from a 9×9 uniform mesh on the ambient domain $[0,R_2]^2 = [0,4]^2$. A good resemblance with the optimal rates of $\mathcal{O}(n^{-\frac{1}{2}k})$ in the velocity H^1-norm and pressure L^2-norm is observed, and, as expected, the rate of the velocity L^2-error is $\mathcal{O}(n^{-\frac{1}{2}(k+1)})$. The error in the energy norm (4.31) is observed to converge with the same rate as the H^1-norm velocity error and L^2-norm pressure error, which is in agreement with the definition of the energy norm. As expected, the error estimator bounds the error in the energy norm from above.

Although optimal convergence rates are obtained using uniform refinements, the adaptive refinement procedure is observed to substantially improve the error for a fixed number of degrees of freedom. This behavior is explained by the observed refinement patterns, as shown in Figure 4.14. Although the exact solution (4.67) is smooth, in particular the steep gradients in the velocity solution lead to local refinements. This effectively reduces the error when compared to a uniform refinement with a similar number of degrees of freedom.

Re-entrant corner

As a final benchmark problem we consider the Stokes problem (4.12) on the re-entrant corner domain with mixed Dirichlet and Neumann boundaries introduced above, as shown in Figure 4.9a. The weakly singular exact solution is taken from Ref. [245] as

$$u_1(x,y) = \quad R^\alpha\big[\sin(\theta)\frac{\partial\psi}{\partial\theta} - (1+\alpha)\cos(\theta)\psi\big] \tag{4.68a}$$

$$u_2(x,y) = -R^\alpha\big[\cos(\theta)\frac{\partial\psi}{\partial\theta} + (1+\alpha)\sin(\theta)\psi\big] \tag{4.68b}$$

$$p(x,y) = -\frac{R^{\alpha-1}}{1-\alpha}\Big[(1+\alpha)^2\frac{\partial\psi}{\partial\theta} + \frac{\partial^3\psi}{\partial\theta^3}\Big], \tag{4.68c}$$

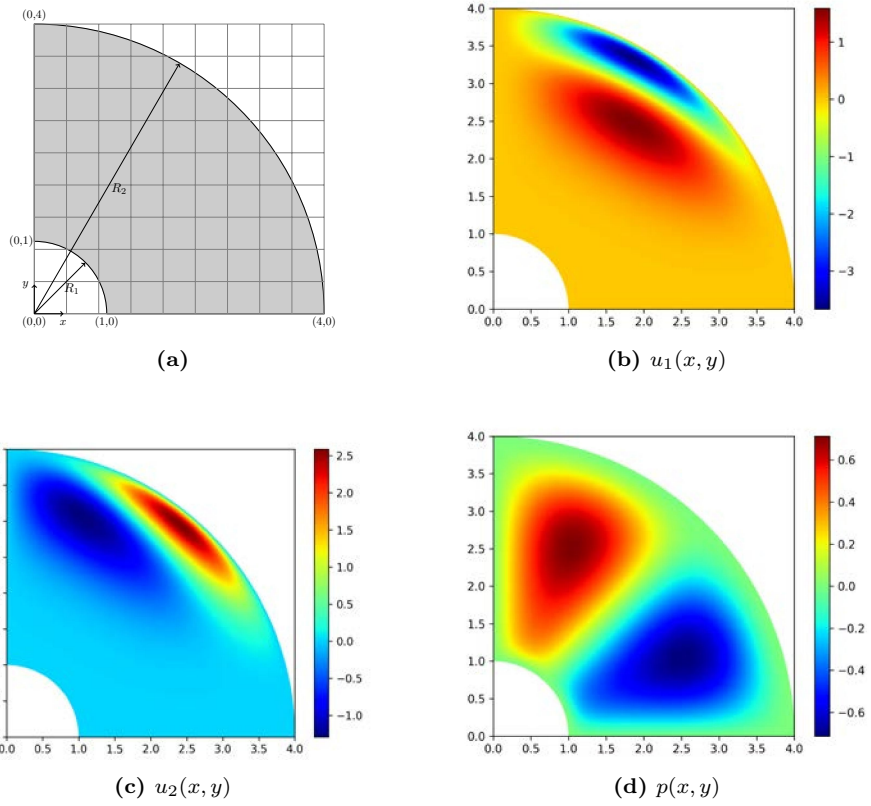
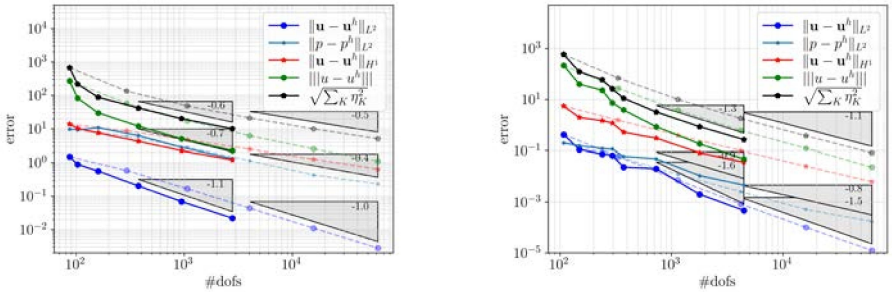


Figure 4.12: (a) Problem setup, and (b)-(d) the exact solution components $u_1(x, y)$, $u_2(x, y)$, and $p(x, y)$, defined in Eq. (4.67), for the Stokes problem on the quarter annulus ring domain.



(a) $k = 1$ **(b)** $k = 2$

Figure 4.13: Error convergence results for the Stokes problem on the quarter annulus ring domain under residual-based adaptive refinement (solid) and uniform refinement (dashed) for linear ($k = 1$) and quadratic ($k = 2$) basis functions.

with constants $\alpha = 856399/1572864$ and $\omega = \frac{3}{2}\pi$, and with

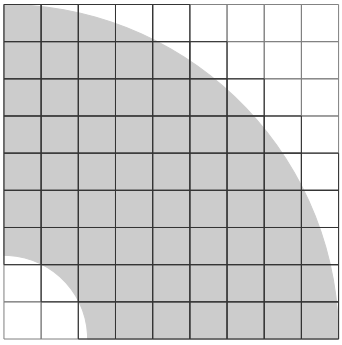
$$\begin{aligned}\psi(\theta) =& \frac{\cos(\alpha\omega)}{1+\alpha}\sin((1+\alpha)\theta) - \frac{\cos(\alpha\omega)}{1-\alpha}\sin((1-\alpha)\theta) \\ &+ \cos((1-\alpha)\theta) - \cos((1+\alpha)\theta).\end{aligned} \tag{4.69}$$

The exact pressure and velocity fields are illustrated in Figure 4.15. The corresponding Stokes problem (4.12) is considered with the viscosity set to $\mu = 1$, no body force $\mathbf{f} = \mathbf{0}$, a no slip condition on Γ_D, such that $\mathbf{u}_D = \mathbf{0}$, and the Neumann data $\mathbf{g}$ on Γ_N matching the exact solution.

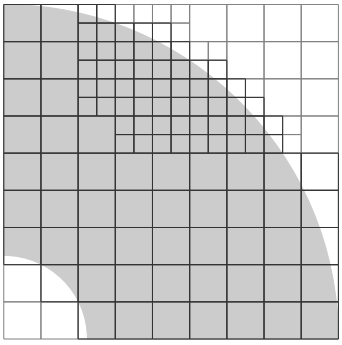
Figure 4.16 displays the error convergence results obtained using uniform and adaptive refinements, for both linear and quadratic (TH)B-splines. As for the Laplace case, the weak singularity in the exact solution (4.68) limits the convergence rate when uniform refinements are considered. Using adaptive mesh refinement results in a recovery of the optimal rates in the case of linear basis functions, with even higher rates observed for the quadratic splines on account of the highly-focussed refinements resulting from the residual-based error estimator as observed in Figure 4.17.

4.5 Scan-based simulations

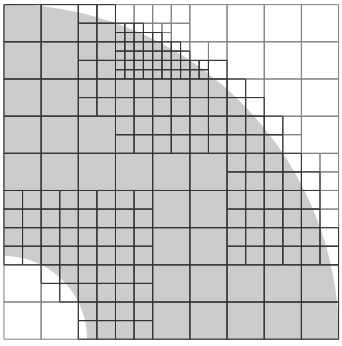
In this section we apply the developed adaptive immersed isogeometric analysis framework in the context of scan-based analysis. We consider the viscous flow



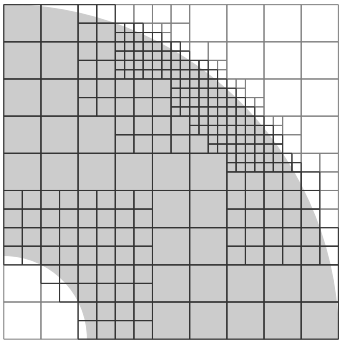
(a) Initial mesh



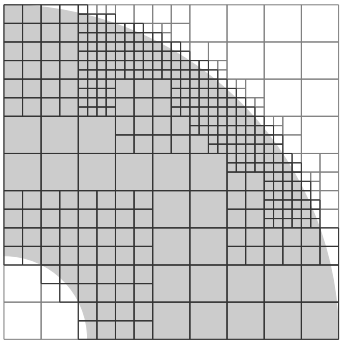
(b) Step 1



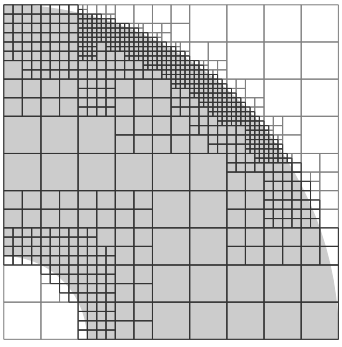
(c) Step 2



(d) Step 3

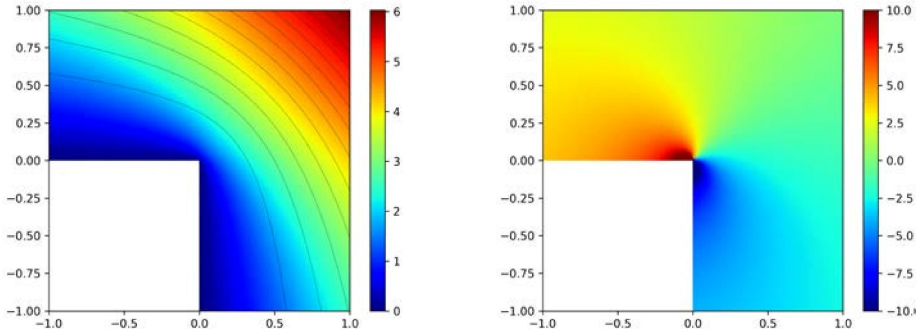


(e) Step 4



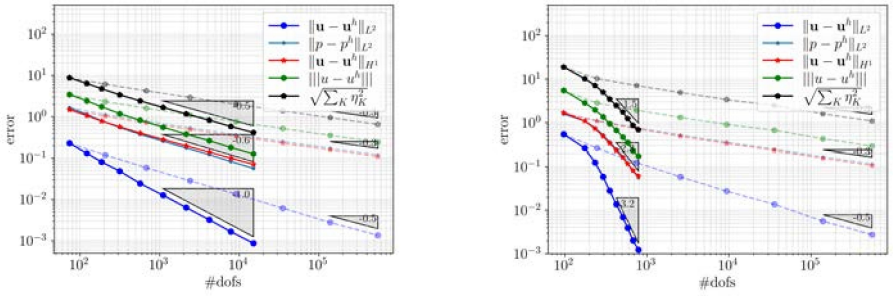
(f) Step 5

Figure 4.14: Evolution of the mesh using the adaptive refinement procedure for the Stokes problem on the quarter annulus ring domain using $k = 2$.



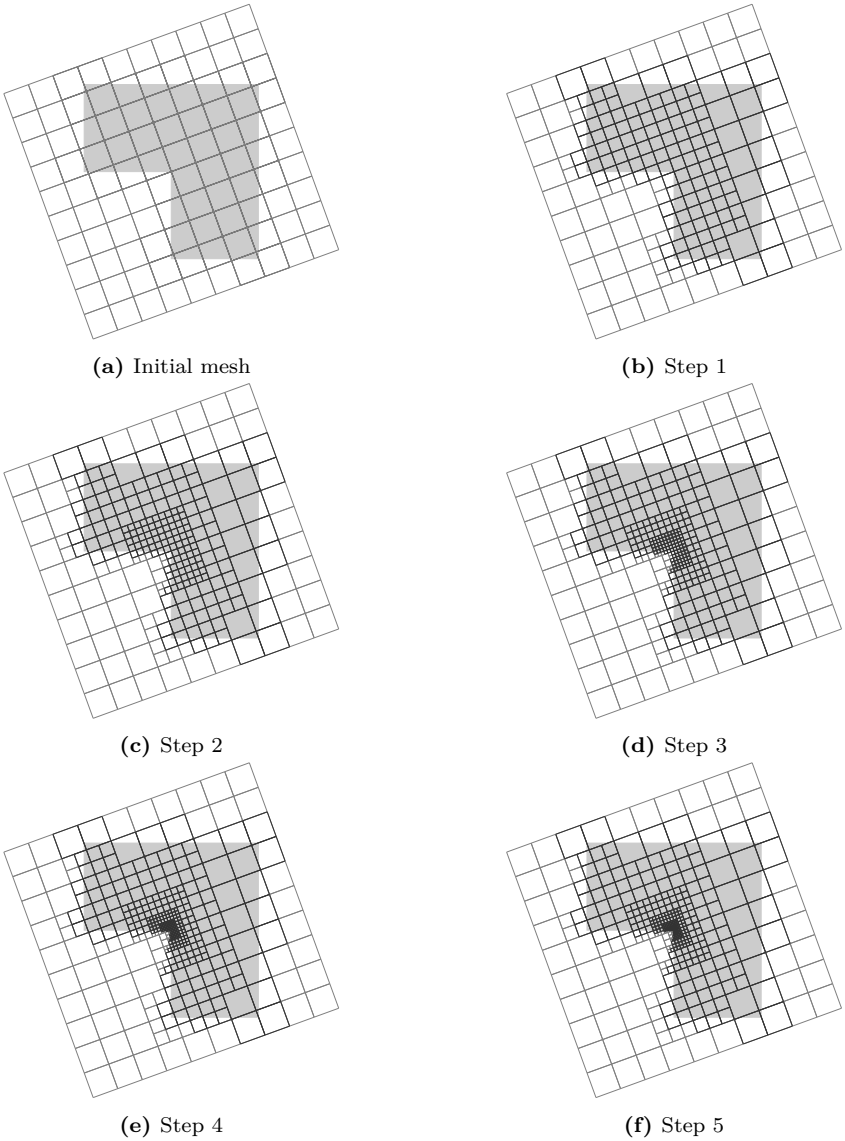
(a) $|\mathbf{u}|$ **(b)** p

Figure 4.15: (a) Velocity magnitude and streamlines, and (b) pressure for the exact solution (4.68) to the Stokes problem on the re-entrant corner domain. Because of the singular solution, the pressure color bar is truncated to the range -10 and 10.



