Title

Interpolation-based methods for computational mechanics

Author

Fromm, Jennifer Ellen

Publication Date

2024

UNIVERSITY OF CALIFORNIA SAN DIEGO

Interpolation-based methods for computational mechanics

A dissertation submitted in partial satisfaction of the
requirements for the degree Doctor of Philosophy

in

Engineering Sciences (Mechanical Engineering)

by

Jennifer E. Fromm

Committee in charge:

Professor Jiun-Shyan Chen, Chair
Professor Boris Kramer
Professor Petr Krysl
Professor Xiaochuan Tian

2024

The Dissertation of Jennifer E. Fromm is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2024

DEDICATION

For all the women who stuck with engineering

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

Every PhD is a unique journey and mine has been no exception. I would like to first thank my current advisor Prof. JS Chen for his guidance and mentorship. I could not have finished this degree without his willingness to step in and advise me towards the end of my degree. I must also thank my previous advisor Prof. David Kamensky for giving me the chance to join his lab and begin working in the field of computational mechanics. David gave me the opportunity to enjoy research again, and his technical expertise and support was invaluable as I found my footing in a new field. In addition, I want to thank Prof. Padmini Rangamani and Prof. Stefan Llewellyn Smith, who brought me to UCSD and advised me for the first years of my degree. Thank you for teaching me how to persevere through difficult research and for empowering me to pursue my passions, even when they meant we parted ways.

I next need to acknowledge the mentorship of Prof. John Evans, who's support has been crucial through the end of my degree. I am incredibly lucky to be a recipient of his guidance, both on matters technical and professions. Along with Prof. Kurt Maute, the EXHUME team from CU Boulder provided me with much needed continuity of research through the end of my program. I am also extremely grateful to have worked with my fellow grad student Nils Wunsch. The technical work in this thesis, particularly in chapters 4 and 5 depends on the contributions he made to the software developed by Kurt's lab. I am also thankful for our many technical discussions and am excited to see what our futures hold.

Having worked with several labs, I have many labmates I want to thank. First and foremost, I need to thank Ru Xiang and Han Zhao. They welcomed me to the Kamensky group and also contributed to the work presented in chapter 4 of this dissertation. Beyond research, they have become wonderful friends and an integral part of my support in San Diego. From the Chen lab, I was to thank Kristen, Iris, Ryan, Sam, and Karan for their support both technical and emotional. From the Kamensky group, thank you to Marco, Josh and Sebastiaan. And from the Rangamani and Llewellyn Smith groups, thank you to Arijit, Miriam, Lucas, Cuncheng, Justin, Michael, Allen, Haleh, Chris, Can, Andrea, Donya, Tianyi, and Todd.

I have made so many wonderful friends during my time in San Diego that have supported me throughout this process. I want to thank Jack, Lysha, Cathleen, Hossein, and Brianna M from my MAE cohort. Thank you to the craft group, Andrea, Elisa, Sara and Brianna S. And a big thank you to my roommates past and present, Jenna, Patches, Chris, Nicole, and Winona.

Finally, I need to thank my incredible family. My parents are so awesome and I am so lucky to have their support in all forms. Thank you for being amazing! Thanks to my brother Jack for showing me its possible to be happy in the real world outside academia, and thanks to my sister Cayla and her fiance Ethan for showing me you can be happy *after* academia to. Cayla, I could not have done this without you having done it first. Thank you so much!

Parts of chapters 1 and 2 and chapter 4, in full, is a reprint of the material as it appears in Computer Methods in Applied Mechanics and Engineering, Fromm, Jennifer E; Wunsch, Nils; Maute, Kurt; Evans, John A; and Kamensky, David; 2023. The dissertation author was the primary researcher and author of this paper. Notation has been changed to maintain consistency within the dissertation.

Parts of chapters 1, 2, and 3 and chapter 5, in full, is a reprint of the material as it appears in Journal of Computational Mechanics, Fromm, Jennifer E; Wunsch, Nils; Maute, Kurt; Evans, John A; and Chen, Jiun-Shyan; 2024. The dissertation author was the primary researcher and author of this paper. Notation has been changed to maintain consistency within the dissertation.

Parts of chapters 1, 2, and 3 and chapter 6, in full, is a reprint of the material being prepared for publication by, Fromm, Jennifer E; Evans, John A; an Chen, Jiun-Shyan; 2024. The dissertation author was the primary researcher and author of this work.

VITA

2018	B.S. in Mechanical Engineering, Union College, Schenectady, New York
2020	M.S. in Engineering Sciences (Mechanical Engineering), University of California San Diego
2024	Ph.D. in Engineering Sciences (Mechanical Engineering), University of California San Diego

PUBLICATIONS

JOURNAL ARTICLES

- **Jennifer E. Fromm**, Nils Wunsch, Ru Xiang, Han Zhao, Kurt Maute, John A. Evans, and David Kamensky. Interpolation-based immersed finite element and isogeometric analysis. Computer Methods in Applied Mechanics and Engineering, 405:115890, February 2023.

- **Jennifer E. Fromm**, Nils Wunsch, Kurt Maute, John A. Evans, and Jiun-Shyan Chen. Interpolation-based immersogeometric analysis methods for multi-material and multi-physics problems. Comput Mech, June 2024

- (under preparation) **Jennifer E. Fromm**, John A. Evans, and Jiun-Shyan Chen. Interpolation-based reproducing kernel particle methods.

ABSTRACT OF THE DISSERTATION

Interpolation-based methods for computational mechanics

by

Jennifer E. Fromm

Doctor of Philosophy in Engineering Sciences (Mechanical Engineering)

University of California San Diego, 2024

Professor Jiun-Shyan Chen, Chair

Computational mechanics tools make up a crucial part of modern engineering analysis. The most widely used softwares typically employ finite element analysis (FEA) to approximate the solutions to partial differential equations (PDEs). FEA relies on the construction of high-quality boundary fitted meshes which can be costly to construct. More specialized numerical methods are widely used within the academic community to not only avoid mesh generation and remeshing, but also to tackle problems with unique requirements not well suited to FEA. This work expands upon the theory of approximate extraction and interpolation-based methods, enabling more advanced computational tools to be implemented within existing finite element softwares.

The first method discussed in this work is the interpolation-based immersed boundary method. Immersed boundary methods avoid mesh generation by embedding a problem's computational domain in a structured background grid. With non-conforming meshes comes additional difficulty in integration steps, limiting the execution of most immersed boundary methods to custom research codes. Interpolation-based immersed boundary methods augment existing FEA software to non-invasively implement immersed-boundary capabilities through extraction. Extraction, which has previously been applied to implement isogeometric analysis in finite element codes, interpolates the structured background basis as a linear combination of Lagrange polynomials which can be easily integrated with existing software. In addition to classic immersed boundary methods, immersed-isogeometric methods are also implemented, using B-splines as background basis functions. B-splines offer higher levels of continuity than classic finite element basis functions, enabling the method to model high-order derivative PDEs, such as shell problems. Heaviside enrichment of the background basis as well as local refinement through truncated hierarchically refined B-splines extends the capability of this method to multi-material problems, including image-based analysis of composite materials.

Extending the theory of interpolation, this work also introduces interpolation-based meshfree methods, specifically interpolation-based reproducing kernel particle method (RKPM) . Developed for similar reasons as immersed boundary methods, meshfree methods do not require a mesh connectivity data structure and are optimal for modeling large deformation, fracture, or extreme events. Also like immersed boundary methods, meshfree methods require sophisticated integration techniques that make them challenging to implement outside a research setting. Interpolation-based RKPM is implemented within existing FEA software, and applied in this work to a variety of problems, including high order derivative PDEs and multi-material problems.

Chapter 1

Introduction

1.1 Motivation

Computational mechanics play a vital roll in modern engineering. Computer aided design (CAD) software has been widely employed in industry and most every product manufactured today starts out as a CAD file. Computer aided analysis is less wide spread, but is standard whenever products are too costly (or dangerous) to build and test prior to widespread manufacture. Computer simulations can improve traditional engineering design and can provide insight not possible to glean from physical experiments. This thesis will summarize the author's contributions to the field of computational mechanics and the technical tools she has created to improve access to computer aided analysis methods.

The industry standard of computational mechanics tools is finite element analysis (FEA), somtimes referred to as simply finite element (FE). FEA is an incredibly powerful tool for solving partial differential equations (PDEs). The method has been applied to many classes of engineering problems, including structural analysis, fluid mechanics, electro-magnetism, plasma physics, and other highly coupled multi-physics systems. A key component of FEA is the discretization of a physical or geometric domain into a computational mesh. This mesh must conform to the domain geometry and accurately represent all physical features, while also adhering to strict quality metrics. The quality of FEA simulation is closely tied to the quality of the mesh, especially when employing high order basis functions, and the oft-cite study of

Sandia engineers [31] found that up to 80% of an engineer's billable time was consumed in the generation and processing of geometric meshes. Especially as computational power becomes cheaper and computers faster, the slowest part of computer aided engineering (CAE) remains the human driven mesh generation steps. In order to increase the access to computational methods, the problems presented by mesh generation must be addressed.

Considerable work has been done in the computational mechanics community to improve mesh generation for more efficient FEA workflows. These methods are crucial to the field and have done much to improve the CAE processes, but they remain cumbersome. There are also applications where the creation of a mesh, or the re-meshing of a highly deformed material, is simply unfeasible. This thesis will instead deal with methods that avoid mesh generation all together. Three classes of methods are discussed: isogeometric analysis (IGA), immersed boundary methods, and meshfree methods.

IGA was created to interface directly with CAD. The method employs the same mathematical basis used to describe geometric entities in CAD in analysis. By using the CAD discretization for analysis, both meshing and the introduction of geometric error are avoided. While classic FEA uses Lagrange polynomial basis functions, IGA utilizes B-splines, or non-uniform rational B-splines (NURBS), to approximate PDEs. While seamless integration within CAD has yet to be fully realized, IGA and the use of IGA bases provides additional benefits over FEA, including higher accuracy per degree of freedom. Due to the higher levels of continuity presented by B-splines, IGA has found use in the modeling of thin shelled structures and other high order derivative problems.

Immersed boundary method were initially developed for fluid-structure interactions and embed the computational domain within a structured non-conforming mesh. Because of their flexibility in representing complex geometric features, immersed boundary methods are popular for simulating interface problems of problems with evolving geometries. The computational mesh remains constant regardless of the deformation of the material being modeled, or changes to geometry made during design iterations. Basis functions on the background mesh may be arbitrarily intersected by domain interfaces, making integration and matrix assembly more challenging than for traditional FEA.

Meshfree methods encompass a broad range of tools that eschew meshes entirely. These methods typically define basis functions on a set of floating points that are not constrained by mesh connectivity, making them ideal for modeling large deformations or extreme events. Major topological changes, such as material fracture or fragmentation, can be modeled through such methods. The reproducing kernel particle method (RKPM), a subset of meshfree methods, also enables for the decoupling of continuity and polynomial reproducibility. This allows for high order methods to be applied to these discontinuous problems. Meshfree methods in general, and RKPM in particular, also allow for convenient local refinement through node insertion [163]. As with immersed boundary methods, enforcement of boundary conditions and integration can be a challenge.

These classes of methods each make important contributions to the field of computational mechanics, and could enable design engineers and engineering analysts to perform simulations not possible through widely available FEA software. However, the use of many of the methods has been limited to academic specialists, and often the only users of the methods are the developers themselves. The aim of this thesis is to broaden the access to specialty methods and their unique benefits through the use of interpolation.

Interpolation-based methods were driven initially by the IGA community. The basis functions utilized by IGA are rational functions with large supports, and are difficult to handle within element-based classical FEA software. The idea of extraction, to represent the B-spline or NURBS basis functions in terms of Bèzier polynomials, was introduced in [34, 148]. The concept was later generalized to Lagrange extraction, where the Bèzier polynomials are further represented in terms of Lagrange polynomials [144]. In this context, extraction is also referred to as interpolation. Lagrange polynomials are traditionally used in classical FEA and a plethora of software, both open-source and commercial, exists to integrate such functions. Seminal works [97, 159] utilized interpolation to retrofit existing FEA software to perform IGA.

1.2 Objectives

The objective of this thesis is to broaden access to specialized computational methods. The methods prioritized in this work reduce the burden of mesh generation or geometric discretization. This goal is realized through the development of interpolation-based methods and the adaptation of existing software tools to implement these methods. The primary products of this work are as follows

- The introduction of the theory of approximate interpolation. Previously existing interpolation-based methods exactly represented target background basis functions as linear combinations of foreground functions. Theory detailed in this work proves special cases of approximate interpolation-based methods still yield optimal error convergence rates, and numerical results demonstrate optimal rates even for cases where theory has yet to be derived.

- Interpolation-based immersed finite element and immersogeometric methods, which employ approximate interpolation to implement immersed methods within existing finite element software. Benchmarking problems demonstrate optimal error convergence rates for these methods for a multitude of PDEs and geometries.

- The extension of interpolation-based immersogeometric methods to multi-material problems through level-set Heaviside enrichment. Optimal error convergence rates are demonstrated through numerical examples on complex geometries. Target application spaces are illustrated via image-based composite material analysis.

- The presentation of interpolation-based RKPM, a specialized example of interpolation-based meshfree methods. In addition to utilizing existing FEA software, the interpolation-based approach demonstrates considerable speed ups in the computation of meshfree derivatives and integration of shape functions.

- The publication of open-source software modules to both reproduce all the numerical examples presented in this thesis, and enable lay-users to apply the methods presented here to additional PDEs and geometries.

1.3 Outline

This thesis contains six remaining chapters. Chapter 2 provides a literature review on the computational methods that interpolation is applied to in this work (immersed boundary methods and RKPM), and an overview of the prior use of interpolation-based methods. Chapter 3 provides the mathematical background on the nonstandard discretizations techniques and basis functions employed by IGA and RKPM. Chapter 4 introduces interpolation-based immersed boundary methods and immersed-isogeometric methods. The theory of approximate extraction is derived. Several benchmarking problems validate the theory. The method's utility is illustrated through the implementation of trimmed shell analysis within open-source FEA software. Chapter 5 extends the body of work on interpolation-based immersed-isogeometric methods to include multi-physics and multi-material problems through Heaviside enrichment truncated hierarchically refined B-splines. The theory behind the enrichment and refinement strategies is given, as well as the developments to the interpolation-based method to accommodate the more complex methods. Image-based thermo-mechanical analysis of a composite sample demonstrates the method's capability. Chapter 6 introduces interpolation-based meshfree methods, focusing on the reproducing kernel particle method. The final chapter will summarize the key points of this thesis and indicate future directions the dissertation's author recommends for the further work on interpolation-based methods.

Acknowledgements

A portion of this chapter has been accepted for publication in Fromm, Jennifer E; Wunsch, Nils; Maute, Kurt; Evans, John A; and Kamensky, David "Interpolation-based immersed finite

element and isogeometric analysis," Computer Methods in Applied Mechanics and Engineering, 2023. The dissertation author was the primary researcher and author of this paper.

A portion of this chapter has been accepted for publication in Fromm, Jennifer E; Wunsch, Nils; Maute, Kurt; Evans, John A; and Chen, Jiun-Shyan; "Interpolation-based immersogeometric analysis methods for multi-material and multi-physics problems", Journal of Computational Mechanics 2024. The dissertation author was the primary researcher and author of this paper.

A portion of this chapter is currently being prepared for submission in Fromm, Jennifer E; Evans, John A; and Chen, Jiun-Shyan; "Interpolation-based reproducing kernel particle methods" 2024. The dissertation author is the primary researcher and author of this work.

Chapter 2

Literature Review

2.1 Immersed finite element methods

Immersed-boundary methods, also called unfitted methods or cut-cell methods, were first introduced in the computational mechanics literature to deal with moving interfaces in finite volume methods by Peskin in [134]. This work presented a method for modeling fluid–structure interactions in the context of blood flow through a heart valve, an application which remains of interest in the immersed-boundary community to this day. In this work, the impenetrable boundary is modeled as a parametric curve embedded in a two-dimensional flow field which exerts force on the flow and is linked with a no-slip boundary condition. Similar methods summarized in the 2005 review [122] were used in fluid–structure interaction and fluid–fluid interface codes with developments to the boundary condition enforcement and boundary tracking.

The method of embedding material in a Cartesian grid also gained traction within the solid mechanics community and was influenced by both developments in meshfree or particle methods [113, 156] and particle in cell methods from the fluids community [35]. These methods differed from previous immersed-boundary methods as they typically dealt with a solid undergoing significant deformation, which would have required remeshing in classical finite element methods, rather than a solid interacting with a surrounding fluid. Immersed-boundary methods were likewise applied to model crack propagation, as in [124].

Immersed boundary methods circumvent conforming mesh generation by embedding the geometric problem domain into a background grid constructed on a geometrically simple domain. Initially proposed to track fluid-structure interfaces in [134], similar classes of immersed methods were likewise developed by the solid mechanics community to accommodate discontinuities in solution fields without remeshing to create boundary-fitted meshes. The partition of unity method (PUM), introduced in [5], leverages the concept of enriching solution functions using *a priori* knowledge of the location of discontinuities. This was combined with classical FEM in [154] and [153] to introduce the generalized finite element method (GFEM). A similar enriched method, characterized by adaptive enrichment schemes, known as the 'eXtended' finite element method (XFEM) [124, 25, 28], was also introduced to model crack propagation and other discontinuous problems.

Immersed boundary methods have also been extended to include high-order methods. The finite cell method[133, 143] utilizes p refinement of Lagrange polynomial basis functions to increase convergence rates, however the issue of integration error remained. Numerous solutions have been proposed to reduce the geometric approximation error, including octree adaptive integration [54] and moment fitting adaptive integration [89].

In the field of meshfree methods, the concepts of immersed or embedded methods have been used to model heterogeneous materials [145] and to enhance solution accuracy and stability near material interfaces in fluid-structure interaction problems [87].

Specific to finite element methods, the generalized finite element method (GFEM) [154] and the subsequent extended finite element method (XFEM) [124] furthered development in immersed-boundary methods. These methods, reviewed in [66], embed a material in a Cartesian grid and "extend" the solution space by enriching it, while XFEM distinguishes itself by enriching only locally near boundaries and discontinuities. These methods rely on a priori knowledge of the partial differential equation (PDE) solution drawn from partition of unity methods (PUM) [65].

The immersed framework has also been extended to the field of meshfree methods, particularly RKPM [145]. Neural networks have been employed to enrich RKPM approximation spaces for localization [10], fractures [9], and problems with local features [11].

TraceFEM methods [131] are a different class of immersed-boundary methods based on surface PDE descriptions (as derived in [56] and adapted in [57]). These methods are elegant formulations for solving PDEs including equations for Stokes flow on surfaces that can be coupled with PDE solutions in bulk [130].

Another recent immersed-boundary method, developed within the open-source software FEniCS, is the multimesh method [94]. This method marks one of the few attempts to integrate immersed-boundary methods into existing FE software distributions rather than research-specific codes. However, the invasive nature of the implementation made it difficult to maintain and it was not retained in the FEniCSx redesign of the software. The difficulty in selective integration is not limited to the multimesh method and remains the major difficulty in the implementation of immersed-boundary methods outside of custom research codes.

Some classes of immersed boundary methods are specially formulated to reduce the difficulty of integrating over cut cells, including approximate domain methods and specifically the shifted boundary method [117, 118]. The shifted boundary method maps the boundaries of a computational domain to a mesh conforming surrogate domain and has been extended to high-order methods in [3]. Similar to the shifted boundary method, interpolation-based immersed methods like the one presented in this work provide the benefits of immersed methods without requiring invasive implementation of custom quadrature methods.

A common issue in most (if not all) immersed boundary methods is the destabilizing effect of small cut cells. This phenomena occurs when a background bases function is only sparsely supported within the computational domain, and can result in ill-conditioning of linear systems and drastically decreased critical timesteps for unsteady problems. Numerous strategieshave been explored to address this issue and its symptoms , such as basis function agglomeration [109], ghost penalty formulations [37], basis function removal [59], and specialized preconditioners [96].

2.1.1 Immersogeometric methods

A historically-separate line of research aimed at circumventing the FE meshing problem is isogeometric analysis (IGA) [90, 51] where function spaces of smooth splines from computer-aided design (CAD) are directly used to approximate PDE solutions. However, these spline spaces often have *more* topological constraints on their construction than classical FE spaces, and industrial CAD models often represent complex geometries by “trimming” [119] spline patches to obtain the desired shapes. Thus instead of offering an alternative to immersed-boundary methods, IGA has intensified interest in them. The symbiotic union of isogeometric and immersed methods is referred to as “immersogeometric analysis” (IMGA) [101].

Since IGA was introduced, numerous methods have been developed to handle trimmed spline patches [106, 139], and IGA was combined with the finite cell method in [140, 137, 141] and with CutFEM to form CutIGA in [59]. As with classic immersed-boundary methods, integrating over cut cells remains a challenge; though it has been addressed for an immersogeometric specific solver using error-estimate based adaptive integration in [54]. Immersogeometric methods were applied to multi-material problems in what was labeled eXtended-IGA, or XIGA [129, 146].

2.2 Meshfree methods

Meshfree methods encompass a large group of computational tools that approximate PDEs with bases constructed on scattered points not connected through a mesh [41, 29]. Meshfree methods stemmed from diffuse element method [126], which was soon followed by the development of element free Galerkin (EFG) methods [26, 27, 115]. Drawing from wavelet methods, the reproducing kernel particle method (RKPM) was introduced in [113] and introduced with discrete reproducing conditions to ensure optimal convergence and extended to large deformation problems in [43]. In addition to circumventing meshing (and re-meshing, in the case of large deformation problems) challenges, meshfree methods offer several advantages over classical FEM. Meshfree methods in general, and RKPM in particular, allow for convenient local refinement through node insertion. While $p-$refinement increases the cost of evaluating RKPM basis functions, it does not increase the number of degrees of freedom (DOFs). And perhaps most notably, the continuity of RKPM bases are independent of polynomial order and controlled solely by the kernel.

As RKPM basis functions are rational and have large overlapping supports, numerical integration of basis functions over domain geometry presents a major challenge in the implementation of RKPM.

The basis functions used in classic finite element methods are compactly supported and can be accurately integrated using Gauss quadrature rules within a desirable order of precision. Gauss quadrature has also be used for meshfree methods [55, 115]. As RKPM functions are by nature rational functions with overlapping supports incompatible with Gauss quadrature integration cells, the number of Gauss quadrature points required to reduce integration error to below approximation error can result in impractical computation times.

Additionally, evaluation algorithms require a mechanism to identify the nearest neighbors of each node to compute basis support size and identify non-zero shape functions at each evaluation point, increasing complexity. Most tasks can be performed off-line as preprocessing steps, but reducing the number of evaluation points is critical to implementing RKPM. Nodal integration schemes with stabilization have thus been developed to enable large scale implementations of meshfree methods [44, 42, 135, 84]. As these integration tools are quite different from those used in conventional mesh-based methods, meshfree methods have yet to be fully adopted in widely used commercial codes.

2.2.1 Meshfree integration techniques

Evaluating RKPM shape functions at a given point (Equation **??**) requires the inversion of the moment matrix, making each evaluation computationally expensive. This computational expense is usually made up for the relative reduction over over-all degrees of freedom when compared with classical FEM, but remains a motivating factor in the development of more efficient nodal integration schemes Direct nodal integration (DNI), where shape functions and their derivatives are only evaluated at nodes presents challenges as the first derivatives of meshfree shape functions vanish at nodes (or nearly vanish at nonuniformly distributed nodes). Vanishing or nearly vanishing derivatives result in an oscillatory mode of wavelength $2h$ (where h is the nodal spacing) leading to instability [84, 29]. Numerous stabilization methods exist, such as the least squares stabilization first proposed for element free Galerkin (EFG) in [24]. In this work, the square of a solution residual is added as a stabilization term to a system's energy functional. As this term contains second order derivatives, which do not vanish at nodes, this serves to stabilize the nodal integration. Unfortunately, the computation of second derivatives, even only at nodal locations, adds considerable computational expense to this method.

Stabilized conforming nodal integration (SCNI), introduced in [44, 45], computes gradients via the divergence of spatially averaged strain, avoiding the computation of derivatives at nodes and also eliminating spurious oscilliatory modes. SCNI with additional stabilization [84] has been introduced to further increase ellipticity and hence stability of nodal integration. SCNI requires the construction conforming representative nodal domains, which can be computational expensive and difficult in the context of large deformations, and has thus be adapted in the form of stabilized non-conforming nodal integration (SNNI)[74, 75]. While SNNI suffers from reduced convergence rates, the addition of variational consistency terms [42] reduces integration error to below method errors.

As nodal integration requires additional stabilization terms added to the variational problem, it is challenging to adapt to the schemes to more complex multi-physics problems. For example, the complex problems investigated in [157] are ill-suited to nodal integration and rely instead on Gaussian integration.

2.3 Weak enforcement of boundary conditions

In both immersed-boundary methods and meshfree methods there is no explicit relationship between nodal degrees of freedom and physical boundaries. Therefore the typical strong enforcement of essential boundary conditions is not possible and essential boundary conditions must be enforced weakly through the problem's variational form.

The simplest boundary condition enforcement method is the penalty method, where deviation from the prescribed value is penalized at the boundary by alteration of the variational form. This method was described in [6] and applied to meshfree methods [166] where strong enforcement of Dirichlet boundary conditions is likewise difficult. While easy to implement, the penalty method results in inconsistent variational forms and can therefore lead to sub-optimal convergence rates.

Boundary condition enforcement using Nitsche's method is more common in the immersed boundary literature and was first proposed in [127] and implemented with formal error estimates in [64]. Nitsche's method was initially derived for the Poisson problem and has been expanded to other PDEs including fluids problems as in [21]. Work done on CutFEM (a subset of immersed-boundary methods similar to the finite cell method but historically separate) includes applications of Nitsche's method for a variety of PDEs, including elliptic problems [79, 81, 22].

A drawback of Nitsche's method is its dependence on a penalty parameter, the magnitude of which must be specified by the user. Several variations of Nitsche's method have been derived to maintain consistency even with a penalty of zero. Notably the non-symmetric method derived for the Laplace problem, Kirchhoff–Love (KL) shells, and elasticity in [142] is stable with zero penalty, as is the penalty-free method for diffusion interface problems in [32].

The Lagrange multiplier method, proposed in [7], adds complexity and is less frequently used as it leads to numeric saddle point problems [36]. It is well-suited for some immersed-boundary problems such as fluid–structure interactions [100] or in multi-physics codes [128] but is not widely adopted.

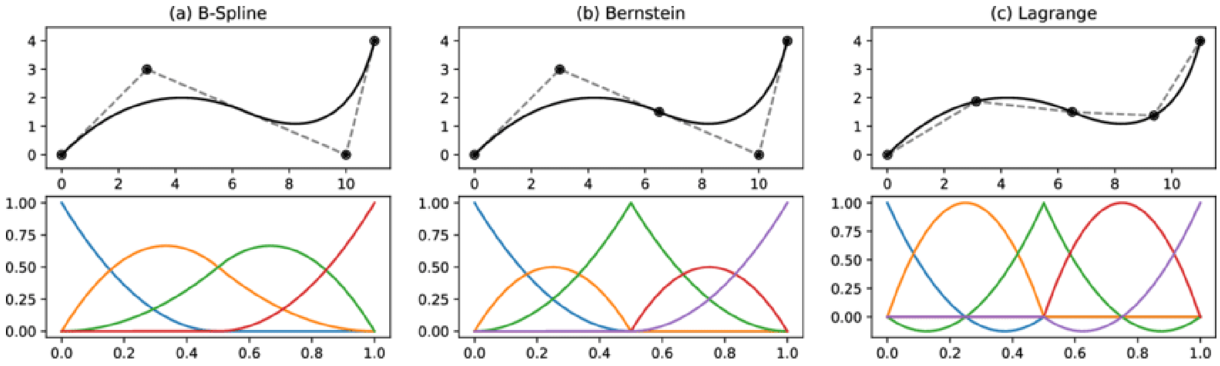


Figure 2.1. A plot from [159], illustrating the implementation of Bézier extraction and Lagrange extraction to exactly interpolate a 1D B-spline curve.

2.4 Interpolation-based methods

A major advance in the design of IGA software was the concept of Bézier extraction [34] where smooth B-spline basis functions are represented as linear combinations of less-regular Bernstein basis functions. This extraction was generalized to T-splines in [148].

The main idea of extraction, as presented in these papers, was to utilize knot insertion to reduce the continuity of the basis representing a given B-spline curve, allowing it to be represented by the less continuious basis of Bézier elements. Each knot insertion decreases the continuity by one at the point of insertion. The crucial characteristic of this insertion is that the continuity of the B-spline curve itself is preserved.

The concept of Bézier extraction was later reformulated as interpolation of IGA basis functions using Lagrange FE bases, referred to as 'Lagrange extraction' [144]. Lagrange extraction allows for the direct reuse of FE software to perform IGA. This has been implemented in both the open-source software package FEniCS [114] in [99] and in the software CodeAster in [159].

This concept is illustrated in Figure 2.1 from [159]. The curve in question and it's highly continuous B-spline basis are shown in (a). The same curve is reproduced in (b) as the linear combination of discontinuous Bernstein polynomials using Bézier extraction. The interpolation is further generalized to Lagrange extraction in (c), where the same curve is represented as a linear combination of Lagrange polynomials. This figure illustrates the extraction performed onto a general curve. With interpolation based methods, the curves being extracted are the B-spline basis functions themselves.

Acknowledgements

A portion of this chapter is a reprint of the material as it appears in Fromm, Jennifer E; Wunsch, Nils; Maute, Kurt; Evans, John A; and Kamensky, David "Interpolation-based immersed finite element and isogeometric analysis," Computer Methods in Applied Mechanics and Engineering, 2023. The dissertation author was the primary researcher and author of this paper.

A portion of this chapter has been accepted for publication in Fromm, Jennifer E; Wunsch, Nils; Maute, Kurt; Evans, John A; and Chen, Jiun-Shyan; "Interpolation-based immersogeometric analysis methods for multi-material and multi-physics problems", Journal of Computational Mechanics 2024. The dissertation author was the primary researcher and author of this paper.

A portion of this chapter is currently being prepared for submission in Fromm, Jennifer E; Evans, John A; and Chen, Jiun-Shyan; "Interpolation-based reproducing kernel particle methods" 2024. The dissertation author is the primary researcher and author of this work.

Chapter 3

Non-standard discretization methods for approximating PDEs

The main contribution of this thesis is the development of interpolation-based tools that increase access to sophisticated computational mechanics methods. The methods covered in this work include isogeometric methods (IGA), reproducing kernel particle methods (RKPM), and immersed boundary methods. The first two, IGA and RKPM, employ basis functions to approximate PDEs that most users of FEA are unfamiliar with. For this reason this chapter

Immersed boundary methods may employ a variety of basis functions, including both classical Lagrange polynomials and the functions utilized in IGA or even RKPM. As this method differs from classical FEA in its treatment of interfaces as opposed to underlying functions, it is discussed alongside the introduction of interpolation-based immersed boundary methods in Chapter 4.

3.1 Isogeometric Analysis (IGA)

Isogeometric analysis (IGA) was first introduced in [90] and developed further in [51] as a method to circumvent time-consuming mesh generation. IGA uses non-uniform rational B-spline (NURBS) basis functions to both create an exact geometric model of a computational domain and to perform computational analysis. NURBS are the standard in the computer-aided design (CAD) and computer graphics communities [138]. By utilizing the native geometric representations used by CAD files for analysis, IGA users avoid the process of mesh creation and eliminate error due to geometric approximations.

A 1D univariate B-spline basis is denoted here as $\left\{B_{i,p}(\xi)\right\}_{i=1}^{N}$, where n is the polynomial order and N is the number of basis functions. The domain is discretized with a knot vector $\Xi = \{\xi_1, \xi_2, ..., \xi_{N+n+1}\}$ such that $\{\xi_i\}_{i=1}^{N+n+1} \subset \mathbb{R}$ and $\xi_1 \leq \xi_2 \leq ... \leq \xi_{N+n+1}$. The functions are then constructed recursively from the piecewise constant basis function

$$B_{i,0}(\xi) = \begin{cases} 1, & \text{if } \xi_i \leq \xi \leq \xi_{i+1} \\ 0, & \text{else} \end{cases}, \tag{3.1}$$

using the Cox-de Boor recursion formula [52]

$$\begin{aligned} B_{i,n}(\xi) = & \frac{\xi - \xi_i}{\xi_{i+n} - \xi_i} B_{i,n-1}(\xi) \\ & + \frac{\xi_{i+n+1} - \xi}{\xi_{i+n+1} - \xi_{i+1}} B_{i+1,n-1}(\xi). \end{aligned} \tag{3.2}$$

If no interior knots are repeated, the basis is C^{n-1} continuous at each knot in the interior domain, and C^{∞} continuous between the knots. The basis will form a partition of unity if the first and last knots are repeated $n+1$ times. Higher dimension basis functions can be constructed by applying tensor-product operations to the univariate functions, such that

$$B_{\boldsymbol{i},\boldsymbol{n}}(\boldsymbol{\xi}) = \prod_{m=1}^{d_p} B^m_{i_m,n_m}(\xi^m), \tag{3.3}$$

where d_p is the parametric space dimension, and there are d_p knot vectors $\Xi^m = \{\xi_1^m, \xi_2^m, ..., \xi_{N_m+n_m+1}^m\}$, where n_m is the number of basis functions and n_m is the polynomial order of the m^{th} parametric direction. Here $\boldsymbol{i} = \{i_1, ..., i_{d_p}\}$ is a multi-index and $\boldsymbol{n} = \{n_1, ..., n_{d_p}\}$ is the vector of polynomial degrees. The B-spline basis of the set of these functions is denoted by $\mathscr{B}_{\boldsymbol{n}} := \{B_{\boldsymbol{i},\boldsymbol{n}}\}$.

For brevity and greater generality, following chapters will drop the polynomial labeling and bolding, referring to functions as only B_i. The notation B_i will also be reused in the context of interpolation-based methods to refer to 'background' basis functions in general, which will, unless otherwise noted, be B-splines.

While IGA development was initially motivated by the difficulty of creating high quality meshes, there are additional benefits to the method. The same mathematical properties that make B-splines attractive to the computer graphics community also enable them to be used for analysis problems outside the scope of traditional finite element analysis. B-splines have been found to be more accurate per degree of freedom than traditional Lagrange polynomials[61, 23], and to have superior spectral properties [91].

Most notably, B-spline basis functions have higher levels of continuity per polynomial degree than Lagrange polynomials. As stated, B-splines of polynomial order n will be C^{n-1} continuous, where traditional FEA basis functions will typically be only C^0 continuous. This continuity can be used to approximate higher order derivative PDEs. IGA has been used with high success for Kirchhoff–Love shell models, which involve 4^{th} order derivatives [103, 105]. This has been applied to a variety of problems, including wind turbines [19, 95, 83] and bio-prosthetic heart valves [101, 102].

The knot-vector structure of multi-variate B-splines can limit discretizations. Most CAD geometries employ multiple B-spline 'patches', thus any meaningful design-through-analysis workflow must provide the means to couple the bases defined upon multiple intersecting patches. This has been accomplished through penalty based enforcement [82], allowing for the analysis of large structures including wind turbines and aircraft[165, 164].

Implementation of IGA remains a significant challenge. Quadrature rules have been developed to tackle to the rational B-spline basis functions [92, 4], but the large support domains of each function hinder efficiency. Bèzier extraction [34, 148] was developed to address these issues, allowing for the rational B-spline function to be exactly represented as a linear combination of Bernstein polynomials. This was later generalized to Lagrange extraction [144], where

B-splines are instead represented as Lagrange polynomials.

The introduction of Lagrange extractions, also known as interpolation, allowed for novel implementations of IGA within existing finite element software. By replacing the difficult to integrate B-spline discretizations with Lagrange polynomials, classic quadrature methods can be employed. This was done using the open-source software package FEniCS in [99, 97] to create the package tIGAr and also by [159].