(a) $k = 1$ **(b)** $k = 2$

Figure 4.16: Error convergence results for the Stokes problem on the re-entrant corner domain under residual-based adaptive refinement (solid) and uniform refinement (dashed) for linear ($k = 1$) and quadratic ($k = 2$) basis functions.

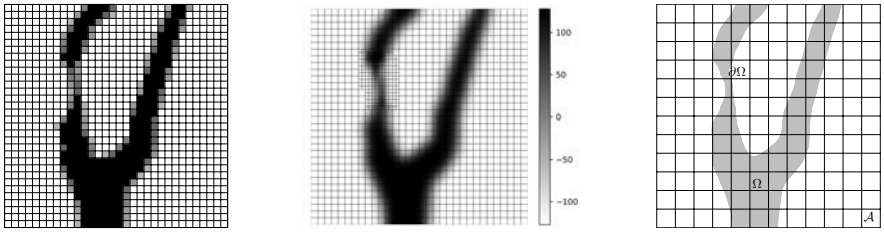


(a) Initial mesh **(b)** Step 1

(c) Step 2 **(d)** Step 3

(e) Step 4 **(f)** Step 5

Figure 4.17: Evolution of the mesh using the adaptive refinement procedure for the Stokes problem on the re-entrant corner domain using $k = 2$.



(a) **(b)** **(c)**

Figure 4.18: Illustration of the scan-based analysis workflow. The original grayscale image in panel (a) is converted to a level set function, shown in panel (b), which is constructed using the topology-preserving segmentation algorithm of Ref. [87]. The trimmed geometry, shown in panel (c), is then extracted using the recursive bi-sectioning strategy with mid-point tessellation of Ref. [86].

problem on a two-dimensional image domain and on a three-dimensional patient-specific problem based on a μCT-scan of a carotid artery, represented by grayscale voxels. The primary purpose of the two-dimensional setting is to test the scan-based analysis framework. For all simulations, the octree subdivision depth is set equal to 8 in two dimensions and 3 in three dimensions. The refinement threshold related to the Dörfler marking is set to $\lambda = 0.8$.

Our scan-based analysis workflow is illustrated in Figure 4.18. The first step in this workflow is to smoothen the original grayscale voxel data using a convolution operation on a B-spline basis formed on the voxel grid [58]. Since this smoothing operator behaves as a Gaussian filter, geometric features that are similar in size to the voxels can be lost [87]. To avoid this loss of features, the topology-preservation procedure proposed in Ref. [87] is employed. This procedure locally refines the convolution basis to retain small geometric features in the smoothing procedure. Once the smooth level set representation has been obtained, the octree segmentation procedure with mid-point tessellation of Ref. [86] is used to obtain the immersed geometry represented on an ambient domain mesh. It is important to note that this ambient domain mesh, on which the solution to the flow problem is computed, can be chosen independently of the voxel size, and hence it is independent of the mesh on which the level set function is constructed.

The considered computational domain is illustrated in Figure 4.19a. Neumann conditions are imposed on the inflow and outflow boundaries, with the traction on the inflow boundary acting in the normal direction with a pressure magnitude, $\boldsymbol{t} = -\bar{p}\boldsymbol{n}$. Homogeneous Dirichlet conditions are imposed along the immersed boundaries to mimic the no slip condition. In all simulations we consider second-order ($k = 2$) (TH)B-splines and set the stabilization parameters to $\beta = 100$, $\gamma_g = 10^{-(k+2)}$ and $\gamma_s = 10^{-(k+1)}$, which have been determined empirically.

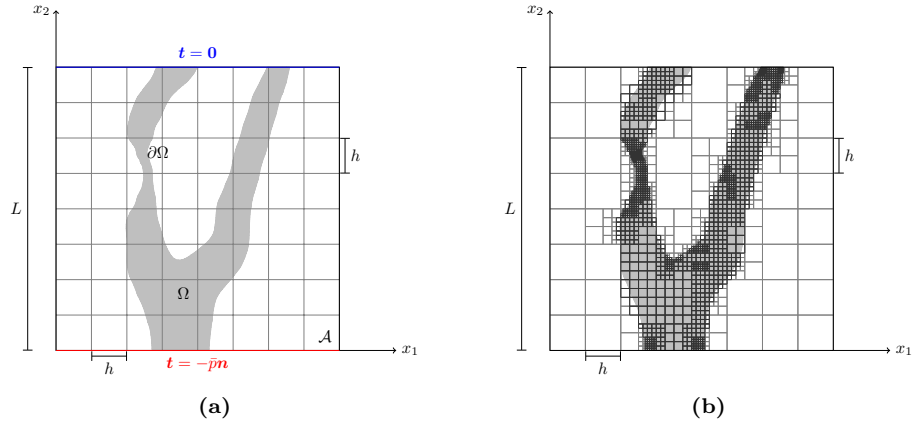


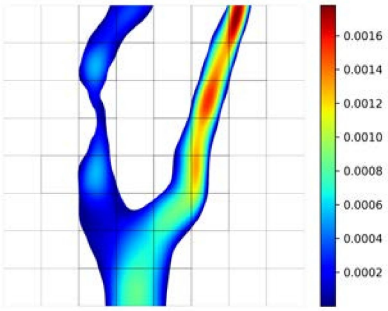
Figure 4.19: (a) Illustration of the domain and boundary conditions for the scan-based viscous flow problem, and (b) a typical locally refined mesh resulting from the adaptive procedure.

4.5.1 Two-dimensional prototypical geometry

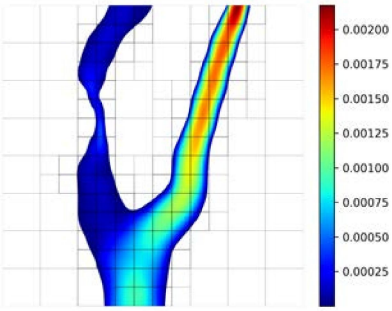
To test the developed methodology in the scan-based setting, we first consider the prototypical two-dimensional geometry shown in Figure 4.19a, which is constructed from 32×32 grayscale voxel data. The ambient domain, which matches the scan window, is taken as a unit square ($L = 1$) which is covered by an 8×8 elements ambient mesh. The viscosity is set equal to $\mu = 1$.

Various steps in the adaptive refinement procedure are illustrated in Figure 4.20. In the first step virtually all elements covering the flow domain are refined, indicating that the initial mesh of only 8×8 elements is too coarse to resolve the solution globally. After the first refinement step, the refinement strategy starts to focus on the regions where the errors are largest, *i.e.*, near boundaries and narrow sections, as also illustrated in Figure 4.19b. Under further refinement, the procedure resolves prominent solution details, most importantly the (Poiseuille-like) profile in the carotid part of the artery and the velocity profiles at the inflow and outflow boundaries.

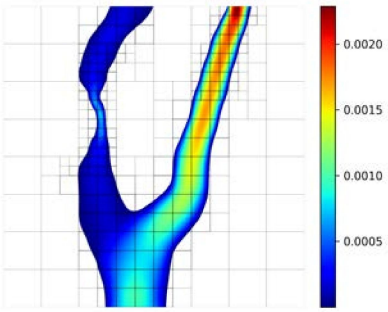
Further results of the viscous flow problem solved using uniform and adaptive refinements are shown in Figure 4.21 in the form of the flux through the left and right outflow channels. Both methods are observed to converge to the same fluxes under refinement, but an excellent approximation of the reference solution (computed on a uniform overkill refinement, consistent with the result reported in Ref. [87]) is obtained using the adaptive mesh refinement procedure using substantially fewer degrees of freedom required than for uniform refinements.



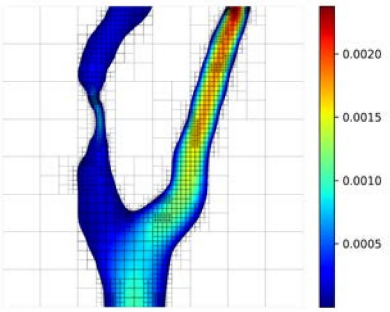
(a) Initial mesh



(b) Step 1

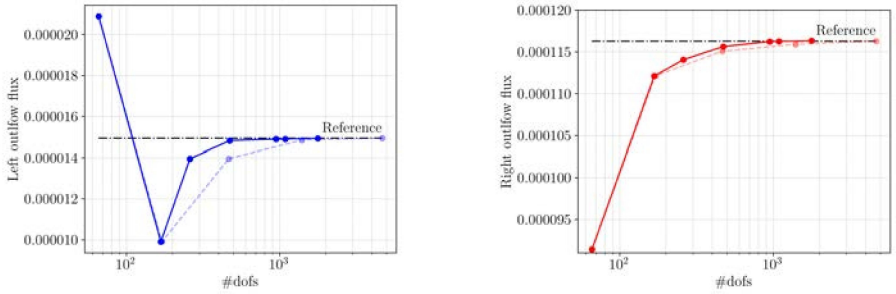


(c) Step 2



(d) Step 6

Figure 4.20: Evolution of the mesh and (magnitude of the) velocity field during the adaptive refinement process for the viscous flow in two dimensions.



(a) **(b)**

Figure 4.21: Mesh convergence of the outflow flux at the (a) left and (b) right channel of the domain in Figure 4.19a using adaptive (solid) and uniform (dashed) mesh refinements.

This is consistent with the observations on the velocity field discussed above, where in particular the ability of the adaptive refinement procedure to resolve the flow in the carotid part is essential.

4.5.2 Three-dimensional patient-specific geometry

To demonstrate the residual-based adaptivity procedure in a real scan-based setting, we consider a part of the patient-specific carotid artery used in Ref. [87]. The geometry of the carotid artery is obtainted from CT-scan data containing 80 slices of 85×70 voxels with each voxel of size $300 \times 300\,\mu\text{m}^2$ and with the distance between the slices being $400\,\mu$m. The total size of the scan domain is $25.6 \times 21.1 \times 32.0\,\text{mm}^3$.

Simulation results for this problem are shown in Figure 4.22. Note that for the considered scan data, the application of the topology-preservation algorithm in Ref. [87] is essential, as otherwise the narrow channel section would disappear. The simulation results are based on a $20 \times 20 \times 20$ ambient domain mesh of $12 \times 9 \times 16\,\text{mm}^3$ and an octree depth of three. In this setting, after two refinements, an element is of a similar size as the voxels. The need to substantially refine beyond the voxel size is, from a practical perspective, questionable, as the dominant error in the analysis will then be related to the scan resolution and the segmentation procedure. In this sense, the constraint of not being able to refine beyond the octree depth is not a crucial problem in the considered simulations.

Different steps in the adaptive refinement procedure are illustrated in Figure 4.22. In all the refinement steps, the refinement strategy starts to focus on the regions where the errors are largest, *i.e.*, near the narrow section due to the

stenosis. Under local mesh refinement, the procedure resolves prominent solution details, most importantly the velocity field in the stenotic part of the artery.

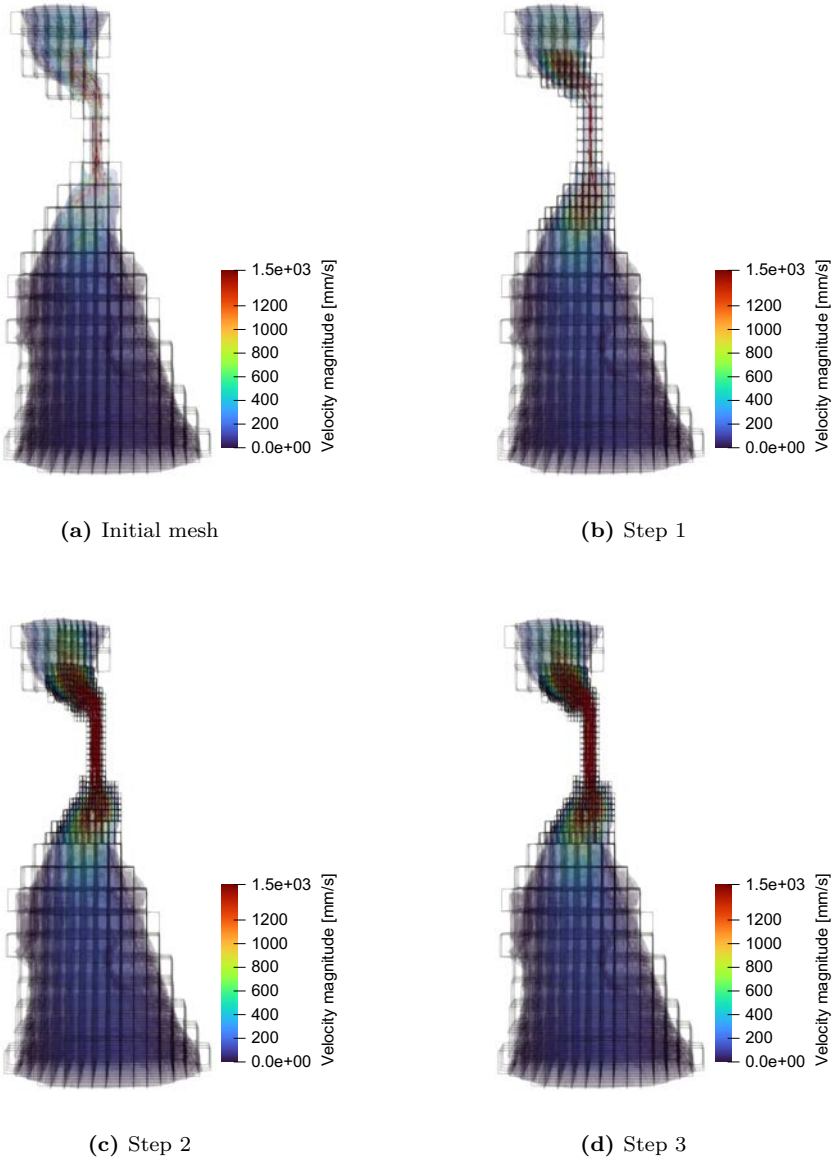
The flux at the outlet of the artery is shown in Figure 4.23, which is computed with the velocity field obtained by solving the flow problem using both uniform and adaptive refinements. Both refinement strategies are observed to approach the reference solution obtained using five uniform refinement steps (approximately 35000 DOFs). It is observed that the adaptive procedure terminates after 3 refinement steps, because of reaching the maximum refinement level in all the elements tagged for refinement. At this point, the adaptive simulation pertains to 6227 DOFs. With this number of DOFs, a better approximation to the reference solution is obtained than using a uniformly refined mesh with 22340 DOFs. This evidently is a consequence of the error estimation and adaptivity procedure being able to focus the refinements on the regions contribution most to the error in the solution.

4.6 Concluding remarks

In the immersed (isogeometric) analysis framework, the geometry representation is decoupled from the discretization. This enables the consideration of spline basis functions on complex volumetric domains, for which boundary-fitting discretizations cannot easily be obtained. Moreover, the decoupling of the geometry and the discretization allows one to have a globally accurate representation of the geometry, but only to refine the mesh in places where the errors are large. Such local mesh refinements have the potential to provide a significant efficiency gain compared to uniform meshes. The adaptive simulation strategy proposed in this work automatically refines the elements in places that significantly contribute to the error in the energy norm.

The developed error estimation and adaptivity strategy is based on residual-based error estimation, which is well-established in traditional finite elements and has been successfully applied in boundary-fitting isogeometric analysis. In the considered immersed setting, the residual-based error estimation and adaptivity framework requires the incorporation of the stabilization terms for the weakly imposed Dirichlet boundary conditions, and, in the case of the (mixed) Stokes flow problem, for the treatment of equal-order discretizations of the velocity-pressure pair. Adequate scaling of the stabilization constants with the mesh size is essential for the adaptive procedure to be effective. In particular the order dependence of the stabilization constants and the definition of the local element sizes must be treated adequately.

In contrast to residual-based error estimation for boundary-fitting finite elements and isogeometric analysis, in the stabilized immersed setting it is not evident that the residual-based error estimator bounds the error in the energy



(a) Initial mesh

(b) Step 1

(c) Step 2

(d) Step 3

Figure 4.22: Evolution of the mesh and (magnitude of the) velocity field during the adaptive refinement process for the patient-specific viscous flow problem.

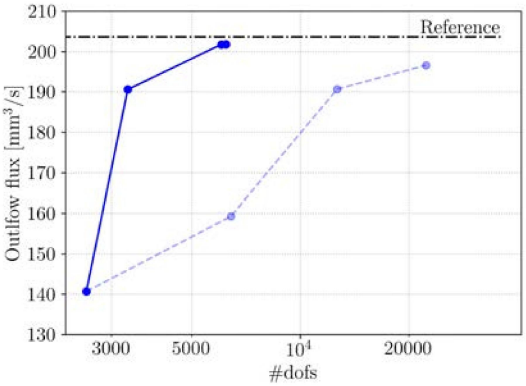


Figure 4.23: Mesh convergence of the flux at the outflow boundary using adaptive (solid) and uniform (dashed) mesh refinements for the patient-specific viscous flow problem.

norm from above. This is a consequence of the absence of an h-independent weak formulation. In this work, it is reasoned, however, that under the assumption of sufficient smoothness, the residual is expected to be useful in the setting of an adaptive refinement strategy. For all numerical simulations considered, including simulations with reduced regularity, it is observed that the error estimator does provide an upper bound to the error in the energy norm. A rigorous study regarding the relation between the residual and the actual error is warranted.

It is demonstrated that the developed adaptive simulation strategy is particularly useful in a scan-based analysis setting, where manual selection of refinement zones is impractical. When used in combination with advanced image segmentation procedures to obtain a smooth geometry representation while preserving small geometric features, the developed adaptive refinement strategy optimally leverages the advantageous approximation properties of splines for geometrically and topologically complex domains. The adaptivity strategy results in a simulation workflow that is capable of obtaining reliable, error-controlled, results with limited user interaction.

The developed adaptive solution strategy is elaborated for the Laplace problem and the Stokes problem. For other problems, such as, for example, Navier-Stokes or Cahn-Hilliard problems, the starting point of the derivation of the error-estimator remains the same. The estimators are problem-specific, however, and hence need to be elaborated for such problems. The same holds for the consideration of additional or alternative stabilization techniques, specifically when these alter the Galerkin form of the problem.

Chapter 5

Conclusions & Recommendations

This chapter discusses the most important conclusions that can be drawn from the work presented in this dissertation. Additionally, based on the understanding gained by the theoretical and numerical analyses, we discuss several concepts and methods that warrant further study.

5.1 Conclusions

The overarching goal of this dissertation is to unlock the potential of immersed isogeometric analysis for scan-based simulations by developing computational solutions for challenges associated with the scan-based analysis workflow. Specifically, we have developed the following computational techniques:

- An efficient integration technique for scan-based immersed isogeometric analysis is required to reduce the computational effort associated with the evaluation of integrals on cut elements. In this dissertation, based on Strang's first lemma, we have developed an algorithm to construct quadrature rules for cut elements in which the integration points are distributed in such a way that the integration error is minimized. The pivotal idea behind the proposed error-estimate-based optimization algorithm is to gradually increase the number of integration points by adding integration points to the octree sub-cells for which the error reduction per added integration point is highest. The presented numerical simulations demonstrate that using this algorithm, the integration error for a fixed number of integration points can be reduced significantly. This is a consequence of the ability of

the algorithm to efficiently distribute integration points over cut elements for a wide range of cut cell configurations. Based on the results obtained from the optimization algorithm, we have also proposed specific choices for distributing the integration points over the octree levels in such a way that the optimal distribution is approached without the need to execute the optimization algorithm.

- A dedicated image segmentation technique is required to obtain a smooth representation of the scan data, without altering the topology of the original voxel data. In this dissertation, based on a Fourier analysis of a B-spline-based smoothing operation, it is proposed to repair smoothing-induced topological anomalies by locally refining the smoothed grayscale function using truncated hierarchical B-splines. In combination with a moving-window strategy to detect topological changes, the local refinement technique is used to develop a topology-preserving image segmentation technique. Based on a comparison of the Euler characteristic between the window view on the original voxel data and that on the smoothed representation, the proposed technique systematically distinguishes shape changes from topological changes. Numerical simulations demonstrate the effectivity of the algorithm, in the sense that a topologically consistent smoothed image is generally obtained after a single topology-correction step.

- To accurately resolve local solution features in scan-based simulations with a minimal number of degrees of freedom, an adaptive local mesh-refinement strategy is required. In this dissertation, an adaptive simulation strategy is proposed which automatically identifies and recursively refines the elements in places that significantly contribute to an a-posteriori residual-based error estimate. Truncated hierarchical B-splines are used to locally refine the background grid, while preserving the geometry of the computational domain. The immersed setting requires a dedicated treatment of the stabilization terms to compute the residual-based error estimate, and adequate scaling of the stabilization constants with the mesh size is found to be essential for the adaptive procedure to be effective. It is observed that the error estimator provides a good indicator for refinement, both for problems with and without reduced regularity. The developed adaptive simulation strategy is particularly useful in a scan-based analysis setting, where manual selection of refinement zones is impractical.

With these three innovations, a highly versatile, efficient and robus immersed isogeometric analysis workflow for scan-based analyses is obtained. Error-controlled simulations can be performed directly based on scan data, without the need for extensive user interactions or interventions. The effectivity of the framework is not fundamentally affected by the geometric and topological complexity of the

scan data, on account of the decoupling of the geometry and computational grid in immersed methods. The robustness of the framework derives from the rigorous mathematical underpinning of the considered methods.

In combination with the earlier developments regarding stabilization techniques [246] and solver performance [247], with the work in this dissertation the most prominent fundamental obstacles for the application of the scan-based immersed isogeometric analysis workflow to complex problems have been resolved. Besides the evident need for problem-specific developments – *e.g.*, possibly requiring additional stabilization techniques or error estimators – further improvements are required to enhance the computational performance of the developed workflow. This mainly pertains to developments in terms of code optimization, which are required to apply the developed workflow to larger scans, time-dependent problems, multi-physics problems, *etc.* The current maturity level of the workflow and the ambition to extend its application to complex problems is also reflected by the emergence of dedicated simulators based on the immersed finite element concept [248].

5.2 Recommendations

The theoretical and numerical analyses presented in this dissertation warrant the further investigation of several aspects. Specifically, we recommend the following:

- In this work we have focused on the evaluation and optimization of the integration error contribution in Strang's first lemma, implicitly assuming that this error contribution is significant in comparison to the approximation error. However, the need to optimize the integration error depends on the approximation error, in the sense that there is no strong need to optimize the integration error if the approximation error is the dominating contribution. The development of a strategy to balance the various error contributions has the potential to substantially enhance the performance of immersed finite element methods.

- In this work, the integration errors are optimized by distributing integration points over the cut elements while keeping the octree depth fixed. However, the octree depth, and with that the accuracy of the geometry approximation, is an adjustable numerical parameter in the simulation framework. Since both the error reduction and the associated computational cost are quantifiable, one can include the octree depth as an additional parameter in the integration error optimization algorithm. Together with the approximation error balancing discussed above, this enhancement would further improve the control over the accuracy in the simulation workflow. Although the error optimization strategy regarding cut element integration

can be conducted as an element-wise pre-processor to the simulations, the involved computational effort can be significant. The development of a machine learning algorithm to obtain integration schemes for cut elements quickly, trained by the developed optimization algorithm, is a topic of further investigation.

- The developed technique to evaluate integration errors on cut elements is, in principle, independent of the considered integration strategy. This implies that it is not only applicable to optimizing octree-based integration schemes, but that it can also be used in combination with alternative integration schemes. In the context of moment-fitting methods, the optimized integration schemes constructed in this work can also serve as a means to compute the positions and weights of the integration points, thereby enhancing the computational performance of the moment-fitting method.

- In this work we have restricted ourselves to adaptivity based on residual-based error estimators. This approach works well in general, in the sense that it systematically reduces the error in the solution field, measured in global norms. In scan-based analysis, it is not uncommon that the analyst is interested in specific quantities of interest. In such cases, steering the mesh adaptivity based on a global norm definition is sub-optimal, and alternative techniques, most prominently goal-oriented error estimation, could be favorable. The concept of goal-oriented adaptivity in immersed (isogeometric) analysis has been explored already [58], but its application in the stabilized (flow) setting considered in this work is largely unexplored.

- In this dissertation we have considered Laplace and Stokes problems to investigate the developed algorithms for stabilized immersed isogeometric analysis. Further investigation of other problem will evidently broaden the horizon of the stabilized framework. We have already considered the extension to moderate/high Reynolds number flows and convection-diffusion problems. Preliminary observations indicate that, with a careful consideration of the non-linear solver, time-integration schemes and stabilization parameters, the general characteristics of the stabilized immersed simulation framework as observed in this work do extend to these problems.

- The current simulation framework is implemented in Nutils [109], an open source Python-based numerical library with support for immersed isogeometric analysis. Nutils contains efficient under-the-hood vectorization and built-in parallellization that allows us to develop novel algorithms in the context of problems from real scan data. With increasing problem (scan) sizes, some of the operations, specifically the mesh-trimming operation, become computationally demanding. Therefore, further code optimization is

required, for example by implementing time-consuming parts in compiled languages (*e.g.*, C, Rust).

- The methods employed in this dissertation, specifically those in Chapter 4, are based on a strong theoretical underpinning. This theoretical basis is, for example, used to derive the scaling relations for the stabilization parameters and for the residual-based error estimator. Further theoretical investigation is warranted with respect to the applicability of the skeleton-stabilization inequality (4.26) to optimal regularity splines, and with respect to the relation between the residual error and the energy norm in Eq. (4.27) in the context of immersed finite elements.

- The mathematical analysis of Chapter 4 elucidates how each stabilization parameter scales with the constants resulting from a specific inequality. In principle, these constants can be computed by solving a (generalized) eigenvalue problem corresponding to the inequality. By doing so, as a preprocessing operation, optimal stabilization parameters can be computed, similar to what is done for the Nitsche parameter [76, 78]. The calculation of optimal ghost and skeleton stability constants along the same lines is a topic worthy of further study.

Bibliography

[1] O.C. Zienkiewicz, R.L. Taylor, P. Nithiarasu, and J.Z. Zhu. *The finite element method*, volume 3. London: McGraw-hill, 1977.

[2] O.C. Zienkiewicz, R.L. Taylor, and J.Z. Zhu. *The finite element method: its basis and fundamentals*. Elsevier, 2005.

[3] O.C. Zienkiewicz and R.L. Taylor. *The finite element method for solid and structural mechanics*. Elsevier, 2005.

[4] J. Donea and A. Huerta. *Finite element methods for flow problems*. John Wiley & Sons, 2003.

[5] O.C. Zienkiewicz and K. Morgan. *Finite elements and approximation*. Courier Corporation, 2006.

[6] T.J.R. Hughes. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*, volume 4. 09 2008.

[7] P. Nithiarasu, R.W. Lewis, and K.N. Seetharamu. *Fundamentals of the finite element method for heat and mass transfer*. John Wiley & Sons, 2016.