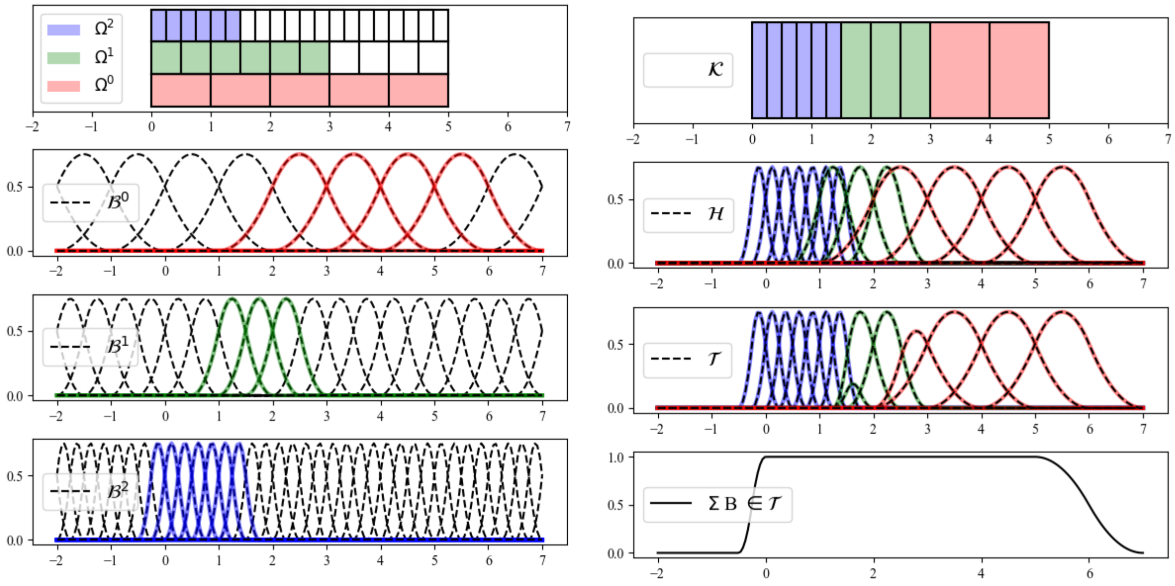
3.1.1 Hierarchical B-splines

As with classical finite element meshes, local refinement with IGA presents a unique set of challenges. Knot insertion, refining the one-dimensional knot vectors in the target region, leads to arbitrary increases in degrees of freedom in surrounding regions. T-splines [149] allow for T-shaped junctions in spline patches and were initially developed for the computer graphics community. T-splines were utilized for computational analysis in IGA [147, 111, 110], reducing system size while still allowing for local refinement. Similarly, hierarchically refined B-splines [141] allow for T-junctions and provide efficient data-organizing structures. Furthermore, truncated hierarchically refined B-splines [72, 73] retain the partition of unity property desirous for approximating PDEs.

Chapter 5 of this thesis will employ hierarchically refined B-splines (HB-splines) for local refinement. HB-splines are constructed using nested sequences of spline spaces created by repeated knot insertion. Following the algorithms presented in [72], an HB-spline basis begins with the construction of a sequence of r tensor-product spline spaces $\mathscr{V}^l$,

$$\mathscr{V}^0 \subset \mathscr{V}^1 \subset ... \subset \mathscr{V}^{r-1}, \tag{3.4}$$

each of which has an accompanying B-spline basis $\mathscr{B}^l = \{B^l_{\boldsymbol{i}}\}$, and tensor-product Cartesian mesh $\mathscr{K}^l$, where elements are denoted by K. A sequence of subdomains Ω^l are chosen, such



(a) A 1D illustration of a sequence of meshed subregions Ω^l and quadratic B-spline bases $\mathscr{B}^l$.

(b) The hierarchically refined mesh $\mathscr{K}$, the HB-spline basis $\mathscr{H}$, the THB-spline basis $\mathscr{T}$, and the verification of the partition of unity. The colors of the elements of $\mathscr{K}$ correspond to the refinement level.

Figure 3.1. Local refinement is applied through truncated hierarchically-refined B-splines (THB).

that Ω^{l-1} is a subregion of Ω^{l},

$$\Omega^0 \supseteq \ \Omega^1 \supseteq ... \supseteq \Omega^{r-1}, \tag{3.5}$$

and each Ω^l can be discretized with mesh elements $K \in \mathscr{K}^l$. Here r is the depth of refinement.

The HB-spline basis $\mathscr{H} := \mathscr{H}^{r-1}$ is constructed recursively by the algorithm:

$$\begin{cases} \mathscr{H}^0 & := \mathscr{B}^0, \\ \mathscr{H}^{l+1} & := \{B \in \mathscr{H}^l \mid \text{supp}(B) \not\subset \Omega^{l+1}\} \cup \\ & \quad \{B \in \mathscr{B}^{l+1} \mid \text{supp}(B) \subset \Omega^{l+1}\}, \\ & l \in \{0, ..., r-2\}. \end{cases} \tag{3.6}$$

In essence, each subsequent level's basis $\mathscr{H}^{l+1}$ is formed from the union of the set of basis functions from the previous level whose support is not in the new level's subdomain Ω^{l+1}, and the set of functions from the new basis $\mathscr{B}^{l+1}$ whose support is within Ω^{l+1}. This is illustrated with a 1D mesh in Figure 3.1.

The hierarchically refined basis $\mathscr{H}$ is associated with a hierarchically refined mesh $\mathscr{K}$, defined as

$$\mathscr{K} := \bigcup_{l=0}^{r-1} \{K \in \mathscr{K}^l \mid K \in \Omega^l \text{ and } K \notin \Omega^{l+1}\}. \tag{3.7}$$

While a useful tool for applying adaptive refinement to IGA, HB-spline bases violate the partition of unity (PU) property. To regain the this property, the hierarchically refined bases are truncated as in [72] and [73]. In addition to forming a partition of unity, truncation reduces the size of some basis functions' supports, thereby reducing the bandwidth of the resulting system of equations when compared to a non-truncated HB-spline basis.

A given multivariate basis function $B^l \in \mathscr{B}^l$ can be represented as a linear combination of the more refined functions of level $\mathscr{B}^{l+1}$:

$$B^l = \sum_{B^{l+1} \in \mathscr{B}^{l+1}} c^{B^{l+1}}\left(B^l\right) B^{l+1}, \tag{3.8}$$

where $c^{B^{l+1}}\left(B^l\right)$ are coefficients relating the coarse basis function B^l to the finer function B^{l+1}.

The truncation of B^l removes the contributions from $B^{l+1} \in \mathscr{B}^{l+1}$ with support contained within Ω^{l+1}, such that

$$\mathrm{trun}^{l+1}(B^l) = B^l - \sum_{\substack{B^{l+1} \in \mathscr{B}^{l+1}, \\ \mathrm{supp}(B^{l+1}) \subseteq \Omega^{l+1}}} c^{B^{l+1}}\left(B^l\right) B^{l+1}. \tag{3.9}$$

Using a similar algorithm to that given in Eq. (3.6), the truncated basis $\mathscr{T} := \mathscr{T}^{r-1}$ can be constructed by

$$\begin{cases} \mathscr{T}^0 & := \mathscr{B}^0, \\ \mathscr{T}^{l+1} & := \{\mathrm{trun}^{l+1}(B) \mid B \in \mathscr{T}^l,\ \mathrm{supp}(B) \not\subset \Omega_{l+1}\} \\ & \quad \cup \{B \in \mathscr{B}^{l+1} \mid \mathrm{supp}(B) \subset \Omega_{l+1}\}, \\ & l \in \{0, ..., r-2\}, \end{cases} \tag{3.10}$$

as illustrated in Figure 3.1b. This truncated basis forms a partition of unity, as is proven in Theorem 10 of [73].

3.2 The Reproducing Kernel Particle Method (RKPM)

RKPM is a meshfree method for solving PDEs. Instead of creating a mesh data structure, a domain $\Omega \subset \mathbb{R}^d$ is discretized by a set of NP nodes of spatial coordinates $\{\boldsymbol{x}_1, \boldsymbol{x}_2, ...\boldsymbol{x}_{NP}\}$. Here d denotes the spatial dimension. For ease of explanation, section work will deal only with the case $d = 2$, with notation $\boldsymbol{x} = [x, y]$. RKPM shape functions Ψ_I are associated with each node.

The RKPM approximation of a function u is then given as

$$u(\boldsymbol{x}) \sim u^h(\boldsymbol{x}) = \sum_I \Psi_I(\boldsymbol{x}) u_I \tag{3.11}$$

The shape functions can be expressed as

$$\Psi_I(\boldsymbol{x}) = \Phi_a(\boldsymbol{x} - \boldsymbol{x}_I) C(\boldsymbol{x}; \boldsymbol{x} - \boldsymbol{x}_I). \tag{3.12}$$

$\Phi_a(\boldsymbol{x})$ is the kernel function with compact support size a_I defined as

$$a_I = c_a h_I, \tag{3.13}$$

where c_a is the normalized support size and h_I is the nodal spacing associated with the node $\boldsymbol{x}_I$. The nodal spacing is determined by

$$h_I = \max\{||\boldsymbol{x}_I - \boldsymbol{x}_J||\}, \forall \boldsymbol{x}_J \ \in B_I \tag{3.14}$$

where B_I is the set of closest nodes to $\boldsymbol{x}_I$. For $d = 2$, B_I will contain four nodes. The kernel determines the local order of continuity. The shape of function support is also determined by the kernel support, and is typically either circular or rectangular.

The function $C(\boldsymbol{x}; \boldsymbol{x} - \boldsymbol{x}_I)$ is the correction function, and is used to impose the reproducing conditions for desired $\bar{\Omega}$ completeness. The reproducing condition states that

$$\sum_I \Psi_I(\boldsymbol{x}) x_I^i y_I^j = x^i y^j, \qquad 0 \leq i + j \leq n \tag{3.15}$$

where n is the order of reproducibility.

The correction function can be computed as

$$C(\boldsymbol{x};\boldsymbol{x}-\boldsymbol{x}_I) = \boldsymbol{H}^{\mathrm{T}}(0)\boldsymbol{M}^{-1}(\boldsymbol{x})\boldsymbol{H}(\boldsymbol{x}-\boldsymbol{x}_I). \tag{3.16}$$

$\boldsymbol{H}(\boldsymbol{x})$ is the basis vector

$$\boldsymbol{H}(\boldsymbol{x}) = [1,x,y,x^2,xy,y^2,...,x^n,x^{n-1}y,...,xy^{n-1},y^n], \tag{3.17}$$

$\boldsymbol{M}(\boldsymbol{x})$ is the moment matrix,

$$\boldsymbol{M}(\boldsymbol{x}) = \sum_{I\in G_{\boldsymbol{x}}} \boldsymbol{H}(\boldsymbol{x}-\boldsymbol{x}_I)\boldsymbol{H}^{\mathrm{T}}(\boldsymbol{x}-\boldsymbol{x}_I)\Phi_a(\boldsymbol{x}-\boldsymbol{x}_I), \tag{3.18}$$

where $G_{\boldsymbol{x}}$ is the set of all nodes with support covering the point $\boldsymbol{x}$. To avoid singular moment matrices, the support of each shape function (and thus kernel function) must be sufficiently large to provide adequate coverage to every evaluation point within the domain. Each point must be covered by at least n_p kernels, where

$$n_p = \begin{pmatrix} n+d \\ d \end{pmatrix}. \tag{3.19}$$

This is ensured by altering the normalized support size c_a.

Examples of RKPM shape functions are plotted in Figures 3.2 and 3.3, for both a uniform and non-uniform (perturbed) nodal distribution. Cubic B-spline kernels are employed, with support sizes such that each kernel covers at least three neighboring nodes. Both linear and quadratic reproducing conditions are applied, to compute the associated RKPM shape functions.

The evaluation of a given shape function at a point within the computational domain requires the inversion of the moment matrix. While this matrix is small relative to the linear systems typically dealt with in computational mechanics, the inversion must be performed at

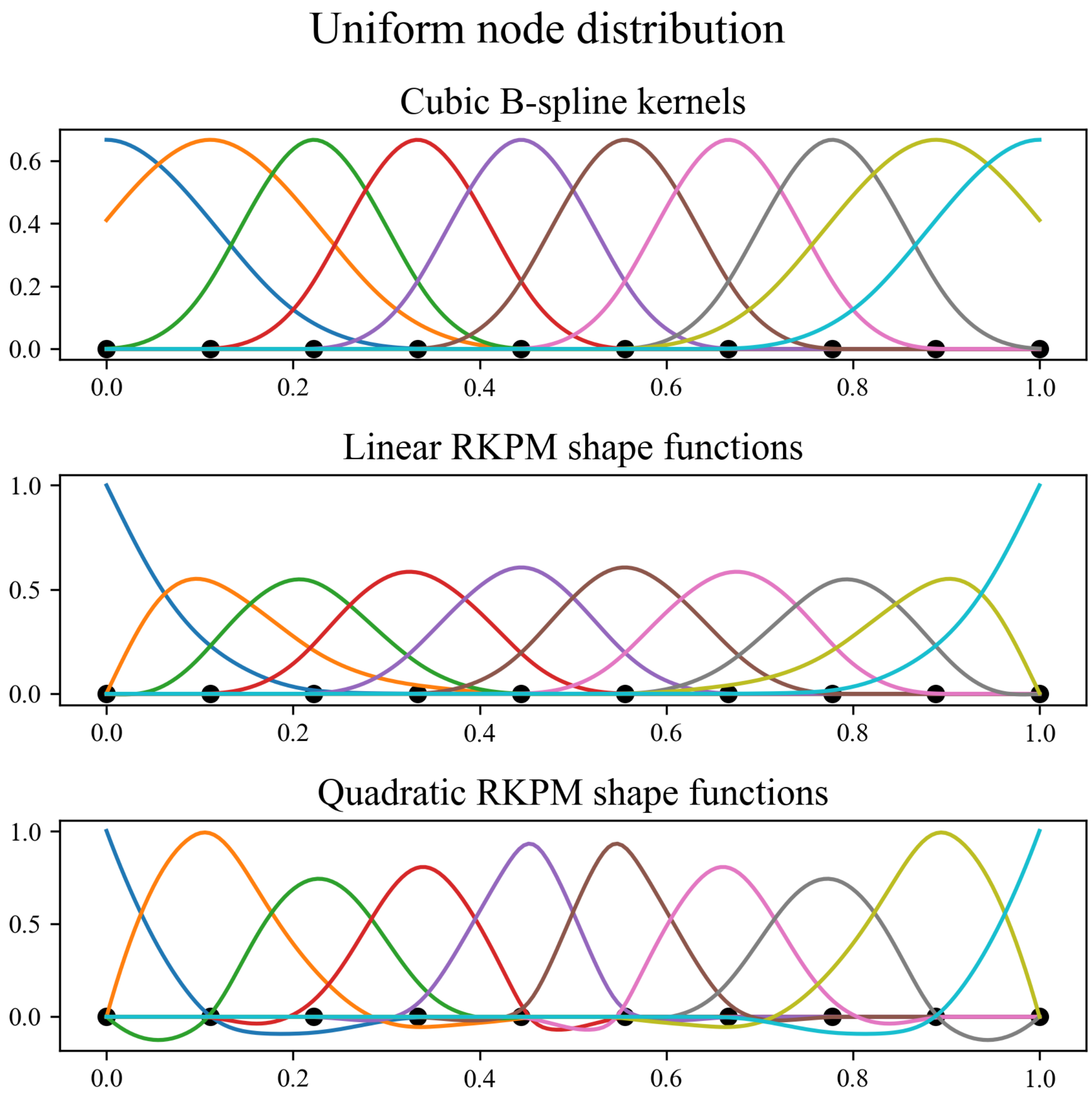


Figure 3.2. Cubic B-spline kernels (top), linear RKPM shape functions (middle) and quadratic RKPM shape functions (bottom) are plotted for a 1D uniform nodal distribution.

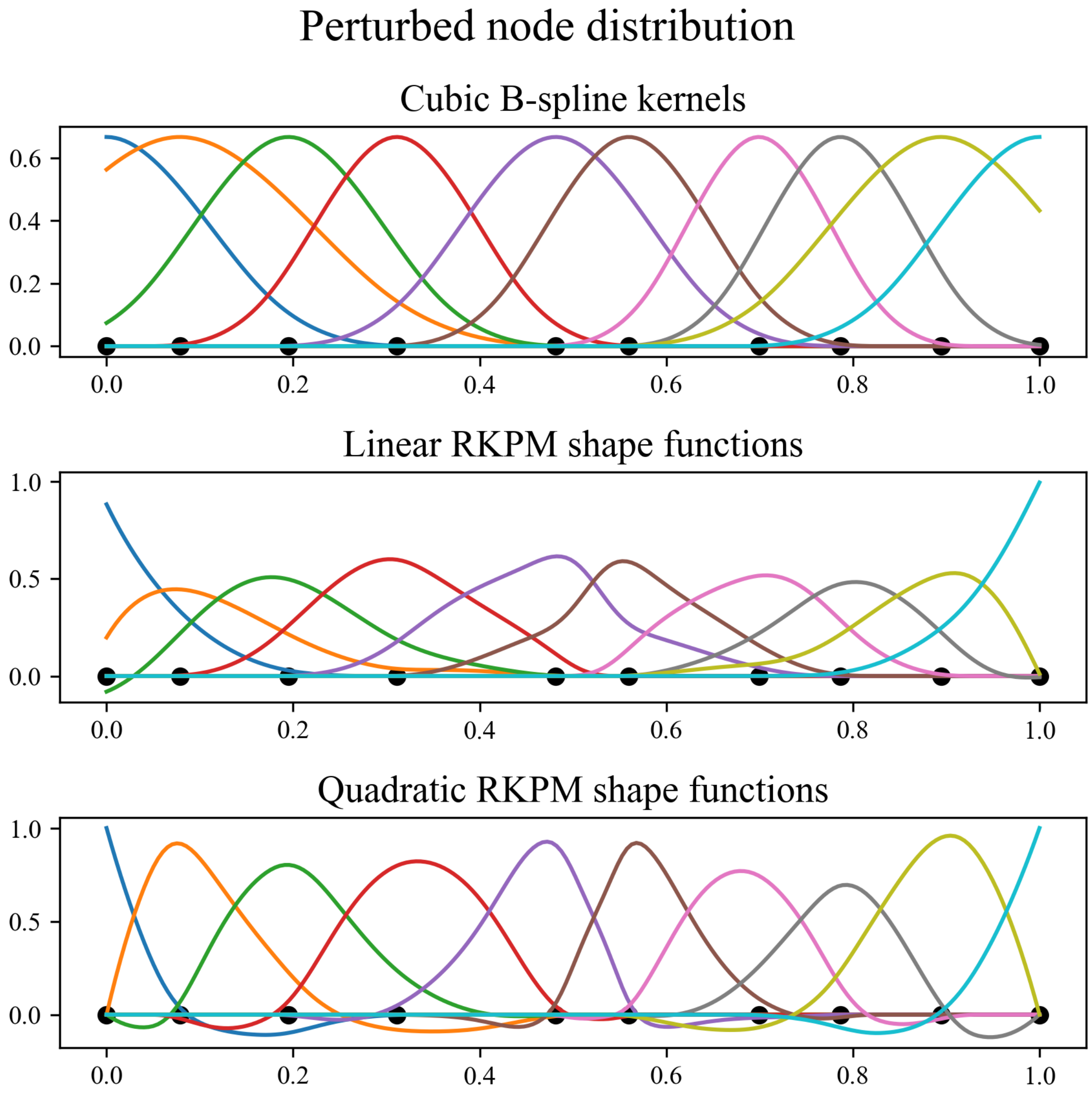


Figure 3.3. Cubic B-spline kernels (top), linear RKPM shape functions (middle) and quadratic RKPM shape functions (bottom) are plotted for a 1D non-uniform nodal distribution.

every evaluation point, for every shape function supporting that point.

The product rule is used to compute the derivatives of shape functions

$$\frac{\partial \Psi_I}{\partial x} = \frac{\partial \Phi_a}{\partial x} C + \Phi_a \frac{\partial C}{\partial x} \tag{3.20}$$

where computing the derivative of the correction function requires additional inversions of the derivatives of moment matrices, adding to this computational cost.

Acknowledgements

A portion of this chapter has been accepted for publication in Fromm, Jennifer E; Wunsch, Nils; Maute, Kurt; Evans, John A; and Chen, Jiun-Shyan; “Interpolation-based immersogeometric analysis methods for multi-material and multi-physics problems”, Journal of Computational Mechanics 2024. The dissertation author was the primary researcher and author of this paper.

A portion of this chapter is currently being prepared for submission in Fromm, Jennifer E; Evans, John A; and Chen, Jiun-Shyan; “Interpolation-based reproducing kernel particle methods” 2024. The dissertation author is the primary researcher and author of this work.

Chapter 4

Interpolation-based immersed finite element analysis

This chapter introduces the concept of approximate interpolation through the application of interpolation-based immersed boundary methods. Section 4.1 introduces the concept with a scalar elliptic model PDE and provides a sketch of how a basic error analysis of the method might proceed. Section 4.2 then describes how to implement interpolation-based immersed analysis for arbitrary PDE systems in a way that efficiently reuses existing FE software. The accuracy of the method is then explored for a variety of PDEs through numerical experiments in Section 4.3 and applied to analysis of shell structures modeled geometrically by trimmed spline surfaces in Section 4.4.

The interpolation-based immersed method employ a foreground integration mesh in addition to a background discretization. The use of multiple meshes with different function spaces can lead to ambiguity in terms like “element size” and “basis function”, so this work is careful to distinguish between background, foreground, and interpolated entities through precise terminology and notation. Notations are defined more comprehensively as they arise in context later, but are also summarized here for convenient reference:

- h : background mesh element size
- η : foreground mesh element size

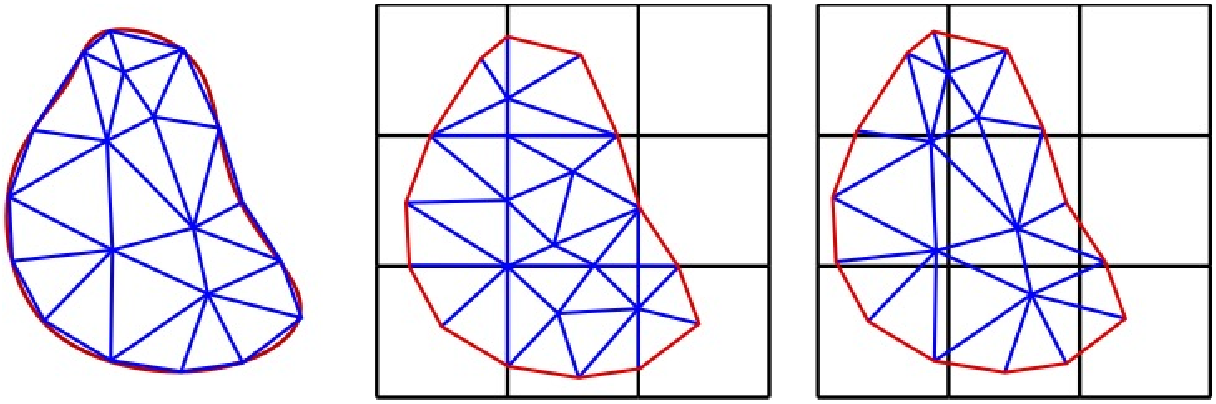


Figure 4.1. (a) Domain discretized using a body-fitted mesh; (b) Domain discretized using a tensor-product background mesh with a background-fitted foreground mesh; (c) Domain discretized using a tensor-grid background mesh with a background-unfitted foreground mesh.

- B_i : background basis functions
- $\widehat{B}_i$: interpolated background basis functions
- N_i : foreground basis functions
- n : polynomial degree of the background FE space
- k : polynomial degree of the foreground FE space
- $\widehat{k}$: limiting degree $\min\{n,k\}$ of the interpolated background FE space
- $\mathscr{P}$: interpolation into the foreground FE space
- $\mathscr{Q}$: interpolation into the background FE space
- $\widehat{\mathscr{Q}}$: interpolation $\mathscr{P} \circ \mathscr{Q}$ into the interpolated background FE space

Terms to distinguish the mesh immersions shown in Figures 4.1 (b) and 4.1 (c) are as follows:

- **Background-fitted foreground mesh:** A foreground mesh which is fitted not just to the domain geometry's boundary but also to interior facets of the background mesh in the sense that the closure of the intersection of a background element with the PDE domain is the closure of a union of foreground elements, as shown in Figure 4.1 (b).

- **Background-unfitted foreground mesh:** A foreground mesh such as the one shown in Figure 4.1 (c) whose elements are fitted to the domain geometry but not to interior facets of the background mesh.

The terms "fitted" and "unfitted" may be used without specifying the prefix "boundary-" or "background-", but only when the omitted prefix should be clear from context.

4.1 Immersed finite element analysis for a scalar elliptic model problem

This section defines the proposed interpolation-based immersed method for a model scalar elliptic problem Following sections will apply the method to several more complicated problems. The model PDE is defined in Section 4.1.1, then a variational formulation suitable for immersed methods is defined in Section 4.1.2. Section 4.1.3 reviews the formulation and challenges of quadrature-based immersed methods, and Section 4.1.4 defines the construction of a function space used for interpolation-based immersed analysis. Section 4.1.5 outlines an *a priori* error estimate for the case of a background-fitted foreground mesh.

4.1.1 The model problem

The chooses model PDE is the Poisson equation with Dirichlet boundary conditions. Let $\Omega \subset \mathbb{R}^d$ be a spatial domain of interest. For simplicity of exposition, Ω is assumed to be polyhedral and can be decomposed into affine simplicial elements exactly. The strong form of the problem is as follows: Find $u : \Omega \to \mathbb{R}$ such that

$$\begin{aligned} -\Delta u &= f \quad \text{in } \Omega\,, \\ u &= g \quad \text{on } \partial\Omega\,, \end{aligned} \tag{4.1}$$

where $f : \Omega \to \mathbb{R}$ is a given source term and $g : \partial\Omega \to \mathbb{R}$ is given Dirichlet boundary data. The weak form used as a starting point for formulating numerical methods is: Find $u \in \mathscr{V}_g$ such that

$$\int_\Omega \nabla u \cdot \nabla v \, d\Omega = \int_\Omega f v \, d\Omega \quad \forall v \in \mathscr{V}_0 \, , \tag{4.2}$$

where

$$\mathscr{V}_g := \left\{ v \in H^1(\Omega) : v|_{\partial\Omega} = g \right\} , \tag{4.3}$$

$$\mathscr{V}_0 := \left\{ v \in H^1(\Omega) : v|_{\partial\Omega} = 0 \right\} , \tag{4.4}$$

in which restriction is understood in the sense of trace, and f and g are in at least $L^2(\Omega)$ and $H^{1/2}(\partial\Omega)$, respectively.

4.1.2 Discretization of the model problem

The most common discretization of the weak problem (4.2) is a conforming one employing finite dimensional subsets of $\mathscr{V}_g$ and $\mathscr{V}_0$. However, the construction of these subsets typically relies on a boundary-fitted mesh to conform to the boundary conditions. Many immersed methods therefore use nonconforming discretizations which enforce the Dirichlet boundary condition in a weak sense. In particular, this work employs Nitsche's method [127] for weak enforcement of Dirichlet boundary conditions, leading to the discrete problem:

Find $u^h \in \mathscr{V}^h$ such that, $\forall v^h \in \mathscr{V}^h$,

$$\begin{aligned}\int_\Omega \nabla u^h \cdot \nabla v^h d\Omega - \int_{\partial\Omega} \left(\nabla u^h \cdot \boldsymbol{n} \right) v^h d\Gamma \mp \int_{\partial\Omega} \left(\nabla v^h \cdot \boldsymbol{n} \right) u^h d\Gamma + \int_{\partial\Omega} \frac{C_{\text{pen}}}{h} u^h v^h d\Gamma \\ = \int_\Omega f v^h d\Omega \mp \int_{\partial\Omega} \left(\nabla v^h \cdot \boldsymbol{n} \right) g d\Gamma + \int_{\partial\Omega} \frac{C_{\text{pen}}}{h} g v^h d\Gamma \, ,\end{aligned} \tag{4.5}$$

where the discrete space $\mathscr{V}^h$ is a finite-dimensional subspace of $H^1(\Omega)$ with no extra restrictions on $\partial\Omega$, h is a length scale indicating the refinement level of the discretizations (e.g., the element diameter in a standard boundary-fitted discretization), $\boldsymbol{n}$ is the outward-pointing unit normal vector to $\partial\Omega$, and $C_{\text{pen}} \geq 0$ is a dimensionless constant. The "$\mp$" in (4.5) must be selected consistently on both sides of the equation, and toggles between the symmetric ("$-$") and non-symmetric ("$+$") variants of the Nitsche method. The original symmetric variant proposed by [127] remains well-posed for C_{pen} sufficiently large, with the lower bound emanating from trace-inverse estimates (but often chosen on an *ad hoc* basis in practice). The non-symmetric variant is well-posed for C_{pen} arbitrarily small (or zero), which has led some authors to advocate its use in immersed discretizations [142] to sidestep technical concerns over the appropriate values for C_{pen} and h. The non-symmetric variant retains optimal accuracy in the H^1 norm, which is often more relevant to applications like stress analysis. However, the symmetric variant is optimal in both H^1 and L^2 norms, and is currently more widely used in practice.

4.1.3 Quadrature-based immersed discretization

In quadrature-based immersed methods the discrete space $\mathscr{V}^h$ is defined by simply restricting functions in an FE (or isogeometric) space defined on a background mesh to the PDE domain Ω:

$$\mathscr{V}^h = \operatorname{span}\{B_i|_\Omega\}_{i=1}^n \ , \tag{4.6}$$

where $\{B_i\}_{i=1}^n$ are the basis functions of the background-mesh FE space whose supports intersect Ω. The principal challenge in implementing such a method is to accurately compute integrals of the forms

$$\int_\Omega (\cdots)\, d\Omega = \sum_{E=1}^{n_{\text{el}}} \int_{\Omega_E \cap \Omega} (\cdots)\, d\Omega \quad \text{and} \quad \int_{\partial\Omega} (\cdots)\, d\Gamma = \sum_{E=1}^{n_{\text{el}}} \int_{\partial\Omega \cap \overline{\Omega_E}} (\cdots)\, d\Gamma \ , \tag{4.7}$$

where Ω_E is the E^{th} element of the background mesh and n_{el} is the total number of background mesh elements. One can use standard Gaussian quadrature on Ω_E when $\Omega_E \subset \Omega$ or on boundary parts of Ω_E when the background mesh is fitted to $\partial\Omega$. Various schemes have been proposed for integrating nontrivial intersections, but we focus in this paper on those which decompose $\Omega_E \cap \Omega$ into a collection of quadrature elements, as shown in Figure 4.1 (b), and we refer to the mesh formed by these quadrature elements as a background-fitted foreground mesh. With such a mesh, one can then decompose integrals like

$$\int_{\Omega}(\cdots)\,d\Omega = \sum_{e=1}^{\nu_{\text{el}}}\int_{\omega_e}(\cdots)\,d\Omega \quad \text{and} \quad \int_{\partial\Omega}(\cdots)\,d\Gamma = \sum_{e=1}^{\nu_{\text{el}}}\int_{\partial\Omega\cap\partial\omega_e}(\cdots)\,d\Gamma\,, \tag{4.8}$$

where $\{\omega_e\}_{e=1}^{\nu_{\text{el}}}$ are the elements of the foreground mesh and standard Gaussian quadrature rules can be accurately used on each one of its boundary parts.

While mathematically-appealing, this presents a major challenge from a software development standpoint as now the mesh over which quadrature rules are defined is a separate data structure from the mesh on which the basis functions are constructed. Implementation of such methods therefore requires either custom software to be written from scratch or complicated invasive modifications to be made in mature FE or IGA software designed for standard boundary-fitted analysis. The primary motivation of the present work is to circumvent these difficulties while retaining the high-order accuracy of quadrature-based immersed methods.

4.1.4 Interpolation-based immersed discretization

The main idea of the present work is to use a discrete space that consists of interpolations of background-mesh basis functions on a Lagrange FE space defined over the foreground mesh:

$$\mathscr{V}^h = \operatorname{span}\left\{\widehat{B}_i\right\}_{i=1}^{n}\,, \tag{4.9}$$

where

$$\widehat{B}_i = \mathscr{P}B_i := \sum_{j=1}^{\nu} B_i(\boldsymbol{x}_j)N_j\,, \tag{4.10}$$

in which $\mathscr{P}$ is the operator that interpolates a function into the Lagrange FE space with basis $\{N_j\}_{j=1}^{\nu}$ and $\boldsymbol{x}_j$ is the nodal point associated with foreground Lagrange basis function N_j, i.e.,

$$N_i(\boldsymbol{x}_j) = \delta_{ij}\,. \tag{4.11}$$

The space $\mathscr{V}^h$ is a subset of the Lagrange FE space defined on the foreground mesh, so its functions can be accurately integrated using Gaussian quadrature rules on each foreground element. However, unlike in quadrature-based immersed methods, this remains true even with a background-unfitted foreground mesh, like the one shown in Figure 4.1 (c), where, for an arbitrary background element Ω_E, $\overline{\Omega_E \cap \Omega}$ is not necessarily equal to the closure of a union of foreground elements.

For either background-fitted or background-unfitted foreground meshes, the basis $\{\widehat{B}_i\}_{i=1}^n$ exhibits the following properties:

1. If $\{B_i\}_{i=1}^n$ forms a partition of unity, then $\{\widehat{B}_i\}_{i=1}^n$ does as well.

2. If the background function space spans polynomials of degree k and the foreground function space spans polynomials of degree κ, then $\{\widehat{B}_i\}_{i=1}^n$ spans polynomials of degree $\widehat{k} := \min\{k, \kappa\}$.

3. If both $\{B_i\}_{i=1}^n$ and $\{N_i\}_{i=1}^{\nu}$ have local support, then $\{\widehat{B}_i\}_{i=1}^n$ also has local support.

4. The continuity of $\{\widehat{B}_i\}_{i=1}^n$ is at least that of $\{N_i\}_{i=1}^{\nu}$. Note that this may be lower than the continuity of $\{B_i\}_{i=1}^n$.

Moreover, for the specific case of a background-fitted foreground mesh, the interpolated background basis $\{\widehat{B}_i\}_{i=1}^n$ is equivalent to the original background basis $\{B_i\}_{i=1}^n$ for a sufficiently high foreground polynomial degree κ. If the background function space consists of piecewise polynomials of degree k defined over a simplicial background mesh, the original background basis is recovered with a Lagrange interpolation of degree $\kappa \geq k$ onto a simplicial background-fitted foreground mesh. If the background function space consists of piecewise tensor-product polynomials of degree k defined over a tensor-product background mesh, the original background basis is recovered with a Lagrange interpolation of degree $\kappa \geq k^d$ onto a simplicial background-fitted foreground mesh.