[8] J.N. Reddy. *Introduction to the finite element method*. McGraw-Hill Education, 2019.

[9] G. Strang and G.J. Fix. *An analysis of the finite element method*, volume 212. Englewood Cliffs, N.J., Prentice-hall Inc., 1973.

[10] S.C. Brenner and L.R. Scott. *The mathematical theory of finite element methods*, volume 3. Springer, 2008.

[11] A. Ern and J.L. Guermond. *Theory and practice of finite elements*, volume 159. Springer Science & Business Media, 2013.

[12] D.L. Pham, C. Xu, and J.L. Prince. Current methods in medical image segmentation. *Annual review of biomedical engineering*, 2(1):315–337, 2000.

[13] Y.J. Zhang. Challenges and advances in image-based geometric modeling and mesh generation. In *Image-Based Geometric Modeling and Mesh Generation*, pages 1–10. Springer, 2013.

[14] D. Schillinger and M. Ruess. The Finite Cell Method: A review in the context of higher-order structural analysis of CAD and image-based geometric models. *Archives of Computational Methods in Engineering*, 22(3):391–455, 2015.

[15] T. Hoang, C.V. Verhoosel, C.Z. Qin, F. Auricchio, A. Reali, and E.H. van Brummelen. Skeleton-stabilized immersogeometric analysis for incompressible viscous flow problems. *Computer Methods in Applied Mechanics and Engineering*, 344:421–450, 2019.

[16] T.J.R. Hughes, J.A. Cottrell, and Y. Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194(39-41):4135–4195, 2005.

[17] P.M. Prenter. *Splines and variational methods*. Wiley, 1975.

[18] L. Piegl and W. Tiller. *The NURBS book*. Springer Science & Business Media, 1996.

[19] E. Catmull and J. Clark. Recursively generated B-spline surfaces on arbitrary topological meshes. *Computer-Aided Design*, 10(6):350–355, 1978.

[20] F. Cirak, M. Ortiz, and P. Schröder. Subdivision surfaces: a new paradigm for thin-shell finite-element analysis. *International Journal for Numerical Methods in Engineering*, 47(12):2039–2072, 2000.

[21] T.J.R. Hughes, A. Reali, and G. Sangalli. Efficient quadrature for NURBS-based isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 199(5-8):301–313, 2010.

[22] F. Auricchio, M. Conti, M. Ferraro, S. Morganti, A. Reali, and R.L. Taylor. Innovative and efficient stent flexibility simulations based on isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 295:347–361, 2015.

[23] J.A. Cottrell, A. Reali, Y. Bazilevs, and T.J.R. Hughes. Isogeometric analysis of structural vibrations. *Computer Methods in Applied Mechanics and Engineering*, 195(41-43):5257–5296, 2006.

[24] Y. Bazilevs, V.M. Calo, Y.J. Zhang, and T.J.R. Hughes. Isogeometric fluid–structure interaction analysis with applications to arterial blood flow. *Computational Mechanics*, 38(4):310–322, 2006.

[25] Y. Bazilevs, J.R. Gohean, T.J.R. Hughes, R.D. Moser, and Y.J. Zhang. Patient-specific isogeometric fluid–structure interaction analysis of thoracic aortic blood flow due to implantation of the Jarvik 2000 left ventricular assist device. *Computer Methods in Applied Mechanics and Engineering*, 198(45-46):3534–3550, 2009.

[26] A. Reali. An isogeometric analysis approach for the study of structural vibrations. *Journal of Earthquake Engineering*, 10(spec01):1–30, 2006.

[27] I. Akkerman, Y. Bazilevs, C.E. Kees, and M.W. Farthing. Isogeometric analysis of free-surface flow. *Journal of Computational Physics*, 230(11):4137–4152, 2011.

[28] J. Kiendl, M.-C. Hsu, M.C.H. Wu, and A. Reali. Isogeometric Kirchhoff–Love shell formulations for general hyperelastic materials. *Computer Methods in Applied Mechanics and Engineering*, 291:280–303, 2015.

[29] S. Morganti, F. Auricchio, D.J. Benson, F.I. Gambarin, S. Hartmann, T.J.R. Hughes, and A. Reali. Patient-specific isogeometric structural analysis of aortic valve closure. *Computer Methods in Applied Mechanics and Engineering*, 284:508–520, 2015.

[30] T. Hoang, C.V. Verhoosel, F. Auricchio, E.H. van Brummelen, and A. Reali. Skeleton-stabilized isogeometric analysis: High-regularity interior-penalty methods for incompressible viscous flow problems. *Computer Methods in Applied Mechanics and Engineering*, 337:324–351, 2018.

[31] T. Terahara, K. Takizawa, T.E. Tezduyar, Y. Bazilevs, and M.-C. Hsu. Heart valve isogeometric sequentially-coupled FSI analysis with the space–time topology change method. *Computational Mechanics*, pages 1–21, 2020.

[32] M. Carraturo, P. Hennig, G. Alaimo, L. Heindel, F. Auricchio, M. Kästner, and A. Reali. Additive manufacturing applications of phase-field-based topology optimization using adaptive isogeometric analysis. *GAMM-Mitteilungen*, 44(3):e202100013, 2021.

[33] Y.J. Zhang, Y. Bazilevs, S. Goswami, C.L. Bajaj, and T.J.R. Hughes. Patient-specific vascular NURBS modeling for isogeometric analysis of blood flow. *Computer Methods in Applied Mechanics and Engineering*, 196(29-30):2943–2959, 2007.

[34] A. Collin, G. Sangalli, and T. Takacs. Analysis-suitable G1 multi-patch parametrizations for C1 isogeometric spaces. *Computer Aided Geometric Design*, 47:93–113, 2016.

[35] D. Toshniwal, H. Speleers, and T.J.R. Hughes. Smooth cubic spline spaces on unstructured quadrilateral meshes with particular emphasis on extraordinary points: Geometric design and isogeometric analysis considerations. *Computer Methods in Applied Mechanics and Engineering*, 327:411–458, 2017.

[36] T.J.R. Hughes, G. Sangalli, T. Takacs, and D. Toshniwal. Smooth multi-patch discretizations in Isogeometric Analysis. In *Handbook of Numerical Analysis*, volume 22, pages 467–543. Elsevier, 2021.

[37] R. Schmidt, R. Wüchner, and K.U. Bletzinger. Isogeometric analysis of trimmed NURBS geometries. *Computer Methods in Applied Mechanics and Engineering*, 241:93–111, 2012.

[38] Y. Guo and M. Ruess. Weak dirichlet boundary conditions for trimmed thin isogeometric shells. *Computers & Mathematics with Applications*, 70(7):1425–1440, 2015.

[39] Y. Guo, M. Ruess, and D. Schillinger. A parameter-free variational coupling approach for trimmed isogeometric thin shells. *Computational Mechanics*, 59(4):693–715, 2017.

[40] Y. Yu, Y.J. Zhang, K. Takizawa, T.E. Tezduyar, and T. Sasaki. Anatomically realistic lumen motion representation in patient-specific space–time isogeometric flow analysis of coronary arteries with time-dependent medical-image data. *Computational Mechanics*, 65(2):395–404, 2020.

[41] M. Bucelli, M. Salvador, L. Dede, and A. Quarteroni. Multipatch Isogeometric Analysis for electrophysiology: Simulation in a human heart. *Computer Methods in Applied Mechanics and Engineering*, 376:113666, 2021.

[42] T. Martin, E. Cohen, and R.M. Kirby. Volumetric parameterization and trivariate B-spline fitting using harmonic functions. *Computer Aided Geometric Design*, 26(6):648–664, 2009.

[43] Y.J Zhang, W. Wang, and T.J.R. Hughes. Solid T-spline construction from boundary representations for genus-zero geometry. *Computer Methods in Applied Mechanics and Engineering*, 249:185–197, 2012.

[44] M.-C. Hsu, C. Wang, A.J. Herrema, D. Schillinger, A. Ghoshal, and Y. Bazilevs. An interactive geometry modeling and parametric design platform for isogeometric analysis. *Computers & Mathematics with Applications*, 70(7):1481–1500, 2015.

[45] A. Pawar, Y.J. Zhang, C. Anitescu, and T. Rabczuk. Joint image segmentation and registration based on a dynamic level set approach using truncated hierarchical B-splines. *Computers & Mathematics with Applications*, 78(10):3250–3267, 2019.

[46] B. Urick, T.M. Sanders, S.S. Hossain, Y.J. Zhang, and T.J.R. Hughes. Review of patient-specific vascular modeling: template-based isogeometric framework and the case for CAD. *Archives of Computational Methods in Engineering*, 26(2):381–404, 2019.

[47] C.S. Peskin. Flow patterns around heart valves: a numerical method. *Journal of Computational Physics*, 10(2):252–271, 1972.

[48] J. Parvizian, A. Düster, and E. Rank. Finite cell method. *Computational Mechanics*, 41(1):121–133, 2007.

[49] A. Düster, J. Parvizian, Z. Yang, and E. Rank. The finite cell method for three-dimensional problems of solid mechanics. *Computer Methods in Applied Mechanics and Engineering*, 197(45-48):3768–3782, 2008.

[50] A. Hansbo and P. Hansbo. An unfitted finite element method, based on Nitsche's method, for elliptic interface problems. *Computer Methods in Applied Mechanics and Engineering*, 191(47-48):5537–5552, 2002.

[51] E. Burman and P. Hansbo. Fictitious domain finite element methods using cut elements: II. A stabilized Nitsche method. *Applied Numerical Mathematics*, 62(4):328–341, 2012.

[52] E. Rank, M. Ruess, S. Kollmannsberger, D. Schillinger, and A. Düster. Geometric modeling, isogeometric analysis and the finite cell method. *Computer Methods in Applied Mechanics and Engineering*, 249:104–115, 2012.

[53] M. Ruess, D. Schillinger, Y. Bazilevs, V. Varduhn, and E. Rank. Weakly enforced essential boundary conditions for NURBS-embedded and trimmed NURBS geometries on the basis of the finite cell method. *International Journal for Numerical Methods in Engineering*, 95(10):811–846, 2013.

[54] M. Ruess, D. Schillinger, A.I. Oezcan, and E. Rank. Weak coupling for isogeometric analysis of non-matching and trimmed multi-patch geometries. *Computer Methods in Applied Mechanics and Engineering*, 269:46–71, 2014.

[55] D. Schillinger and E. Rank. An unfitted *hp*-adaptive finite element method based on hierarchical B-splines for interface problems of complex geometry. *Computer Methods in Applied Mechanics and Engineering*, 200(47-48):3358–3380, 2011.

[56] D. Kamensky, M.-C. Hsu, D. Schillinger, J.A. Evans, A. Aggarwal, Y. Bazilevs, M.S. Sacks, and T.J.R. Hughes. An immersogeometric variational framework for fluid–structure interaction: Application to bioprosthetic heart valves. *Computer Methods in Applied Mechanics and Engineering*, 284:1005–1053, 2015.

[57] M.-C. Hsu, D. Kamensky, F. Xu, J. Kiendl, C. Wang, M.C.H. Wu, J. Mineroff, A. Reali, Y. Bazilevs, and M.S. Sacks. Dynamic and fluid–structure interaction simulations of bioprosthetic heart valves using parametric design with T-splines and Fung-type material models. *Computational Mechanics*, 55(6):1211–1225, 2015.

[58] C.V. Verhoosel, G.J. van Zwieten, B. van Rietbergen, and R. de Borst. Image-based goal-oriented adaptive isogeometric analysis with application to the micro-mechanical modeling of trabecular bone. *Computer Methods in Applied Mechanics and Engineering*, 284:138–164, 2015.

[59] M. Ruess, D. Tal, N. Trabelsi, Z. Yosibash, and E. Rank. The finite cell method for bone simulations: verification and validation. *Biomechanics and Modeling in Mechanobiology*, 11(3-4):425–437, 2012.

[60] F. de Prenter, C.V. Verhoosel, E.H. van Brummelen, J.A. Evans, C. Messe, J. Benzaken, and K. Maute. Multigrid solvers for immersed finite element methods and immersed isogeometric analysis. *Computational Mechanics*, 65(3):807–838, 2020.

[61] A. Düster, H.G. Sehlhorst, and E. Rank. Numerical homogenization of heterogeneous and cellular materials utilizing the finite cell method. *Computational Mechanics*, 50(4):413–431, 2012.

[62] J.N. Jomo, F. de Prenter, M. Elhaddad, D. D'Angella, C.V. Verhoosel, S. Kollmannsberger, J.S. Kirschke, V. Nübel, E.H. van Brummelen, and E. Rank. Robust and parallel scalable iterative solutions for large-scale finite cell analyses. *Finite Elements in Analysis and Design*, 163:14–30, 2019.

[63] M. Carraturo, J.N. Jomo, S. Kollmannsberger, A. Reali, F. Auricchio, and E. Rank. Modeling and experimental validation of an immersed thermo-mechanical part-scale analysis for laser powder bed fusion processes. *Additive Manufacturing*, 36:101498, 2020.

[64] T. Belytschko, R. Gracie, and G. Ventura. A review of extended/generalized finite element methods for material modeling. *Modelling and Simulation in Materials Science and Engineering*, 17(4):043001, 2009.

[65] A. Düster, E. Rank, and B. Szabó. The *p*-version of the Finite Element and Finite Cell Methods. *Encyclopedia of Computational Mechanics Second Edition*, pages 1–35, 2017.

[66] V. Varduhn, M.-C. Hsu, M. Ruess, and D. Schillinger. The tetrahedral finite cell method: higher-order immersogeometric analysis on adaptive non-boundary-fitted meshes. *International Journal for Numerical Methods in Engineering*, 107(12):1054–1079, 2016.

[67] A. Stavrev, L.H. Nguyen, R. Shen, V. Varduhn, M. Behr, S. Elgeti, and D. Schillinger. Geometrically accurate, efficient, and flexible quadrature techniques for the tetrahedral finite cell method. *Computer Methods in Applied Mechanics and Engineering*, 310:646–673, 2016.

[68] A. Abedian, J. Parvizian, A. Düster, and E. Rank. The finite cell method for the J2 flow theory of plasticity. *Finite Elements in Analysis and Design*, 69:37–47, 2013.

[69] R. Sevilla, S. Fernández Méndez, and A. Huerta. NURBS-Enhanced Finite Element Method (NEFEM). *International Journal for Numerical Methods in Engineering*, 76(1):56–83, 2008.

[70] T.P. Fries and S. Omerović. Higher-order accurate integration of implicit geometries. *International Journal for Numerical Methods in Engineering*, 106(5):323–371, 2016.

[71] L. Kudela, N. Zander, S. Kollmannsberger, and E. Rank. Smart octrees: Accurately integrating discontinuous functions in 3D. *Computer Methods in Applied Mechanics and Engineering*, 306:406–426, 2016.

[72] S. Hubrich and A. Düster. Numerical integration for nonlinear problems of the finite cell method using an adaptive scheme based on moment fitting. *Computers & Mathematics with Applications*, 77(7):1983–1997, 2019.

[73] A. Abedian and A. Düster. Equivalent Legendre polynomials: Numerical integration of discontinuous functions in the finite element methods. *Computer Methods in Applied Mechanics and Engineering*, 343:690–720, 2019.

[74] T. Jonsson, M.G. Larson, and K. Larsson. Cut finite element methods for elliptic problems on multipatch parametric surfaces. *Computer Methods in Applied Mechanics and Engineering*, 324:366–394, 2017.

[75] P. Antolin and T. Hirschler. Quadrature-free immersed isogeometric analysis. *arXiv preprint arXiv:2107.09024*, 2021.

[76] F. de Prenter, C. Lehrenfeld, and A. Massing. A note on the stability parameter in Nitsche's method for unfitted boundary value problems. *Computers & Mathematics with Applications*, 75(12):4322–4336, 2018.

[77] J. Nitsche. Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind. In *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, volume 36, pages 9–15. Springer, 1971.

[78] A. Embar, J. Dolbow, and I. Harari. Imposing dirichlet boundary conditions with Nitsche's method and spline-based finite elements. *International Journal for Numerical Methods in Engineering*, 83(7):877–898, 2010.

[79] E. Burman. Ghost penalty. *Comptes Rendus Mathematique*, 348(21-22):1217–1220, 2010.

[80] F. de Prenter, C.V. Verhoosel, G. van Zwieten, and E.H. van Brummelen. Condition number analysis and preconditioning of the finite cell method. *Computer Methods in Applied Mechanics and Engineering*, 316:297–327, 2017.

[81] B. Marussig and T.J.R. Hughes. A review of trimming in isogeometric analysis: challenges, data exchange and simulation aspects. *Archives of Computational Methods in Engineering*, 25(4):1059–1127, 2018.

[82] F. de Prenter, C.V. Verhoosel, and E.H. van Brummelen. Preconditioning immersed isogeometric finite element methods with application to flow problems. *Computer Methods in Applied Mechanics and Engineering*, 348:604–631, 2019.

[83] E. Burman, S. Claus, P. Hansbo, M.G. Larson, and A. Massing. CutFEM: discretizing geometry and partial differential equations. *International Journal for Numerical Methods in Engineering*, 104(7):472–501, 2015.

[84] W.G. Dettmer, C. Kadapa, and D. Perić. A stabilised immersed boundary method on hierarchical B-spline grids. *Computer Methods in Applied Mechanics and Engineering*, 311:415–437, 2016.

[85] S. Badia, F. Verdugo, and A.F. Martín. The aggregated unfitted finite element method for elliptic problems. *Computer Methods in Applied Mechanics and Engineering*, 336:533–553, 2018.

[86] S.C. Divi, C.V. Verhoosel, F. Auricchio, A. Reali, and E.H. van Brummelen. Error-estimate-based adaptive integration for immersed isogeometric analysis. *Computers & Mathematics with Applications*, 80(11):2481–2516, 2020.

[87] S.C. Divi, C.V. Verhoosel, F. Auricchio, A. Reali, and E.H. van Brummelen. Topology-preserving Scan-based Immersed Isogeometric Analysis. *Computer Methods in Applied Mechanics and Engineering*, 392:114648, 2022.

[88] S.C. Divi, P.H. van Zuijlen, T. Hoang, F. de Prenter, F. Auricchio, A. Reali, E.H. van Brummelen, and C.V. Verhoosel. Residual-based error estimation and adaptivity for stabilized immersed isogeometric analysis using truncated hierarchical B-splines. *arXiv preprint arXiv:2202.08763*, 2022.

[89] D. Schillinger, L. Dede, M.A. Scott, J.A. Evans, M.J. Borden, E. Rank, and 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. *Computer Methods in Applied Mechanics and Engineering*, 249:116–150, 2012.

[90] A.M. Bauer, M. Breitenberger, B. Philipp, R. Wüchner, and K.U. Bletzinger. Embedded structural entities in NURBS-based isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 325:198–218, 2017.

[91] G. Beer, B. Marussig, and J. Zechner. A simple approach to the numerical simulation with trimmed CAD surfaces. *Computer Methods in Applied Mechanics and Engineering*, 285:776–790, 2015.

[92] F. Xu, S. Morganti, R. Zakerzadeh, D. Kamensky, F. Auricchio, A. Reali, T.J.R. Hughes, M.S. Sacks, and M.-C. Hsu. A framework for designing patient-specific bioprosthetic heart valves using immersogeometric fluid–structure interaction analysis. *International Journal for Numerical Methods in Biomedical Engineering*, 34(4):e2938, 2018.

[93] K.W. Cheng and T.P. Fries. Higher-order XFEM for curved strong and weak discontinuities. *International Journal for Numerical Methods in Engineering*, 82(5):564–590, 2010.

[94] R. Sevilla, S. Fernández Méndez, and A. Huerta. Comparison of high-order curved finite elements. *International Journal for Numerical Methods in Engineering*, 87(8):719–734, 2011.

[95] E. Nadal, J.J. Ródenas, J. Albelda, M. Tur, J.E. Tarancón, and F.J. Fuenmayor. Efficient finite element methodology based on cartesian grids: application to structural shape optimization. In *Abstract and Applied Analysis*, volume 2013. Hindawi, 2013.

[96] S. Hubrich, P. Di Stolfo, L. Kudela, S. Kollmannsberger, E. Rank, A. Schröder, and A. Düster. Numerical integration of discontinuous functions: moment fitting and smart octree. *Computational Mechanics*, 60(5):863–881, 2017.

[97] S.E. Mousavi and N. Sukumar. Numerical integration of polynomials and discontinuous functions on irregular convex polygons and polyhedrons. *Computational Mechanics*, 47(5):535–554, 2011.

[98] M. Joulaian, S. Hubrich, and A. Düster. Numerical integration of discontinuities on arbitrary domains based on moment fitting. *Computational Mechanics*, 57(6):979–999, 2016.

[99] A. Düster and O. Allix. Selective enrichment of moment fitting and application to cut finite elements and cells. *Computational Mechanics*, pages 1–22, 2019.

[100] G. Ventura. On the elimination of quadrature subcells for discontinuous functions in the extended finite-element method. *International Journal for Numerical Methods in Engineering*, 66(5):761–795, 2006.

[101] G. Ventura, R. Gracie, and T. Belytschko. Fast integration and weight function blending in the extended finite element method. *International Journal for Numerical Methods in Engineering*, 77(1):1–29, 2009.

[102] G. Strang. Variational crimes in the finite element method. In *The mathematical foundations of the finite element method with applications to partial differential equations*, pages 689–710. Elsevier, 1972.

[103] E. Burman, P. Hansbo, M.G. Larson, and S. Zahedi. Cut finite element methods for coupled bulk–surface problems. *Numerische Mathematik*, 133(2):203–231, 2016.

[104] E. Burman, P. Hansbo, M.G. Larson, and A. Massing. A cut discontinuous Galerkin method for the Laplace–Beltrami operator. *IMA Journal of Numerical Analysis*, 37(1):138–169, 2017.

[105] O.C. Zienkiewicz, R.L. Taylor, and J.M. Too. Reduced integration technique in general analysis of plates and shells. *International Journal for Numerical Methods in Engineering*, 3(2):275–290, 1971.

[106] T.J.R. Hughes, M. Cohen, and M. Haroun. Reduced and selective integration techniques in the finite element analysis of plates. *Nuclear Engineering and Design*, 46(1):203–222, 1978.

[107] D.S. Malkus and T.J.R. Hughes. Mixed finite element methods—reduced and selective integration techniques: a unification of concepts. *Computer Methods in Applied Mechanics and Engineering*, 15(1):63–81, 1978.

[108] A. Taghipour, J. Parvizian, S. Heinze, and A. Düster. The finite cell method for nearly incompressible finite strain plasticity problems with complex geometries. *Computers & Mathematics with Applications*, 75(9):3298–3316, 2018.

[109] G. van Zwieten, J. van Zwieten, C.V. Verhoosel, E. Fonn, T. van Opstal, and W. Hoitinga. Nutils, 2020.

[110] J.A. Cottrell, T.J.R. Hughes, and Y. Bazilevs. *Isogeometric analysis: Toward integration of CAD and FEA*. John Wiley & Sons, 2009.

[111] S. Marschke, L. Wunderlich, W. Ring, K. Achterhold, and F. Pfeiffer. An approach to construct a three-dimensional isogeometric model from μ-CT scan data with an application to the bridge of a violin. *Computer Aided Geometric Design*, 78:101815, 2020.

[112] B. van Rietbergen, H. Weinans, R. Huiskes, and B.J.W. Polman. Computational strategies for iterative solutions of large FEM applications employing voxel data. *International Journal for Numerical Methods in Engineering*, 39(16):2743–2767, 1996.

[113] D.A. Rajon and W.E. Bolch. Marching cube algorithm: review and trilinear interpolation adaptation for image-based dosimetric models. *Computerized Medical Imaging and Graphics*, 27(5):411–435, 2003.

[114] Z.L. Wang, J.C.M. Teo, C.K. Chui, S.H. Ong, C.H. Yan, S.C. Wang, H.K. Wong, and S.H. Teoh. Computational biomechanical modelling of the lumbar spine using marching-cubes surface smoothened finite element voxel meshing. *Computer Methods and Programs in Biomedicine*, 80(1):25–35, 2005.

[115] M.S. Pigazzini, Y. Bazilevs, A. Ellison, and H. Kim. Isogeometric analysis for simulation of progressive damage in composite laminates. *Journal of Composite Materials*, 52(25):3471–3489, 2018.

[116] M.-C. Hsu, C. Wang, F. Xu, A.J. Herrema, and A. Krishnamurthy. Direct immersogeometric fluid flow analysis using B-rep CAD models. *Computer Aided Geometric Design*, 43:143–158, 2016.

[117] D. Schillinger, M. Ruess, N. Zander, Y. Bazilevs, A. Düster, and E. Rank. Small and large deformation analysis with the *p*- and B-spline versions of the Finite Cell Method. *Computational Mechanics*, 50(4):445–478, 2012.

[118] M. Elhaddad, N. Zander, T. Bog, L. Kudela, S. Kollmannsberger, J. Kirschke, T. Baum, M. Ruess, and E. Rank. Multi-level *hp*-finite cell method for embedded interface problems with application in biomechanics. *International Journal for Numerical Methods in Biomedical Engineering*, 34(4):e2951, 2018.

[119] M. Würkner, S. Duczek, H. Berger, H. Köppe, and U. Gabbert. A software platform for the analysis of porous die-cast parts using the finite cell method. In *Analysis and Modelling of Advanced Structures and Smart Systems*, pages 327–341. Springer, 2018.

[120] S. Duczek, H. Berger, and U. Gabbert. The finite pore method: a new approach to evaluate gas pores in cast parts by combining computed tomography and the finite cell method. *International Journal of Cast Metals Research*, 28(4):221–228, 2015.

[121] H.J. Kim, Y.D. Seo, and S.K. Youn. Isogeometric analysis for trimmed CAD surfaces. *Computer Methods in Applied Mechanics and Engineering*, 198(37-40):2982–2995, 2009.

[122] H.J. Kim, Y.D. Seo, and S.K. Youn. Isogeometric analysis with trimming technique for problems of arbitrary complex topology. *Computer Methods in Applied Mechanics and Engineering*, 199(45-48):2796–2812, 2010.

[123] M.-C. Hsu, D. Kamensky, Y. Bazilevs, M.S. Sacks, and T.J.R. Hughes. Fluid–structure interaction analysis of bioprosthetic heart valves: significance of arterial wall deformation. *Computational Mechanics*, 54(4):1055–1071, 2014.

[124] D. Kamensky, M.-C. Hsu, Y. Yu, J.A. Evans, M.S. Sacks, and T.J.R. Hughes. Immersogeometric cardiovascular fluid–structure interaction analysis with divergence-conforming B-splines. *Computer Methods in Applied Mechanics and Engineering*, 314:408–472, 2017.

[125] L. Kudela, N. Zander, T. Bog, S. Kollmannsberger, and E. Rank. Efficient and accurate numerical quadrature for immersed boundary methods. *Advanced Modeling and Simulation in Engineering Sciences*, 2(1):1–22, 2015.

[126] T. Boiveau and E. Burman. A penalty-free Nitsche method for the weak imposition of boundary conditions in compressible and incompressible elasticity. *IMA Journal of Numerical Analysis*, 36(2):770–795, 2016.

[127] E. Burman and P. Hansbo. Fictitious domain methods using cut elements: III. A stabilized Nitsche method for Stokes' problem. *ESAIM: Mathematical Modelling and Numerical Analysis*, 48(3):859–874, 2014.

[128] E. Burman and M.A. Fernández. An unfitted Nitsche method for incompressible fluid–structure interaction using overlapping meshes. *Computer Methods in Applied Mechanics and Engineering*, 279:497–514, 2014.

[129] D. Schillinger, A. Düster, and E. Rank. The *hp-d*-adaptive finite cell method for geometrically nonlinear problems of solid mechanics. *International Journal for Numerical Methods in Engineering*, 89(9):1171–1202, 2012.