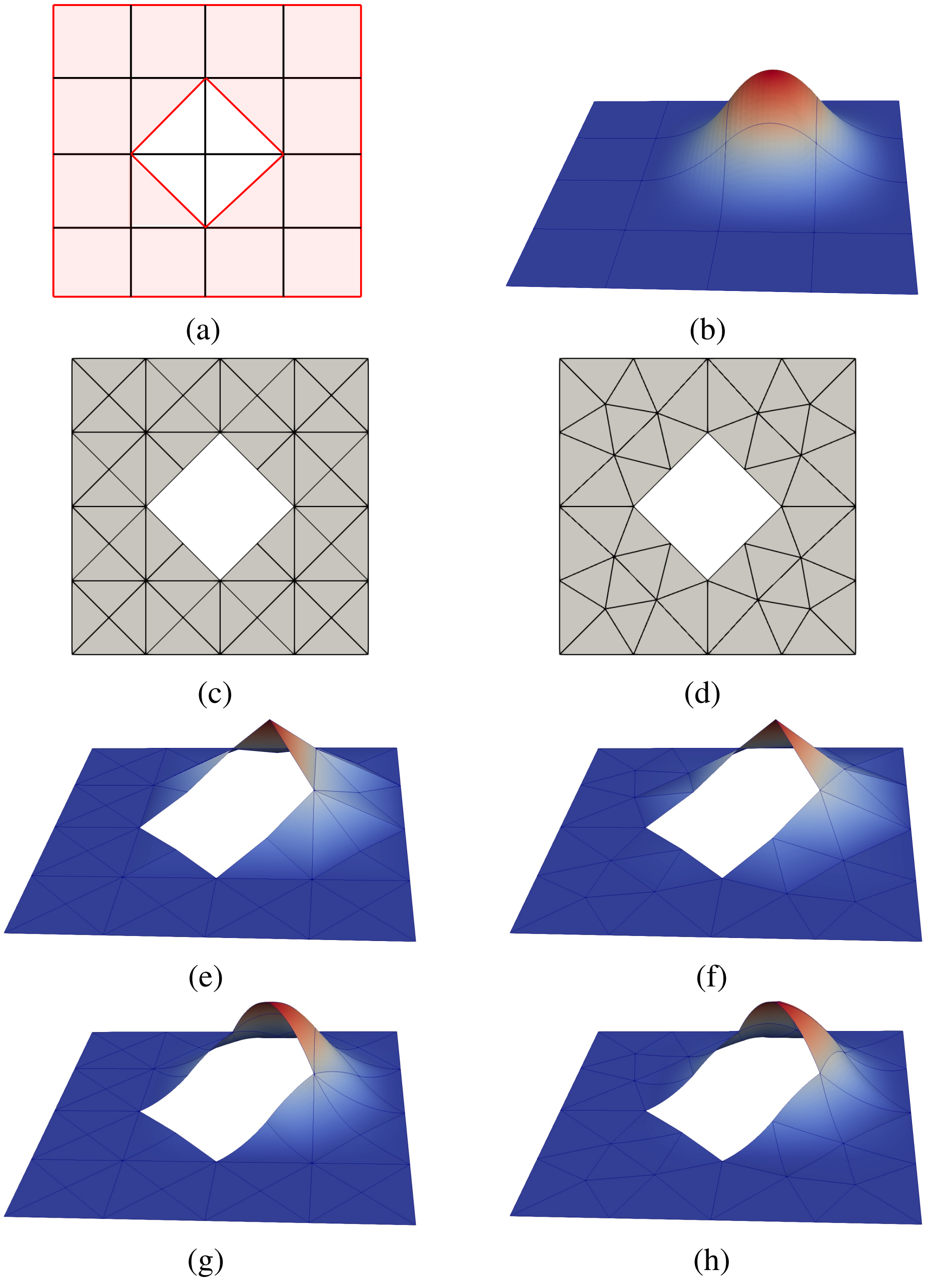


Figure 4.2. (a) Geometric domain (red) and background mesh (black). (b) Background basis function. (c) Background-fitted foreground mesh. (d) Background-unfitted foreground mesh. (e) Linear Lagrange interpolation of background basis function on background-fitted mesh. (f) Linear Lagrange interpolation of background basis function on background-unfitted mesh. (g) Quadratic Lagrange interpolation of background basis function on background-fitted mesh. (h) Quadratic Lagrange interpolation of background basis function on background-unfitted mesh.

To better convey the concept of interpolated background basis function, several interpolations of an example background basis function are displayed in Figure 4.2. Figure 4.2 (a) shows the geometric domain and background mesh in red and black respectively, while Figure 4.2 (b) shows the example background basis function. The background basis function is a bi-quadratic ($k = 2$) B-spline basis function of maximal C^{k-1} continuity defined over the background mesh. Figure 4.2 (c) and Figure 4.2 (d) show representative background-fitted and background-unfitted foreground meshes. Figure 4.2 (e) and Figure 4.2 (f) show linear ($\kappa = 1$) Lagrange interpolations of the example background basis function onto the background-fitted and background-unfitted foreground meshes. These two interpolated background basis functions share the same shape as the background basis function, but they are both inaccurate approximations of the background basis function. Alternatively, Figure 4.2 (g) and Figure 4.2 (h) show quadratic ($\kappa = 2$) Lagrange interpolations of the background basis function onto the background-fitted and background-unfitted foreground meshes. These two interpolated background basis functions are much more accurate approximations of the background basis function than the interpolated background basis functions displayed in Figure 4.2 (e) and Figure 4.2 (f).

4.1.5 Best approximation estimates for the background-fitted case

Having presented the new paradigm for immersed finite element and isogeometric analysis, this work now derives the H^1-norm best approximation error

$$\inf_{v^h \in \operatorname{span}\left\{\widehat{B}_i\right\}_{i=1}^{n}} \left\| u - v^h \right\|_{H^1(\Omega)} \tag{4.12}$$

associated with the interpolated background FE space where u is the exact solution of our model problem. Best approximation estimates of the form given by (4.12) can be combined with coercivity and continuity results to arrive at *a priori* error estimates for a given immersed finite element method (as done, e.g., for quadrature-based CutFEM with ghost penalty stabilization in [38]). In this mathematical analysis, two assumptions regarding the foreground mesh must be made:

Assumption 1: *The foreground mesh is background-fitted.*

Assumption 2: *The foreground mesh is free of hanging nodes.*

Assumption 1 ensures that the background basis functions are smooth within every element of the foreground mesh, while Assumption 2 ensures that the interpolated background FE space is H^1-conforming and thus the H^1-norm best approximation error is well-defined. Note that an immediate consequence of Assumption 1 is that the foreground mesh size η is less than or equal to the background mesh size h. Two further assumptions must be made regarding the exact solution u and the background FE space:

Assumption 3: *The exact solution u lies in the space $H^{\widehat{k}+1}(\Omega)$.*

Assumption 4: *There exists an interpolation operator $\mathscr{Q} : H^1(\Omega) \to \operatorname{span}\{B_i\}_{i=1}^n$ onto the background FE space such that*

$$\|v - \mathscr{Q}v\|_{H^1(\Omega)} \leq C_1 h^{\widehat{k}} \|v\|_{H^{\widehat{k}+1}(\Omega)} \tag{4.13}$$

and

$$\sum_{e=1}^{\nu_{el}} \|\mathscr{Q}v\|^2_{H^{\widehat{k}+1}(\omega^e)} \leq C_2 \|v\|^2_{H^{\widehat{k}+1}(\Omega)} \tag{4.14}$$

for all $v \in H^{\widehat{k}+1}(\Omega)$ where $C_1 > 0$ and $C_2 > 0$ depend on the background FE space polynomial degree k and the shape regularity of the background mesh but not the background mesh element size h.

Assumption 3 holds provided the problem geometry and parameters are sufficiently smooth, while Assumption 4 holds when the background FE space consists of classical finite element functions or smooth spline functions provided there exists a continuous extension operator $E : H^{k+1}(\Omega) \to H^{k+1}(\mathbb{R}^d)$ (cf. [38, (9)]).

Now define $\widetilde{\mathscr{Q}} = \mathscr{P} \circ \mathscr{Q}$. The H^1-norm best approximation error associated with the interpolated background FE space by the H^1-norm error associated with the interpolation operator $\widetilde{\mathscr{Q}}$ can be bound by:

$$\inf_{v^h \in \operatorname{span}\left\{\widehat{B}_i\right\}_{i=1}^{n}} \left\| u - v^h \right\|_{H^1(\Omega)} \le \left\| u - \widetilde{\mathscr{Q}} u \right\|_{H^1(\Omega)}. \tag{4.15}$$

By the triangle inequality,

$$\left\| u - \widetilde{\mathscr{Q}} u \right\|_{H^1(\Omega)} \le \| u - \mathscr{Q} u \|_{H^1(\Omega)} + \| \mathscr{Q} u - \mathscr{P} \mathscr{Q} u \|_{H^1(\Omega)}. \tag{4.16}$$

Invoking Assumption 3, the first term on the right hand side of the above equation can be immediately bound:

$$\| u - \mathscr{Q} u \|_{H^1(\Omega)} \le C_1 h^{\widehat{k}} \| u \|_{H^{\widehat{k}+1}(\Omega)}. \tag{4.17}$$

To bound the second term, first recognize that

$$\| \mathscr{Q} u - \mathscr{P} \mathscr{Q} u \|^2_{H^1(\Omega)} = \sum_{e=1}^{\nu_{\text{el}}} \| \mathscr{Q} u - \mathscr{P} \mathscr{Q} u \|^2_{H^1(\omega^e)}. \tag{4.18}$$

By Assumption 1, $\mathscr{Q} u|_{\omega^e}$ is smooth for every foreground mesh element ω^e. Thus

$$\| \mathscr{Q} u - \mathscr{P} \mathscr{Q} u \|^2_{H^1(\omega^e)} = C_3 \eta^{2\widehat{k}} \| \mathscr{Q} u \|^2_{H^{\widehat{k}+1}(\omega^e)} \tag{4.19}$$

for every foreground element ω^e where C_3 is a constant that depends on the foreground FE space polynomial degree κ and the shape regularity of the foreground mesh but not the foreground mesh element size η. Then combine (4.18), (4.19), Assumption 1, and Assumption 4 to arrive at

$$\|\mathscr{Q}u - \mathscr{P}\mathscr{Q}u\|_{H^1(\Omega)} = \sqrt{C_2C_3}h^{\widehat{k}}\|u\|_{H^{\widehat{k}+1}(\Omega)}. \tag{4.20}$$

Combine (4.15), (4.16), (4.17), and (4.18) to arrive at the following estimate for the H^1-norm best approximation error associated with the interpolated background FE space:

$$\inf_{v^h \in \operatorname{span}\left\{\widehat{B}_i\right\}_{i=1}^{n}} \left\|u - v^h\right\|_{H^1(\Omega)} \leq C_{\text{interp}} h^{\widehat{k}} \|u\|_{H^{\widehat{k}+1}(\Omega)} \tag{4.21}$$

where $C_{\text{interp}} = C_1 + \sqrt{C_2C_3}$. It is evident that the above estimate is optimal with respect to both the background mesh size h and the limiting degree $\hat{k}$ of the interpolated background space. Note that the convergence rate does not improve by decreasing the foreground mesh size η faster than the background mesh size h or increasing the foreground polynomial degree k above that of the background polynomial degree n.[1]

In the above analysis, the rather stringent assumption that the foreground mesh be background-fitted was made. However, numerical experiments suggest that best approximation estimates of the form given by (4.21) also hold when the foreground mesh is background-unfitted. This is perhaps not too unexpected as the interpolated background FE space is complete up to polynomial degree $\hat{n}$ even when the foreground mesh is background-unfitted, but the preceding analysis does not extend to this case as it require that the background basis functions be smooth over each element of the foreground mesh.

The analysis also made the assumption that the foreground mesh be free of hanging nodes. This was to ensure that the interpolated background FE space was H^1-conforming. That being said, later numerical experiments suggest the method may not need an interpolated background FE space that is H^1-conforming provided that the original background FE space is H^1-conforming. In particular, this method is able to attain optimal convergence rates for the

[1] If there is geometry error in the foreground mesh, then there can be some benefit to over-refining η near boundaries (cf. Section 4.3.3), but geometry error is outside the scope of the analysis in this section.

biharmonic problem using an interpolated background FE space that is only H^1-conforming when the original background FE space is H^2-conforming. For the analysis of non-conforming methods, note that a similar interpolation estimate to (4.21) could be derived element-wise, using a weakened version of Assumption 2, where the mesh only needs to be free of hanging nodes within each background element. However, this work is focused on the globally $H^1(\Omega)$-conforming case for a more direct analogy to prior convergence analysis of quadrature-based immersed methods.

Finally, the best approximation estimate given by (4.21) depends on the shape regularity of the foreground mesh through the interpolation constant C_{interp}, but later numerical experiments suggest that high quality results are attained with interpolation-based immersed finite element and isogeometric analysis even when the foreground mesh is of poor quality.

4.2 Reusing FE software for interpolation-based immersed methods

As each basis function $\widehat{B}_i$ of $\mathscr{V}^h$ in the proposed method is a linear combination of basis functions defined on the foreground mesh used for element-by-element assembly, one may implement interpolation-based immersed methods using existing FE software in a similar manner as one would implement IGA via Lagrange extraction [144]. The cited paper emphasizes an implementation based on modifying element-level shape function routines which could also be followed for immersed methods. However, the method outlined here is based on global linear algebra operations applied to the full, assembled system of equations, similar to what is done in [99, 159] for IGA. The latter approach requires fewer modifications to the existing FE code and is referred to as “non-invasive” by [159].

4.2.1 Non-invasive implementation using extraction matrices

Consider a scalar variational problem of the form: Find $u^h \in \mathscr{V}^h$ such that, $\forall v^h \in \mathscr{V}^h$,

$$a\left(u^h, v^h\right) = L\left(v^h\right) \ , \tag{4.22}$$

where a and L are a bilinear and linear form, i.e., the left- and right-hand sides of the Nitsche formulation (4.5) for the Poisson model problem. $\mathscr{V}^h = \text{span}\{\widehat{B}_i\}$ is the solution space for the interpolation-based immersed method. The state variable field is discretized by the interpolated basis functions as

$$u^h = \sum_{i=1}^{n} d_i \widehat{B}_i \ , \tag{4.23}$$

where $\{d_i\}_{i=1}^n$ are unknown coefficients. The linear system

$$\mathbf{Kd} = \mathbf{F} \ , \tag{4.24}$$

must be solved, where the global stiffness matrix and forcing vector are

$$K_{ij} = a\left(\widehat{B}_j, \widehat{B}_i\right) \quad \text{and} \quad F_i = L\left(\widehat{B}_i\right) \ . \tag{4.25}$$

In the interpolation-based immersed method,

$$\widehat{B}_j = \sum_{i=1}^{v} M_{ij} N_i \quad \text{with} \quad M_{ij} = B_j(\boldsymbol{x}_i) \ . \tag{4.26}$$

M_{ij} projects the background functions onto the space of the foreground elements and can be obtained by evaluating the background basis B_j at the nodal points of the foreground mesh $\boldsymbol{x}_i$. Using the (bi-) linearity of a and L, re-write (4.25) as

$$K_{ij} = \sum_{k,\ell} M_{ki} a(N_\ell, N_k) M_{\ell j} \quad \text{and} \quad F_i = \sum_{k} M_{ki} L(N_k) \ . \tag{4.27}$$

In matrix form, this reads as

$$\mathbf{K} = \mathbf{M}^T\mathbf{A}\mathbf{M} \quad \text{and} \quad \mathbf{F} = \mathbf{M}^T\mathbf{B} \,, \tag{4.28}$$

where

$$A_{ij} = a(N_j, N_i) \quad , \quad B_i = L(N_i) \,, \tag{4.29}$$

M will be referred to as the “extraction matrix”. With some additional straightforward bookkeeping, this method extends to linear PDE systems with multiple scalar solution fields, as presented for extraction-based IGA in [99], and detailed in Chapter 5. It can also be applied to each step of Newton iteration to solve a nonlinear problem, where a is the Gateaux derivative of a residual form $-L$.

Crucially, the matrix **A** and vector **B** can be assembled using a standard boundary-fitted FE code in a loop over elements in the foreground mesh. Thus, given a foreground mesh and an extraction matrix **M**, one can directly reuse an existing FE code for interpolation-based immersed analysis without making any modifications to the assembly procedures. The post processing features of existing FE codes can also be reused by leveraging a foreground FE representation of the solution:

$$u^h = \sum_{j=1}^{\nu} c_j N_j \,, \tag{4.30}$$

where the foreground FE coefficients $\{c_i\}$ are obtained from the solution **d** of (4.24) by

$$\mathbf{c} = \mathbf{M}\mathbf{d} \,. \tag{4.31}$$

4.2.2 Example implementation using FEniCS

For the numerical results in this paper, the method outlined in Section 4.2.1 is applied to the existing FE automation toolchain FEniCS [114]. FEniCS allows arbitrary linear and bilinear forms to be specified over a variety of classical FE function spaces using the Python-based

Unified Form Language (UFL) [1]. UFL forms are compiled into efficient element-level routines [107] which are used by the FE library DOLFIN [114] to assemble matrices and vectors. In the notation of Section 4.2.1, UFL would be used to specify the forms a and L, and DOLFIN would be used to assemble the matrix $\mathbf{A}$ and vector $\mathbf{B}$, applying the compiled element-level routines to elements $\{\omega_e\}_{i=1}^{\nu_{\mathrm{el}}}$ of the foreground mesh. DOLFIN's application–programmer interface (API) provides access to the linear algebra objects $\mathbf{A}$ and $\mathbf{B}$ as data structures in the parallel linear algebra library PETSc [12, 13, 14], which is used to form $\mathbf{K}$ and $\mathbf{F}$ via (4.28), solve the linear system (4.24) for $\mathbf{d}$, and compute the foreground FE coefficients $\mathbf{c}$ via (4.31). DOLFIN's output and visualization routines can then be applied to this foreground FE representation of the solution.

This non-invasive reuse of FEniCS provides great flexibility for implementing formulations of a variety of different PDE systems as we illustrate in the sequel. What remains is to generate the foreground- and background meshes alongside the extraction operators $\mathbf{M}$. The tools and methods used to do so are outlined in the following subsection.

4.2.3 Mesh generation

This section describes the techniques used in the numerical examples of Sections 4.3 and 4.4 to generate both background-fitted and -unfitted foreground meshes.

Background-fitted foreground meshes

To generate background-fitted foreground meshes and their extraction matrices $\mathbf{M}$, the XFEM Tool Kit of the research code presented in [129] was used. Within this toolkit, the foreground elements are created by repeated subdivision of a structured background grid. First, each background cell is subdivided into a set of triangular or tetrahedral elements as is shown in Figure 4.3 (b). Then, all edges of the elements are checked for intersections with the geometry. Using a set of subdivision templates, elements with intersected edges are then again subdivided into straight-edged triangular and tetrahedral elements such that new element vertices are created

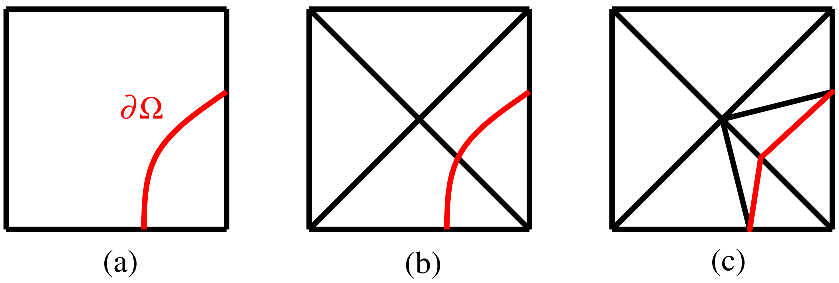


Figure 4.3. Generation of background fitted foreground meshes. (a) Single background element with description of the boundary shown by a red curve; (b) regular subdivision of the background element into triangles; (c) subdivision of triangles based on intersections of the interface with edges; interface approximated by red lines.

at the intersection locations. The new element facets (i.e. edges in 2D and faces in 3D) are a first-order geometric approximation of the domain boundary as shown in Figure 4.3 (c). The XFEM tool kit constructs the extraction matrices $\mathbf{M}$ such that basis functions are enriched following a Heaviside enrichment strategy [129]. Enrichment is needed when the support of a basis contains two or more geometrically disconnected domains. Note that for the numerical examples studied in this chapter, enrichment is not required to ensure optimal convergence. Enrichment will be further investigated in chapter 5

While the procedure outlined enables the robust background-fitted triangulation of the material domain, it may lead to elements of arbitrarily small size and arbitrarily large aspect ratio. As an example, the 2D foreground meshes used in Subsection 4.4.2 contain elements with aspect ratios as high as ~ 300 and elements with volumes as small as $\sim 10^{-4}$ of the biggest elements. The poor mesh quality is exacerbated in 3D. The 3D foreground meshes used for the example in Subsection 4.3.1 contain elements that differ by a factor of $\sim 10^{27}$ in volume, and aspect ratios as high as $\sim 10^{29}$. An example for the poor elements resulting from this mesh generation process in 3D is shown in Figure 4.4.

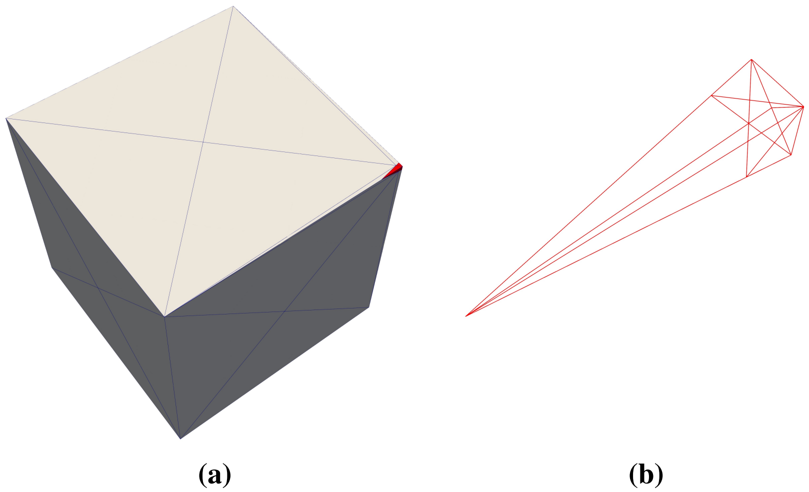


Figure 4.4. (a) A tessellation of a hexahedral background element from the 3D mesh shown in Figure 4.5, highlighting the sliver element in the rightmost corner. (b) Only the red section consisting of small tetrahedral elements intersects the PDE domain Ω.

Background-unfitted foreground meshes

Mesh cutting tools, like the XFEM tool kit discussed in Section 4.2.3, generate background-fitted meshes and do not require an additional mesh generator to create foreground meshes. However, background-unfitted analysis allows substantially greater flexibility in the generation of foreground meshes; essentially any existing mesh generator can be used, with the only necessary information from the background mesh being an approximate element size. This work employs two different approaches within the following examples, with the particular details largely motivated by convenience. The background-unfitted foreground meshes used in Section 4.3.5 were generated using FEniCS's built in structured mesh functions, as were the corresponding background meshes. For these examples, the background function spaces were C^0 Lagrange FE spaces, and the extraction operators were generated directly in FEniCS, using its `PETScDMCollection` functionality. The background-unfitted foreground meshes used in Sections 4.4.1 and 4.4.2 use an isogeometric background basis. These meshed were created with FEniCS's mshr module, which generates unstructured simplicial meshes from geometric descriptions. The extraction operators were generated using the FEniCS-based isogeometric analysis library tIGAr [99] by simply overriding methods that normally generate structured meshes for exact Lagrange extraction.

4.3 Numerical results

This chapter now explore the properties of interpolation-based immersed analysis numerically using a variety of benchmark problems.

4.3.1 Comparing quadrature-based and interpolation-based methods

The first question this chapter seeks to answer is: How do interpolation-based immersed methods with background-fitted foreground meshes compare with the more thoroughly-studied class of quadrature-based immersed methods? (We defer the comparison of background-fitted and -unfitted foreground meshes to Section 4.3.5.) In particular, this section uses the Poisson equation as a model problem and consider a background function space of tensor-product B-splines of degree n (at maximal continuity) and a foreground Lagrange FE space of equal polynomial degree $n = k$ on a simplicial mesh. The background function space contains monomials of degree greater than n which cannot be represented exactly by the foreground function space, so interpolation-based methods differ from quadrature-based ones.

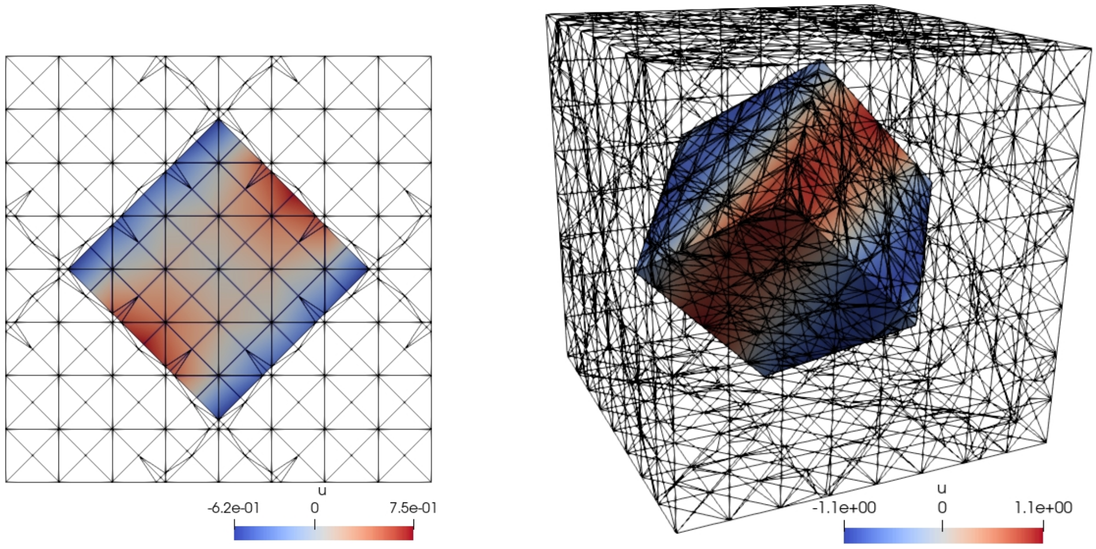
To carry out this test, an instance of the Poisson problem was constructed using the method of manufactured solutions, where the exact solutions were selected as

$$u(\boldsymbol{x}) = \sin(\pi(x_1^2 + x_2^2))\cos(\pi(x_1 - x_2)) \tag{4.32}$$

and

$$u(\boldsymbol{x}) = \sin(\pi(x_1^2 + x_2^2 + x_3^2))\cos(\pi(x_1 + x_2 + x_3)) \tag{4.33}$$

in 2D and 3D respectively, and source term is defined as $f := -\Delta u$. For the 2D problem, Ω is taken to be a unit square rotated by 45° about the origin (i.e., $\Omega = \{\boldsymbol{x} \in \mathbb{R}^2 \ : \ \|\boldsymbol{x}\|_{\ell^1} < 1/2\}$), and the manufactured solution u is applied as Dirichlet data on the boundary of this rotated



(a)

(b)

Figure 4.5. Numerical solution to the Poisson problem using interpolation-based immersed analysis, visualized on the foreground mesh of refinement level $R = 2$. Tests were performed on both the 2D square (a) and 3D rotated cube (b).

square. Ω is then immersed into a structured background mesh of an axis-aligned bi-unit square on which the background B-spline spaces are constructed. A suite of progressively-refined pairs of background and background-fitted foreground meshes are used with background element size $h = 2^{-(R+1)}$, where $R \in \{0, \cdots, 6\}$ denotes the refinement level and "h" is the square root of background element area.

For the 3D case, Ω is taken to be an analogous rotated unit cube (rotated by 45° about the x_3 axis and then by 45° about the x_2 axis) and a sequence of structured hexahedral background meshes are defined with element size $h = 2^{-(R+1)}$, where $R \in \{0, \cdots, 4\}$ is the refinement level and h is the cube root of the element volume. The foreground meshes in 3D are background-fitted tetrahedral meshes.

Dirichlet boundary conditions are enforced weakly using the non-symmetric Nitsche formulation (4.5), with $C_{\text{pen}} = 0$. A representative foreground mesh and numerical solution are shown in Figure 4.5.

Figures 4.6 and 4.7 compare the convergence of interpolation-based and quadrature-based

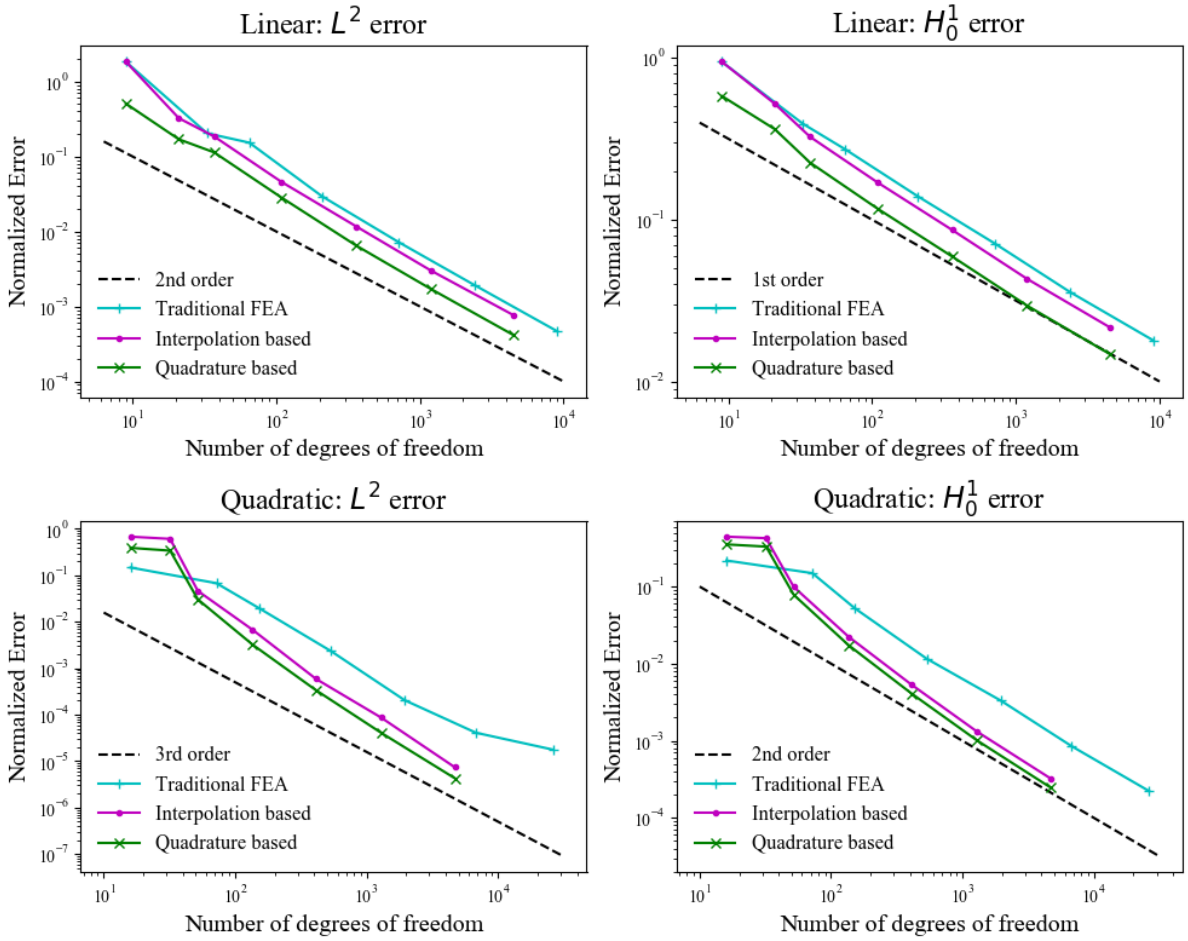


Figure 4.6. Convergence data for Poisson's problem in 2D, comparing the results of traditional FEA on the foreground mesh with the Lagrange FE space, the interpolation based immersed method developed in this paper, and the quadrature based immersed method described in Section 4.1.3.

immersed calculations.[2] Both types of immersed method are seen to converge optimally for $n = 1$ and $n = 2$ in both H_0^1 and L^2 norms. Figures 4.6 and 4.7 also include convergence results of standard FE analysis using the Lagrange FE space defined on the foreground mesh. This also converges optimally with respect to h, but is less accurate per degree of freedom due to excess foreground mesh refinement in cut background elements and less-efficient approximation of smooth functions than the maximally-smooth B-splines of the background function space.

[2] The interpolation-based calculations use the FEniCS-based implementation described in Section 4.2, whereas the quadrature-based calculations use a different research code with the requisite functionality which is outlined in [129]

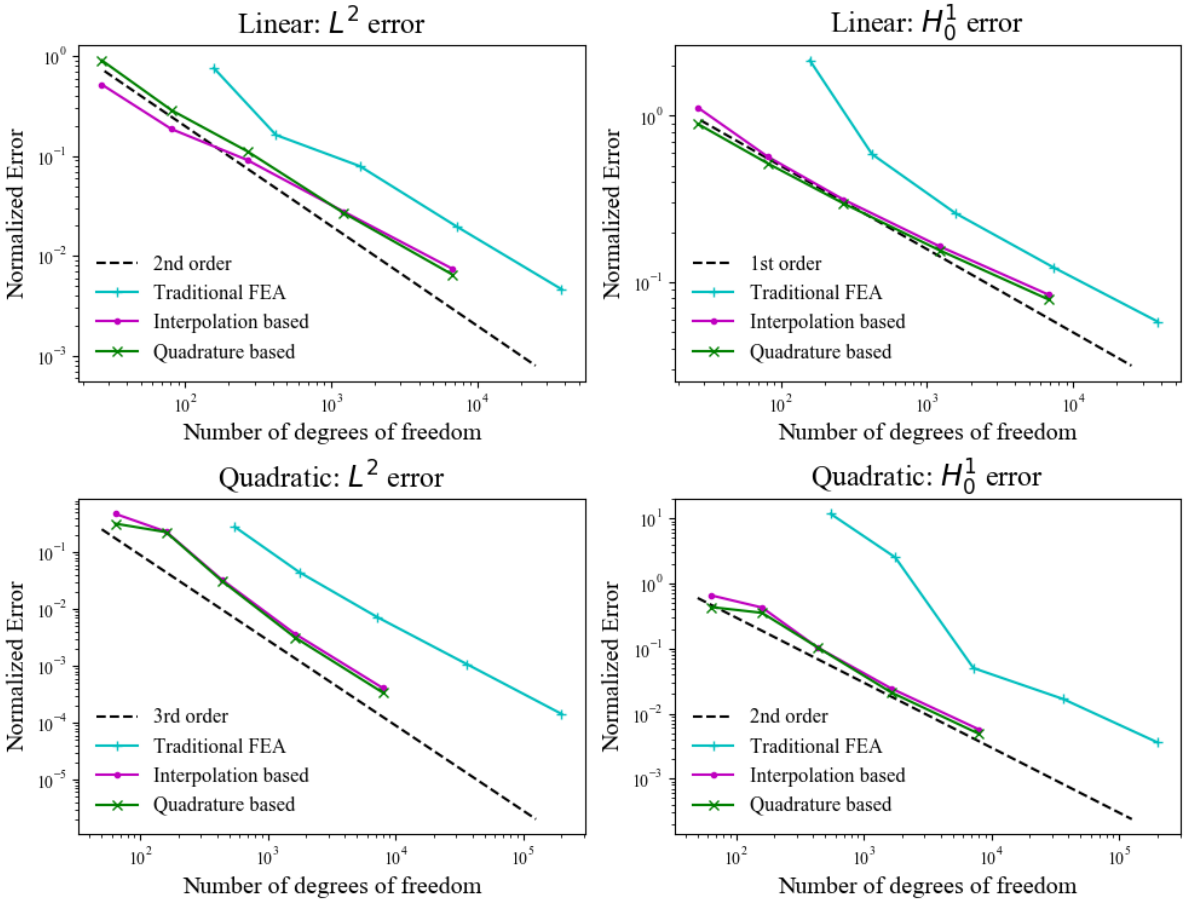


Figure 4.7. Convergence data for Poisson's problem in 3D, comparing the results of traditional FEA on the foreground mesh with the Lagrange FE space, the interpolation based immersed method developed in this paper, and the quadrature based immersed method described Section 4.1.3.

4.3.2 Non-conforming background spaces

The numerical analysis sketched in Section 4.1.5 made the assumption of a conforming discrete space, i.e., that the discrete space $\mathscr{V}^h$ of interpolated background functions was a subset of the infinite dimensional space used to define the continuous variational problem (viz., H^1 in the case of the Poisson equation). However, one may wish to relax that requirement in practice to eliminate topological constraints on the foreground mesh and/or interpolate smooth background functions inexactly with less-regular foreground FE functions.