[130] S. Heinze, M. Joulaian, and A. Düster. Numerical homogenization of hybrid metal foams using the finite cell method. *Computers & Mathematics with Applications*, 70(7):1501–1517, 2015.

[131] W. Garhuom, S. Hubrich, L. Radtke, and A. Düster. A remeshing strategy for large deformations in the finite cell method. *Computers & Mathematics with Applications*, 80(11):2379–2398, 2020.

[132] J.F. Mangin, V. Frouin, I. Bloch, J. Régis, and J. López-Krahe. From 3D magnetic resonance images to structural representations of the cortex topography using topology preserving deformations. *Journal of Mathematical Imaging and Vision*, 5(4):297–318, 1995.

[133] A.M. Dale, B. Fischl, and M.I. Sereno. Cortical surface-based analysis: I. Segmentation and surface reconstruction. *Neuroimage*, 9(2):179–194, 1999.

[134] C. Xu, D.L. Pham, M.E. Rettmann, D.N. Yu, and J.L. Prince. Reconstruction of the human cerebral cortex from magnetic resonance images. *IEEE Transactions on Medical Imaging*, 18(6):467–480, 1999.

[135] B. Fischl, A. Liu, and A.M. Dale. Automated manifold surgery: constructing geometrically accurate and topologically correct models of the human cerebral cortex. *IEEE Transactions on Medical Imaging*, 20(1):70–80, 2001.

[136] F. Ségonne, J. Pacheco, and B. Fischl. Geometrically accurate topology-correction of cortical surfaces using nonseparating loops. *IEEE Transactions on Medical Imaging*, 26(4):518–529, 2007.

[137] H. Lester and S.R. Arridge. A survey of hierarchical non-linear medical image registration. *Pattern Recognition*, 32(1):129–149, 1999.

[138] Z. Xie and G.E. Farin. Image registration using hierarchical B-splines. *IEEE Transactions on Visualization and Computer Graphics*, 10(1):85–94, 2004.

[139] G. Falqui and C. Reina. BRS cohomology and topological anomalies. *Communications in Mathematical Physics*, 102(3):503–515, 1985.

[140] T. Qaiser, K. Sirinukunwattana, K. Nakane, Y.W. Tsang, D. Epstein, and N. Rajpoot. Persistent homology for fast tumor segmentation in whole slide histology images. *Procedia Computer Science*, 90:119–124, 2016.

[141] W. Bae, J. Yoo, and J. Chul Ye. Beyond deep residual learning for image restoration: Persistent homology-guided manifold simplification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 145–153, 2017.

[142] R. Assaf, A. Goupil, V. Vrabie, T. Boudier, and M. Kacim. Persistent homology for object segmentation in multidimensional grayscale images. *Pattern Recognition Letters*, 112:277–284, 2018.

[143] X. Hu, L. Fuxin, D. Samaras, and C. Chen. Topology-preserving deep image segmentation. *arXiv preprint arXiv:1906.05404*, 2019.

[144] J. Clough, N. Byrne, I. Oksuz, V.A. Zimmer, J.A. Schnabel, and A.P. King. A topological loss function for deep-learning based image segmentation using persistent homology. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[145] P. de Dumast, H. Kebiri, C. Atat, V. Dunet, M. Koob, and M.B. Cuadra. Segmentation of the cortical plate in fetal brain MRI with a topological loss. *arXiv preprint arXiv:2010.12391*, 2020.

[146] N. Byrne, J.R. Clough, G. Montana, and A.P. King. A persistent homology-based topological loss function for multi-class CNN segmentation of cardiac MRI. *arXiv preprint arXiv:2008.09585*, 2020.

[147] A.A. Novotny, R.A. Feijóo, E. Taroco, and C. Padra. Topological sensitivity analysis. *Computer Methods in Applied Mechanics and Engineering*, 192(7-8):803–829, 2003.

[148] M. Burger, B. Hackl, and W. Ring. Incorporating topological derivatives into level set methods. *Journal of Computational Physics*, 194(1):344–362, 2004.

[149] L. He, C.Y. Kao, and S. Osher. Incorporating topological derivatives into shape derivatives based level set methods. *Journal of Computational Physics*, 225(1):891–909, 2007.

[150] I. Larrabide, R.A. Feijóo, A.A. Novotny, and E.A. Taroco. Topological derivative: a tool for image processing. *Computers & Structures*, 86(13-14):1386–1403, 2008.

[151] M. Hintermüller and A. Laurain. Multiphase image segmentation and modulation recovery based on shape and topological sensitivity. *Journal of Mathematical Imaging and Vision*, 35(1):1–22, 2009.

[152] M. Droske, B. Meyer, M. Rumpf, and C. Schaller. An adaptive level set method for medical image segmentation. In *Biennial International Conference on Information Processing in Medical Imaging*, pages 416–422. Springer, 2001.

[153] V. Sochnikov and S. Efrima. Level set calculations of the evolution of boundaries on a dynamically adaptive grid. *International Journal for Numerical Methods in Engineering*, 56(13):1913–1929, 2003.

[154] M. Xu, P.M. Thompson, and A.W. Toga. An adaptive level set segmentation on a triangulated mesh. *IEEE Transactions on Medical Imaging*, 23(2):191–201, 2004.

[155] G. Varadhan, S. Krishnan, T.V.N. Sriram, and D. Manocha. Topology preserving surface extraction using adaptive subdivision. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, pages 235–244, 2004.

[156] Y. Bai, X. Han, and J.L. Prince. Octree-based topology-preserving isosurface simplification. In *2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06)*, pages 81–81. IEEE, 2006.

[157] Y. Bai, X. Han, and J.L. Prince. Octree grid topology preserving geometric deformable model for three-dimensional medical image segmentation. In *Biennial International Conference on Information Processing in Medical Imaging*, pages 556–568. Springer, 2007.

[158] C. Giannelli, B. Jüttler, and H. Speleers. THB-splines: The truncated basis for hierarchical splines. *Computer Aided Geometric Design*, 29(7):485–498, 2012.

[159] D. D'Angella, S. Kollmannsberger, E. Rank, and A. Reali. Multi-level Bézier extraction for hierarchical local refinement of Isogeometric Analysis.

Computer Methods in Applied Mechanics and Engineering, 328:147–174, 2018.

[160] D. D'Angella and A. Reali. Efficient extraction of hierarchical B-splines for local refinement and coarsening of Isogeometric Analysis. *Computer Methods in Applied Mechanics and Engineering*, 367:113131, 2020.

[161] X. Han, C. Xu, and J.L. Prince. A topology preserving level set method for geometric deformable models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(6):755–768, 2003.

[162] H. Lee, H. Kang, M.K. Chung, B.N. Kim, and D.S. Lee. Persistent brain network homology from the perspective of dendrogram. *IEEE Transactions on Medical Imaging*, 31(12):2267–2277, 2012.

[163] D. Lee, J. Yoo, and J.C. Ye. Deep residual learning for compressed sensing MRI. In *2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017)*, pages 15–18. IEEE, 2017.

[164] L. Piegl and W. Tiller. Curve and surface constructions using rational B-splines. *Computer-Aided Design*, 19(9):485–498, 1987.

[165] L. Shapiro. *Computer vision and image processing*. Academic Press, 1992.

[166] L. Ortiz-Gracia and C.W. Oosterlee. A highly efficient Shannon wavelet inverse Fourier technique for pricing European options. *SIAM Journal on Scientific Computing*, 38(1):B118–B143, 2016.

[167] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[168] M.H.F. Wilkinson, T. Wijbenga, G. De Vries, and M.A. Westenberg. Blood vessel segmentation using moving-window robust automatic threshold selection. In *Proceedings 2003 International Conference on Image Processing (Cat. No. 03CH37429)*, volume 2, pages II–1093. IEEE, 2003.

[169] T. Häikiö, R. Bertram, J. Hyönä, and P. Niemi. Development of the letter identity span in reading: Evidence from the eye movement moving window paradigm. *Journal of Experimental Child Psychology*, 102(2):167–181, 2009.

[170] J. Ohser, W. Nagel, and K. Schladitz. The Euler number of discretized sets - on the choice of adjacency in homogeneous lattices. In *Morphology of Condensed Matter*, pages 275–298. Springer, 2002.

[171] S. Lichtenbaum. The Weil-étale topology on schemes over finite fields. *Compositio Mathematica*, 141(3):689–702, 2005.

[172] T. Leinster. The Euler characteristic of a category. *arXiv preprint math/0610260*, 2006.

[173] C. Berger and T. Leinster. The Euler characteristic of a category as the sum of a divergent series. *Homology, Homotopy and Applications*, 10(1):41–51, 2008.

[174] T.M. Fiore, W. Lück, and R. Sauer. Finiteness obstructions and Euler characteristics of categories. *Advances in Mathematics*, 226(3):2371–2469, 2011.

[175] S. van der Walt, J.L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J.D. Warner, N. Yager, E. Gouillart, and T. Yu. Scikit-image: image processing in Python. *PeerJ*, 2:e453, 6 2014.

[176] van BrummelenE.H., T.H.B. Demont, and G.J. van Zwieten. An adaptive isogeometric analysis approach to elasto-capillary fluid-solid interaction. *International Journal for Numerical Methods in Engineering*, 2020.

[177] E.M. Garau and R. Vázquez. Algorithms for the implementation of adaptive isogeometric methods using hierarchical B-splines. *Applied Numerical Mathematics*, 123:58–87, 2018.

[178] M. Conti, C. Long, M. Marconi, R. Berchiolli, Y. Bazilevs, and A. Reali. Carotid artery hemodynamics before and after stenting: A patient specific CFD study. *Computers & Fluids*, 141:62–74, 2016.

[179] P.A. Yushkevich, J. Piven, H.C. Hazlett, R.G. Smith, S. Ho, J.C. Gee, and G. Gerig. User-guided 3D active contour segmentation of anatomical structures: Significantly improved efficiency and reliability. *Neuroimage*, 31(3):1116–1128, 2006.

[180] Y. Bazilevs and T.J.R. Hughes. Weak imposition of Dirichlet boundary conditions in fluid mechanics. *Computers & Fluids*, 36(1):12–26, 2007.

[181] A. Massing, M.G. Larson, A. Logg, and M.E. Rognes. A stabilized Nitsche fictitious domain method for the Stokes problem. *Journal of Scientific Computing*, 61(3):604–628, 2014.

[182] K. Höllig, U. Reif, and J. Wipper. Weighted extended B-spline approximation of Dirichlet problems. *SIAM Journal on Numerical Analysis*, 39(2):442–462, 2001.

[183] K. Höllig, C. Apprich, and A. Streit. Introduction to the Web-method and its applications. *Advances in Computational Mathematics*, 23(1-2):215–237, 2005.

[184] T. Rüberg and F. Cirak. Subdivision-stabilised immersed B-spline finite elements for moving boundary flows. *Computer Methods in Applied Mechanics and Engineering*, 209:266–283, 2012.

[185] T. Rüberg and F. Cirak. A fixed-grid B-spline finite element technique for fluid–structure interaction. *International Journal for Numerical Methods in Fluids*, 74(9):623–660, 2014.

[186] B. Marussig, J. Zechner, G. Beer, and T.P. Fries. Stable isogeometric analysis of trimmed geometries. *Computer Methods in Applied Mechanics and Engineering*, 316:497–521, 2017.

[187] S. Badia, A.F. Martin, and F. Verdugo. Mixed aggregated finite element methods for the unfitted discretization of the Stokes problem. *Scientific Computing*, 40(6):B1541–B1576, 2018.

[188] I. Babuvska. The finite element method with Lagrangian multipliers. *Numerische Mathematik*, 20(3):179–192, 1973.

[189] F. Brezzi. On the existence, uniqueness and approximation of saddle-point problems arising from Lagrangian multipliers. *Publications Mathématiques et Informatique de Rennes*, pages 1–26, 1974.

[190] C. Taylor and P. Hood. A numerical solution of the Navier-Stokes equations using the finite element technique. *Computers & Fluids*, 1(1):73–100, 1973.

[191] A. Buffa, C. De Falco, and G. Sangalli. Isogeometric analysis: stable elements for the 2D Stokes equation. *International Journal for Numerical Methods in Fluids*, 65(11-12):1407–1422, 2011.

[192] A. Bressan and G. Sangalli. Isogeometric discretizations of the Stokes problem: stability analysis by the macroelement technique. *IMA Journal of Numerical Analysis*, 33(2):629–651, 2013.

[193] P.A. Raviart and J.M. Thomas. A mixed finite element method for 2^{nd} order elliptic problems. In *Mathematical aspects of Finite Element Methods*, pages 292–315. Springer, 1977.

[194] V. Girault and P.A. Raviart. *Finite element approximation of the Navier-Stokes equations*, volume 749. Springer Berlin, 1979.

[195] J.A. Evans and T.J.R. Hughes. Isogeometric divergence-conforming B-splines for the Darcy–Stokes–Brinkman equations. *Mathematical Models and Methods in Applied Sciences*, 23(04):671–741, 2013.

[196] T.J.R. Hughes, L.P. Franca, and G.M. Hulbert. A new finite element formulation for computational fluid dynamics: VIII. the Galerkin/least-squares method for advective-diffusive equations. *Computer Methods in Applied Mechanics and Engineering*, 73(2):173–189, 1989.

[197] J. Douglas and J.P. Wang. An absolutely stabilized finite element method for the Stokes problem. *Mathematics of Computation*, 52(186):495–508, 1989.

[198] T.E. Tezduyar. Stabilized finite element formulations for incompressible flow computations. *Advances in Applied Mechanics*, 28:1–44, 1991.

[199] T.J.R. Hughes, G.R. Feijóo, L. Mazzei, and J.B. Quincy. The variational multiscale method — a paradigm for computational mechanics. *Computer Methods in Applied Mechanics and Engineering*, 166(1-2):3–24, 1998.

[200] T.J.R. Hughes, L. M., A.A. Oberai, and A.A. Wray. The multiscale formulation of large eddy simulation: Decay of homogeneous isotropic turbulence. *Physics of Fluids*, 13(2):505–512, 2001.