As a model problem to illustrate non-conforming interpolation-based immersed analysis, consider the biharmonic problem: Find u such that

$$\Delta^2 u = f \;, \tag{4.34}$$

with boundary conditions

$$u = \sigma \text{ on } \partial\Omega \;, \tag{4.35}$$

$$\nabla u \cdot \boldsymbol{n} = \nabla\sigma \cdot \boldsymbol{n} \text{ on } \partial\Omega \;, \tag{4.36}$$

where $f : \Omega \to \mathbb{R}$ is a given source term and $\sigma : \Omega \to \mathbb{R}$ is an auxiliary function used to define Dirichlet boundary data for u and $\nabla u \cdot \boldsymbol{n}$ on $\partial\Omega$.[3] Unlike the nonsymmetric Nitsches method used in the Poisson problem in (4.5), this problem is discretized using a symmetric Nitsche-like method: Find $u^h \in \mathscr{V}^h$ such that, $\forall v^h \in \mathscr{V}^h$,

$$\begin{aligned}
&\int_\Omega \Delta u^h \Delta v^h \, d\Omega + \int_{\partial\Omega} \nabla\Delta u^h \cdot \boldsymbol{n} v^h - \Delta u^h \nabla v^h \cdot \boldsymbol{n} \, d\Gamma \\
&+ \int_{\partial\Omega} (\nabla\Delta v^h) \cdot \boldsymbol{n}(u^h - \sigma) - \Delta v (\nabla u^h \cdot \boldsymbol{n} - \nabla\sigma \cdot \boldsymbol{n}) \, d\Gamma \\
&+ \int_{\partial\Omega} \frac{\alpha}{h^3}(u^h - \sigma) v^h + \frac{\beta}{h}(\nabla u^h \cdot \boldsymbol{n} - \nabla\sigma \cdot \boldsymbol{n}) \nabla v^h \cdot \boldsymbol{n} \, d\Gamma = \int_\Omega f v^h \, d\Omega \;,
\end{aligned} \tag{4.37}$$

where $\alpha > 0$ and $\beta > 0$ are independent of h and η, but must be sufficiently large to ensure stability of the formulation.[4] In the computations of this paper $\alpha = \beta = 5$. In the case of immersed methods, the integrals $\int_\Omega (\cdots) \, d\Omega$ and $\int_{\partial\Omega} (\cdots) \, d\Gamma$ should be understood as being split up over elements of the foreground mesh, as in (4.7). This distinction is important when non-conforming spaces are used as some spatial derivatives may have distributional contributions

[3] One could instead define separate functions to prescribe boundary data for u and its normal derivative, but an equivalent σ must exist for the problem to be well-posed and the present perspective is more convenient for applying the method of manufactured solutions.

[4] For the method to be usable with piecewise-quadratic function spaces, α must be strictly positive, even if a non-symmetric variant is used, as $\nabla\Delta v^h \equiv 0$ and only the penalty term is left to enforce $u^h \approx \sigma$.

at interior mesh facets that are ignored, so (4.37) technically involves some abuse of notation.

As with the Poisson problem in Section 4.3.1, the method of manufactured solutions is used with exact solutions

$$u(\boldsymbol{x}) = \cos(0.05\pi x_1 + 0.1)\cos(0.05\pi x_2 + 0.1)) \tag{4.38}$$

and

$$u(\boldsymbol{x}) = \cos(\pi x_1 + 0.5)\cos(\pi x_2 + 0.5)\cos(\pi x_3 + 0.5) \tag{4.39}$$

in 2D and 3D, respectively, and setting the problem data to $f := \Delta^2 u$ and $\sigma := u$.

The rotated square and cube domains and mesh sequences from Section 4.3.1 are reused for this problem. The *background* function spaces are selected to be H^2-conforming. In particular, quadratic B-splines of maximal continuity are used. However, the *foreground* function spaces are taken to be quadratic Lagrange FE spaces, which, on simplicial meshes, do not represent all multivariate monomials of the background B-spline spaces. Further, interpolations of background functions on this Lagrange FE space are only C^0, containing jumps in their first derivatives (so they can be at most $H^{3/2-\varepsilon}$-conforming, for $\varepsilon > 0$ arbitrarily-small). Interpolation-based immersed analysis will therefore be non-conforming in this case. 2D and 3D convergence results are shown in Figure 4.8. Optimal rates of 1 and 2 are achieved for the H^2 and H^1 norms respectively, where "H^2 norm" is understood in a broken sense (i.e., the square root of a sum over squared foreground-element H^2 norms) in this non-conforming setting. A rate of 2 is observed v for the L^2 norm, but this is expected, since an Aubin–Nitsche-type duality argument to obtain an optimal L^2 error bound breaks down for quadratic B-spline discretizations of the biharmonic problem [152].

Remark. *The 3D foreground meshes used in this section include some elements where $\eta \ll h$, such as the example shown in Figure 4.4. The harmful effects of small background element–domain intersections on linear system conditioning have been studied previously in the context* *of quadrature-based immersed methods, along with a variety of possible remedies, but we leave the adaptation of such techniques to the interpolation-based setting to future work, as discussed further in Section 7.1. However, small foreground elements, where* $\eta \to 0$*, present a distinct challenge unique to interpolation-based methods, due to the blow-up of foreground basis function derivatives .*[5] *The resulting large numbers in the matrix* $\mathbf{A}$ *defined by* (4.29) *should, in exact arithmetic, be canceled by small entries in the extraction matrix* $\mathbf{M}$ *when forming* $\mathbf{K}$ *by* (4.28)*. However, this cancellation is not exact due to finite precision. For the fourth-order biharmonic problem in 3D, this can prevent convergence even with direct solvers. In the 3D biharmonic examples from the present work, this issue is avoided by removing near-zero volume elements from the assembly of* $\mathbf{A}$*, skipping integration over foreground elements (and their associated boundary facets) if their volume falls below* 0.001% *of the largest foreground element. The resulting geometric perturbation of the domain has no observable effect on solutions, as illustrated by the convergence results of Figure 4.8. One can imagine pathological domain–mesh intersections where such a strategy would lead to significant errors, but these do not arise in the tests performed for this chapter, and the solution of fourth-order problems on topologically-3D domains is itself relatively uncommon in applications.*

4.3.3 Geometric approximations of domain geometry

The convergence tests of Sections 4.3.1 and 4.3.2 deliberately used polyhedral PDE domains so that a foreground mesh of affine simplex elements could fit them exactly, allowing for the isolation of errors due to interpolation of the background basis functions. This chapter now considers a problem with a curved boundary where there is some geometry error between Ω and the foreground mesh.

In particular, the classic linear elasticity benchmark of an infinite 2D plate with a circular hole subjected to biaxial tension is choosen. The plane strain variant of the problem is considered.

[5] This blow-up does not manifest in quadrature-based methods since they directly evaluate background basis function derivatives.

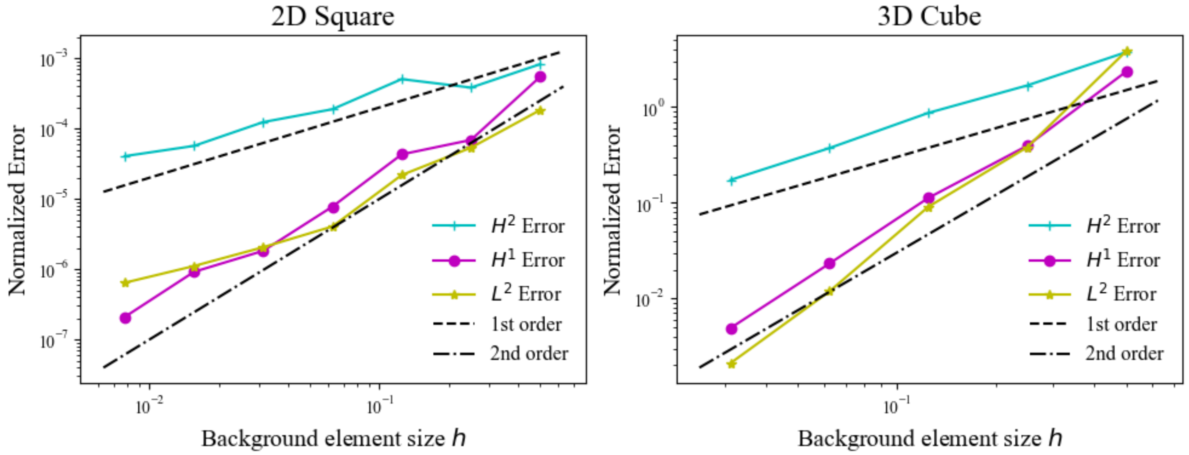


Figure 4.8. Convergence data for the biharmonic problem.

An exact solution for the stress field is readily available in elasticity textbooks, e.g., [116, Section 7.7.5]. The problem is truncated to a finite square domain whose side length is four times the diameter of the hole and the traction is applied from the exact solution to the infinite-domain problem on the boundary. This finite-domain problem is further truncated by restricting it to the upper-right quadrant of the x_1–x_2 plane and applying symmetry boundary conditions on the axes, which eliminates rigid-body modes from the displacement solution of the original pure Neumann problem. Linear elasticity is formulated as a problem for displacement, viz., Navier's equations, which also provide an opportunity to demonstrate interpolation-based immersed analysis for a system of PDEs: Find a displacement field $\boldsymbol{u} : \Omega \to \mathbb{R}^2$ such that

$$-\nabla \cdot \boldsymbol{\sigma}(\boldsymbol{u}) = \mathbf{0} \tag{4.40}$$

subject to boundary conditions

$$\boldsymbol{u} \cdot \boldsymbol{n} = 0 \quad \text{on } \Gamma_{\text{sym}} \,, \tag{4.41}$$

$$(\boldsymbol{I} - \boldsymbol{n} \otimes \boldsymbol{n})(\boldsymbol{\sigma} \cdot \boldsymbol{n}) = \mathbf{0} \quad \text{on } \Gamma_{\text{sym}} \,, \tag{4.42}$$

$$\boldsymbol{\sigma} \cdot \boldsymbol{n} = \boldsymbol{t} \quad \text{on } \Gamma_t \,, \tag{4.43}$$

where $\Gamma_{\text{sym}} \subset \partial\Omega$ is the union of the symmetry planes where sliding boundary conditions are applied, $\Gamma_t = \partial\Omega \setminus \Gamma_{\text{sym}}$ is the Neumann boundary on which traction is prescribed, $\boldsymbol{t}$ is the traction on Γ_t taken from the exact solution, and the Cauchy stress $\boldsymbol{\sigma}$ is

$$\boldsymbol{\sigma}(\boldsymbol{u}) = 2\mu\boldsymbol{\varepsilon}(\boldsymbol{u}) + \lambda\,\mathrm{tr}(\boldsymbol{\varepsilon}(\boldsymbol{u}))\boldsymbol{I} \tag{4.44}$$

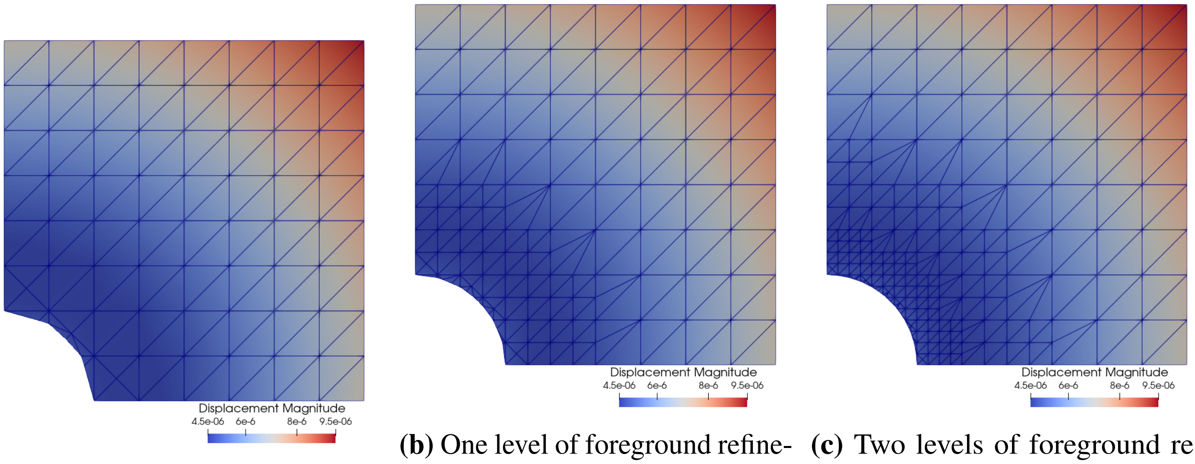
in terms of the symmetric gradient operator $\boldsymbol{\varepsilon}(\cdot)$ (which gives strain when applied to the displacement) and Lamé parameters μ and λ, which are set from Young's modulus $E = 200 \times 10^9$ and Poisson's ratio $\nu = 0.3$ by standard formulas.

Navier's equations are discretized with slip boundary conditions using a symmetric Nitsche-like formulation analogous to (4.5): Find $\boldsymbol{u}^h \in \mathscr{V}^h$ such that, $\forall \boldsymbol{v}^h \in \mathscr{V}^h$,

$$\begin{aligned}
&\int_\Omega \boldsymbol{\sigma}(\boldsymbol{u}^h) : \nabla \boldsymbol{v}^h \, d\Omega \\
&\quad - \int_{\Gamma_{\text{sym}}} (\boldsymbol{u}^h \cdot \boldsymbol{n})\boldsymbol{n} \cdot \boldsymbol{\sigma}(\boldsymbol{v}^h) \cdot \boldsymbol{n} \, d\Gamma - \int_{\Gamma_{\text{sym}}} (\boldsymbol{v}^h \cdot \boldsymbol{n})\boldsymbol{n} \cdot \boldsymbol{\sigma}(\boldsymbol{u}^h) \cdot \boldsymbol{n} \, d\Gamma \\
&\quad + \int_{\Gamma_{\text{sym}}} \frac{\beta\mu}{h} (\boldsymbol{u}^h \cdot \boldsymbol{n})\boldsymbol{n} \cdot \boldsymbol{v}^h \, d\Gamma = \int_{\Gamma_t} \boldsymbol{t} \cdot \boldsymbol{v}^h \, d\Gamma \,,
\end{aligned} \tag{4.45}$$

where $\beta > 0$ is a penalty parameter associated with Nitsche's method (in this section $\beta = 10$) and integrals should be understood in the sense of (4.7) (which is technically an abuse of notation in the present case because it ignores the geometry error in the foreground mesh). Similarly, the outward-facing unit normal $\boldsymbol{n}$ depends on the geometry of the foreground mesh, but this dependence has been suppressed in the notation.

To study the behavior of interpolation-based immersed methods, a sequence of uniform, structured background meshes of quadrilateral elements, covering the truncated plate and hole are constructed. Several corresponding sequences of foreground meshes are considered, with different levels of local refinement around the curved boundary of the hole. The three different foreground mesh refinement levels are shown for an 8 × 8 background mesh in Figure 4.9, along with representative displacement solutions. In all cases, the foreground mesh elements are affine



(a) No foreground refinement

(b) One level of foreground refinement

(c) Two levels of foreground refinement

Figure 4.9. Plots showing the magnitude of displacement. To resolve curved surfaces, the foreground mesh is locally refined, while the same uniform background mesh is used.

triangles. For the background function spaces, we select B-spline spaces of degrees $n = 1$ and $n = 2$, with maximal continuity for each, and, for the foreground function spaces Lagrange FE spaces of corresponding polynomial degrees $k = n$ are used.

Figure 4.10 shows the convergence of the L^2 norm of error in the Cauchy stress for different polynomial degrees and levels of foreground refinement. Even without foreground refinement the case of $n = 1$ attains the optimal convergence rate. For $n = 2$, geometry error dominates in the case without foreground refinement and limits the rate of convergence. However, even with just one level of foreground refinement near the curved boundary, an optimal effective convergence rate is observed. For affine foreground elements and a fixed level of local foreground refinement, one still expects the geometry error to dominate in the ultimate asymptotic limit (as occurs also with the even lower-order stair-step approximations of curved boundaries in finite cell adaptive quadrature), but this phenomenon is unlikely to be encountered in practical problems where the model error would dominate well before this asymptotic regime. The authors hypothesize that the use of high-order isoparametric elements instead of local foreground refinement would provide true optimal convergence, but high-order foreground mesh generation is currently outside the capabilities of the software framework used for this paper.

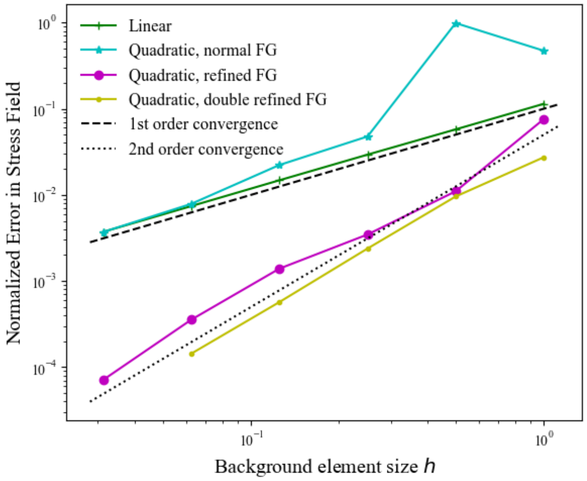


Figure 4.10. Frobenius norm of stress error tensor plotted against the background element size shown with both linear basis functions and quadratic basis functions. In the case of quadratic basis functions, the foreground mesh was refined as shown in Figure 4.9.

4.3.4 Nonlinear and unsteady problems

This chapter now considers the unsteady incompressible Navier–Stokes equations as an example of a more complicated nonlinear PDE system. This problem also illustrates the use of interpolation-based immersed finite element analysis in conjunction with stabilized finite element formulations that use mesh-dependent variational forms, originally designed with the standard body-fitted FE setting in mind.

In particular, consider the PDE system: Find velocity and pressure fields $(\boldsymbol{u}, p) : \Omega \times (0,T) \to \mathbb{R}^d \times \mathbb{R}$ such that

$$\begin{aligned} \rho\left(\partial_t \boldsymbol{u} + \boldsymbol{u}\cdot\nabla\boldsymbol{u}\right) &= -\nabla p + \mu\Delta\boldsymbol{u} \\ \nabla\cdot\boldsymbol{u} &= 0 \end{aligned} \tag{4.46}$$

subject to the boundary condition

$$\boldsymbol{u} = \boldsymbol{g} \quad \text{on } \partial\Omega\,, \tag{4.47}$$

where d is the spatial dimension of Ω, $(0,T)$ is the time interval on which the problem is posed, ρ is mass density, μ is dynamic viscosity, and $\boldsymbol{g}$ is given Dirichlet boundary data. This in space is discretized using a stabilized variational multiscale (VMS) formulation [17] with non-symmetric Nitsche-like weak enforcement of the Dirichlet boundary condition [21], leading to the semi-discrete problem: Find $(\boldsymbol{u}^h, p^h) \in \boldsymbol{\mathscr{V}}^h \times \mathscr{Q}^h$ such that, $\forall(\boldsymbol{v}^h, q^h) \in \boldsymbol{\mathscr{V}}^h \times \mathscr{Q}^h$,

$$\begin{aligned}
\int_\Omega \left(\rho\left(\partial_t \boldsymbol{u}^h + \boldsymbol{u}^h \cdot \nabla \boldsymbol{u}^h\right) \cdot \boldsymbol{v}^h + \boldsymbol{\sigma}(\boldsymbol{u}^h, p^h) : \nabla \boldsymbol{v}^h + \nabla \cdot \boldsymbol{u}^h q^h\right) d\Omega \\
+ \int_{\partial\Omega} \left(-(\boldsymbol{\sigma}(\boldsymbol{u}^h, p^h)\boldsymbol{n}) \cdot \boldsymbol{v}^h + (\boldsymbol{\sigma}(\boldsymbol{v}^h, q^h) \cdot \boldsymbol{n}) \cdot (\boldsymbol{u}^h - \boldsymbol{g})\right) d\Gamma \\
- \int_{\partial\Omega} \left(\rho \min\{\boldsymbol{u}^h \cdot \boldsymbol{n}, 0\}(\boldsymbol{u}^h - \boldsymbol{g}) \cdot \boldsymbol{v}^h\right) d\Gamma \\
+ \int_\Omega \tau_{\mathrm{M}} \left(\left(\boldsymbol{u}^h \cdot \nabla \boldsymbol{v}^h + \frac{1}{\rho} \nabla q^h\right) \cdot \boldsymbol{r}_{\mathrm{M}} - \boldsymbol{v}^h \cdot \boldsymbol{r}_{\mathrm{M}} \cdot \nabla \boldsymbol{u}^h\right) d\Omega \\
- \int_\Omega \frac{\tau_{\mathrm{M}}^2}{\rho} (\nabla \boldsymbol{v}^h) : (\boldsymbol{r}_{\mathrm{M}} \otimes \boldsymbol{r}_{\mathrm{M}})\, d\Omega \\
+ \int_\Omega \tau_{\mathrm{C}} \left(r_{\mathrm{C}} \nabla \cdot \boldsymbol{v}^h\right) d\Omega = 0
\end{aligned} \tag{4.48}$$

where

$$\boldsymbol{\sigma}(\boldsymbol{u}, p) = 2\mu \boldsymbol{\varepsilon}(\boldsymbol{u}) - p\boldsymbol{I} \tag{4.49}$$

gives the functional form of the Cauchy stress in terms of the velocity and pressure fields,

$$\boldsymbol{r}_{\mathrm{M}} = \rho(\partial_t \boldsymbol{u}^h + \boldsymbol{u}^h \cdot \nabla \boldsymbol{u}^h) - \nabla \cdot \boldsymbol{\sigma}(\boldsymbol{u}^h, p^h) \tag{4.50}$$

is the residual of the strong momentum balance equation,

$$r_{\mathrm{C}} = \rho \nabla \cdot \boldsymbol{u}^h \tag{4.51}$$

is the residual of the strong continuity equation, and τ_{M} and τ_C are the momentum and continuity stabilization parameters, respectively. A number of definitions for these stabilization parameters have been studied in the literature. This section follows [18] in defining

$$\tau_{\mathrm{M}} = \left(\boldsymbol{u}^h \cdot \boldsymbol{G} \cdot \boldsymbol{u}^h + C_{\mathrm{I}} \nu^2 \boldsymbol{G} : \boldsymbol{G} + \frac{C_t}{\Delta t^2} \right)^{-1/2}, \tag{4.52}$$

and

$$\tau_{\mathrm{C}} = \frac{1}{\tau_{\mathrm{M}} \mathrm{tr}(\mathbf{G})}, \tag{4.53}$$

with $\nu = \mu/\rho$ the kinematic viscosity, Δt the time step size of the temporal discretization, $C_{\mathrm{I}} > 0$ and $C_t > 0$ are dimensionless constants (set to $C_{\mathrm{I}} = 60$ and $C_t = 4$ in this work), and $\boldsymbol{G}$ is an anisotropic mesh size tensor, which simplifies to

$$\boldsymbol{G} = 4h^{-2}\boldsymbol{I} \tag{4.54}$$

in the case of a uniform isotropic mesh with element size h. In the computations of this chapter, this element size is taken from the uniform background mesh. The system of ordinary differential equations emanating from the semi-discrete problem (4.48) is discretized by the implicit midpoint rule in time, with a time step Δt proportional to the background element size h.

As a representative instance of the Navier–Stokes problem, this chapters considers the 2D Taylor–Green vortex. The statement and spatially-periodic exact solution of this problem are well-known. In numerical studies, it is often restricted to the square $(0,\pi)^2$ with symmetry boundary conditions (e.g., [62, Section 9.10.1]). However, in the present work, we take Ω to be the rotated unit square used as the domain of the Poisson problem in Section 4.3.1, apply the exact solution as the boundary data $\boldsymbol{g}$, and use the same sequence of background and foreground mesh pairs defined in Section 4.3.1. The stabilized formulation (4.48) permits the stable use of equal-order discretizations, i.e.,

$$\boldsymbol{\mathscr{V}}^h = (\mathscr{V}^h)^d \quad \text{and} \quad \mathscr{Q}^h = \mathscr{V}^h \tag{4.55}$$

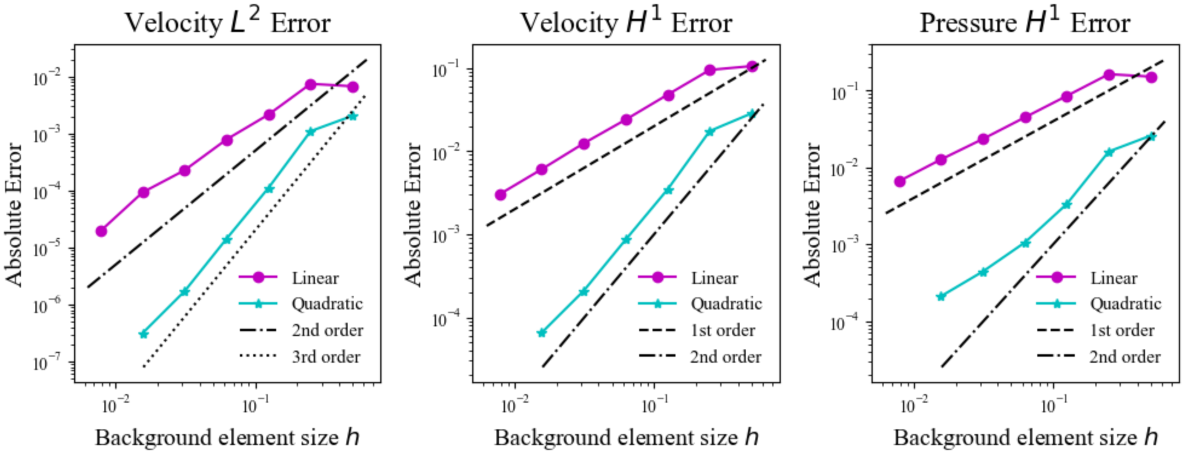


Figure 4.11. Convergence data for the velocity and pressure fields of the Taylor–Green vortex problem using both linear and quadratic discretizations. Kinematic viscosity was set to $1/100$ yielding a Reynolds number of 100, and error is evaluated at time $T = 1$.

for a single scalar function space $\mathscr{V}^h$. Our numerical tests in this section use the same linear and quadratic choices for $\mathscr{V}^h$ as used for the scalar Poisson problem in Section 4.3.1.

The convergence of velocity and pressure as $h \sim \Delta t \to 0$ (resulting in a CFL number of approximately 1) is shown in Figure 4.11. Effectively optimal rates of velocity convergence with respect to the polynomial degree of the spatial discretizations in the L_2 and H^1 norms are seen. In principle, using $h \sim \Delta t$ and the implicit midpoint rule should limit convergence to second-order in the L_2 norm,[6] but the quadratic spatial discretizations appear to be third-order in velocity L^2 error within the range of resolutions tested. Pressure H^1 error convergence is also nearly optimal, but with a drop in convergence rate on the finest meshes when using quadratic spatial discretizations. This sub-optimal H^1 pressure convergence is consistent with the analysis of related stabilized methods for Stokes and Oseen flow, e.g., [93, 63], where error is bounded in a norm on the velocity–pressure product space in which the pressure gradient is scaled by element size, weakening its contribution.

[6] The assumption of $h \sim \Delta t$ is tacit in the stabilized formulation used; refining Δt faster than h would weaken stabilization in the asymptotic regime. See [85] for additional discussion.

4.3.5 Background-unfitted foreground meshes

The analysis of Section 4.1.5 and preceding numerical examples in Sections 4.3.1–4.3.4 all used background-fitted foreground meshes. Background-fitted foreground meshes are a necessity for quadrature-based immersed methods, since the quadrature rules defined on foreground elements require smoothness of the integrand for accuracy. However, interpolations of background basis functions on a foreground mesh are, by construction, smooth within each foreground element and can be accurately integrated with foreground element-by-element quadrature rules. Of course, the solution space for interpolation-based immersed methods consists of interpolations of background basis functions that are non-smooth within each foreground element, so the analysis of Section 4.1.5 is not applicable. However, the numerical results of the current section suggest that optimal accuracy is maintained with background-unfitted foreground meshes.

In the preceding examples, the background function space was taken to be a B-spline space of maximal continuity. However, as a stress test of background-unfitted interpolation-based immersion, this section instead tests it with C^0 Lagrange background function spaces defined on simplicial meshes. As test problems, consider the Poisson equation with a manufactured solution (cf. Section 4.3.1) and the Taylor–Green vortex (cf. Section 4.3.4). For both of these test problems, the domain is taken to be a rotated square, and the sequence of pairs of background and background-unfitted foreground meshes are constructed similarly to the representative pair shown in Figure 4.12 but at varying levels of refinement. In all tests, the foreground element size η is kept proportional to the background element size h, and the polynomial degrees n and k of the background and foreground are equal. For the Taylor–Green vortex problem, the same time implicit midpoint rule with time step Δt proportional to the background element size h was used as in Section 4.3.4. The convergence results for both the Poisson and Navier–Stokes examples are shown in Figures 4.13 and 4.14 respectively. Optimal convergence rates of L^2 and H^1 error (of velocity in the case of Navier–Stokes) for linear and quadratic discretizations are seen.

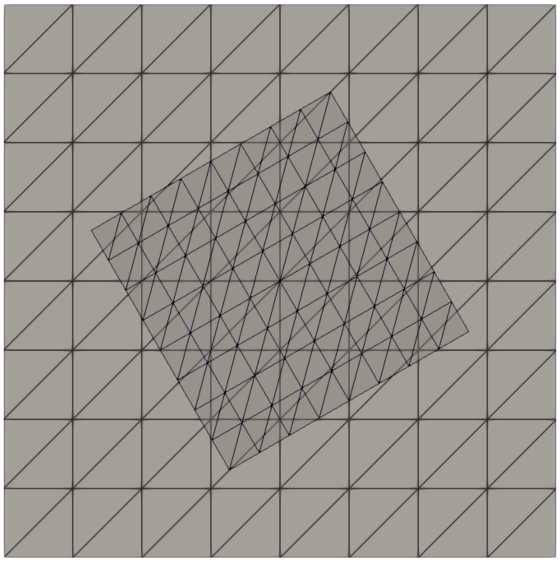


Figure 4.12. A representative example of a background mesh and a background-unfitted foreground mesh.

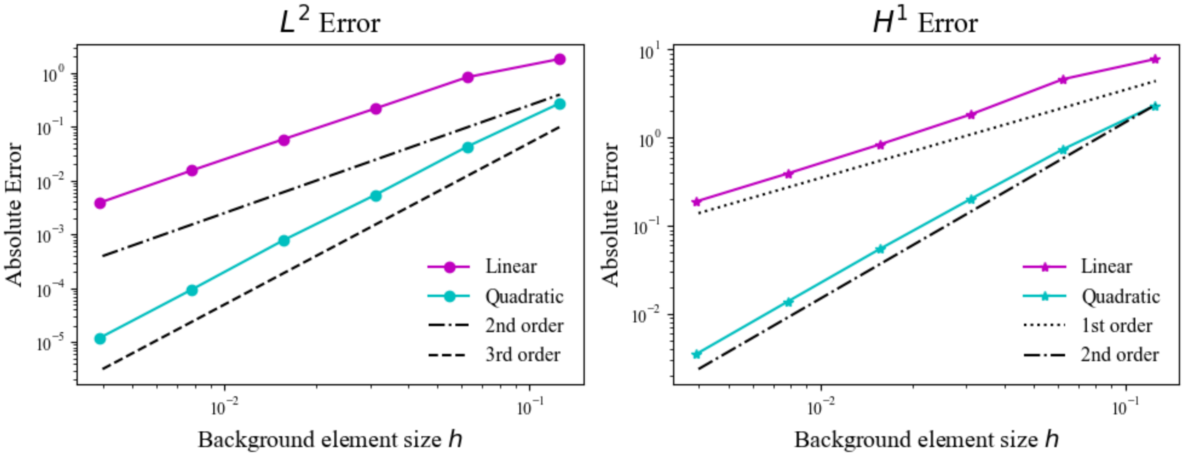


Figure 4.13. Convergence data for the Poisson problem, generated using extraction onto unfitted background meshes (Figure 4.12).

4.4 Application to IMGA of trimmed shells

A major success of IGA has been its application to thin shell analysis, originating in the work of Kiendl and collaborators [104, 103, 105]. Smooth discrete function spaces from IGA make it possible to apply the Bubnov–Galerkin method directly to Kirchhoff–Love theory where

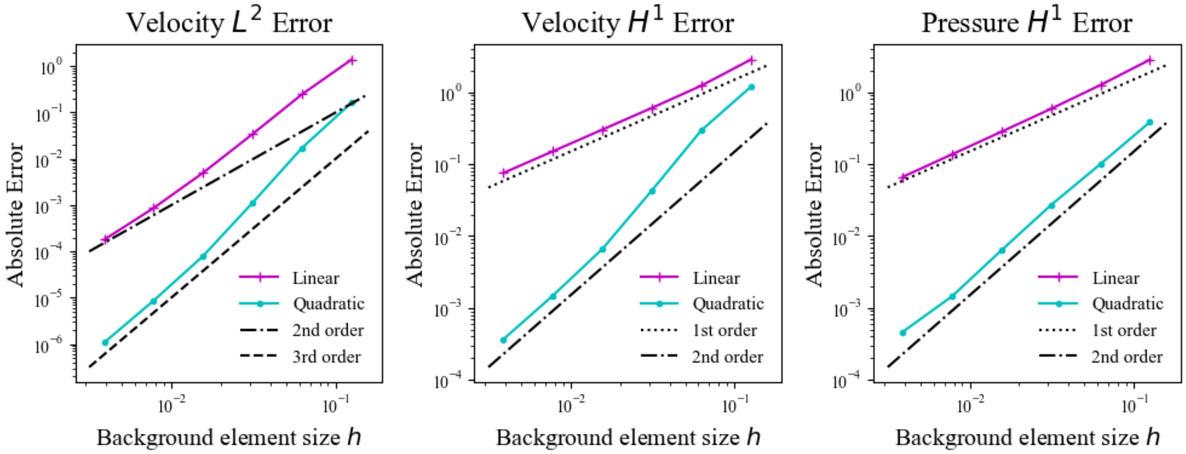


Figure 4.14. Convergence data for the Taylor–Green vortex problem generated using extraction onto unfitted meshes (Figure 4.12), with Reynolds number 100, after elapsed time $T = 1$.

the displacement satisfies a fourth-order PDE system. This provides much better per-degree-of-freedom accuracy than classical FE discretizations of shells, sometimes resolving important qualitative solution features with orders of magnitude less computation [123]. However, the original isogeometric Kirchhoff–Love shell formulation of [104] is restricted to boundary-fitted spline representations of the midsurface geometry. With suitably-flexible spline technologies, such as T-splines [149], it is practical to design such "analysis-suitable" geometries for industrial applications as demonstrated for thin shell analysis of composite wind turbine blades [20], prosthetic heart valve leaflets [86], and automobile parts [40]. However, the vast majority of industrial CAD software defines surface geometries in an immersed way by cutting them out of unfitted parametric coordinate charts along arbitrary trim curves [119]. Thus, immersogeometric formulations are needed to directly analyze CAD models of shell structures. Quadrature-based immersed methods for thin shells have previously been used in a number of studies [78, 49, 50]. In this section of the present work, interpolation-based IMGA is applied to Kirchhoff–Love analysis of shells whose geometries are defined by subsets of a B-spline patch's parameter space.

The specific variational formulation that used in this section is that of a geometrically-nonlinear Kirchhoff–Love shell of uniform thickness h_{th} whose material behavior follows a homogeneous isotropic St. Venant–Kirchhoff model defined by a Young's modulus E and Poisson ratio ν. A complete mathematical statement of this problem is given by [103] and its translation into FEniCS UFL is available online in the library ShNAPr [98] developed to support the work in [97]. The extensive kinematic definitions are therefore omitted for brevity as they are neither novel nor essential to following the present study.

Pinned boundary conditions are implemented on the edges of trimmed shells using a penalty formulation based on [82]. This consists of adding

$$+\frac{1}{2}\int_{\mathscr{L}_{\text{pin}}} \frac{\alpha E h_{\text{th}}}{h} \left| \boldsymbol{u}^h - \boldsymbol{g} \right|^2 \, d\mathscr{L} \tag{4.56}$$

to the shell structure's elastic energy functional where $\mathscr{L}_{\text{pin}}$ is the portion of the trim curve (in the reference configuration) on which pinned boundary conditions are to be applied, $\alpha > 0$ is a dimensionless constant, h is the background-mesh element size, $\boldsymbol{u}^h$ is the unknown discrete displacement field of the shell structure's midsurface, and $\boldsymbol{g}$ is the prescribed midsurface displacement on the pinned boundaries. For quadratic spline discretizations, this "naive" penalty method for pinned boundaries is essentially as accurate as a full Nitsche-type formulation (cf. [30]) because the associated consistency term involves third-order derivatives. The use of (4.56) in conjunction with an analogous rotational penalty has been demonstrated to be accurate for coupling non-matching shell structures in practical problem settings [82, 94, 112, 165].

The examples of this section are limited to trimmed single-patch shell geometries. However the authors see no significant conceptual barrier preventing the application of the above-cited patch coupling methodologies to collections of multiple trimmed patches, modeling shell geometries with sharp crease-like features. Given a foreground FE mesh of the physical part of each patch, the software framework of [165] could be applied more-or-less directly, with minimal modifications, since it is already based on Lagrange extraction.

4.4.1 Verification with boundary-fitted IGA

Interpolation-based IMGA of thin shells is first tested by trimming out a rectangular domain which can easily be modeled with an untrimmed B-spline to provide a reference solution for verification purposes. In particular, for each Cartesian component of the shell midsurface displacement, the $n = 2$ sequence of scalar B-spline background and Lagrange-interpolated foreground spaces from the 2D Poisson and biharmonic tests of Sections 4.3.1 and 4.3.2 are reused, where the parametric-space domain is a rotated unit square trimmed out of an axis-aligned bi-unit square, shown in Figure 4.15. Additionally, a new background-unfitted $n = 2$ sequence of scalar B-spline background and Lagrange-interpolated foreground spaces are employed, constructed as described in Section 4.2.3. In this test, the mapping from the B-spline parameter space to the physical-space reference configuration of the shell's midsurface are given by

$$\boldsymbol{X}(\boldsymbol{\xi}) = \xi_1 \boldsymbol{e}_1 + \xi_2 \boldsymbol{e}_2 \ , \tag{4.57}$$

where $\boldsymbol{\xi} = (\xi_1, \xi_2)$ is a point in the B-spline parameter space, $\boldsymbol{X}$ is the corresponding 3D point on the reference configuration midsurface, and $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$ is the standard orthonormal basis for a Cartesian coordinate system of 3D physical space into which the reference configuration is embedded. The construction of an analogous sequence of untrimmed reference B-spline discretizations is straightforward due to the square geometry.

Homogeneous (i.e., $\boldsymbol{g} = \mathbf{0}$) pinned boundary conditions are applied to the entire boundary of the unit square domain, using penalty coefficient $\alpha = 10^5$, and material parameters of $E = 4.8 \times 10^5$ and $\nu = 0.38$ are assumed, and thickness $h_{\text{th}} = 0.1$ is set. A uniform force density per unit reference area of $90\boldsymbol{e}_3$ is applied, leading to an "inflated" equilibrium configuration, of which a representative numerical approximation using interpolation-based IMGA is shown in Figures 4.15a and 4.15b. The out-of-plane displacement component is extracted at the center of the plate using IMGA at several levels of refinement and is plotted as a function of number of degrees of freedom in Figure 4.15c. This displacement converges to the reference value obtained from a highly-refined computation using standard untrimmed IGA. (This is a geometrically-nonlinear problem, for which there is no analytical Kirchhoff–Love solution available.)

4.4.2 Background-fitted and -unfitted IMGA for a complicated trimmed geometry

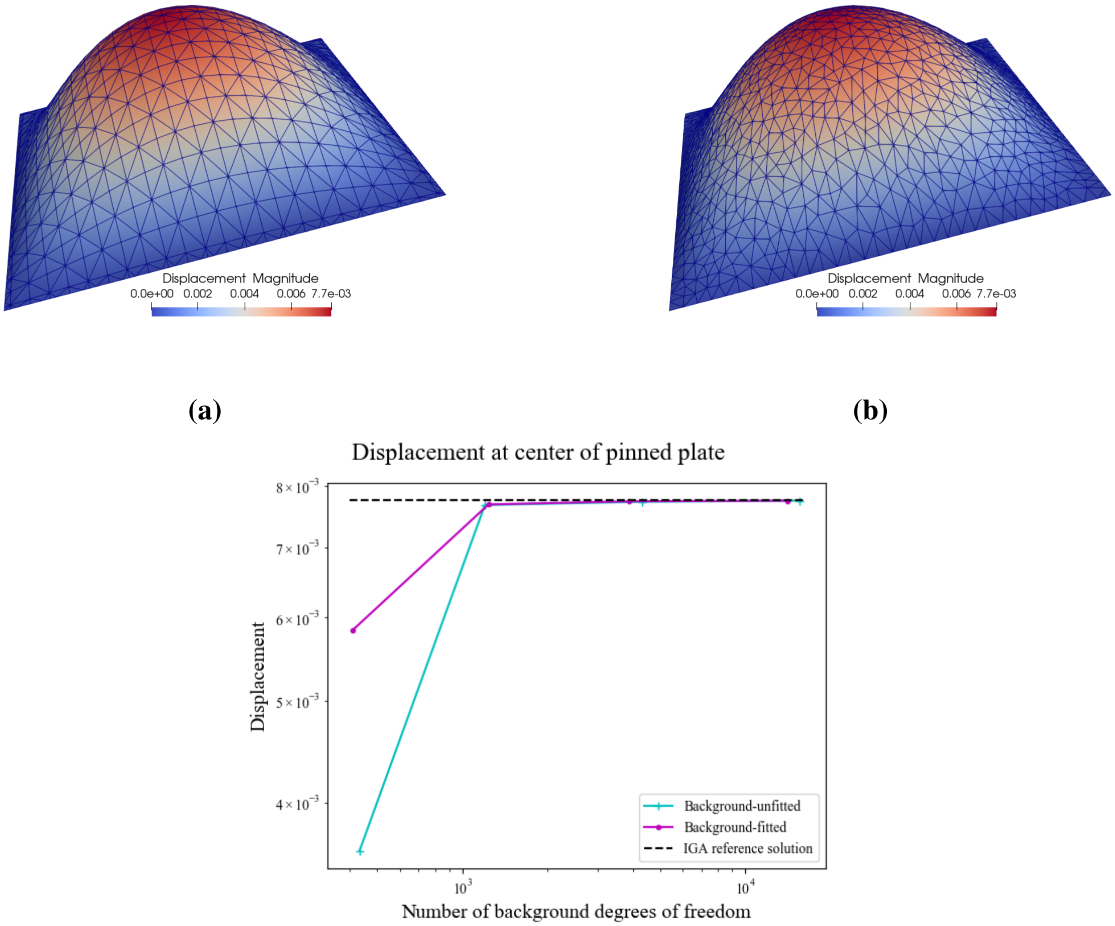
As a final example a more complicated trimmed spline geometry is considered, for which a boundary-fitted IGA model would require unstructured spline technologies that are not currently part of standard design workflows. Background-fitted and -unfitted meshes of the trimmed geometry are shown in the spline parameter space in Figures 4.16a and 4.16b. The parametric space is then mapped to a curved physical-space reference configuration by

$$\boldsymbol{X}(\boldsymbol{\xi}) = \xi_1 \boldsymbol{e}_1 + \xi_2 \boldsymbol{e}_2 + A(1-\xi_1^2)\boldsymbol{e}_3 \ , \tag{4.58}$$

as visualized in Figures 4.16c and 4.16d where $A = 1/2$. The physical problem solved on this geometry is a static Kirchhoff–Love shell analysis with $E = 3 \times 10^4$, $\nu = 0.3$, and $h_{\text{th}} = 0.03$, pinned boundary conditions applied to the outer left and right sides of the parametric domain, and a pressure follower load with force density per unit reference midsurface area of $\boldsymbol{f} = 90\boldsymbol{n}$ where $\boldsymbol{n}$ is the unit normal vector to the deformed midsurface. Representative numerical approximations of the resulting equilibrium configuration are shown in Figures 4.16e and 4.16f.

Remark. *The static equilibrium configurations shown in Figures 4.16e and 4.16f were arrived at by computing the steady-state solution of a dynamic Kirchhoff–Love shell problem with strong mass damping and numerical dissipation from under-resolved backward-Euler time stepping. This pseudo-time continuation method was necessary to avoid bifurcations that arise on some meshes when attempting to apply static load stepping.*

To study convergence with respect to mesh refinement quantitatively, the displacement magnitude is extracted at the parametric point $\boldsymbol{\xi} = (0, -1/4)$ (corresponding to the bottom of



(a)

(b)

(c)

Figure 4.15. Examples of the background-fitted (a) and background-unfitted (b) discretizations used to verify the IMGA formulation. (c) Displacement of the center of the plate is plotted against the number of degrees of freedom in each mesh..

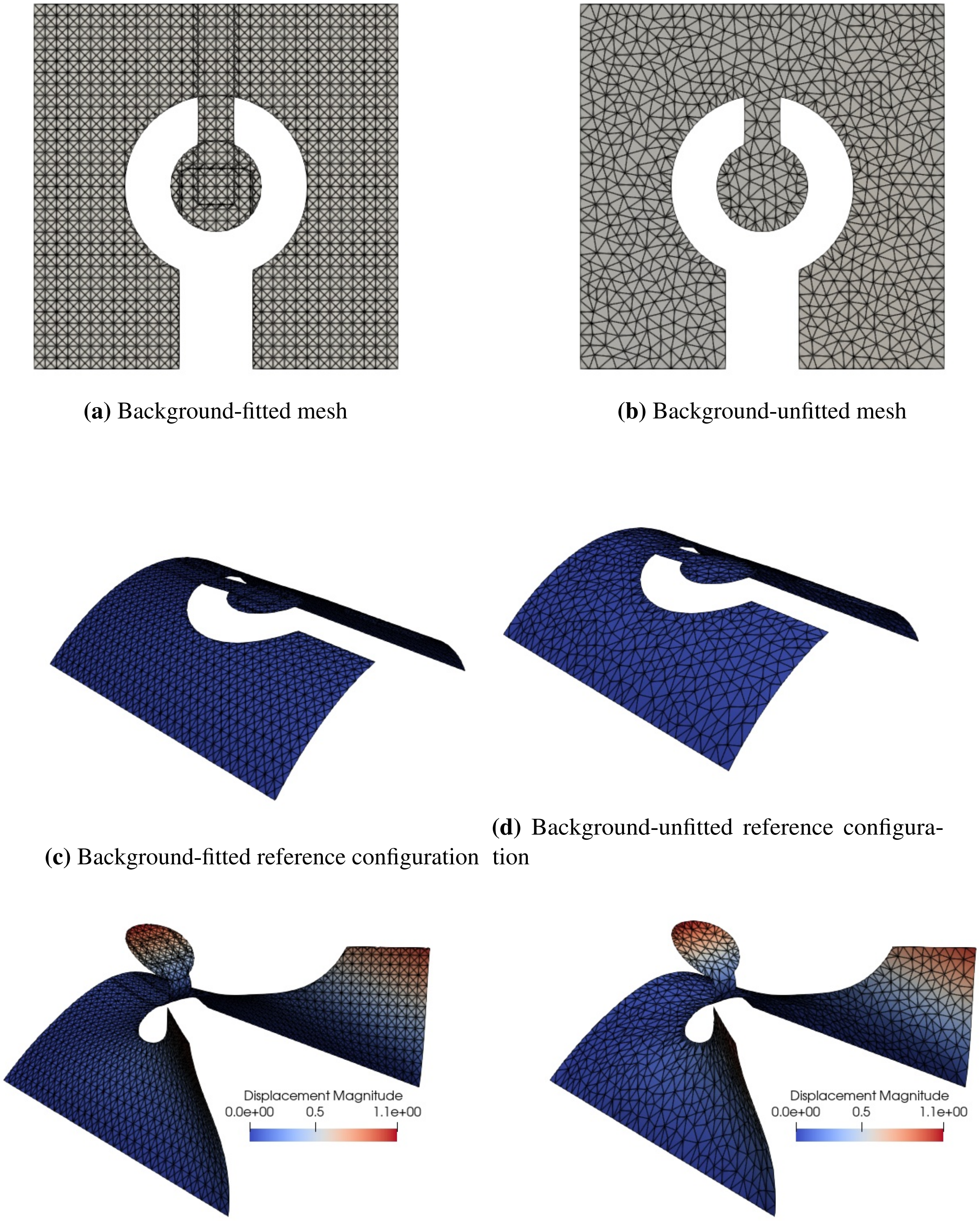
the inner circle) as a quantity of interest. This displacement is shown as a function of number of degrees of freedom for both the background-fitted and -unfitted discretizations in Figure 4.17. Both types of discretizations converge to a consistent value as the number of degrees of freedom increases, where we estimate the converged value of 0.0078 based on a highly-refined background-unfitted discretization.

Chapter summary

This chapter presents a new interpolation-based immersed boundary method for approximating PDEs. This method is implementing within the existing finite element software FEniCS with non-invasive modifications to the linear algebra solvers.

The new method is presented in contrast to a classical quadrature-based immersed boundary method. A PDE is defined upon a computational domain, and basis of approximation functions is created that does not conform to the geometry of the domain. While classical quadrature based method integrates the nonconforming basis functions by shifting the quadrature points to align with the domain geometry, the novel interpolation-based method instead employs a boundary conforming foreground-basis. This foreground basis is defined upon a mesh that does not have the same quality metrics of typical FEA and can be rapidly generated. The background basis is replaced with an interpolated background basis, constructed of a linear combination of foreground basis functions. The weights of this linear combination are organized into what is called the extraction operator. Integration can thus be performed with the boundary conforming foreground basis, eliminating the need to recompute quadrature points.

The interpolation-based immersed boundary method was developed to be implemented within existing finite element software with minimal modifications. Foreground integration meshes are used to define foreground function spaces, and a PDE's variational form is assembled with the foreground space. Prior to the linear solve step, the extraction operator is applied to project the problem into the interpolated background basis. After the linear system is solved, the



(a) Background-fitted mesh

(b) Background-unfitted mesh

(c) Background-fitted reference configuration

(d) Background-unfitted reference configuration

(e) Background-fitted deformed configuration

(f) Background-unfitted deformed configuration

Figure 4.16. Background-fitted and -unfitted meshes (in B-spline parameter space) and reference and deformed configurations (in physical space) of a trimmed shell structure subjected to a follower load.

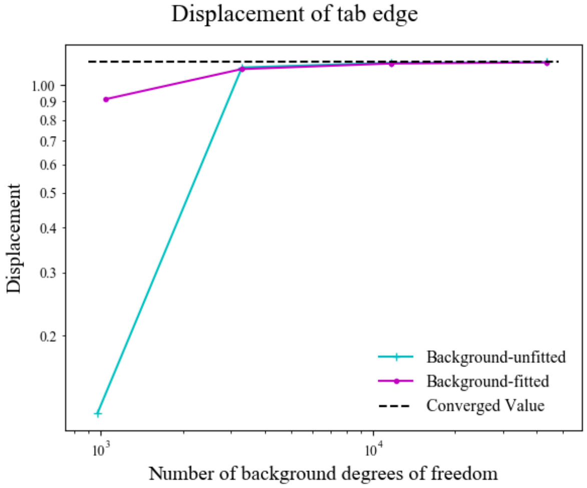


Figure 4.17. Convergence of background-fitted and -unfitted analyses of a trimmed shell subject to a follower load.

solution is projected back onto the foreground space to utilize the FEA software's post-processing and visualization capabilities. This workflow is demonstrated in this chapter with the open-source FEA software FEniCS.

Several benchmark problems validated this method. The interpolation-based immersed boundary method was compared to both a classical quadrature-based immersed-boundary method and classical FEA on a series of poorly conditioned foreground meshes. The method was extended to employ non-conforming foreground function spaces to model the biharmonic problem. Approximate geometries were modeled with additional foreground refinement to reduce geometric error to below method error, employing linear elasticity. The Navier–Stokes equations demonstrated the methods ability to tackle nonlineat and unsteady problems. With each problem, the method demonstrated optimal error convergence rates.

Lastly, the methods application to trimmed shell analysis is demonstrated. A benchmarking problem using an inflated plate is used to verify the numerical behavior with comparison to classical IGA. A trimmed shell geometry of a parabolic plate with tab is subjected to a uniform follower load and the method was tested for both background-fitted and background-unfitted foreground meshes.

Acknowledgements

The above chapter, in full, is a reprint of the material as it appears in Fromm, Jennifer E; Wunsch, Nils; Maute, Kurt; Evans, John A; and Kamensky, David "Interpolation-based immersed finite element and isogeometric analysis," Computer Methods in Applied Mechanics and Engineering, 2023. The dissertation author was the primary researcher and author of this paper. Notation has been changed to maintain consistency within the dissertation.

The dissertation author was supported by National Science Foundation award number 1650112. R. Xiang and D. Kamensky were partially supported by National Science Foundation award number 2103939. N. Wunsch, K. Maute, and J. A. Evans were supported by National Science Foundation award number 2104106. H. Zhao was partially supported by National Aeronautics and Space Administration award number 80NSSC21M0070.

Chapter 5

Interpolation-based immersogeometric analysis methods for multi-material and multi-physics problems

The previous chapter of this examined interpolation-based immersed boundary methods using a single uniform background basis for single material and single physics problems. This chapter expands the applications of this method to multi-material and multi-physics problems with the following contributions:

- Discontinuous state variable fields are approximated by background bases with Heaviside enrichment applied at material interfaces.
- Material interfaces are described by level sets, and hierarchical refinement allows for local refinement.
- Local foreground refinement is applied independently of background refinement, improving geometric approximation of material interfaces without increasing the number of system degrees of freedom.
- Separate background bases are used to approximate different fields in multi-physics applications, which are interpolated using a single foreground basis, allowing for easy coupling.

Furthermore, the following numerical examples are implemented in the next generation open-source software library FEniCSx [15], with code available at [68]. Multi-material heat conduction and linear elasticity are modeled to demonstrate the convergence rates of the proposed workflow. A coupled thermo-mechanical problem illustrates the combined multi-material and multi-physics capabilities of the interpolation-based immersed boundary framework.

The outline of this chapter is as follows: Section 5.1 describes this method's treatment of the geometric description of material interfaces and the Heaviside enrichment of solution spaces with discontinuities at material interfaces. Section 5.2 details the novel interpolation-based immersed boundary method's application to multi-material and multi-physics problems and its implementation workflow within existing FEM codes. Section 5.3 provides numerical results validating and expanding upon this method.

5.1 Immersed material interfaces

Workflows to solve multi-material PDEs require functionalities to both describe the geometry of material interfaces and to represent the associated discontinuities in the state variable fields. In this chapter, level set functions (LSFs) are utilized to implicitly describe the geometry of material interfaces, and a generalized Heaviside enrichment strategy in conjunction with a set of interface terms is employed to represent the required discontinuities at material interfaces.

5.1.1 Representing interface geometry through level set functions

The level set method, developed in [132], has been used to describe interfaces in the extended finite element method (XFEM) [124, 25] and extended isogeometric analysis (XIGA) [129, 146]. Following these works, the domain geometry is implicitly represented using LSFs $\phi_i(\boldsymbol{x})$. An iso-level ϕ_t of the LSF describes the interface $\Gamma_\pm$ between two subdomains Ω_+ and

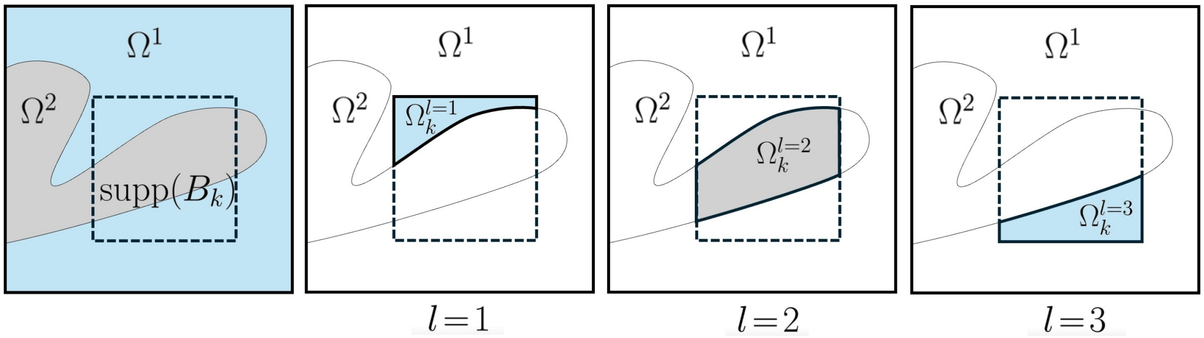


Figure 5.1. The function-wise enrichment strategy considers the material connectivity within a basis functions support. The function B_k spans two material subdomains, denoted Ω^1 and Ω^2, forming three disconnected material subregions $\Omega_k^{l=1}$, $\Omega_k^{l=2}$, and $\Omega_k^{l=3}$.

Ω_- such that

$$\begin{aligned} &\phi(\boldsymbol{x}) < \phi_t,\ \boldsymbol{x} \in \Omega_+, \\ &\phi(\boldsymbol{x}) > \phi_t,\ \boldsymbol{x} \in \Omega_-, \\ &\phi(\boldsymbol{x}) = \phi_t,\ \boldsymbol{x} \in \Gamma_\pm. \end{aligned} \tag{5.1}$$

With n_L LSFs, this method can represent up to 2^{n_L} subdomains. Materials are then associated with these subdomains using a multi-phase level set model as in [160], where phases are identified by phase indices $\mathbf{P}$. Phase indices $\mathbf{P}$ are assigned with characteristic functions f_i,

$$f_i(\boldsymbol{x}) = \begin{cases} 0, & \phi_i(\boldsymbol{x}) < \phi_t, \\ 1, & \phi_i(\boldsymbol{x}) \geq \phi_t, \end{cases} \tag{5.2}$$

such that

$$\mathbf{P}(\boldsymbol{x}) = \sum_{j=1}^{n} 2^{j-1} f_j(\boldsymbol{x}). \tag{5.3}$$

Phases are then mapped onto material subregions.

In this work LSFs are discretized using linear basis functions from a THB-spline basis

$B_k \in \mathscr{T}$,

$$\phi_i^h(\boldsymbol{x}) = \sum_k B_k(\boldsymbol{x}) \phi_i^k, \tag{5.4}$$

where ϕ_i^k are the coefficients associated with LSF ϕ_i. The LSF are linearly interpolated such that the coefficients are the nodal values $\phi_i^k \equiv \phi_i(\boldsymbol{x}_k)$. This discretization is used to construct material characteristic functions and to enrich background basis functions.

5.1.2 Heaviside enrichment of basis functions at material interfaces

Heaviside enrichment have been widely used in PUM [5], GFEM [154], and XFEM [28] as a means to represent strong discontinuities within elements. The enrichment strategies presented in most existing literature, such as in [158, 80], add enriched basis functions for each material domain.

While effective, these global enrichment strategies can lead to artificial numerical stiffening around small geometric features. This stiffening is caused by interpolation of a state variable field in locally disconnected domains of the same material by the same basis function as shown in Fig. 5.1. The high-order, higher-continuity B-spline basis functions with large supports employed in this work, alongside the complex material layouts presented in Section 5.4, would exacerbate local stiffening effects and lead to an increased solution error. Typically, h-refinement is used to avoid locally disconnected same-material domains within the region of support of any given basis function, increasing overall system size. This work instead adopts the enrichment strategy presented by [129] which considers the material connectivity in the individual basis functions' supports.

As shown in Figure 5.1, for a given function B_k with support $\text{supp}(B_k)$, the phase IDs, defined in Eq. (5.3), are used to identify the L_k distinct but connected material subregions Ω_k^m, such that $\text{supp}(B_k) = \cup_{m=1}^{L_k} \Omega_k^m$. For L_k distinct subregions, the basis function B_k requires L_k enrichment levels.

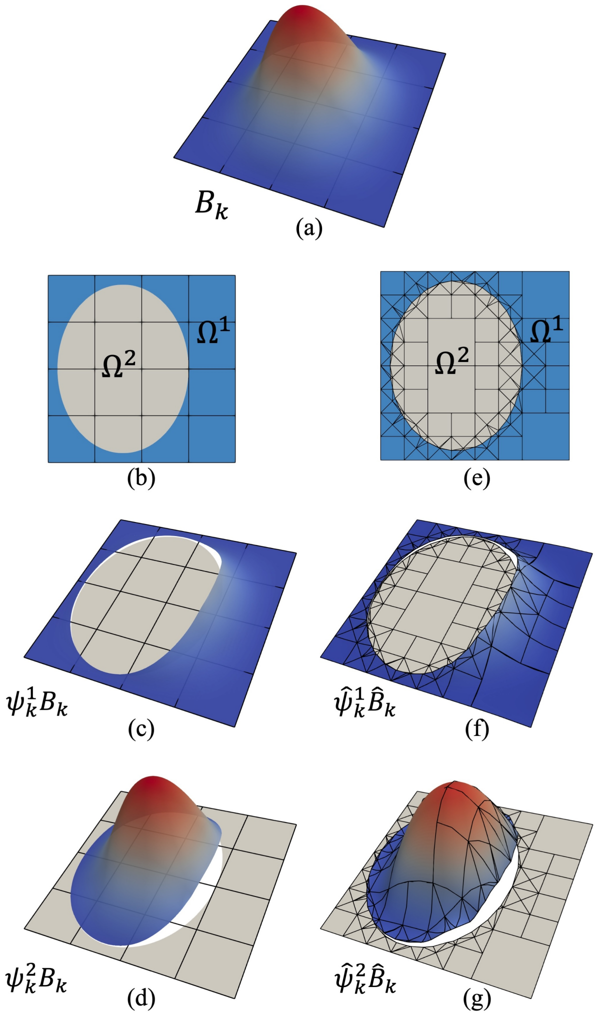


Figure 5.2. A B-spline basis is defined on a structured background mesh and an example function B_k is depicted in (a). Using the geometry description of the material subdomains in (b), Heaviside enrichment is applied to form the discontinuous functions $\psi_k^1 B_k$ and $\psi_k^2 B_k$, depicted in (c) and (d) respectively. A Lagrange foreground function space is defined on the boundary-fitted mesh in (e). The function space is used to interpolate the enriched background functions $\hat{\psi}_k^1 \hat{B}_k$ and $\hat{\psi}_k^2 \hat{B}_k$, depicted in (f) and (g), respectively.

This enrichment is then achieved through characteristic functions ψ_k^m,

$$\psi_k^m(\boldsymbol{x}) = \begin{cases} 1, \text{ if } \boldsymbol{x} \in \Omega_k^m \\ 0, \text{ else,} \end{cases} \tag{5.5}$$

such that the enriched basis functions can be expressed as

$$B_k^m(\boldsymbol{x}) = \psi_k^m(\boldsymbol{x}) B_k(\boldsymbol{x}), \;\; \forall\, m \in \{1, ..., L_k\}. \tag{5.6}$$

The enriched basis functions constructed from a single non-enriched basis function, are shown in Figure 5.2 for a two-material configuration. The bi-quadratic B-spline B_k depicted in Figure 5.2(a) is enriched assigning one material to the inside and one material to the outside of the ellipse shown in Figure 5.2(b). The basis function B_k is split into two enriched functions, $B_k^1(\boldsymbol{x}) = \psi_1^m(\boldsymbol{x}) B_k(\boldsymbol{x})$ in Figure 5.2(c) and $B_k^2(\boldsymbol{x}) = \psi_k^2(\boldsymbol{x}) B_k(\boldsymbol{x})$ in Figure 5.2(d) allowing for the representation of strong discontinuities at the material interface. Interface conditions are enforced weakly; for example C^0 continuity can be enforced at the interface using Nitsche's method [2].

5.2 The interpolation-based immersed boundary method for multi-material PDEs

One of the core challenges associated with classic immersed methods is the construction of custom quadrature rules for the various material regions within each intersected background element. Numerous solutions exist and have been used in custom research codes, such as octree refinement [58, 143, 101], interface reconstruction and tessellation [121, 46, 67], moment-fitting [125, 155], and quadrature schemes using generalized Stokes theorem [151, 76, 77]. However, the generation of such quadrature rules can generally not be implemented within existing finite element software without major changes to the software itself.

The interpolation-based immersed approach is instead designed to utilize the integration subroutines of existing FEM codes. To this end, a boundary-fitted foreground mesh is constructed with only minimal requirements on mesh quality. Element formation is then performed on the poor quality boundary-fitted mesh using existing standard finite element routines. Using Lagrange extraction operators [144] the resulting tangent matrix and force vector are projected into the enriched THB-spline space. This can be done either on an elemental level during assembly, or globally afterwards. The resulting final problem uses an approximation of the enriched function space of the background mesh which is interpolated by the basis functions of the foreground mesh.

The following Subsection 5.2.1 will first introduce the thermo-elastic model problem. Subsection 5.2.2 will provide an overview of the interpolation-based approach specific to the multi-material, multi-physics problems presented in this paper. For a more general and comprehensive introduction to the approach, we refer the reader to the authors' previous work on the topic [71]. The generation of the boundary-fitted foreground mesh is discussed in Subsection 5.2.3.

5.2.1 Multi-material and multi-physics model problem

To illustrate the application of interpolation-based immersed boundary methods to multi-material and multi-physics problems, a thermo-elastic problem is introduced. This problem can be broken into a thermal subproblem and a structural subproblem.

Let a domain of interest Ω with closure denoted $\overline{\Omega}$ be composed of N material subdomains $\mathscr{M} = \{1, \cdots, N\}$

$$\overline{\Omega} = \bigcup_{m \in \mathscr{M}} \overline{\Omega}^m \quad \subset \mathbb{R}^{d_p}. \tag{5.7}$$

A different thermal conductivity κ^m may be associated with each material m

$$\kappa(\boldsymbol{x}) = \kappa^m, \;\; \boldsymbol{x} \in \Omega^m, \tag{5.8}$$

for $m \in \mathscr{M}$. A source term $f : \Omega \to \mathbb{R}$, a boundary heat flux term $\overline{q} : \partial\overline{\Omega} \to \mathbb{R}$ on $\Gamma_{\overline{q}} \subset \partial\overline{\Omega}$, and Dirichlet boundary data $\overline{T} : \partial\overline{\Omega} \to \mathbb{R}$ on $\Gamma_{\overline{T}} \subset \partial\overline{\Omega}$ are ascribed. The strong form for the thermal problem then reads as: Find $T : \Omega \to \mathbb{R}$ such that $\forall\; m \in \mathscr{M}$

$$\begin{aligned}
-\nabla \cdot (\kappa(\boldsymbol{x})\nabla T) = f \quad & \text{in } \Omega^m, \\
[\![T]\!] = 0 \quad & \text{on all } \Gamma_{km}, \\
[\![\boldsymbol{q}]\!] = 0 \quad & \text{on all } \Gamma_{km}, \\
-\kappa(\boldsymbol{x})\nabla T \cdot \boldsymbol{n} = \overline{q} \quad & \text{on } \Gamma^m_{\overline{q}}, \\
T = \overline{T} \quad & \text{on } \Gamma^m_{\overline{T}},
\end{aligned} \tag{5.9}$$

where $\Gamma_{km} = \overline{\Omega}^k \cap \overline{\Omega}^m \neq \emptyset$, with $k \in \mathscr{M}$ and $k \neq m$, are the material interfaces, and $[\![\cdot]\!] = (\cdot)^k - (\cdot)^m$ is the jump of a given quantity over an interface Γ_{km}. The material fields are defined $T^m = T(\boldsymbol{x}), \boldsymbol{x} \in \Omega^m$, and $\boldsymbol{q}^m = -\kappa^m \nabla T^m$. The domains $\Gamma^m_{\overline{q}} = \Gamma_{\overline{q}} \cup \partial\overline{\Omega}^m$, and $\Gamma^m_{\overline{T}} = \Gamma_{\overline{T}} \cup \partial\overline{\Omega}^m$ are the intersections of the domain boundaries with the material subdomain boundaries. $\boldsymbol{n}$ denotes the surface normal.

The domain of interest $\overline{\Omega}$ is embedded into a hierarchically refined background mesh $\mathscr{K}_T$, generated using the sequence of refined meshes $\mathscr{K}^l$ and subdomains Ω^l_T.[1] Note that the largest subdomain Ω^0_T must be chosen such that the closure of the domain of interest is a subset, $\overline{\Omega} \subset \Omega^0_T$. The mesh $\mathscr{K}_T$ is associated with the enriched THB-spline basis $\mathscr{T}_T = \{B^T_i\}$. The

[1] Here the subscript $(\cdot)_T$ refers to entities associated with the thermal subproblem. The subscript $(\cdot)_u$ will refer to entities associated with the structural subproblem. T and u are used as superscripts when referring to entities that require subscripts for indices, such as B-spline basis functions (B^T_i and B^u_i).

temperature field is discretized using the function space

$$\mathscr{V}_T^h = \text{span}\{B_i^T \mid \text{supp}(B_i^T) \cap \overline{\Omega} \neq \emptyset\}. \tag{5.10}$$

The discrete form can then be defined as: Find $T^h \in \mathscr{V}_T^h$ such that $\forall\ \theta^h \in \mathscr{V}_T^h$,

$$\begin{aligned}\sum_{m=1}^{n} \left[\int_{\Omega^m} \kappa \nabla T^h \cdot \nabla \theta^h d\Omega \right] - \int_{\Omega} f \theta^h d\Omega - \int_{\Gamma_{\overline{q}}} \overline{q} \theta^h d\Gamma \\ = \mathscr{R}_T^D + \mathscr{R}_T^I\ ,\end{aligned} \tag{5.11}$$

where $\mathscr{R}_T^D$ and $\mathscr{R}_T^I$ are Dirichlet and interface residual terms. The temperature Dirichlet residual is the result of Nitsche's method [127] enforcement of the Dirichlet boundary condition,

$$\begin{aligned}\mathscr{R}_T^D = \sum_{m=1}^{n} \Bigg[&\mp \int_{\Gamma_{\overline{T}}^m} \kappa (T^h - \overline{T})(\nabla \theta^h \cdot \boldsymbol{n})\, d\Gamma \\ &- \int_{\Gamma_{\overline{T}}^m} \kappa \theta^h (\nabla T^h \cdot \boldsymbol{n})\, d\Gamma \\ &+ \int_{\Gamma_{\overline{T}}^m} \frac{\beta_T^D \kappa}{h} (T^h - \overline{T}) \theta^h \, d\Gamma \Bigg],\end{aligned} \tag{5.12}$$

where $\beta_T^D \geq 0$ is a user defined constant. The first integral of Eq. (5.12) will be negative for the symmetric version of Nitsche's method (which is employed in numerical examples in this work) or positive for the non-symmetric version. h is taken as the characteristic element size on the foreground mesh, differing from the usual implementation where h is the element size on the background mesh.

The temperature interface conditions, lines 2 and 3 of Eq. (5.9), are also enforced through a Nitsche-like method, resulting in the temperature interface residual

$$
\begin{aligned}
\mathscr{R}_T^I = \sum_{i=1}^{n} \sum_{j=i+1}^{n} \Bigg[& - \int_{\Gamma_{ij}} [\![T^h]\!] \{ \kappa \nabla \theta^h \} \cdot \boldsymbol{n}) \, d\Gamma \\
& - \int_{\Gamma_{ij}} [\![\theta^h]\!] \{ \kappa \nabla T^h \} \cdot \boldsymbol{n}) \, d\Gamma \\
& + \int_{\Gamma_{ij}} \gamma_T^{ij} [\![T^h]\!] [\![\theta^h]\!] \, d\Gamma \Bigg],
\end{aligned} \tag{5.13}
$$

where $\{\cdot\} = w^i (\cdot)^i - w^j (\cdot)^j$ is the weighted average of a given quantity. Motivated by the formulation in [2], these weights are defined as

$$
\begin{aligned}
w^i &= \frac{(h^i)^{d_p} / \omega^i}{(h^i)^{d_p} / \omega^i + (h^j)^{d_p} / \omega^j} \quad \text{and} \\
w^j &= \frac{(h^j)^{d_p} / \omega^j}{(h^i)^{d_p} / \omega^i + (h^j)^{d_p} / \omega^j},
\end{aligned} \tag{5.14}
$$

where h^m is the characteristic size of the foreground element in domain Ω^m bordering the interface facet, ω^m is the characteristic material parameter, which for the thermal subproblem is κ^m, and d_p is the domain dimension. The penalty parameter γ_T^{ij} is defined as

$$
\gamma_T^{ij} = 2 \beta_T^I \frac{(h^i)^{d_p - 1} + (h^j)^{d_p - 1}}{(h^i)^{d_p} / \omega^i + (h^j)^{d_p} / \omega^j}, \tag{5.15}
$$

where $\beta_T^I \geq 0$ is a user specified constant.