[201] Y. Bazilevs, V.M. Calo, J.A. Cottrell, T.J.R. Hughes, A. Reali, and G. Scovazzi. Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows. *Computer Methods in Applied Mechanics and Engineering*, 197(1-4):173–201, 2007.

[202] R. Becker and M. Braack. A finite element pressure gradient stabilization for the Stokes equations based on local projections. *Calcolo*, 38(4):173–199, 2001.

[203] M. Braack and E. Burman. Local projection stabilization for the Oseen problem and its interpretation as a variational multiscale method. *SIAM Journal on Numerical Analysis*, 43(6):2544–2566, 2006.

[204] T. Hoang, C.V. Verhoosel, F. Auricchio, E.H. van Brummelen, and A. Reali. Mixed isogeometric finite cell methods for the Stokes problem. *Computer Methods in Applied Mechanics and Engineering*, 316:400–423, 2017.

[205] N. Zander, T. Bog, S. Kollmannsberger, D. Schillinger, and E. Rank. Multi-level *hp*-adaptivity: high-order mesh adaptivity without the difficulties of constraining hanging nodes. *Computational Mechanics*, 55(3):499–517, 2015.

[206] D. D'Angella, N. Zander, S. Kollmannsberger, F. Frischmann, E. Rank, A. Schröder, and A. Reali. Multi-level *hp*-adaptivity and explicit error estimation. *Advanced Modeling and Simulation in Engineering Sciences*, 3(1):1–18, 2016.

[207] R.E. Bank and R.K. Smith. A posteriori error estimates based on hierarchical bases. *Numerical Analysis*, 30(4):921–935, 1993.

[208] M. Ainsworth and J.T. Oden. A posteriori error estimation in finite element analysis. *Computer Methods in Applied Mechanics and Engineering*, 142(1-2):1–88, 1997.

[209] T. Grätsch and K.J. Bathe. A posteriori error estimation techniques in practical finite element analysis. *Computers & Structures*, 83(4-5):235–265, 2005.

[210] M.R. Dörfel, B. Jüttler, and B. Simeon. Adaptive isogeometric analysis by local *h*-refinement with T-splines. *Computer Methods in Applied Mechanics and Engineering*, 199(5-8):264–275, 2010.

[211] A.V. Vuong, C. Giannelli, B. Jüttler, and B. Simeon. A hierarchical approach to adaptive local refinement in isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 200(49-52):3554–3567, 2011.

[212] A. Buffa and C. Giannelli. Adaptive isogeometric methods with hierarchical splines: error estimator and convergence. *Mathematical Models and Methods in Applied Sciences*, 26(01):1–25, 2016.

[213] G. Kuru, C.V. Verhoosel, K.G. van der Zee, and E.H. van Brummelen. Goal-adaptive isogeometric analysis with hierarchical splines. *Computer Methods in Applied Mechanics and Engineering*, 270:270–292, 2014.

[214] C. Bracco, A. Buffa, C. Giannelli, and R. Vázquez. Adaptive isogeometric methods with hierarchical splines: An overview. *Discrete & Continuous Dynamical Systems*, 39(1):241, 2019.

[215] L. Coradello, P. Antolin, R. Vázquez, and A. Buffa. Adaptive isogeometric analysis on two-dimensional trimmed domains based on a hierarchical approach. *Computer Methods in Applied Mechanics and Engineering*, 364:112925, 2020.

[216] P. Hansbo. Nitsche's method for interface problems in computational mechanics. *GAMM-Mitteilungen*, 28:183 – 206, 11 2005.

[217] F. Chouly, M. Fabre, P. Hild, R. Mlika, J. Pousin, and Y. Renard. An overview of recent results on Nitsche's method for contact problems. *Geometrically unfitted finite element methods and applications*, pages 93–141, 2017.

[218] A. Hansbo, P. Hansbo, and M.G. Larson. A finite element method on composite grids based on Nitsche's method. *ESAIM: Mathematical Modelling and Numerical Analysis*, 37(3):495–514, 2003.

[219] R. Becker, P. Hansbo, and R. Stenberg. A finite element method for domain decomposition with non-matching grids. *ESAIM: Mathematical Modelling and Numerical Analysis*, 37(2):209–225, 2003.

[220] M. Juntunen and R. Stenberg. Nitsche's method for general boundary conditions. *Mathematics of computation*, 78(267):1353–1374, 2009.

[221] F. Chouly, M. Fabre, P. Hild, J. Pousin, and Y. Renard. Residual-based a posteriori error estimation for contact problems approximated by Nitsche's method. *IMA Journal of Numerical Analysis*, 38(2):921–954, 2018.

[222] K. Bandara, T. Rüberg, and F. Cirak. Shape optimisation with multiresolution subdivision surfaces and immersed finite elements. *Computer Methods in Applied Mechanics and Engineering*, 300:510–539, 2016.

[223] T. Kanduvc, C. Giannelli, F. Pelosi, and H. Speleers. Adaptive isogeometric analysis with hierarchical box splines. *Computer Methods in Applied Mechanics and Engineering*, 316:817–838, 2017.

[224] P. Di Stolfo, A. Düster, S. Kollmannsberger, E. Rank, and A. Schröder. A posteriori error control for the finite cell method. *PAMM*, 19(1):e201900419, 2019.

[225] P. Di Stolfo, A. Rademacher, and A. Schröder. Dual weighted residual error estimation for the finite cell method. *Numerical Mathematics*, 27(2):101–122, 2019.

[226] E. Burman, C. He, and M.G. Larson. A posteriori error estimates with boundary correction for a cut finite element method. *arXiv preprint arXiv:1906.00879*, 2019.

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

[228] Y. Bazilevs, V.M. Calo, J.A. Cottrell, J.A. Evans, T.J.R. Hughes, S. Lipton, M.A. Scott, and T.W. Sederberg. Isogeometric analysis using T-splines.

Computer Methods in Applied Mechanics and Engineering, 199(5-8):229–263, 2010.

[229] M.A. Scott, M.J. Borden, C.V. Verhoosel, T.W. Sederberg, and T.J.R. Hughes. Isogeometric finite element data structures based on Bézier extraction of T-splines. *International Journal for Numerical Methods in Engineering*, 88(2):126–156, 2011.

[230] M.A. Scott, X. Li, T.W. Sederberg, and T.J.R. Hughes. Local refinement of analysis-suitable T-splines. *Computer Methods in Applied Mechanics and Engineering*, 213:206–222, 2012.

[231] A. Buffa, G. Sangalli, and R. Vázquez. Isogeometric methods for computational electromagnetics: B-spline and T-spline discretizations. *Computational Physics*, 257:1291–1320, 2014.

[232] K.A. Johannessen, T. Kvamsdal, and T. Dokken. Isogeometric analysis using LR B-splines. *Computer Methods in Applied Mechanics and Engineering*, 269:471–514, 2014.

[233] K.A. Johannessen, F. Remonato, and T. Kvamsdal. On the similarities and differences between Classical Hierarchical, Truncated Hierarchical and LR B-splines. *Computer Methods in Applied Mechanics and Engineering*, 291:64–101, 2015.

[234] D. Thomas, L. Engvall, S. Schmidt, K. Tew, and M.A. Scott. U-splines: Splines over unstructured meshes. *Preprint, https://coreform.com/usplines*, 2018.

[235] S. Badia, E. Neiva, and F. Verdugo. Linking ghost penalty and aggregated unfitted methods. *Computer Methods in Applied Mechanics and Engineering*, 388:114232, 2022.

[236] E. Burman and P. Hansbo. Edge stabilization for the generalized Stokes problem: a continuous interior penalty method. *Computer Methods in Applied Mechanics and Engineering*, 195(19-22):2393–2410, 2006.

[237] J.A. Evans and T.J.R. Hughes. Explicit trace inequalities for isogeometric analysis and parametric hexahedral finite elements. *Numerische Mathematik*, 123(2):259–290, 2013.

[238] A. Kroó. On Bernstein–Markov-type inequalities for multivariate polynomials in Lq-norm. *Approximation Theory*, 159(1):85–96, 2009.

[239] P. Hansbo, M.G. Larson, and S. Zahedi. A cut finite element method for a Stokes interface problem. *Applied Numerical Mathematics*, 85:90–114, 2014.

[240] M.G. Larson and S. Zahedi. Stabilization of high order cut finite element methods on surfaces. *IMA Journal of Numerical Analysis*, 40(3):1702–1745, 2020.

[241] L. Schumaker. *Spline Functions: Basic Theory*. Cambridge Mathematical Library. Cambridge University Press, 3 edition, 2007.

[242] L.B. da Veiga, A. Buffa, G. Sangalli, and R. Vázquez. Mathematical analysis of variational isogeometric methods. *Acta Numerica*, 23:157–287, 2014.

[243] W. Dörfler. A convergent adaptive algorithm for Poisson's equation. *SIAM Journal on Numerical Analysis*, 33(3):1106–1124, 1996.

[244] I. Babuvska and B.Q. Guo. Approximation properties of the hp version of the finite element method. *Computer Methods in Applied Mechanics and Engineering*, 133(3-4):319–346, 1996.

[245] R. Verfürth. *A Review of a Posteriori Error Estimation Adaptive Mesh-Refinement Techniques*, volume 50. 05 1996.

[246] T. Hoang. *Isogeometric and immersogeometric analysis of incompressible flow problems*. PhD thesis, PhD thesis, Department of Mechanical Engineering, 2018. Proefschrift, 2018.

[247] F. de Prenter. *Preconditioned iterative solution techniques for immersed finite element methods*. PhD thesis, PhD thesis, Department of Mechanical Engineering, 2019. Proefschrift, 2019.

[248] D. D'Angella, N. Korshunova, and L. Kudela. DirectFEM - ReveaL, 2021.

Appendix A

Midpoint tessellation procedure

As discussed in Section 2.3.1, at the lowest level of bisectioning we apply a tessellation procedure that provides an order $\mathcal{O}(h^2/2^{2\varrho_{\max}})$ approximation of the interior volume [58], with h the size of the background element in which the sub-cell resides and with $\varrho_{\max}$ the number of octree bisections. The considered procedure provides an explicit parametrization of both the interior of the trimmed sub-cell and its immersed boundary.

The employed tessellation procedure is illustrated for a two-dimensional sub-cell in Figure A.1. In order to acquire a tessellation of the immersed boundary and the corresponding trimmed interiors, the following steps are taken:

(a) The immersed boundary, *i.e.*, the blue curve in Figure A.1a, is assumed to corresponds to the zero level set of a continuous function.

(b) The level set function is evaluated in the vertices of the quadrilateral cell, and a bi-linear interpolation is used to approximate the level set value in the center point. In Figure A.1b positive level set values are indicated by green circles with a plus sign, and negative ones by a red circle with a minus sign.

(c) For all edges of the sub-cell we consider a linear interpolation of the level set function along the edge, and, if the level set values at the edge vertices have opposite signs, we determine the approximate zero level set point based on this linear interpolation. If an edge is intersected by the immersed boundary, it is split up in a positive part and a negative part based on the determined edge intersection. The approximate zero level set points are

indicated by the blue circles with a 0 in Figure A.1c, and the splitting of the edges is visualized by the edge colors.

(d) Along each of the 4 lines connecting the center of the square with its vertices, linearly interpolated zero level set points are determined. For the case considered in Figure A.1d, the two additionally obtained zero level set points are indicated by the blue squares on the yellow diagonals.

(e) An additional zero level set point is formed by taking the arithmetic mean of the coordinates of the zero level set points along the diagonals. This point, which we refer to as the (approximate) midpoint of the trimmed boundary, is illustrated by the blue circle in the interior of the cell in Figure A.1e.

(f) The trimmed boundary is then constructed by extruding the edge intersections toward the computed approximate midpoint. As can be seen from Figure A.1f, a piece-wise linear approximation of the trimmed boundary in Figure A.1a is obtained.

(g-i) Interior cells are then constructed by extruding the edges toward the midpoint, thereby creating triangular integration sub-cells. The collection of sub-cells pertaining to the positive side of the trimmed elements are shown in green in Figure A.1h, whereas the negative side of the element is shown in red in Figure A.1i.

From this tessellation procedure applied in two dimensions it is evident that it traverses through the dimensions of the problem, in the sense that it first trims the edges (a–c), after which the quadrilateral is trimmed through the extrusion of the edges to the computed mid-point (d–i). This approach directly extends to the three-dimensional case, an illustration of which is presented in Figure A.2. In Figure A.2b the level set function is evaluated in all vertices of the cube. For each of the six faces of the cube the two-dimensional procedure as described above is applied, as illustrated on the unfolded cubes in Figures A.2d–A.2e. Finally, the trimmed faces are extruded toward the computed mid-point of the cube, as illustrated in Figures A.2g–A.2h. Note that in the three-dimensional case both tetrahedrons and pyramids are created, based on whether or not a face is trimmed. Integration rules are available for both these shapes.