The thermal subproblem can be stated compactly as the variational problem: Find $T^h \in \mathscr{V}_T^h$ such that $\forall \ \theta^h \in \mathscr{V}_T^h$

$$
a_T(T^h, \theta^h) = L_T(\theta^h), \tag{5.16}
$$

where $a_T(T, \theta)$ and $L_T(\theta)$ can be computed from Eq. (5.11).

The structural subproblem may utilize a differently refined background discretization. In this case a separate sequence of refined domains Ω_u^l may be selected. The structural subproblem's background mesh $\mathscr{K}_u$ is constructed with this sequence of subdomains and the same sequence of refined tensor-product Cartesian grids $\mathscr{K}^l$. The basis $\mathscr{T}_u = \{B_i^u\}$ associated with the mesh is used for the components of the displacement field $\boldsymbol{u}$. Each displacement component is discretized using the function space

$$\mathscr{V}_u^h = \operatorname{span}\{B_i^u \mid \operatorname{supp}(B_i^u) \cap \Omega \neq \emptyset\}. \tag{5.17}$$

Using a similar derivation (given in full in Appendix 5.A) as applied to the temperature subproblem the mechanical variation problem is compactly written as: Find $\boldsymbol{u}^h \in \boldsymbol{\mathscr{V}}_u^h = [\mathscr{V}_u^h, \mathscr{V}_u^h]$ such that, $\forall \boldsymbol{v}^h \in \boldsymbol{\mathscr{V}}_u^h$

$$a_u(\boldsymbol{u}^h, \boldsymbol{v}^h) = L_u(\boldsymbol{v}^h). \tag{5.18}$$

The two subproblems are coupled through a constitutive model accounting for the thermal expansion by computing the mechanical strain $\boldsymbol{\varepsilon}_m$ as

$$\begin{aligned} \boldsymbol{\varepsilon}_m(\boldsymbol{u}, T) &= \boldsymbol{\varepsilon}_u(\boldsymbol{u}) - \boldsymbol{\varepsilon}_T(T) \\ &= \frac{1}{2}\left(\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^{\mathrm{T}}\right) - \alpha(T - T_0)\boldsymbol{I}, \end{aligned} \tag{5.19}$$

where α is the thermal expansion coefficient and T_0 is the temperature in the reference configuration. $\boldsymbol{I}$ is the identity matrix.

The fully coupled system is then given by the variational problem: Find $(\boldsymbol{u}^h, T^h) \in [\boldsymbol{\mathscr{V}}_u^h, \mathscr{V}_T^h]$ such that, $\forall$ $(\boldsymbol{v}^h, \theta^h) \in [\boldsymbol{\mathscr{V}}_u^h, \mathscr{V}_T^h]$

$$\begin{aligned} a_T(T^h, \theta^h) &= L_T(\theta^h) \text{ and} \\ a_u(\boldsymbol{u}^h, \boldsymbol{v}^h) - b(T^h, \boldsymbol{v}^h) &= L_u(\boldsymbol{v}^h), \end{aligned} \tag{5.20}$$

where the form $b(T, \boldsymbol{v})$ is a result of the coupling conditions and is given by

$$\begin{aligned} b(T, \boldsymbol{v}) = & \sum_{m=1}^{n} \left[\int_{\Omega^m} \boldsymbol{\varepsilon}^u(\boldsymbol{v}) : \boldsymbol{\varepsilon}^T(T) \mathrm{d}\Omega \right] \\ & + \int_{\Gamma_{\overline{u}}} \boldsymbol{C} : \boldsymbol{\varepsilon}^T(T) \cdot \boldsymbol{n} \cdot \boldsymbol{v}^h \mathrm{d}\Gamma \\ & - \sum_{i=1}^{n} \sum_{j=i+1}^{n} \left[\int_{\Gamma_{ij}} [\![\boldsymbol{v}]\!] \cdot \{\boldsymbol{C} : \boldsymbol{\varepsilon}^T(T)\} \cdot \boldsymbol{n} d\Gamma \right]. \end{aligned} \tag{5.21}$$

5.2.2 Interpolated basis functions

In traditional immersed boundary methods custom quadrature rules would be used to evaluate the integral in the weak form of the coupled problem, given in Eq. (5.20). The main idea of the interpolation-based immersed paradigm is to interpolate the background basis functions using a space of Lagrange functions defined on a foreground mesh, which can be integrated with classical quadrature methods. This workflow thus introduces an interpolated background function space for the thermal subproblem

$$\widehat{\mathscr{V}}_T^h = \operatorname{span}\{\widehat{B}_i^T \mid \operatorname{supp}(\widehat{B}_i^T) \cap \Omega \neq \emptyset\}, \tag{5.22}$$

where the interpolated basis functions are defined as

$$\widehat{B}_i^T := \sum_{j=1}^{\nu} M_{ij}^T N_j \tag{5.23}$$

where

$$M_{ij}^T := B_i^T(\boldsymbol{x}_j) \tag{5.24}$$

is the Lagrange extraction operator. $\{N_j\}_{j=1}^{\nu}$ is the basis of a Lagrange FE space with nodal points $\boldsymbol{x}_j$ such that $N_i(\boldsymbol{x}_j) = \delta_{ij}$. Here ν is the number of foreground basis functions. The same foreground space is used to interpolate the background bases for both the temperature and the displacement state variables.

The approximations of the temperature field becomes

$$T^h = \sum_{i=1}^{n_T} \widehat{B}_i^T d_i^T = \sum_{i=1}^{n_T} \sum_{j=1}^{\nu} M_{ij}^T N_j d_i^T \,, \tag{5.25}$$

where $\{d_i^T\}_{i=1}^{n_T}$ are the unknown coefficients and n_T is the number of basis functions in the temperature field's interpolated background B-spline basis. The displacement is similarly discretized with vector valued function spaces as

$$\boldsymbol{u}^h = \sum_{I=1}^{(d_p \cdot n_u)} \widehat{\boldsymbol{B}}_I^{\boldsymbol{u}} d_I^{\boldsymbol{u}} = \sum_{I=1}^{(d_p \cdot n_u)} \sum_{J=1}^{(d_p \cdot \nu)} M_{IJ}^{\boldsymbol{u}} \boldsymbol{N}_J d_I^{\boldsymbol{u}}. \tag{5.26}$$

Details regarding this discretization are given in Appendix 5.A.

The variational problem in Eq. (5.20) can be assembled using the interpolated bases in Eqs. (5.25) and (5.26) to form the linear system

$$\begin{bmatrix} \boldsymbol{K}^{\theta\theta} & \boldsymbol{0} \\ \boldsymbol{K}^{\theta\nu} & \boldsymbol{K}^{\nu\nu} \end{bmatrix} \begin{bmatrix} \boldsymbol{d}^T \\ \boldsymbol{d}^{\boldsymbol{u}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}^\theta \\ \boldsymbol{f}^\nu \end{bmatrix}, \tag{5.27}$$

where

$$K_{ij}^{\theta\theta} = a_T(\widehat{B}_i^T, \widehat{B}_j^T), \tag{5.28}$$

$$K_{iJ}^{\theta\nu} = b(\widehat{B}_i^T, \widehat{\boldsymbol{B}}_J^{\boldsymbol{u}}), \tag{5.29}$$

$$K_{IJ}^{\nu\nu} = a_{\boldsymbol{u}}(\widehat{\boldsymbol{B}}_I^{\boldsymbol{u}}, \widehat{\boldsymbol{B}}_J^{\boldsymbol{u}}), \tag{5.30}$$

$$f_i^\theta = L_T(\widehat{B}_i^T), \quad \text{and} \tag{5.31}$$

$$f_i^\nu = L_u(\widehat{\boldsymbol{B}}_I^{\boldsymbol{u}}) \tag{5.32}$$

are the matrix entries.

By consolidating the extraction operators for all components in a single matrix

$$\boldsymbol{M} = \begin{bmatrix} \boldsymbol{M}^T & \mathbf{0} \\ \mathbf{0} & \boldsymbol{M}^{\boldsymbol{u}} \end{bmatrix} \tag{5.33}$$

the linear system can be rewritten as

$$\boldsymbol{M}^{\mathrm{T}} \begin{bmatrix} \boldsymbol{A}^{\theta\theta} & \mathbf{0} \\ \boldsymbol{A}^{\theta \boldsymbol{v}} & \boldsymbol{A}^{\boldsymbol{v}\boldsymbol{v}} \end{bmatrix} \boldsymbol{M} \begin{bmatrix} \boldsymbol{d}^T \\ \boldsymbol{d}^{\boldsymbol{u}} \end{bmatrix} = \boldsymbol{M}^{\mathrm{T}} \begin{bmatrix} \boldsymbol{b}^{\theta} \\ \boldsymbol{b}^{\boldsymbol{v}} \end{bmatrix}, \tag{5.34}$$

where

$$A_{ij}^{\theta\theta} = a_T(N_i, N_j), \tag{5.35}$$

$$A_{iJ}^{\theta \boldsymbol{v}} = b(N_i, \boldsymbol{N}_J), \tag{5.36}$$

$$A_{IJ}^{\boldsymbol{v}\boldsymbol{v}} = a_{\boldsymbol{u}}(\boldsymbol{N}_I, \boldsymbol{N}_J), \tag{5.37}$$

$$b_i^{\theta} = L_T(N_i), \quad \text{and} \tag{5.38}$$

$$b_I^{\boldsymbol{v}} = L_u(\boldsymbol{N}_I). \tag{5.39}$$

The quantities $\boldsymbol{A}^{\theta\theta}$, $\boldsymbol{A}^{\theta \boldsymbol{v}}$, $\boldsymbol{A}^{\boldsymbol{v}\boldsymbol{v}}$, $\boldsymbol{b}^{\theta}$, and $\boldsymbol{b}^{\boldsymbol{v}}$ are evaluated and assembled on the boundary-fitted foreground mesh. As the foreground basis is the typical conforming Lagrange polynomial basis, existing commercial or open-source FE software applying standard quadrature rules may be utilized.

The extraction operators can either be computed globally, as suggested in Eq. (5.35), or on an element level for more efficient implementation. However, the latter option requires a modification of the assembly maps in existing FE software. After solving the linear system in Eq. (5.34) with the interpolated basis, the solution is post-processed by projecting the solution for each field onto the foreground basis.

A visual example of the interpolation of an enriched B-spline basis function is given in

Figure 5.2. The boundary-fitted foreground mesh in Figure 5.2(e) is used to define a discontinuous Lagrange polynomial foreground function space. The enriched background functions B_k^1 and B_k^2 are interpolated with this function space, as shown in Figure 5.2(f) and (g).

In this work, B-spline background spaces are utilized to exploit their superior approximation properties as compared to traditional finite element function spaces [61] and to build upon previous work done with enriched THB-splines in XIGA implementations [146]. The interpolation scheme presented here can be utilized with other background basis functions as in the previous work by the authors [71], where both B-splines and classic Lagrange polynomial bases were used. The method can be applied to multi-material problems provided the background basis used is sufficiently enriched at material interfaces.

5.2.3 Foreground mesh generation

Both state variable fields are interpolated using the same Lagrange FE space $\{N_j\}_{j=1}^{v}$ which is constructed on a boundary-fitted foreground mesh. To generate this foreground mesh, the elements of a background mesh intersected by interfaces are decomposed into triangles and tetrahedrons whose facets approximately reconstruct the interfaces.

To ensure the foreground mesh is sufficiently refined, a background mesh for decomposition $\mathscr{K}_D$ is constructed from the series of hierarchically refined meshes $\mathscr{K}^l$,

$$\mathscr{K}_D := \bigcup_{l=0}^{r-1} \{K \in \mathscr{K}^l | K \in \Omega_D^l \text{ and } K \notin \Omega_D^{l+1}\}, \tag{5.40}$$

where $\Omega_D^l \supseteq \Omega_T^l \cup \Omega_u^l$ is a series of subdomains containing the union of the subdomains used for each subproblem's discretization. This ensures that $\mathscr{K}_D$ is at least as refined as the meshes used for each subproblem's discretization and allows for additional refinement of the foreground to improve geometric resolution. The construction of a decomposition mesh $\mathscr{K}_D$ is illustrated in Figure 5.3.

The decomposition mesh $\mathscr{K}_D$ is triangulated by first applying a pre-defined triangulation

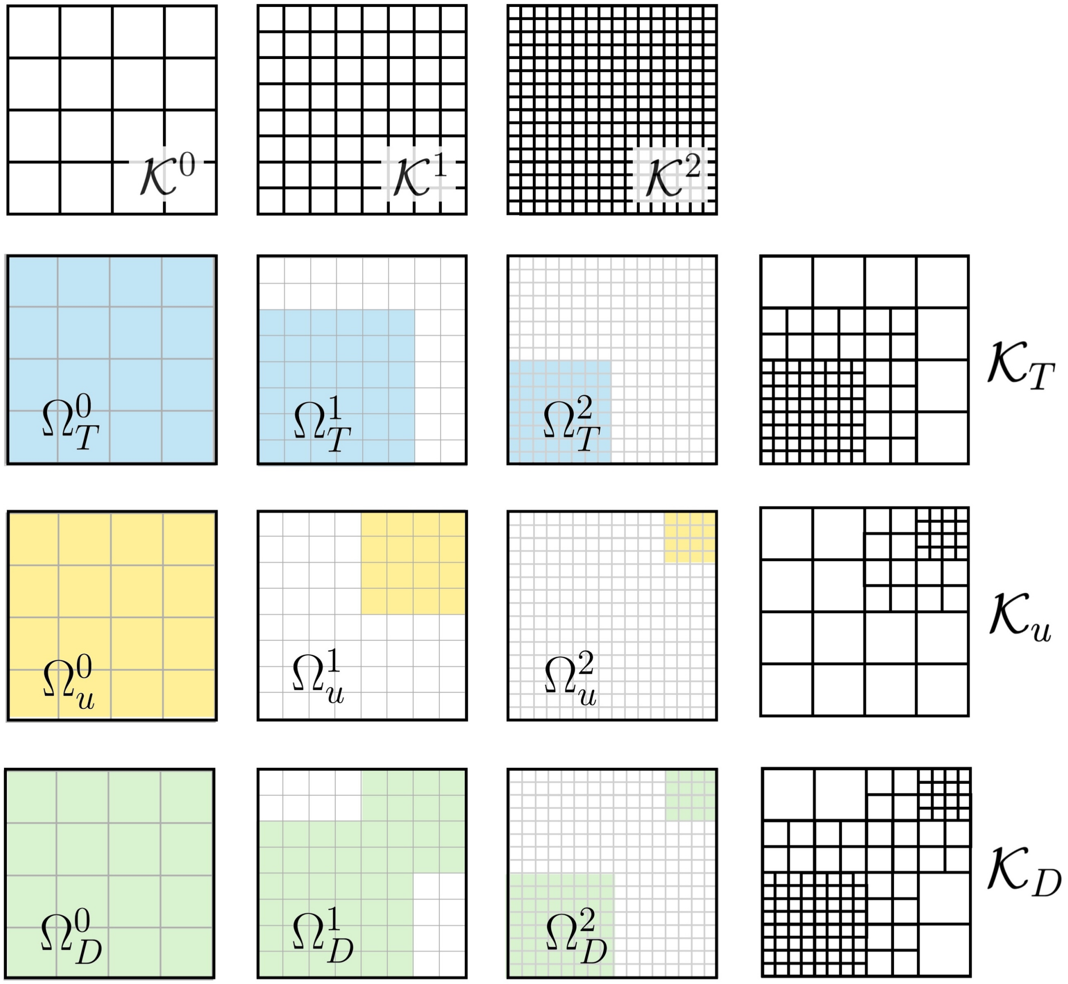


Figure 5.3. A series of uniformly refined meshes $\mathscr{K}^l$ is shown in the top row. Series of nested subdomains, $\Omega_T^{l+1} \subseteq \Omega_T^l$, shown in the second row, and $\Omega_u^{l+1} \subseteq \Omega_u^l$, shown in the third row, are defined for each state variable. For the decomposition mesh the series of subdomains Ω_D^l, shown in the last row, is defined such that the domain on each level l contains the union of the l^{th} level domains for both variable fields. The hierarchically refined meshes $\mathscr{K}_T$, $\mathscr{K}_u$, and $\mathscr{K}_D$, shown in the rightmost column, are constructed with the mesh series $\mathscr{K}^l$ in the top row and their respective subdomain series.

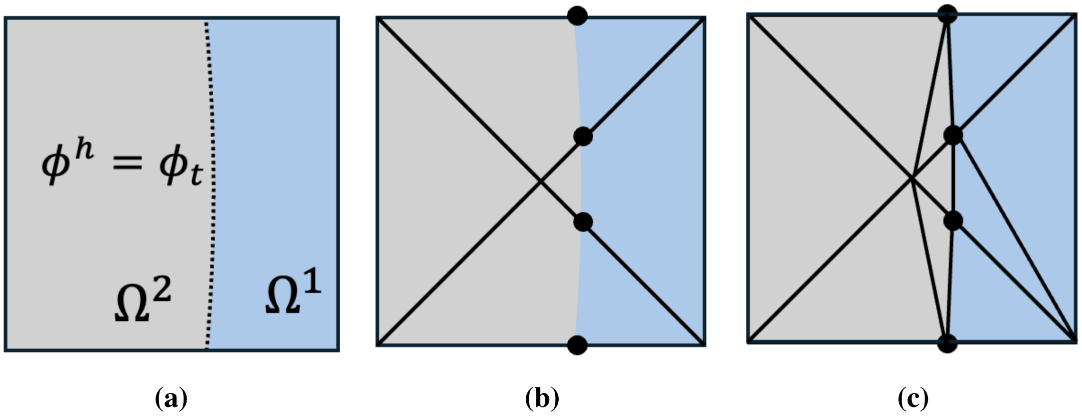


Figure 5.4. Foreground integration meshes are formed by triangulating cut elements of the decomposition mesh $\mathscr{K}_D$. (a) A cell is intersected by the isocontour of the discretized level set function $\phi^h = \phi_t$ defining a material interface. (b) The cell is subdivided into triangular cells and isocontour-edge intersections (indicated by black cicles) are computed. (c) Using the intersection points as nodal points, the cell is further subdivided.

to the intersected background elements, as shown in Figure 5.4 (a-b). This pre-defined triangulation forms 4 triangular elements in 2D domains or 24 hexahedral elements in 3D. Through root finding along elemental edges, the location of the isocontour is found, indicated by the black dots in Figure 5.4 (b). A subdivision template is then applied to each intersected triangle/tetrahedron, as shown in Figure 5.4 (c), to further subdivide the triangles/tetrahedrons into a set of triangles and tetrahedrons whose facets follow the LS isocontour. The last step is repeated recursively for each individual LSF ϕ_i^h which enables sharp geometric corners and edges to be captured where multiple interfaces meet. This approach has been used previously by, e.g., [150]. The resulting approximation of the interface is piecewise linear and depends on the resolution of the decomposition mesh $\mathscr{K}_D$.

Note that the sliver elements and poor aspect ratios in meshes produced by this method are still suitable for the interpolation of the background basis functions which is not bound by the typical mesh quality constraints of traditional FEM [108]. The resulting foreground mesh is of mixed element type (triangles and quadrilaterals in 2D, or hexahedrons and tetrahedrons in 3D) and contains hanging nodes. To accommodate the hanging nodes on the foreground mesh and to adequately interpolate the discontinuous enriched background basis functions described in Subsection 5.1.2, this method employs discontinuous Galerkin type elements for foreground function spaces. As the original THB-spline background basis maintains at minimum C^{p-1} continuity within each material domain, the continuity of the interpolated basis is likewise C^{p-1} continuous where interpolation is exact [144]. This work expands upon previous results using approximate extraction methods [71], where the constraints placed on the foreground Lagrange basis are lessened while numerical accuracy is maintained.

The sliver elements resulting from this procedure do not themselves present problems with the interpolation-based workflow. However, due to the arbitrary location of material interfaces with respect to cell boundaries, it is possible for cells to be cut such that only a small portion of a basis function's support resides inside a given material domain. These small cell cuts result in sparsely supported basis functions, which can present issues with stability and linear conditioning.

Numerous strategies exist to mitigate these issues and were recently reviewed in [53]. Strategies include basis function removal [59], ghost stabilization [37], and basis function agglomeration [8] or extension [39], and have the potential to be implemented within the presented interpolation-based framework. The benchmark problems presented in this work do not require special treatment of sparsely support basis functions and the authors leave the implementation of these stabilization strategies to future work.

5.3 Numerical Results

The accuracy of the proposed method is demonstrated through the study of several benchmark problems. Problems were defined in the Python-based open-source FE code FEniCSx, using foreground meshes and extraction operators generated by the open-source XIGA code MORIS, available at github.com/kkmaute/moris [120]. The source code with which the results were generated is also available at github.com/jefromm/EXHUME_dolfinX [68].

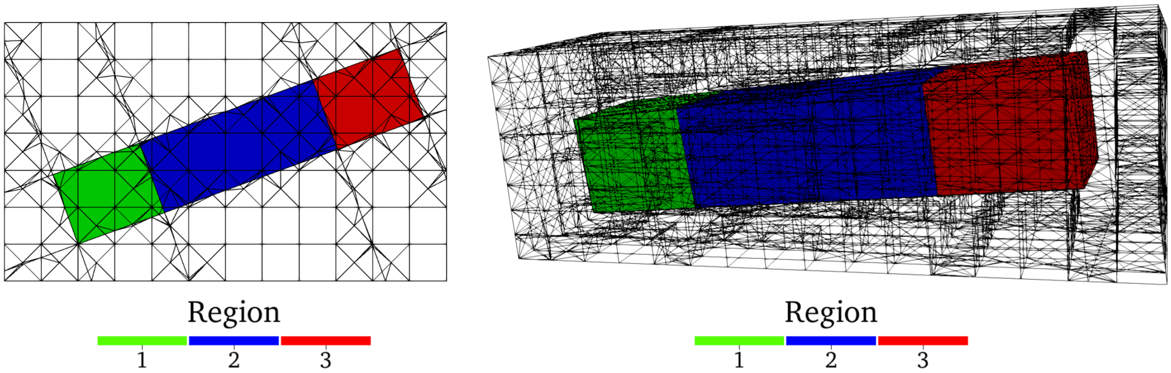


Figure 5.5. The geometric configurations for the 2D (left) and 3D (right) domains. A three-material beam is embedded in a structured background grid and rotated such that material interfaces do not align with element edges (in 2D) or facets (in 3D). Elements intersected by the level set functions defining the beam geometry are triangulated to form a boundary-fitted foreground mesh.

5.3.1 Resolving discontinuities in solution fields through Heaviside enrichment

In this Subsection, a multi-material beam undergoing a spatially varying heat load presents a weakly discontinuous temperature solution field. In this work, weak discontinuities refer to discontinuities in the gradients of solution fields. The solution is approximated with an interpolated Heaviside-enriched background basis, using the interpolation-based immersed boundary workflow. The convergence results from this study demonstrate the accuracy of this method's enrichment scheme for multi-material problems. Beams in both 2D and 3D domains are considered.

The beam is initially defined with corner coordinates $(0,0)$ and (L,H) in 2D, and $(0,0,0)$ and (L,H,H) in 3D, with $L=5$ and $H=1$. The beam is divided into 3 sections, with interfaces at $x=L/4$ and $x=3L/4$. Each section of the beam is assigned a thermal conductivity $\kappa_1=1.0$, $\kappa_2=0.1$, and $\kappa_3=1.0$. To ensure the non-conformity of the material interfaces with respect to the background mesh facets, the 2D beam is rotated about the origin by angle $\phi=20^\circ$, while the 3D beam is rotated about y- and z-axes by angles $\phi_y=-5^\circ$ and $\phi_z=5^\circ$, respectively. The 2D beam is embedded into rectangle with corner coordinates (-1.0, -0.5) and (5,3) and the

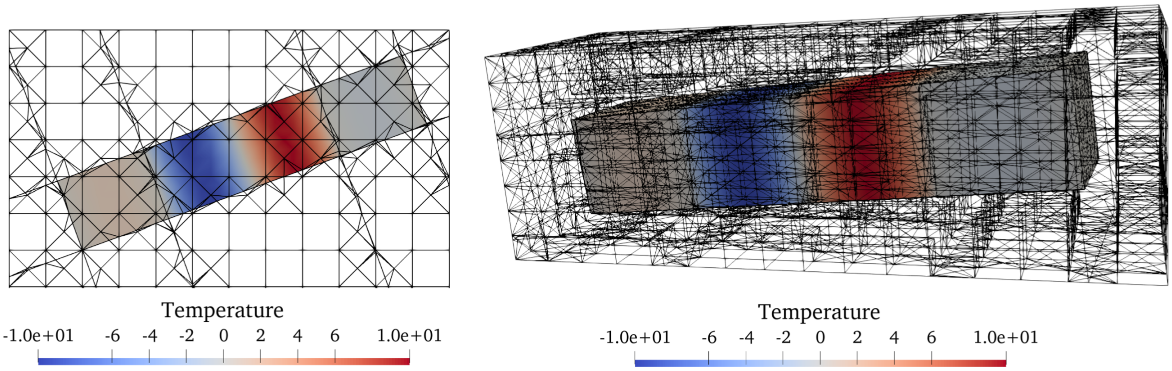


Figure 5.6. The temperature solution field plotted for both domain geometries, using a bi(tri)-quadratic B-spline basis interpolated with a bi(tri)-quadratic foreground Lagrange space. The solution is weakly discontinuous at material interfaces.

3D beam is embedded into a rectangular prism with corner coordinates (-0.5, -0.25,-0.25) and (5.5,1.75,1.75). The geometric configurations are shown in Figure 5.5. Each edge (in 2D) or plane (in 3D) of the beam is defined by a level set function which extends beyond the beam domain in the mesh. The functions are extended through the entire mesh to fully resolve the corners (in 2D) or edges (in 3D) of the beam.

Thermal diffusion is governed by the Poisson equation. The strong and weak forms of this problem are given by Eqs. (5.9) and (5.11) in Subsection 5.2.1. The source term

$$q_B = -\nabla \cdot (\kappa \nabla T_{ex}) \tag{5.41}$$

is constructed from the exact solution

$$T_{ex}(x') = \frac{1}{\kappa} \sin\left(\frac{4\pi}{L} x'\right). \tag{5.42}$$

In 2D the beam-aligned coordinate is expressed in global coordinates as

$$x' = x\cos(\text{-}\phi) - y\sin(\text{-}\phi), \tag{5.43}$$

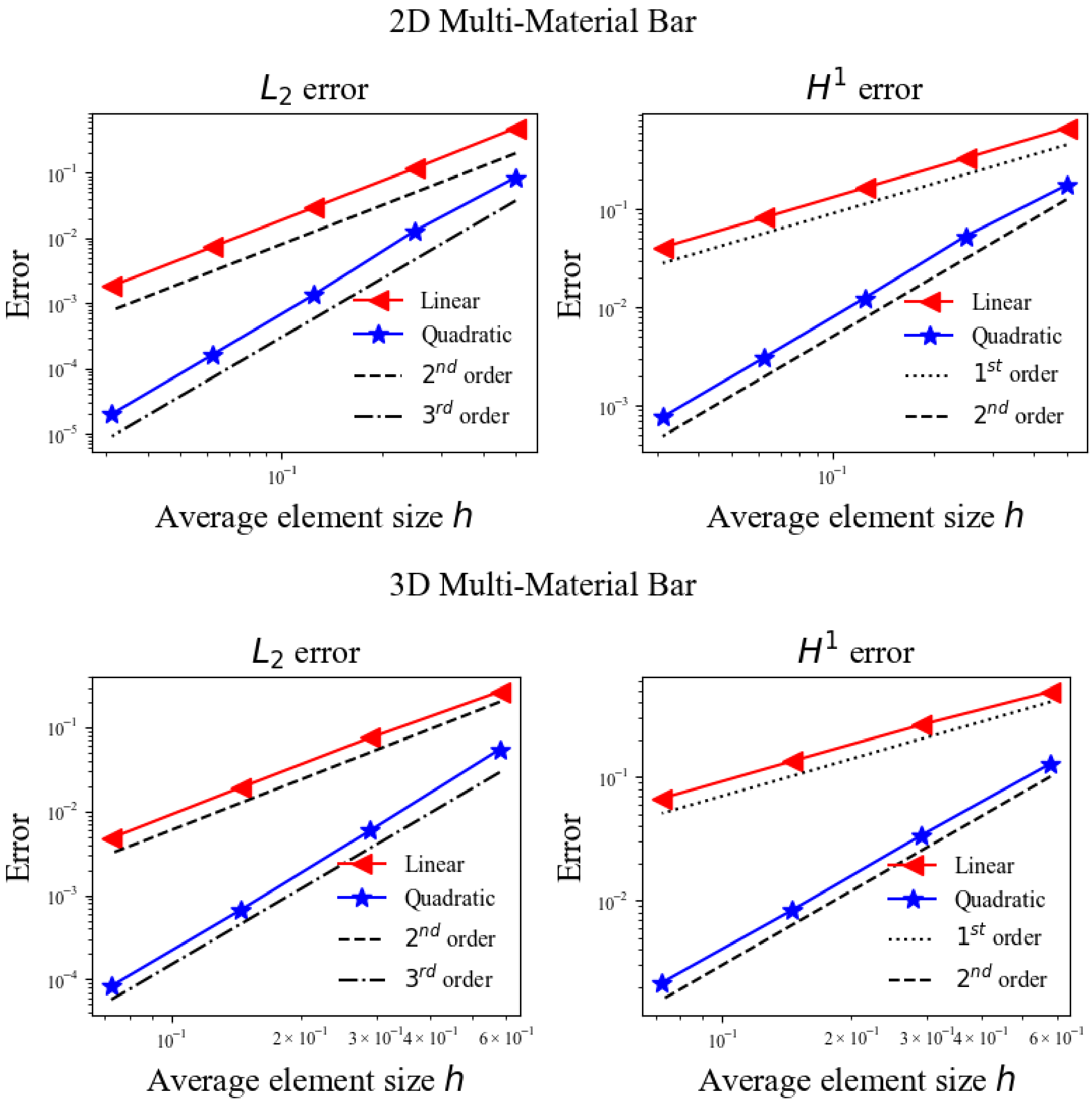


Figure 5.7. Ideal convergence rates are seen for both bi(tri)-linear and quadratic B-spline basis functions, which were interpolated with equal order foreground Lagrange function spaces. The convergence rates are ideal for both the 2D domain (top) and 3D domain (bottom).

and in 3D

$$x' = z\sin(\text{-}\phi_y) + \big(x\cos(\text{-}\phi_z) - y\sin(\text{-}\phi_z)\big)\cos(\text{-}\phi_y), \tag{5.44}$$

where $\boldsymbol{x} = [x,y,z]$ are the mesh coordinates. The exact solution is imposed as Dirichlet boundary data on the ends of the bar, $x' = 0$ and $x' = L$.

For the 2D domains a suite of meshes with average background element sizes $h \in 0.5 \times [1, 0.5, 0.25, 0.125, 0.0625]$ was used. In 3D, the background element sizes $h \in 0.58 \times [1, 0.5, 0.25, 0.125]$ were used. The foreground meshes were constructed with the workflow described in Subsection 5.2.3. Results from the 2D mesh with background element size $h = 0.5$ and from the 3D mesh with background element size $h = 0.29$ using bi(tri)-quadratic Lagrange foreground bases to interpolate bi(tri)-quadratic enriched B-spline bases are shown in Figure 5.6. Convergence rates are plotted in Figure 5.7 for both bi(tri)-linear and quadratic enriched background spline spaces interpolated with equal-order foreground bases. Ideal convergence rates validate the interpolation-based immersed boundary workflow for multi-material problems.

5.3.2 Approximating curved geometries through local foreground refinement

A major challenge in the modeling of multi-material problems is the discretization of material interfaces. In this work, local refinement of the foreground mesh is performed to increase geometric resolution without affecting the number of degrees of freedom in the solution space. In this example the linear elastic behavior of an infinite plate with an embedded circular inclusion of radius $R = 0.5$ is modeled, with local refinement employed to improve the approximation of the inclusion geometry.

The inclusion is comprised of Material 1 with Lamé constants $\lambda_1 = 497.16$ and $\mu_1 = 390.63$, while the exterior plate is made of Material 2 with $\lambda_2 = 656.79$, and $\mu_2 = 338.35$. A uniform isotropic eigenstrain of $\varepsilon_0 = 0.1$ is imposed on the inclusion. The weakly discontinuous

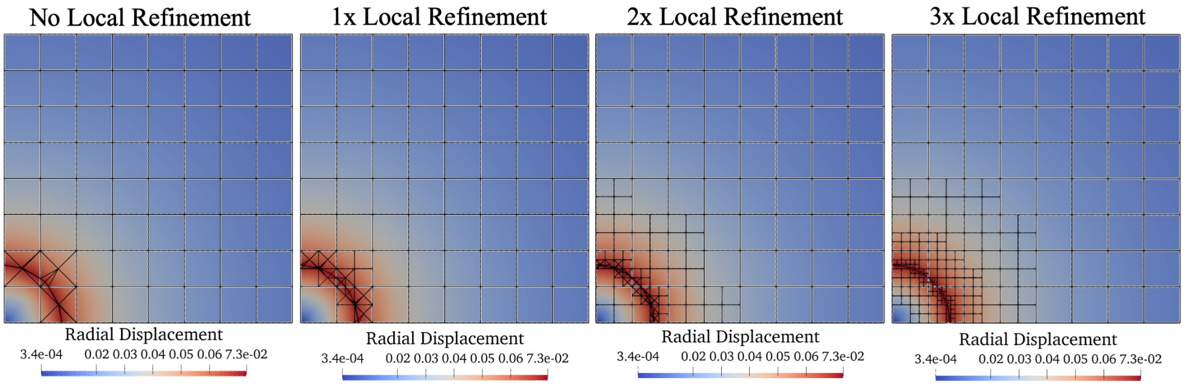


Figure 5.8. Approximated radial displacement of the eigenstrain problem. The background mesh is shown in white, while the foreground integration mesh is overload in black. The background mesh remains the same as local refinement is applied to the foreground for improved geometric resolution. The images correspond to the coarsest refinement level with background element size $h = 0.625$.

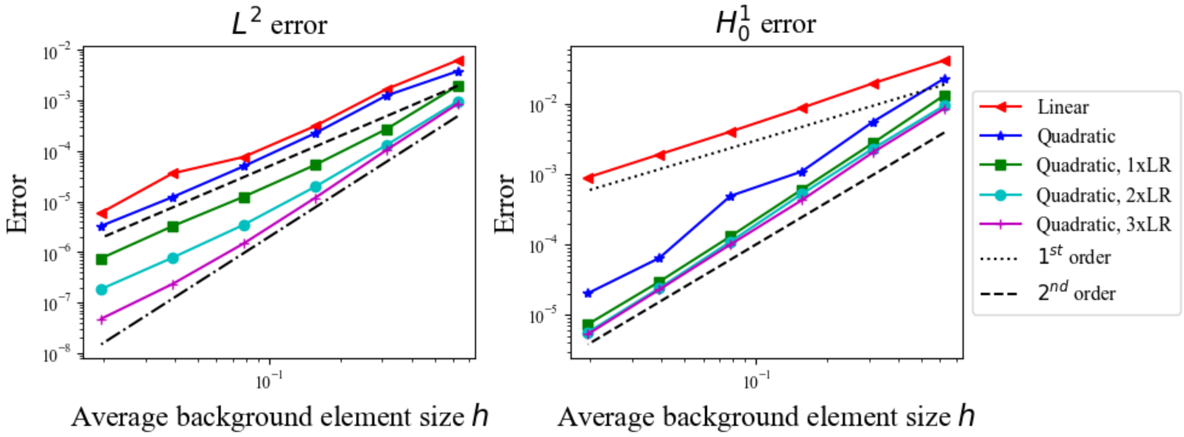


Figure 5.9. Error convergence data for the eigenstrain problem, illustrating the efficacy of foreground refinement. With no foreground refinement, the L^2 convergence rate is limited by the geometric error. With sufficient foreground refinement (3x LR), the convergence rate approaches the ideal of 3.

analytic solution for the radial displacement is given in [161] as

$$u_r = \begin{cases} C_1 r, & r \leq R, \\ C_1 \dfrac{R^2}{r}, & r \geq R, \end{cases} \tag{5.45}$$

where

$$C_1 = \frac{(\lambda_1 + \mu_1)\varepsilon_0}{\lambda_1 + \mu_1 + \mu_2}. \tag{5.46}$$

Exploiting symmetry, only the upper right quadrant of the plate is modeled as shown in Figure 5.8. Symmetry conditions are enforced on the left and bottom edges of the domain, and the exact displacement is prescribed on the right and top edges. The solution domain is a 5×5 square, with a quarter circle at the lower left corner.

The strong and discrete forms of the multi-material linear elasticity PDE are detailed in Appendix 5.A. For this example the mechanical strain is computed by

$$\boldsymbol{\varepsilon}_m(\boldsymbol{u}) = \begin{cases} \boldsymbol{\varepsilon}_u(\boldsymbol{u}) - \varepsilon_0 \boldsymbol{I}, & \boldsymbol{x} \in \Omega_1 \\ \boldsymbol{\varepsilon}_u(\boldsymbol{u}), & \boldsymbol{x} \in \Omega_2 \end{cases} \tag{5.47}$$

where the total strain $\boldsymbol{\varepsilon}_u(\boldsymbol{u}) = \frac{1}{2}(\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^{\mathrm{T}})$.

The domain Ω is immersed into an axis-aligned 5×5 square on which the enriched background B-spline spaces are constructed. A suite of background meshes with characteristic element lengths $h \in 0.625 \times [1, 0.5, 0.25, 0.125, 0.0625, 0.03125]$ are used to generate convergence data.

The foreground meshes used here are locally refined about the material interfaces, as seen in Figure 5.8. The background basis functions remain constant, meaning that there is no increase to the number of solution degrees of freedom with foreground refinement.

The convergence plots in Figure 5.9 show the expected convergence rates for the bi-linear B-spline function space. With the bi-quadratic function space and no foreground refinement, the dominating effects of geometric error are shown with degraded L^2 convergence rates. Geometric error is reduced through local foreground refinement, allowing the function approximation error to dominate the overall convergence. With local foreground refinement the convergence rates approach the ideal rates exhibited by the method for problems without geometric approximation error, as in Subsection 5.3.1.

The foreground meshes generated using this octree local refinement strategy include hanging nodes, which are difficult for most commercial or open-source finite element software to handle. To accommodate these hanging nodes, this interpolation-based workflow employs C^{-1} discontinuous foreground Lagrange polynomial basis functions, . Classical discontinuous Galerkin methods require augmentation of the variational form to enforce continuity at cell interfaces [47], but this augmentation is not done within this workflow.

The results in this work suggest that for a problem where conforming FE methods would require C^0 continuity, a C^{-1} interpolated basis can be employed provided the background basis is at least C^0 continuous. More generally, so long as the background basis is sufficiently continuous it may be interpolated with a less-continuous foreground basis. This attribute of interpolated-based methods was first exploited in [71] to approximate fourth-order PDEs with a background basis of quadratic B-spline bases, which are C^1 continuous, interpolated with a foreground basis of quadratic Lagrange polynomials, which are only C^0 continuous.

5.4 Image-based thermo-mechanical analysis of composite materials, utilizing multiple levels of local refinement

The capability of this method to tackle distinct discretization requirements of state variable fields within a multi-physics problem is illustrated here in a coupled thermo-elastic problem. This problem is posed on an alumina-epoxy composite sample undergoing simultaneous heating and loading conditions. Separate background discretizations are interpolated with a single foreground discretization for the two fields.

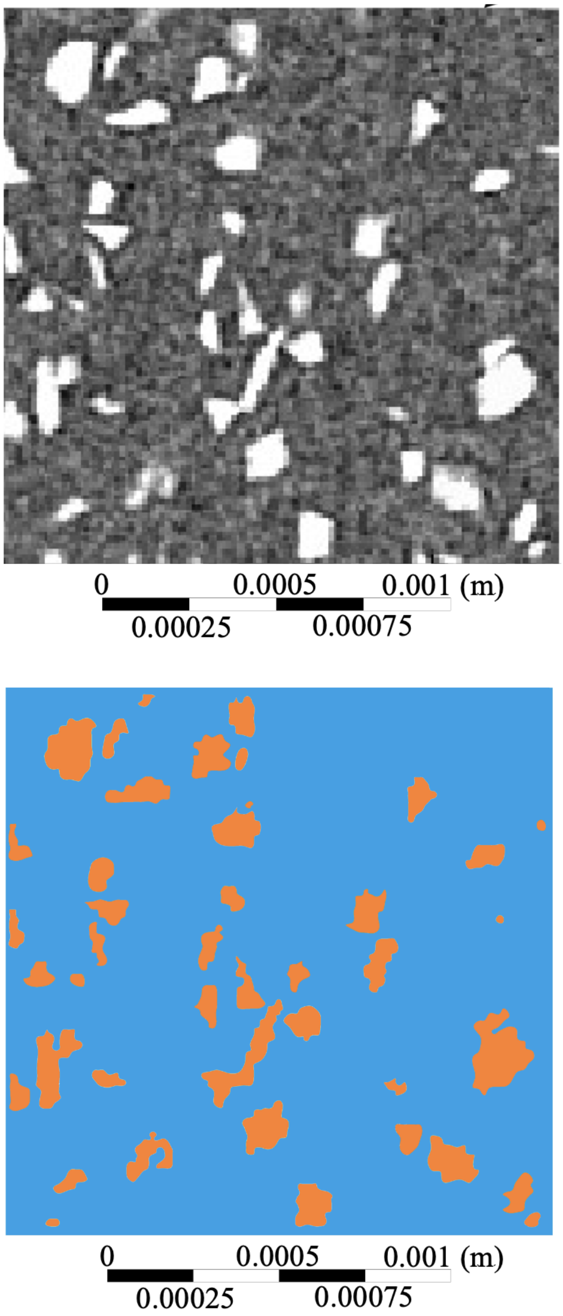
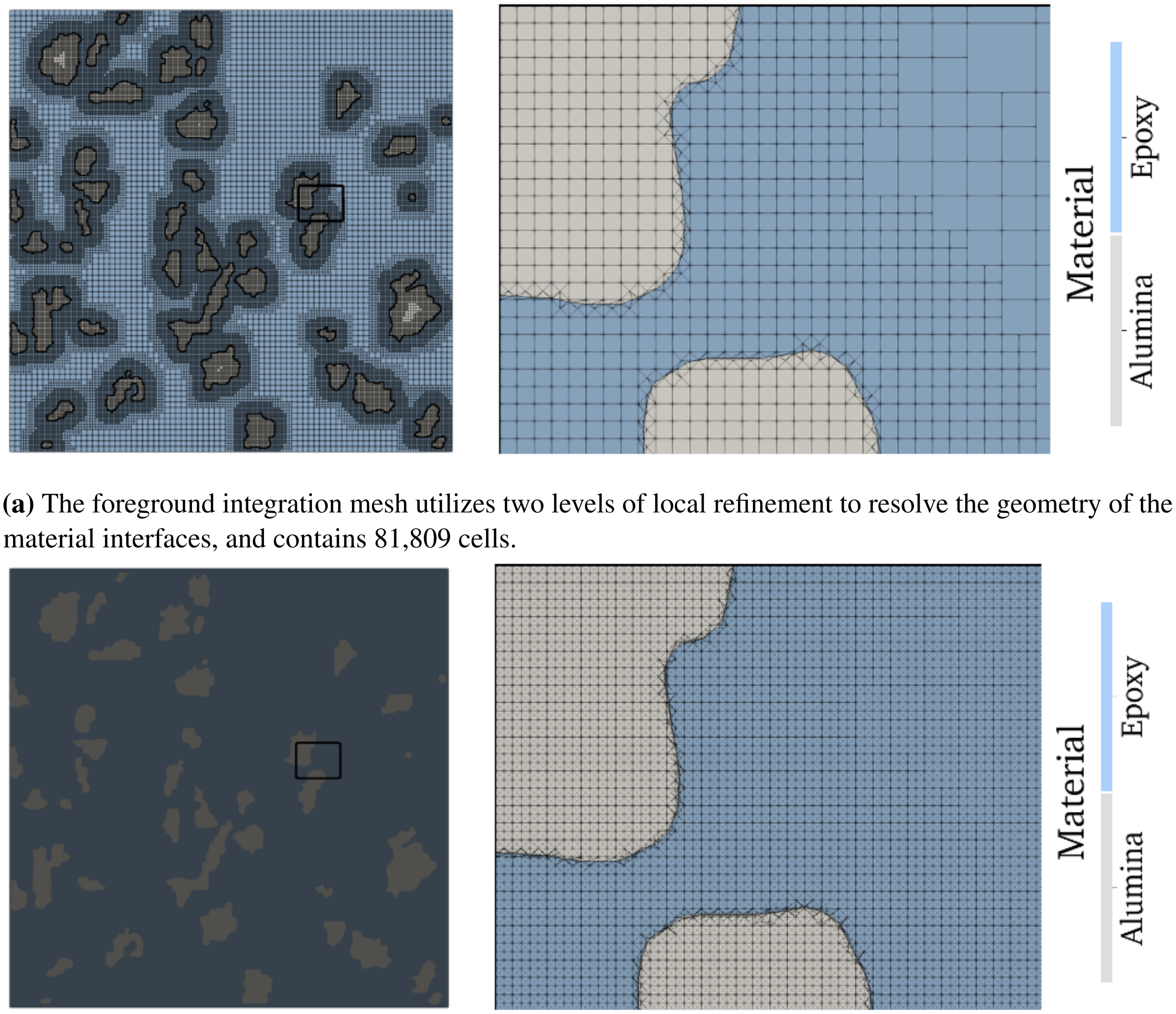


Figure 5.10. Top:Micro-CT image of alumina-epoxy composite, where the white sections signify alumina particles and the grey is the surrounding epoxy. Bottom: Smoothed image used to generate the LSF geometric description



(a) The foreground integration mesh utilizes two levels of local refinement to resolve the geometry of the material interfaces, and contains 81,809 cells.

(b) The mesh used for the classical boundary-fitted FEM comparison was generated with 3 levels of uniform refinement. The mesh elements were uniformly triangulated and modified to improve mesh quality. The mesh contains 1,675,860 cells.

Figure 5.11. The whole domain is shown in the images on the left with the box indicating the region shown in the zoomed in view on the left.

The geometry of the composite sample is taken from a real micro-CT image converted to an implicit level-set description. The micro-CT image,from [162] and shown in Figure 5.10a, is made up of 200×200 pixels, with a pixel size of 8μm, and the specimen is 1.6mm by 1.6mm. The epoxy is represented by the grey background while the alumina particles appear white. The image was then manually processed to generate the smoothed image shown in 5.10b, which was then converted to the implicit level-set description. The following material properties, from [33, 16], were used: Poisson ratios $\nu_{\mathrm{Al}} = 0.23$ and $\nu_{\mathrm{Ep}} = 0.358$, elastic moduli $E_{\mathrm{Al}} = 320e^9$ Pa and $E_{\mathrm{Ep}} = 3.66e^9$ Pa, thermal conductivities $\kappa_{\mathrm{Al}} = 25.0$ W/mK and $\kappa_{\mathrm{Ep}} = 0.14$ W/mK, and thermal expansion coefficients $\alpha_{\mathrm{Al}} = 15e^{-6}$ 1/C^o and $\alpha_{\mathrm{Ep}} = 65e^{-6}$ 1/C^o.

The process described in Subsection 5.2.3 was used to construct a foreground integration mesh, beginning with a uniform 80 element by 80 element axis aligned decomposition mesh. Two levels of local refinement were applied about the material interfaces, and the refined decomposition mesh was triangulated. The resulting foreground mesh, shown in 5.11a, contains 81,809 cells.

For comparison purposes, a similar workflow was used to generate a boundary-fitted mesh for use in classical FEM. Classical FEM with FEniCSx requires a single element type mesh without hanging nodes. To avoid hanging nodes and to sufficiently resolve the gradients of the state variable fields the decomposition mesh was uniformly refined three times forming a 640 element by 640 element square mesh. The cut cells were triangulated to create a boundary-fitted mesh, and then the remaining quadrilateral elements were triangulated. The mesh was then modified with the software package Coreform Cubit 2023.11 to improve mesh quality metrics. The resulting mesh, shown in Figure 5.11b, contains 1,675,860 cells. The method described here was used to ensure the level set descriptions of the material interfaces matched between this mesh and the foreground mesh used with the interpolation-based immersed boundary method.

With this sample, a heated compression-shear test was simulated. The top and bottom displacements were imposed as $\boldsymbol{u}_{top} = [-0.01, -0.01]$mm and $\boldsymbol{u}_{bottom} = [0,0]$mm. The temperature at the top and bottom edges were specified as $T_{top} = 0^o$C and $T_{bottom} = 100^o$C. The

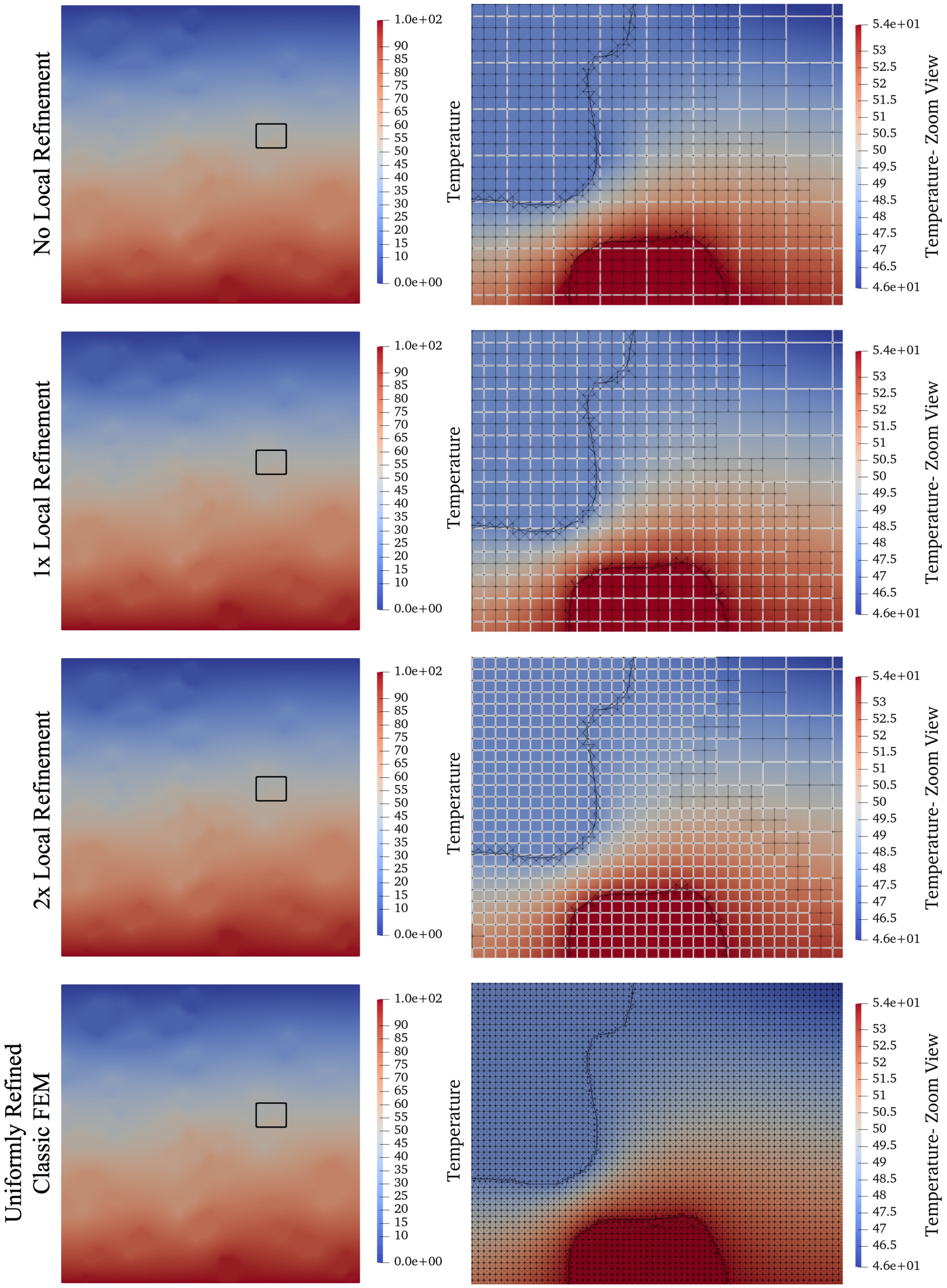


Figure 5.12. The temperature field results are compared for four different mesh configurations. The left shows the entire domain with the box indicating the region shown in the zoomed in view on the right.

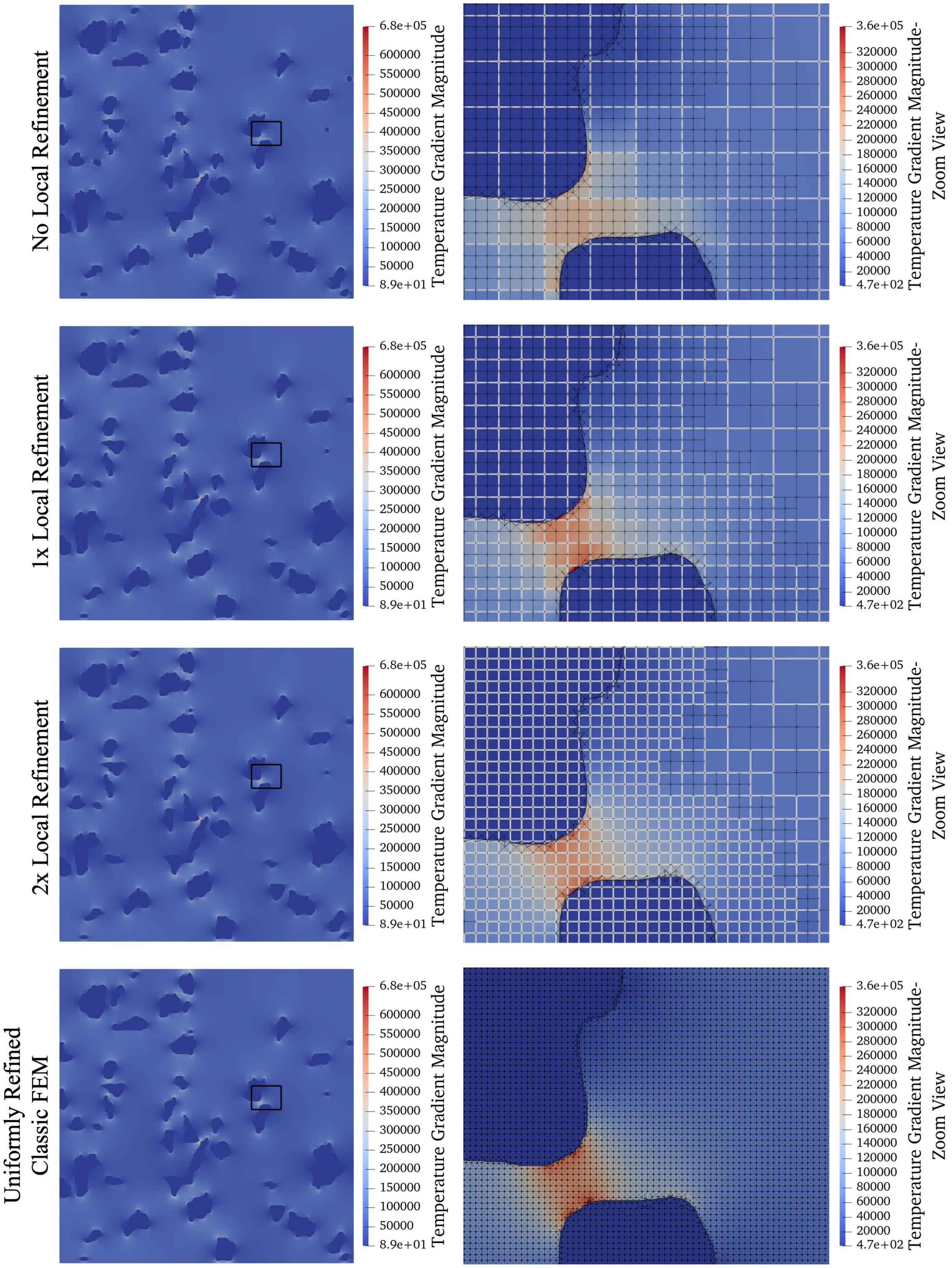


Figure 5.13. The the temperature gradient magnitude field is compared for four different mesh configurations. The left shows the entire domain with the box indicating the region shown in the zoomed in view on the right.

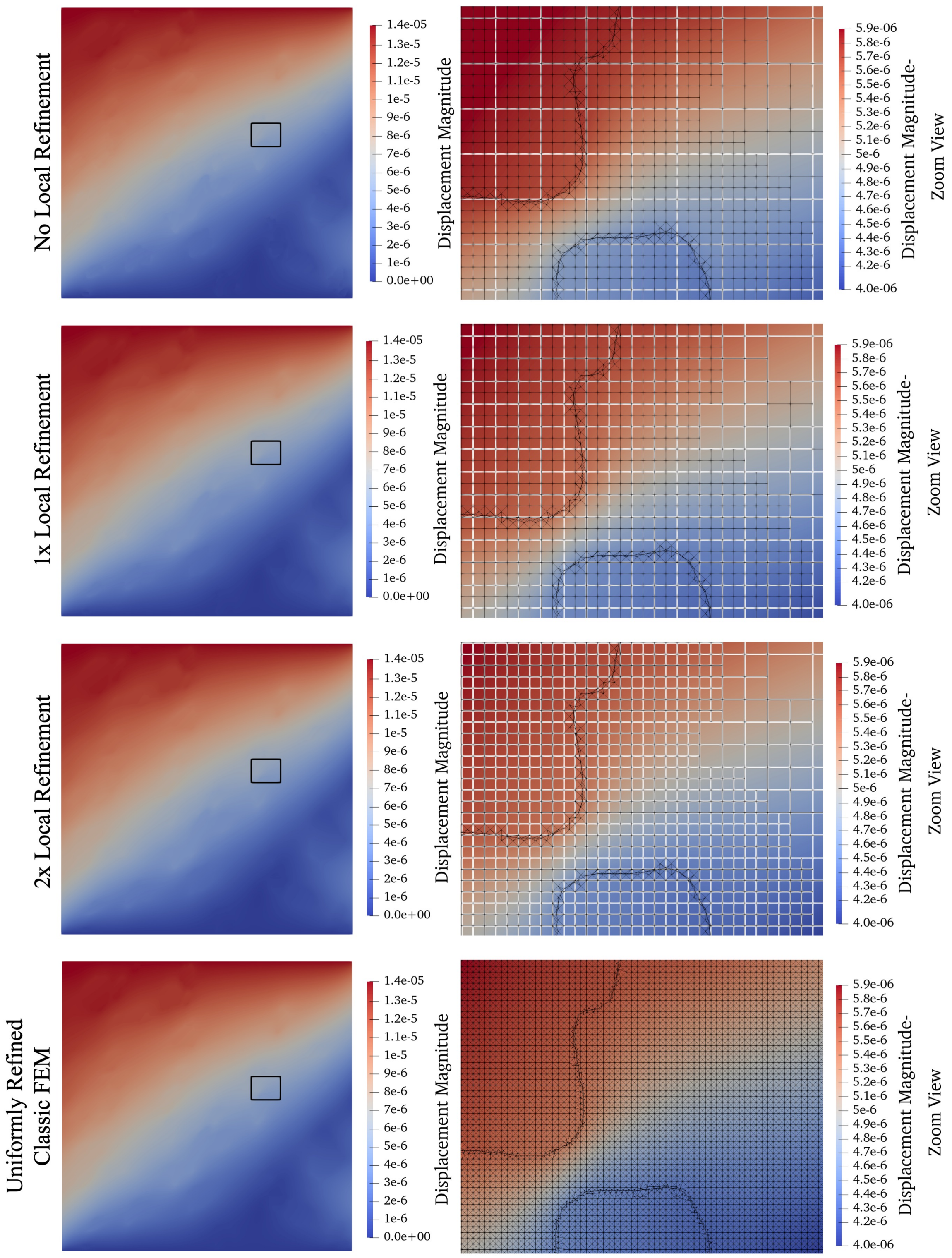


Figure 5.14. The displacement magnitude field results are compared for four different mesh configurations. The left shows the entire domain with the box indicating the region shown in the zoomed in view on the right.

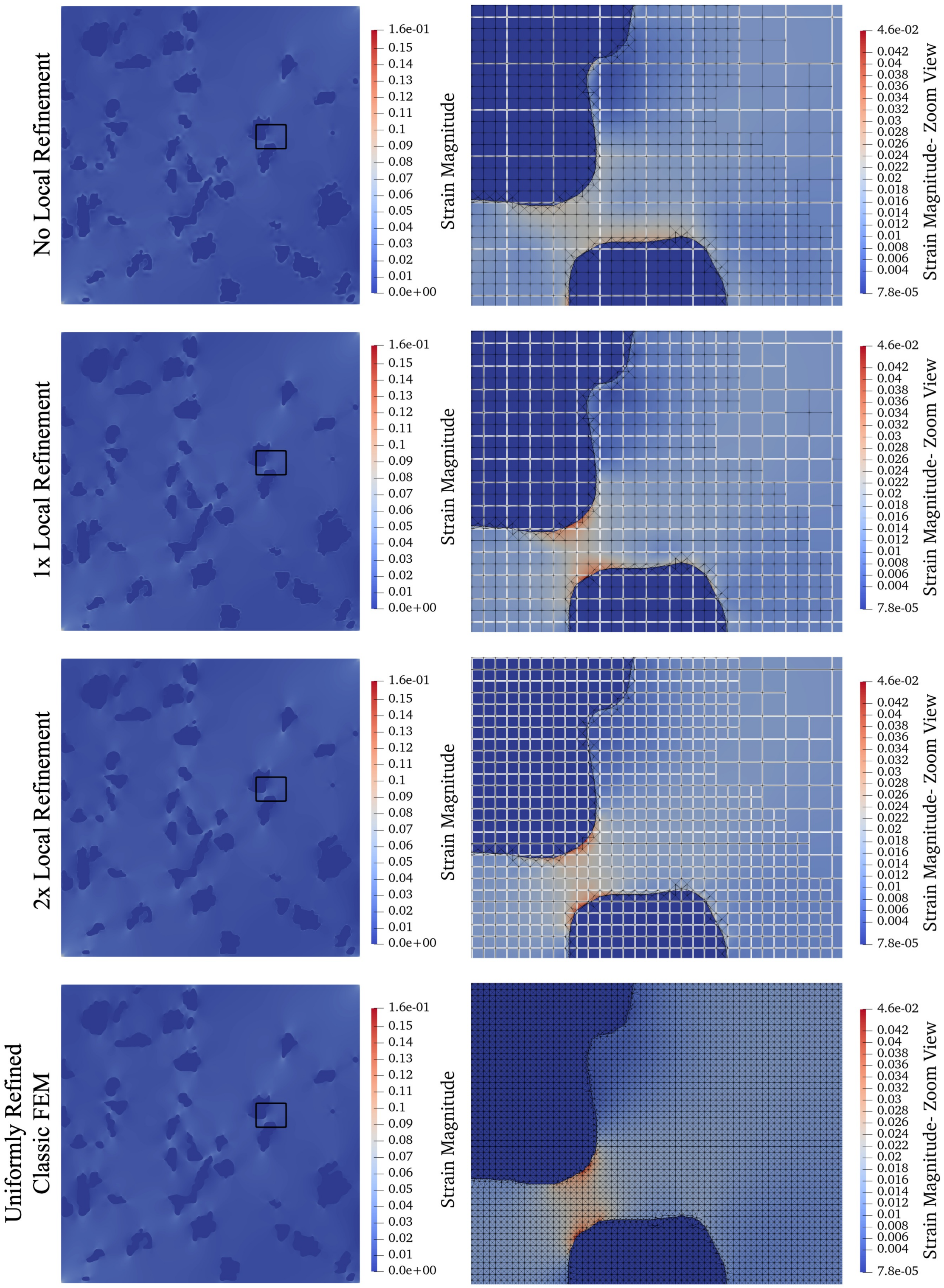


Figure 5.15. The mechanical strain magnitude field results are compared for four different mesh configurations. The left shows the entire domain with the box indicating the region shown in the zoomed in view on the right.

environmental temperature was $T_0 = 0^o C$. The sides were left both traction and heat flux free.

The governing equations for heat conduction and linear elasticity were coupled via the constitutive law adding a thermal expansion component to the total strain as in Eq. (5.19). Dirichlet boundary conditions for each field were imposed using Nitsche's method. Nitsche's terms enforcing continuity in T and $\boldsymbol{u}$ were also imposed upon the allumina-epoxy interfaces.

Bi-linear B-splines are used for the temperature field and bi-quadratic B-splines are used for each component of the displacement field, and both fields are interpolated using bi-quadratic Lagrange polynomials. For the boundary-fitted FEM comparison, Dirichlet boundary conditions were likewise enforced using Nitsches method, a bi-linear Lagrange basis was used for the temperature field, and a bi-quadratic Lagrange basis was used for the displacement. The results for the temperature field, the temperature gradient magnitude, the displacement magnitude, and the strain magnitude are shown in Figures 5.12, 5.13, 5.14, and 5.15, respectively. Within the figures, the foreground meshes are drawn in black while the background meshes are overlaid in white. The same foreground mesh, with two levels of local refinement to resolve the composite geometries, is used for each discretization.

Table 5.1. Degrees of freedom (DOFs) associated with the background function spaces at various levels of local refinement and with the uniformly refined FEM function spaces.

	No Local Refinement	1x Local Refinement	2x Local Refinement	Uniformly Refined Boundary-Fitted Mesh
Bi-linear basis DOFs	8,321	17,247	48,191	5,027,580
Bi-quadratic basis DOFs	9,381	22,166	60,729	10,055,160

Progressive levels of local refinement were applied to the background B-spline function spaces, as seen in the rows of Figures 5.12, 5.13, 5.14, and 5.15. This local refinement was implemented with THB-splines, previously defined in chapter 3. The regions of refinement with high-order splines are larger than those for the low-order to adequately support the truncated basis. This can be seen by comparing the white background meshes used for the temperature field depicted in Figures 5.12 and 5.13 with those used for the displacement field in Figures 5.14 and

5.15. The number of degrees of freedom associated with the various background discretizations are shown in Table 5.1.

With two levels of local refinement, the results were in qualitative agreement with the ones of the uniformly refined classical FEM example with far fewer degrees of freedom. As previously noted, the uniformly refined meshes were initially generated with the same workflow used to create the boundary conforming foreground meshes, a ‘hands-off’ method requiring only a bitmap image file. A more efficient mesh with fewer elements could be created for use in classical FEM but only with either significant user intervention or with sophisticated meshing software. Additionally, the workflow used here can easily be extended to 3D image stacks, which are not easily processed for classical FEM.

Chapter summary

This chapter presents a new interpolation-based immersed boundary method for multi-material and multi-physics problems. This method employs enriched truncated hierarchically refined B-spline background spaces and discontinuous hierarchically refined Lagrange integration spaces.

Domain geometry and material interfaces are represented by level set functions, which can be generated by, for example, geometric primitives or from 2D or 3D images. The domain is embedded in a grid with an associated B-spline basis and the level set function is discretized and used to compute Heaviside characteristic functions. Basis functions are inspected and individually enriched with respect to the disconnected material subregions within their supports.

The enriched and refined background bases are interpolated by a discontinuous foreground basis. This foreground basis requires a boundary-fitted mesh, but is not subject to usual mesh conditioning constraints. The foreground meshes are thus constructed by the discretized level set function boundary descriptions and hierarchically refined quadrilateral meshes. Cells in the refined quadrilateral mesh intersected by the domain and material interface geometry are triangulated to form a mixed-element type foreground mesh, which can be used by classical finite element codes to define a foreground basis. The level of refinement used to create the foreground mesh may exceed the level of refinement used for the background basis, allowing for greater geometric resolution without increasing the system's degrees of freedom associated with the background basis.

The background basis is interpolated using extraction operators. Extraction operators are constructed by evaluating the background basis at the locations of the foreground basis nodes. Existing classical finite element codes assemble the linear systems using the Lagrange foreground basis. The linear system is then projected onto the interpolated basis with the extraction operators and the system is solved with the interpolated basis. Interpolation allows the enriched and refined B-splines basis to be utilized in traditional finite element codes without the addition of complex integration subroutines, broadening the applicability of this method.

Several benchmark problems validated this method. Interpolated enriched bases were used in both 2D and 3D for a multi-material thermal diffusion problem with exact geometric representation. The numerical approximations were compared with analytic solutions and errors were computed. The L_2 and H_1 errors converged at ideal rates with mesh refinement. Foreground only refinement was implemented for a multi-material linear elasticity PDE involving a circular domain. When compared to the analytic solution, errors from a non-refined foreground mesh converged ideally with a bi-linear basis, but the L_2 convergence rate of the bi-quadratic basis was limited to 2^{nd} order due to geometric error in the discretization of the domain boundary. With sufficient foreground refinement, the ideal 3^{rd} order convergence for the bi-quadratic basis was observed.

Lastly, micro-CT images were used to generate a geometric discretization of an alumina-epoxy composite sample. Thermo-elastity was simulated, coupling separately discretized temperature and displacement fields through a thermal expansion component. Unique THB-spline bases were used for each field, bi-linear for temperature, and bi-quadratic for displacement. The results were in qualitative agreement with the ones of a uniformly refined traditional boundary conforming finite element simulation.

5.A Discretization of multi-material equations for linear elasticity

Let a domain of N material subdomains $\overline{\Omega} = \overline{\Omega}^1 \cup \overline{\Omega}^2 \cup ... \cup \overline{\Omega}^N \subset \mathbb{R}^{d_p}$ be the domain of interest, with varied material properties μ and λ:

$$\mu(\boldsymbol{x}) = \mu^m, \boldsymbol{x} \in \Omega^m \text{ and} \tag{5.48}$$

$$\lambda(\boldsymbol{x}) = \lambda^m, \boldsymbol{x} \in \Omega^m. \tag{5.49}$$

Following the problem set up in Subsection 5.3.2, a source term $\boldsymbol{b} : \Omega \rightarrow \mathbb{R}^{d_p}$, a traction term $\overline{\boldsymbol{h}} : \partial\overline{\Omega} \rightarrow \mathbb{R}^{d_p}$ on $\Gamma_{\overline{\boldsymbol{h}}} \subset \partial\overline{\Omega}$ and Dirichlet boundary data $\overline{\boldsymbol{u}} : \partial\overline{\Omega} \rightarrow \mathbb{R}^{d_p}$ on $\Gamma_{\overline{\boldsymbol{u}}} \subset \partial\overline{\Omega}$ are ascribed.

The strong form of this problem is then: Find $\boldsymbol{u} : \Omega \rightarrow \mathbb{R}^{d_p}$ such that $\forall\ m \in \mathscr{M}$

$$\begin{aligned} -\nabla \cdot \boldsymbol{\sigma}^m = \boldsymbol{b} &\quad \text{in } \Omega^m \ , \\ [\![\boldsymbol{u}]\!] = 0 &\quad \text{on all } \Gamma_{km} \\ [\![\boldsymbol{\sigma}]\!] \cdot \boldsymbol{n} = 0 &\quad \text{on all } \Gamma_{km} \\ \boldsymbol{\sigma}^m \cdot \boldsymbol{n} = \overline{\boldsymbol{h}} &\quad \text{on } \Gamma^m_{\overline{\boldsymbol{h}}} \ , \\ \boldsymbol{u} = \overline{\boldsymbol{u}} &\quad \text{on } \Gamma_{\overline{\boldsymbol{u}}} \end{aligned} \tag{5.50}$$

where $\Gamma_{km} = \overline{\Omega}^k \cap \overline{\Omega}^m \neq \emptyset$, $k \in \mathscr{M}$ and $k \neq m$ are the material interfaces. $\Gamma^m_{\overline{\boldsymbol{h}}} = \Gamma_{\overline{\boldsymbol{h}}} \cup \partial\overline{\Omega}^m$, and $\Gamma^m_{\overline{\boldsymbol{u}}} = \Gamma_{\overline{\boldsymbol{u}}} \cup \partial\overline{\Omega}^m$ are intersections of the domain boundaries with the material subdomain boundaries, and $\boldsymbol{n}$ denotes the surface normal. Here $[\![\cdot]\!] = (\cdot)^k - (\cdot)^k$ is again the jump of a given quantity over the Γ_{km} interface, and $\boldsymbol{n}$ is the surface normal. For each material subregion the displacement is $\boldsymbol{u}^m = \boldsymbol{u}(\boldsymbol{x})$, $\boldsymbol{x} \in \Omega^m$. The Cauchy stress tensor in each material subdomain is defined

$$\boldsymbol{\sigma}^m = \boldsymbol{C}^m : \boldsymbol{\varepsilon}^m = 2\mu^m\boldsymbol{\varepsilon}^m + \lambda^m \text{tr}(\boldsymbol{\varepsilon}^m)\boldsymbol{I} \tag{5.51}$$

in terms of the strain $\boldsymbol{\varepsilon}^m$, whose definition will vary depending on application, and $\boldsymbol{\sigma}(\boldsymbol{x}) = \boldsymbol{\sigma}^m$, $\boldsymbol{x} \in \Omega^m$.