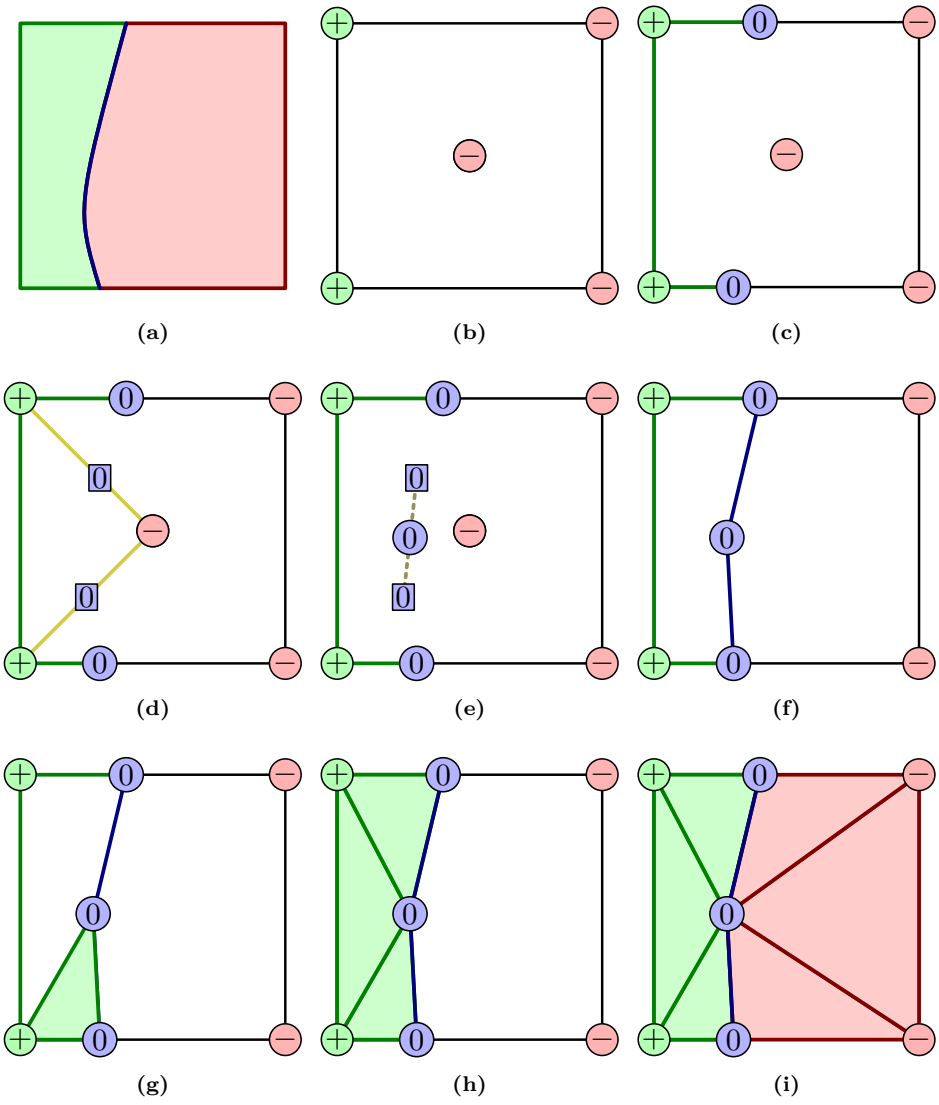


Figure A.1: Schematic representation of the mid-point tessellation procedure for a two-dimensional case.

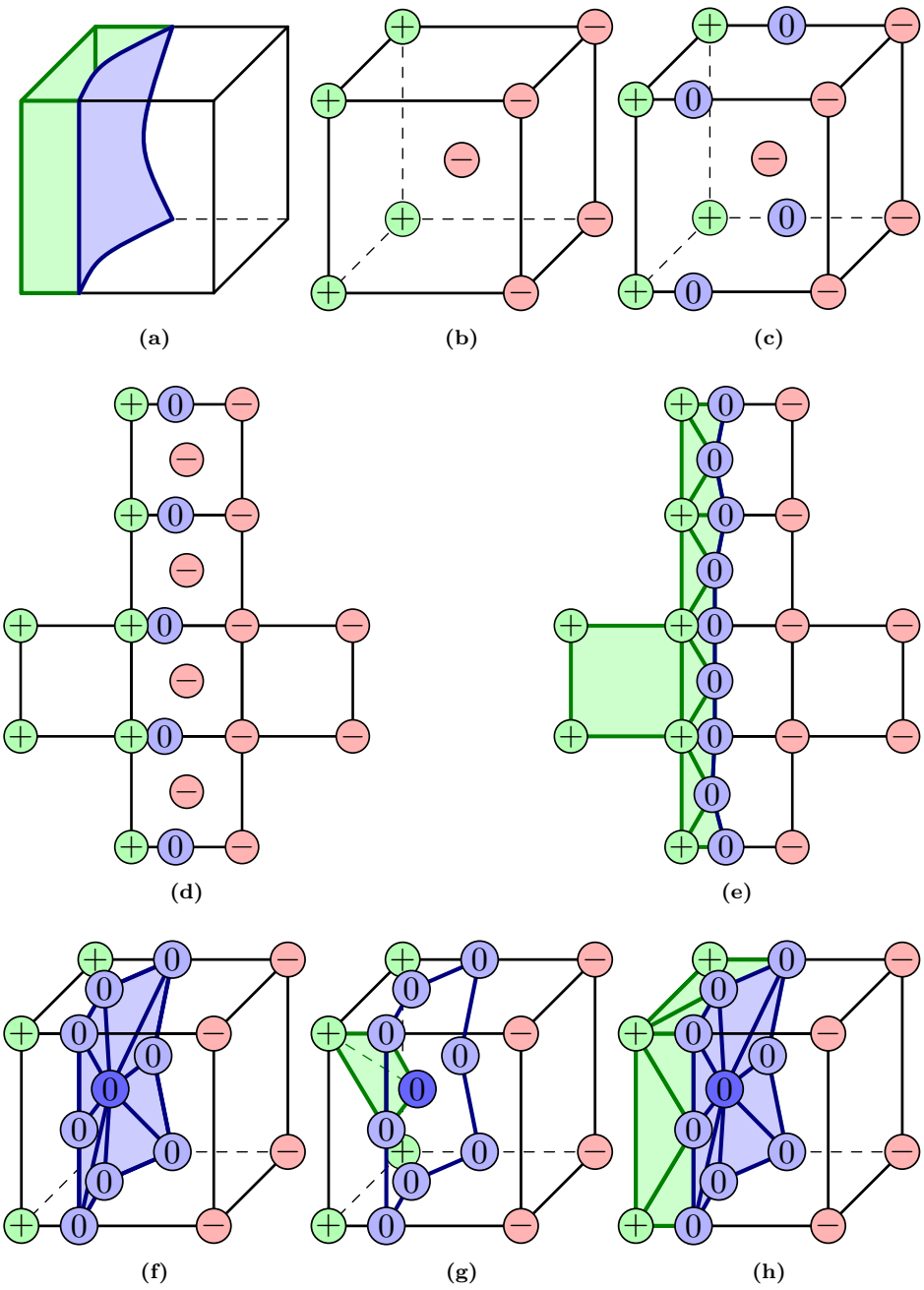


Figure A.2: Schematic representation of the mid-point tessellation procedure for a three-dimensional case.

Acknowledgements

The work in this dissertation would not have been possible without the help of certain people. Firstly, I would like to express my greatest gratitude to Clemens for being an exceptional supervisor who helped me with a perfect and practical planning towards the challenges addressed in this dissertation. I would like to extend my gratitude to Harald for providing me with a deep insight into theoretical aspects with many interesting discussions. I really appreciate your swift review of all sections of this dissertation over the past four years. Additionally, thanks to both of you for encouraging my interest in developing my teaching skills.

Charles Darwin said that, in the long history of humankind, those who learned to collaborate most effectively have prevailed. The nature of the SEED PhD program gave me a splendid opportunity to collaborate with Ale and Auricchio, who, always, helped improve my research output and encouraged me to discover suitable applications for my research work. I would like to extend my gratitude to Giancarlo Sangalli.

Unlike most people, I did not start this journey alone. My office mates, Elisa and Tristan, and I started on the same day. A big thanks to Elisa for creating a very welcoming environment from the first day of our PhD and giving me moral and emotional support. Thanks to Tristan for the initial discussions, ensuring a smooth start to my PhD project, and helping me resolve certain mathematical misunderstandings. Our shared office gave us opportunities for both technical and nontechnical discussions. Thanks to both of you for introducing me to the Dutch culture.

I also would like to thank Gertjan for resolving many Nutils programming issues and indirectly helping me in improving my programming skills. A very special thanks to Frits for helping me understand the theoretical aspects of the project and for giving me a very detailed review of Chapter 4.

I am very grateful to Hasini, who was not only a vibrant colleague but also a true 3am friend – literally. I would like to extend thanks to her husband Vinoth for feeding me home (Indian) food on multiple occasions. A special thanks to

family Verhoosel for letting me work at your home. My wholehearted thanks to Daan, Robin, Stein, Daniel, Herman, Tuong, Chaozhong, Benjamin, Timo, Eivind, Pieter and Michael for creating a good working atmosphere and for the many joyful discussions in the group. Furthermore, I gratefully acknowledge the Computational Mechanics team at UniPv for their great hospitality. Special thanks to Alessia, Laura, Gian, Alfredo and Mauro for creating lively environment at the office in Pavia. Michele Conti is acknowledged for providing the scan-data presented in Chapters 3 and 4 and Alice Finotello for the help in understanding the biomedical scan data. Thanks to Fankai and Anke for discussions regarding their Master's projects which gave me a deeper insight and understanding of certain concepts in the field. My deepest gratitude to the secretaries Linda at TU/e, Sonia at UniPv and Lelia at CIMNE for making all bureaucratic work effortless.

Before building a personal life in the Netherlands, a few people helped in filling the empty space in my life outside work. I am always grateful to Sindhu, Sush, Kiki, Veena, Nikki, Dada (Karthik), Shreeja, Ann-Katherin and Manasi who remotely gave me enormous strength and emotional support in various situations. Many thanks to the friends I made in the Netherlands over this period: Anke, Christos, Elsa, Emma, Janet, Monika, Stefan, Torre and Vera.

I want to thank my father Saigopal and mother Vasanthi for giving me unconditional support and freedom to pursue my career dreams. The essential values they taught me in life helped me in pushing myself and trying to give my best. I would like to extend my gratitude to my sister Harini and her family.

The reward of being accepted is often greater than the reward of winning something. I am very grateful to my bonus parents, Ria and Rien, for accepting me as a part of their family. I would like to extend thanks to the family Reekoert. Last but not least, thanks to Ellen - my love, for giving me a stable life, home, family and friends. Without your support I wouldn't have been able to finish this dissertation.

Summary

The rapid evolution of the field of scientific computing over the past decades has driven the development of various numerical techniques for mathematical-physical modeling. In particular, the Finite Element Method (FEM) has been an astonishing success in both solid mechanics and fluid dynamics. In recent years, immersed finite element methods have been proven to be suitable for problems for which the performance of standard, mesh-fitting, FEM is impeded by complications in the meshing procedure. Immersed techniques have been successfully combined with Isogeometric Analysis (IGA) – a spline-based higher-order finite element framework originally targeting the integration of FEM and Computer Aided Design (CAD) – enabling its application to complex three-dimensional problems. Specifically, immersed IGA can be applied for scan-based analyses, which is, for example, of interest in the fields of biomechanics (patient-specific analysis), geomechanics and material science.

Scan-based simulations contain innate topologically and geometrically complex three-dimensional domains, represented by large data sets in formats which are not directly suitable for analysis. Consequently, performing high-fidelity scan-based simulations at practical computational costs is still very challenging. The main objective of this dissertation is to develop an efficient and robust scan-based simulation strategy by acquiring a profound understanding of three prominent challenges in scan-based IGA, *viz.*: *i)* balancing the accuracy and computational effort associated with numerical integration; *ii)* the preservation of topology in the spline-based segmentation procedure; and *iii)* the control of accuracy using error estimation and adaptivity techniques.

In three-dimensional immersed isogeometric simulations, the computational effort associated with integration can be the critical component. A myriad of integration strategies has been proposed over the past years to ameliorate the difficulties associated with integration, but a general optimal integration framework that suits a broad class of engineering problems is not yet available. In this dissertation we provide a thorough investigation of the accuracy and computational effort of the octree integration technique. We quantify the contribution of

the integration error using the theoretical basis provided by Strang's first lemma. Based on this study we propose an error-estimate-based adaptive integration procedure for immersed IGA.

To exploit the advantageous properties of IGA in a scan-based setting, it is important to extract a smooth geometry. This can be established by convoluting the voxel data using B-splines, but this can induce problematic topological changes when features with a size similar to that of the voxels are encountered. This dissertation presents a topology-preserving segmentation procedure using truncated hierarchical (TH)B-splines. A moving-window-based topological anomaly detection algorithm is proposed to identify regions in which (TH)B-spline refinements must be performed. The criterion to identify topological anomalies is based on the Euler characteristic, giving it the capability to distinguish between topological and shape changes. A Fourier analysis is presented to explain the effectiveness of the developed procedure.

An additional computational challenge in the context of immersed IGA is the construction of optimal approximations using locally refined splines. For scan-based volumetric domains, hierarchical splines are particularly suitable, as they optimally leverage the advantages offered by the availability of a geometrically simple background mesh. Although truncated hierarchical B-splines have been successfully applied in the context of IGA, their application in the immersed setting is largely unexplored. In this dissertation we propose a computational strategy for the application of error estimation-based mesh adaptivity for stabilized immersed IGA.

The conducted analyses and developed computational techniques for scan-based immersed IGA are interrelated, and together constitute a significant improvement in the efficiency and robustness of the analysis paradigm. In combination with other state-of-the-art developments regarding immersed FEM/IGA (*e.g.*, iterative solution techniques, parallel computing), the research in this thesis opens the doors to scan-based simulations with more sophisticated physical behavior, geometries of increased complexity, and larger scan-data sizes.

Curriculum Vitæ

Sai Chandana Divi
Born on January 24, 1993 in Tirupati, Andhra Pradesh, India.

Jan. 2022 - present	Post-doctoral candidate, Technische Universiteit Eindhoven, The Netherlands.
Sept. 2017 - Dec. 2021	Joint doctoral candidate, Technische Universiteit Eindhoven, The Netherlands, and, University of Pavia, Italy.
Sept. 2015 - Aug. 2017	Joint M.Sc. in Computational Mechanics, Swansea University, UK, and, Universitat Politècnica de Catalunya, Spain.
Nov. 2014 - May 2015	Senior project assistant, Department of Mechanical Engineering, Indian Institute of Technology Madras, Chennai, India.
Aug. 2010 - Aug. 2014	Bachelor of Engineering in Mechanical Engineering, Sri Chandrasekharendra Saraswathi Viswa Maha Vidyalaya, Kanchipuram, India.
June 1995 - July 2010	Secondary school & intermediate education Sri Venkateswara Children's high school, and Bhupal junior college (resp.), Tirupati, India.