The computational domain Ω is embedded into a hierarchically refined background mesh $\mathscr{K}_u$, generated using a sequence of refined meshes $\mathscr{K}^l$ and subdomains Ω_u^l, and associated with the THB basis $\mathscr{T}_u = \{B_i^u\}$. Each component of the displacement is then discretized with the function space

$$\mathscr{V}_u^h = \mathrm{span}\{B_i^u | \ \mathrm{supp}(B_i^u) \cap \Omega \neq \emptyset\}, \tag{5.52}$$

where $B_i^u \in \mathscr{T}_u$, the basis of enriched THB-splines.

The discrete form is then: find $\boldsymbol{u}^h \in \boldsymbol{\mathscr{V}}_u^h = [\mathscr{V}_u^h, \mathscr{V}_u^h]$ such that all $\boldsymbol{v}^h \in \boldsymbol{\mathscr{V}}_u^h$

$$\begin{aligned} \sum_{m=1}^{n} \left[\int_{\Omega}^{m} \boldsymbol{\sigma}(\boldsymbol{u}^h) : \boldsymbol{\varepsilon}^u(\boldsymbol{v}^h) \, d\Omega \right] - \int_{\Omega} \boldsymbol{b} \cdot \boldsymbol{v}^h \, d\Omega & \\ - \int_{\Gamma_{\overline{\boldsymbol{h}}}} \overline{\boldsymbol{h}} \cdot \boldsymbol{v}^h \, d\Gamma &= \mathscr{R}_u^D + \mathscr{R}_u^I \, , \end{aligned} \tag{5.53}$$

where $\mathscr{R}_u^D$ and $\mathscr{R}_u^I$ are the Dirichlet and interface residuals,

$$\mathscr{R}_u^D = \sum_{m=1}^{n} \Bigg[\mp \int_{\Gamma_{\bar{\boldsymbol{u}}}^m} (\boldsymbol{u}^h - \bar{\boldsymbol{u}}) \cdot \boldsymbol{\sigma}(\boldsymbol{v}^h) \cdot \boldsymbol{n} \, d\Gamma \\ - \int_{\Gamma_{\bar{\boldsymbol{u}}}^m} \boldsymbol{v}^h \cdot \boldsymbol{\sigma}(\boldsymbol{u}^h) \cdot \boldsymbol{n} \, d\Gamma \\ + \int_{\Gamma_{\bar{\boldsymbol{u}}}^m} \frac{\beta_u^D E}{h} (\boldsymbol{u}^h - \bar{\boldsymbol{u}}) \cdot \boldsymbol{v}^h \, d\Gamma \Bigg] \tag{5.54}$$

and

$$\mathscr{R}_u^I = \sum_{i=i}^{n} \sum_{j=1+i}^{n} \Bigg[- \int_{\Gamma_{ij}} [\![\boldsymbol{u}^h]\!] \cdot (\{\boldsymbol{\sigma}(\boldsymbol{v}^h)\} \cdot \boldsymbol{n}) \, d\Gamma \\ - \int_{\Gamma_{ij}} [\![\boldsymbol{v}^h]\!] \cdot (\{\boldsymbol{\sigma}(\boldsymbol{u}^h)\} \cdot \boldsymbol{n}) \, d\Gamma \\ + \int_{\Gamma_{ij}} \gamma_u^{ij} [\![\boldsymbol{u}^h]\!] \cdot [\![\boldsymbol{v}^h]\!] \, d\Gamma \Bigg]. \tag{5.55}$$

As with the temperature Dirichlet residual, the first line of Eq. (5.54) is either negative for symmetric Nitsche's method, which is used in this work, or positive for non-symmmetric Nitsche's method. Once again, $\{\cdot\} = w_i(\cdot)_i - w_j(\cdot)_j$ is the weighted average of a given quantity, with weights as defined in Eq. (5.14) using the elastic modulus as the material parameter ω. The penalty parameter γ_u^{ij} is defined as

$$\gamma_u^{ij} = 2\beta_u^I \frac{(h^i)^{d_p - 1} + (h^j)^{d_p - 1}}{(h^i)^{d_p}/E^i + (h^j)^{d_p}/E^j}, \tag{5.56}$$

where $\beta_u^I \geq 0$ is a user specified constant which controls the accuracy of the the interface condition, and $E(\boldsymbol{x}) = E^m$, $\boldsymbol{x} \in \Omega^m$.

The linear elastic subproblem can be compactly written as the variational problem: Find $\boldsymbol{u}^h \in \mathscr{V}_u^h = [\mathscr{V}_u^h, \mathscr{V}_u^h,]$ such that, $\forall \boldsymbol{v}^h \in \mathscr{V}_u^h$

$$a_u(\boldsymbol{u}^h, \boldsymbol{v}^h) = L_u(\boldsymbol{v}^h), \tag{5.57}$$

where $a_u(\boldsymbol{u},\boldsymbol{u})$ and $L_u(\boldsymbol{v})$ can be derived from Eq. (5.53).

To approximate the solutions to multi-material linear elasticity problems, this workflow introduces an interpolated background function space

$$\widehat{\mathscr{V}}_u^h = \operatorname{span}\{\widehat{B}_i^u \mid \operatorname{supp}(\widehat{B}_i^u) \cap \Omega \neq \emptyset\}, \tag{5.58}$$

where the interpolated background basis functions are defined

$$\widehat{B}_i^u := \sum_{j=1}^{\nu} M_{ij}^u N_j, \tag{5.59}$$

where

$$M_{ij}^u := B_i^u(\boldsymbol{x}_j) \tag{5.60}$$

is the displacement component Lagrange extraction operator. $\{N_j\}_{j=1}^{\nu}$ is the basis of a Lagrange FE space with nodal points $\boldsymbol{x}_j$ such that $N_i(\boldsymbol{x}_j) = \delta_{ij}$. Here ν is the number of foreground basis functions. Note that for multi-physics problems, the same foreground space is used to interpolate the background bases for both the temperature and the displacements.

The approximation of each displacement component is then

$$u_k^h = \sum_{i=1}^{n_u} \widehat{B}_i^u d_i^{u_k} = \sum_{i=1}^{n_u} \sum_{j=1}^{\nu} M_{ij}^u N_j d_i^{u_k}, \tag{5.61}$$

where $\{d_i^{u_k}\}_{i=1}^{n_u}$ are the unknown coefficients associated with each state variable field and n_u is the number of basis functions in the interpolated background B-spline basis. $k \in \{1,...,d_p\}$ are the indices associated with each of the displacement components, with d_p denoting the physical dimension. The vector-valued approximation of displacement is defined as

$$\boldsymbol{u}^h = \sum_{k=1}^{d_p} \sum_{i=1}^{n_u} \widehat{B}_i^u d_i^{u_k} \boldsymbol{e}^k \tag{5.62}$$

where $\boldsymbol{e}^k$ are the directional unit vectors. For brevity in notation, new capital letter indices $I = d_p(i-1)+k$ and $J = d_p(j-1)+k$ are defined such that the vector value basis functions are

$$\widehat{\boldsymbol{B}}_I^u = \widehat{B}_i^u \boldsymbol{e}^k \text{ and } \boldsymbol{N}_J = N_j^u \boldsymbol{e}^k, \tag{5.63}$$

and the approximation of displacement can be written

$$\boldsymbol{u}^h = \sum_{I=1}^{(d_p \cdot n_u)} \boldsymbol{B}_I^u d_I^u = \sum_{I=1}^{(d_p \cdot n_u)} \sum_{J=1}^{(d_p \cdot \nu)} M_{IJ}^{\boldsymbol{u}} \boldsymbol{N}_J d_I^u, \tag{5.64}$$

where $d_I^u = d_i^{u_k}$, and $M_{IJ}^{\boldsymbol{u}}$ are the components of the vector valued displacement field extraction operator

$$M_{IJ}^{\boldsymbol{u}} = \widehat{\boldsymbol{B}}_I^u (\boldsymbol{x}_J). \tag{5.65}$$

The variational form in Eq. (5.57) assembled using the interpolated basis forms the linear system

$$\boldsymbol{K}^{\boldsymbol{vv}} \boldsymbol{d}^{\boldsymbol{u}} = \boldsymbol{f}^{\boldsymbol{v}}, \tag{5.66}$$

where

$$K_{IJ}^{\boldsymbol{vv}} = a_{\boldsymbol{u}}(\widehat{\boldsymbol{B}}_I^u, \widehat{\boldsymbol{B}}_J^u), \text{ and} \tag{5.67}$$

$$f_I^{\boldsymbol{v}} = L_{\boldsymbol{u}}(\widehat{\boldsymbol{B}}_I^u). \tag{5.68}$$

Applying extraction, the linear system in Eq. (5.66) is rewritten as

$$(\boldsymbol{M}^{\boldsymbol{v}})^{\mathrm{T}}\boldsymbol{A}^{\boldsymbol{vv}}\boldsymbol{d}^{\boldsymbol{u}} = (\boldsymbol{M}^{\boldsymbol{u}})^{\mathrm{T}}\boldsymbol{b}^{u}, \tag{5.69}$$

where $\boldsymbol{A}^{\boldsymbol{vv}}$ and $\boldsymbol{b}^{\boldsymbol{v}}$ are computed with the foreground basis

$$A_{IJ}^{\boldsymbol{vv}} = a_{\boldsymbol{u}}(\boldsymbol{N}_I, \boldsymbol{N}_J) \text{ and} \tag{5.70}$$

$$b_I^{\boldsymbol{v}} = L_{\boldsymbol{u}}(\boldsymbol{N}_I). \tag{5.71}$$

The quantities $\boldsymbol{A}^{\boldsymbol{vv}}$ and $\boldsymbol{b}^{\boldsymbol{v}}$ are evaluated and assembled with the boundary-fitted foreground mesh, following the workflow outlined in Subsection 5.2.2.

Acknowledgements

The above chapter, in full, is a reprint of the material as it appears in Fromm, Jennifer E; Wunsch, Nils; Maute, Kurt; Evans, John A; and Chen, Jiun-Shyan; “Interpolation-based immersogeometric analysis methods for multi-material and multi-physics problems”, Journal of Computational Mechanics 2024. The dissertation author was the primary researcher and author of this paper. Notation has been changed to maintain consistency within the dissertation.

The authors acknowledge the support for this work from the National Science Foundation under Grants 2103939 and 2104106. The authors also acknowledge and thank Dr. David Kamensky for his assistance in developing the interpolation workflow.

Chapter 6

Interpolation-based reproducing kernel particle method

The previous two chapters have introduced and applied interpolation-based immersed-boundary and immersogeometric methods to a wide class of problems. The work presented has gone beyond past interpolation-based implementations through the application of approximate, as opposed to exact, extraction. With approximate extraction, the interpolated basis is not everywhere equivalent to the background basis.

In the first sections of chapter 4, the interpolation error arising from this approximation is limited to the regions surrounding material or domain interfaces, and is thus bounded by the method error, allowing for interpolation-based immersed boundary methods to maintain ideal error convergence rates. Venturing beyond what has been mathematically shown, chapter 4 also presents numerical evidence that approximate extraction can be extended to loosen the constraints placed on the foreground. The local refinement techniques explored in 5 also employed foreground meshes with hanging nodes, further violating the assumptions made in chapter 4.

This chapter builds upon the concept of approximate extraction and non-background fitted interpolation with the extension of interpolation-based methods to meshfree methods, specifically the reproducing kernel particle method (RKPM). Recall that RKPM was introduces alongside IGA in chapter 3. The new method presented here will be referred to as Interpolation-based

RKPM, or Int-RKPM.

The outline of the chapter is as follows: Section 6.1 introduces the novel computational method combining interpolation with RKPM; Section 6.2 provides numerical data supporting the efficacy of the proposed method when applied to (1) a single material poisson problem, (2) a single material biharmonic problem, (3) a linear elastic problem with approximate geometry, (4) a tri-material poisson problem and (5) a bi-material linear elasticity problem with approximate geometry.

6.1 Interpolation-based RKPM for implementation within existing finite element software frameworks

Despite the significant development of nodal integration techniques, efficient integration and assembly remains a challenge in the implementation of RKPM. This work introduces a novel computational method employing interpolation to represent RKPM bases as linear combinations of Lagrange polynomial bases. Interpolation-based RKPM (Int-RKPM) reduces assembly costs when compared to classical RKPM methods, and is implemented within existing finite element software.

6.1.1 Properties of Int-RKPM functions

The main idea of interpolation-based methods is to replace a basis function, which for some reason may be difficult to integrate or otherwise work with, with an approximation of that function. This approximation is constructed of a secondary ‘foreground’ basis $\{N_i\}_{i=0}^{\nu}$, where ν is the number of basis functions. Thus, an RKPM function Ψ_I is approximated as

$$\Psi_I(\boldsymbol{x}) \sim \hat{\Psi}_I(\boldsymbol{x}) = \sum_{j=0}^{\nu} M_{Ij} N_j(\boldsymbol{x}), \tag{6.1}$$

where the tensor M_{Ij} is referred to as the extraction operator.

The properties of a basis of interpolated functions $\{\hat{\Psi}_I\}_{I=0}^{NP}$ will depend on both the properties of both the original 'background' basis $\{\Psi_I\}_{I=0}^{NP}$ and the foreground basis $\{N_i\}_{i=0}^{v}$. In this work, $\{N_i\}_{i=0}^{v}$ is composed of Lagrange polynomials and thus satisfies the kronecker delta property $N_i(\boldsymbol{x}_j) = \delta_{ij}$. Due to this interpolatory property, the components of the extraction operator are computed by evaluating the original background function at the nodal locations of the foreground basis,

$$M_{Ij} = \Psi_I(\boldsymbol{x}_j). \tag{6.2}$$

Following [71], several properties of the interpolated basis can be derived. Provided the original basis $\{\Psi_I\}_{I=0}^{NP}$ forms a partition of unity, the interpolated basis $\{\hat{\Psi}_I\}_{I=0}^{NP}$ will as well. The function space spanned by the interpolated basis will span polynomials of degree κ, where $\kappa = min\{k,n\}$, n being the polynomial order spanned by the background basis and k being the polynomial order spanned by the foreground basis. If both the foreground and background basis have local support, then the interpolated basis will as well.

Furthermore, if both foreground and background basis satisfy the reproducing condition

$$\sum_{I=0}^{NP} \Psi_I(\boldsymbol{x}) f^{\alpha}(\boldsymbol{x}_I) = f^{\alpha}(\boldsymbol{x}), \text{ and } \sum_{i=0}^{v} N_i(\boldsymbol{x}) f^{\alpha}(\boldsymbol{x}_i) = f^{\alpha}(\boldsymbol{x}), \tag{6.3}$$

where f^{α} is a polynomial of degree α, then the interpolated basis also satisfies the condition

$$\sum_{I=0}^{NP} \hat{\Psi}_I(\boldsymbol{x}) f^{\alpha}(\boldsymbol{x}_I) = \sum_{I=0}^{NP} \sum_{j=0}^{v} M_{Ij} N_j(\boldsymbol{x}) f^{\alpha}(\boldsymbol{x}_I) = \tag{6.4}$$

$$\sum_{j=0}^{v} N_j(\boldsymbol{x}) \left(\sum_{I=0}^{NP} \Psi_I(\boldsymbol{x}_j) f^{\alpha}(\boldsymbol{x}_I) \right) = \sum_{j=0}^{v} N_j(\boldsymbol{x}) f^{\alpha}(\boldsymbol{x}_j) = f^{\alpha}(\boldsymbol{x}). \tag{6.5}$$

The derivatives of the interpolation functions can be computed using the derivative of the

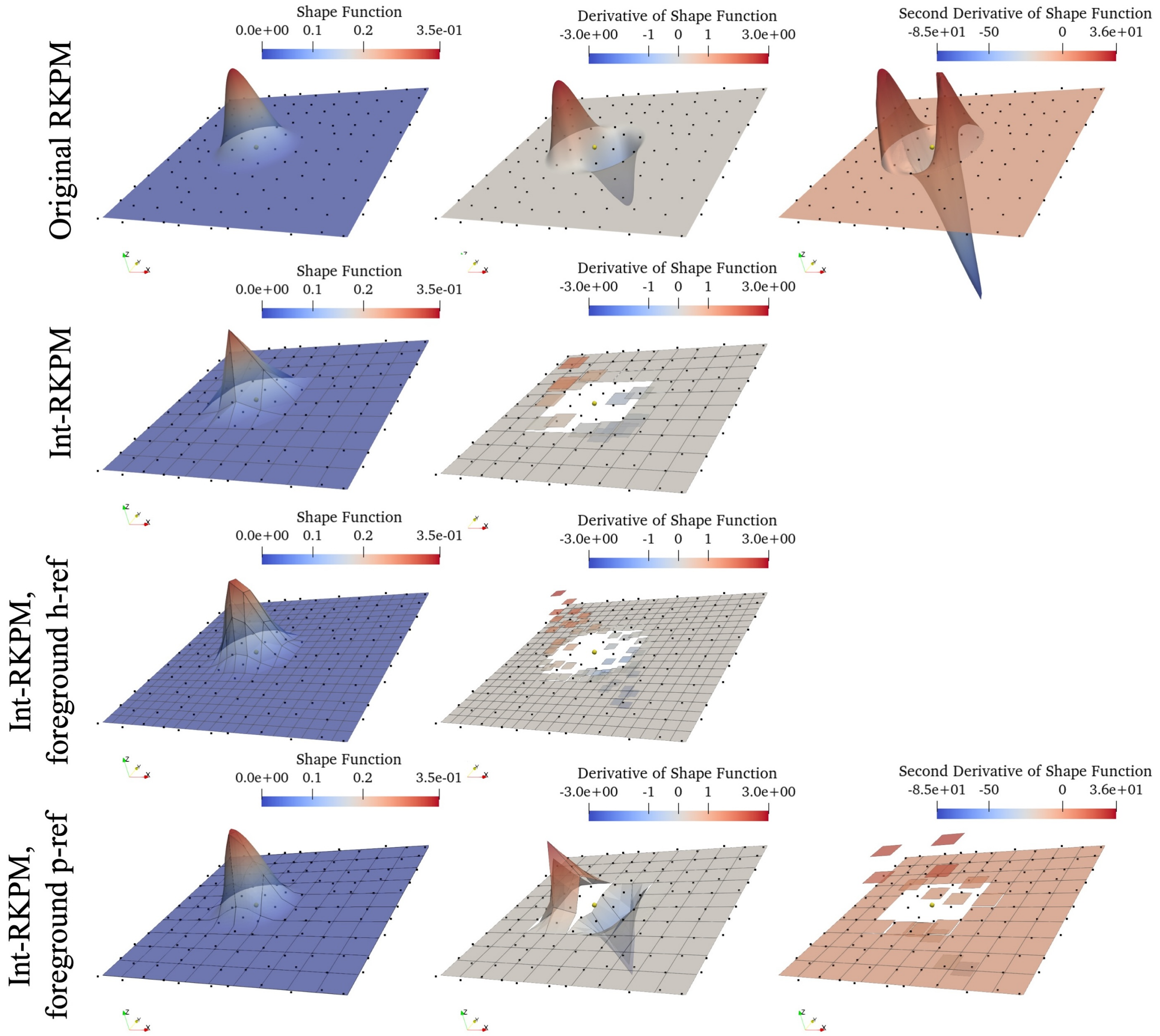


Figure 6.1. A linear RKPM shape function Ψ_I, using a cubic B-spline kernel is plotted along with its first derivative $\partial_x \Psi_I$ and second derivative $\partial_{xx} \Psi_I$. The second row shows the analogous Int-RKPM function $\hat{\Psi}_I$ and its derivatives $\partial_x \hat{\Psi}_I$ and $\partial_{xx} \hat{\Psi}_I$, interpolated with a linear foreground function space with element size equal to the average background nodal spacing. *h*-refinement and *p*-refinement are performed on the foreground basis N_i, and are shown in the third and fourth rows, respectively.

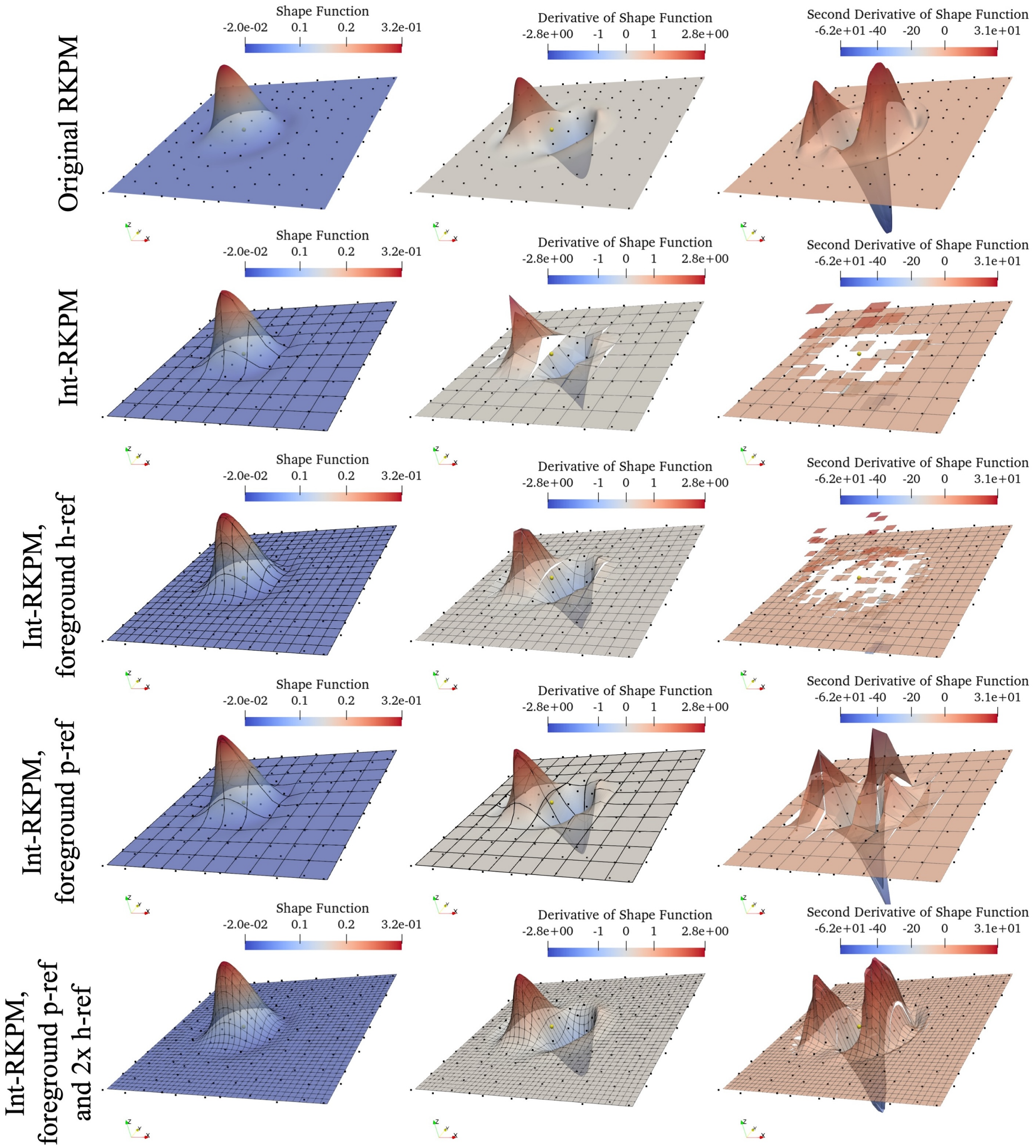

Original RKPM
Int-RKPM
Int-RKPM, foreground h-ref
Int-RKPM, foreground p-ref
Int-RKPM, foreground p-ref and 2x h-ref
Shape Function
-2.0e-02 0.1 0.2 3.2e-01
Derivative of Shape Function
-2.8e+00 -1 0 1 2.8e+00
Second Derivative of Shape Function
-6.2e+01 -40 -20 0 3.1e+01

Figure 6.2. A quadratic RKPM shape function Ψ_I, using a cubic B-spline kernel is plotted along with its first derivative $\partial_x \Psi_I$ and second derivative $\partial_{xx} \Psi_I$. The second row shows the analogous Int-RKPM function $\hat{\Psi}_I$ and its derivatives $\partial_x \hat{\Psi}_I$ and $\partial_{xx} \hat{\Psi}_I$, interpolated with a quadratic foreground function space with element size equal to the average background nodal spacing. h-refinement and p-refinement are performed on the foreground basis N_i, and are shown in the third and fourth rows, respectively. The fifth row shows the results of combined p-refinement and two levels of h-refinement, which is more accurately able to interpolate the second derivatives

foreground functions:

$$\partial_x \hat{\Psi}_I(\boldsymbol{x}) = \sum_{j=0}^{v} M_{Ij} \partial_x N_j(\boldsymbol{x}). \tag{6.6}$$

Again, provided the foreground and background bases satisfy the gradient reproducing condition

$$\sum_{I=0}^{NP} \partial_x \Phi_I(\boldsymbol{x}) f^{\alpha}(\boldsymbol{x}_I) = \partial_x f^{\alpha}(\boldsymbol{x}), \text{ and } \sum_{i=0}^{v} \partial_x N_i(\boldsymbol{x}) f^{\alpha}(\boldsymbol{x}_i) = \partial_x f^{\alpha}(\boldsymbol{x}), \tag{6.7}$$

the interpolated basis will as well

$$\sum_{I=0}^{NP} \partial_x \hat{\Phi}_I(\boldsymbol{x}) f^{\alpha}(\boldsymbol{x}_I) = \sum_{I=0}^{NP} \sum_{j=0}^{v} M_{Ij} \partial_x N_j(\boldsymbol{x}) f^{\alpha}(\boldsymbol{x}_I) = \tag{6.8}$$

$$\sum_{j=0}^{v} \partial_x N_j(\boldsymbol{x}) \left(\sum_{I=0}^{NP} \Phi_I(\boldsymbol{x}_j) f^{\alpha}(\boldsymbol{x}_I) \right) = \sum_{j=0}^{v} \partial_x N_j(\boldsymbol{x}) f^{\alpha}(\boldsymbol{x}_j) = f^{\alpha}(\boldsymbol{x}). \tag{6.9}$$

RKPM shape functions Ψ and a series of Int-RKPM shape functions $\hat{\Psi}$, are shown in Figures 6.1 and 6.2.

Previous interpolation work deals primarily with exact interpolation, where the foreground basis is sufficiently refined to exactly interpolate every function in the background basis. In [71] and [70], approximate extraction was implemented to utilize interpolation for immersed boundary analysis, where the interpolation is exact in the majority of a computational domain but approximate in the lower dimensional surface where background elements were cut. Also investigated in [71] were so-called ‘background-unfitted’ foreground meshes. In background-fitted foreground meshes, the foreground cells conform to the boundary of each background basis function support. These constraints are not present with background-unfitted foreground meshes, which nevertheless yielded optimal error convergence rates when employed to approximate PDEs. With RKPM and other meshfree methods the construction of background-fitted foreground meshes, where foreground cells would conform to the support of each RKPM function, presents a considerable challenge and would prove counter to the goal of Int-RKPM which is to decrease the computational cost and implementation overhead of meshfree methods.

Int-RKPM thus represents a new iteration of background-unfitted foreground interpolation techniques. With the loosening of constraints on the foreground mesh discretization comes increased choice on the part of the user, and considerable space is dedicated in this work to quantifying the effect of foreground space on the properties of Int-RKPM. In general, the discretization used to generate the foreground basis should have average element size equal to or less than the average nodal space of the background RKPM point distribution, and should be of equal polynomial order (i.e. $k = n$). This work will discuss both foreground $h-$ refinement, where the foreground mesh is refined relative to the background RKPM discretization, and foreground $p-$ refinement, where the polynomial order of the foreground basis is increased relative to the background RKPM order of reproducibility. Both types of foreground refinement are shown in Figures 6.1 and 6.2. The effects of these refinements and will be discussed in following sections 6.2.

6.1.2 Using Int-RKPM bases to solve PDEs

The poisson problem is used to model the use of Int-RKPM functions to solve PDEs. The strong form of this problem is simply: Find $u : \Omega \to \mathbb{R}$ such that

$$\begin{aligned} -\Delta u &= f \quad \text{in}\, \Omega\,, \\ u &= g \quad \text{on}\, \partial\Omega\,, \end{aligned} \tag{6.10}$$

where g is Dirichlet boundary data. Nitsche's method [127] is used to enforce boundary conditions such that the approximate form can be written: Find $u^h \in \mathscr{V}^h$ such that $\forall v^h \in \mathscr{V}^h$,

$$\begin{aligned} \int_\Omega \nabla u^h \cdot \nabla v^h d\Omega - \int_{\partial\Omega} \left(\nabla u^h \cdot \boldsymbol{n}\right) v^h d\Gamma \mp \int_{\partial\Omega} \left(\nabla v^h \cdot \boldsymbol{n}\right) u^h d\Gamma + \int_{\partial\Omega} \frac{C_{\text{pen}}}{h} u^h v^h d\Gamma \\ = \int_\Omega f v^h d\Omega \mp \int_{\partial\Omega} \left(\nabla v^h \cdot \boldsymbol{n}\right) g d\Gamma + \int_{\partial\Omega} \frac{C_{\text{pen}}}{h} g v^h d\Gamma\,, \end{aligned} \tag{6.11}$$

where $\boldsymbol{n}$ is the normal vector, C_{pen} is a user defined constant, and h is understood to be the background function nodal spacing. This work employs the symmetric Nitsche's method, such that the $\mp$ signs in equation 6.11 are taken as $-$. The left and right hand side of equation 6.11 can be grouped into a bilinear and linear form, respectively,

$$a(u^h, v^h) = L(v^h). \tag{6.12}$$

For classic RKPM, the function space $\mathscr{V}^h = \mathscr{V}_{RKPM} = \text{span}\{\Psi_I\}$, where each shape function Ψ_I can be computed with Equation 3.20 . Functions $u^h_{RKPM} \in \mathscr{V}_{RKPM}$ are then expressed as

$$u^h_{RKPM}(\boldsymbol{x}) = \sum_I \Psi_I(\boldsymbol{x}) u_I. \tag{6.13}$$

These shape functions can be integrated either using Gauss-quadrature schemes or some flavor of stabilized nodal integration, to assemble the linear system

$$\boldsymbol{Kd} = \boldsymbol{F} \tag{6.14}$$

where $K_{IJ} = a(\Psi_J, \Psi_I)$, $d_I = u_I$, and $F_I = L(\Psi_I)$ are defined in terms of the bilinear and linear form.

With interpolation-based RKPM the function space $\mathscr{V}^h = \mathscr{V}_{Int-RKPM} = \text{span}\{\hat{\Psi}_I\}$, where each shape function $\hat{\Psi}_I$ is an approximation of the corresponding function Ψ_I, as in equation 6.1. With the extraction operator given in equation 6.2, functions $u^h_{Int-RKPM} \in \mathscr{V}_{Int-RKPM}$ are expressed as

$$u^h_{Int-RKPM}(\boldsymbol{x}) = \sum_I \hat{\Psi}_I(\boldsymbol{x}) \hat{u}_I = \sum_I \sum_j M_{Ij} N_j(\boldsymbol{x}) \hat{u}_I. \tag{6.15}$$

To assemble the linear system in equation 6.14, allow $\hat{K}_{IJ} = a(\hat{\Psi}_J, \hat{\Psi}_I)$, $\hat{d}_I = \hat{u}_I$, and

$\hat{F}_I = L(\hat{\Psi}_I)$. The stiffness matrix and force vector are thus

$$\hat{K}_{IJ} = a(\hat{\Psi}_J, \hat{\Psi}_I) = M_{Jl} a(N_l, N_k) M_{kI} \text{ and } \hat{F}_I = L(\hat{\Psi}_I) = M_{Ik} L(N_k). \tag{6.16}$$

The quantities $a(N_l, N_k)$ and $L(N_k)$ can be easily computed with the foreground Lagrange polynomial basis $\{N_i\}_{i=0}^{\nu}$. The only evaluation of the original RKPM basis resides within the computation of the extraction operator, see equation 6.2. This feature of Int-RKPM allows existing finite element software to be easily augmented to use meshfree methods.

6.1.3 Implementation through open-source finite element software

Efficient integration techniques have been developed for meshfree methods, but these techniques are difficult to implement within existing finite element codes. Open source implementations of RKPM [88] and other meshfree methods [136] rely on considerable custom software packages.

As Int-RKPM replaces the integration of RKPM basis functions with integration of classic Lagrange polynomials, it is naturally implemented within existing FEM software. Following previous works on interpolation-based IGA [99, 97] and immersogeometric methods [71, 70], this chapter employs the popular open source software FEniCSx [15] to illustrate the potential of Int-RKPM.

6.2 Numerical Results

6.2.1 Comparison of interpolation-based RKPM with classic RKPM

To demonstrate the efficacy of Int-RKPM a simple poisson problem on a unit square was modeled. A suite of RKPM point sets was used to investigate error convergence rates. Each point set was create by perturbing a uniform grid of points in a jittered grid [48] with initial uniform spacing of h and perturbations of $\varepsilon h \eta$ in each coordinate. $\eta \in (0,1)$ was randomly generated for each point and coordinate and ε is referred to as the perturbation parameter. The accuracy of integration schemes is shown to be linked to to the uniformity of points [42] thus multiple perturbation parameters $\varepsilon \in [0,1]$ were investigated. However, provided points did not overlap, the performance of Int-RKPM was not dependent on the point distribution, and for all RKPM point sets in this chapter $\varepsilon = 0.5$.

With the point sets, RKPM bases were defined using a cubic B-spline kernel and circular supports. The normalized support size of each function was $n+1$, where n is the polynomial order of reproducibility. Both linear, $n = 1$, and quadratic, $n = 2$, RKPM bases were investigated.

Interpolation was performed using continuous-Galerkin type Lagrange polynomial foreground bases defined upon a suite of uniform quadrilateral boundary conforming meshes. The effects of both $p-$ and $h-$refinement on the foreground basis were investigated. Initial foreground element sizes are proportional to background nodal spacing. With $h-$refinement, the ratio of foreground element size to average RKPM nodal spacing was decreased to 2:1 (for 1x $h-$ref) and 4:1 (for 2x $h-$ref). The polynomial order of the foreground basis, k, is set initially to n, the polynomial order of reproducibility of the background basis. With $p-$refinement, $k = n+p$ where p is the level of refinement.

Int-RKPM is here compared with a classic implementation of RKPM where integrals are computed using a Gauss-Quadrature grid. The grid size is equal to the average nodal spacing of the RKPM pointset. A 6×6 Gauss scheme is used for the linear RKPM basis, and an 8×8 Gauss scheme is used for the quadratic RKPM basis.

The Poisson problem introduced in 6.1.2 is modeled, using the method of manufactured solutions to set $f = -\Delta u_{ex}$ in Equation 6.39 and $g = u_{ex}$ in Equation 6.39, where

$$u_{ex} = \sin(0.1x+0.1)\sin(0.1y+0.1) \tag{6.17}$$

The results are shown in Figure 6.3.

For the linear basis, results shown in the first row of Figure 6.3, interpolation-error increases error magnitude but does not impact the convergence rate of Int-RKPM. $h-$refinement

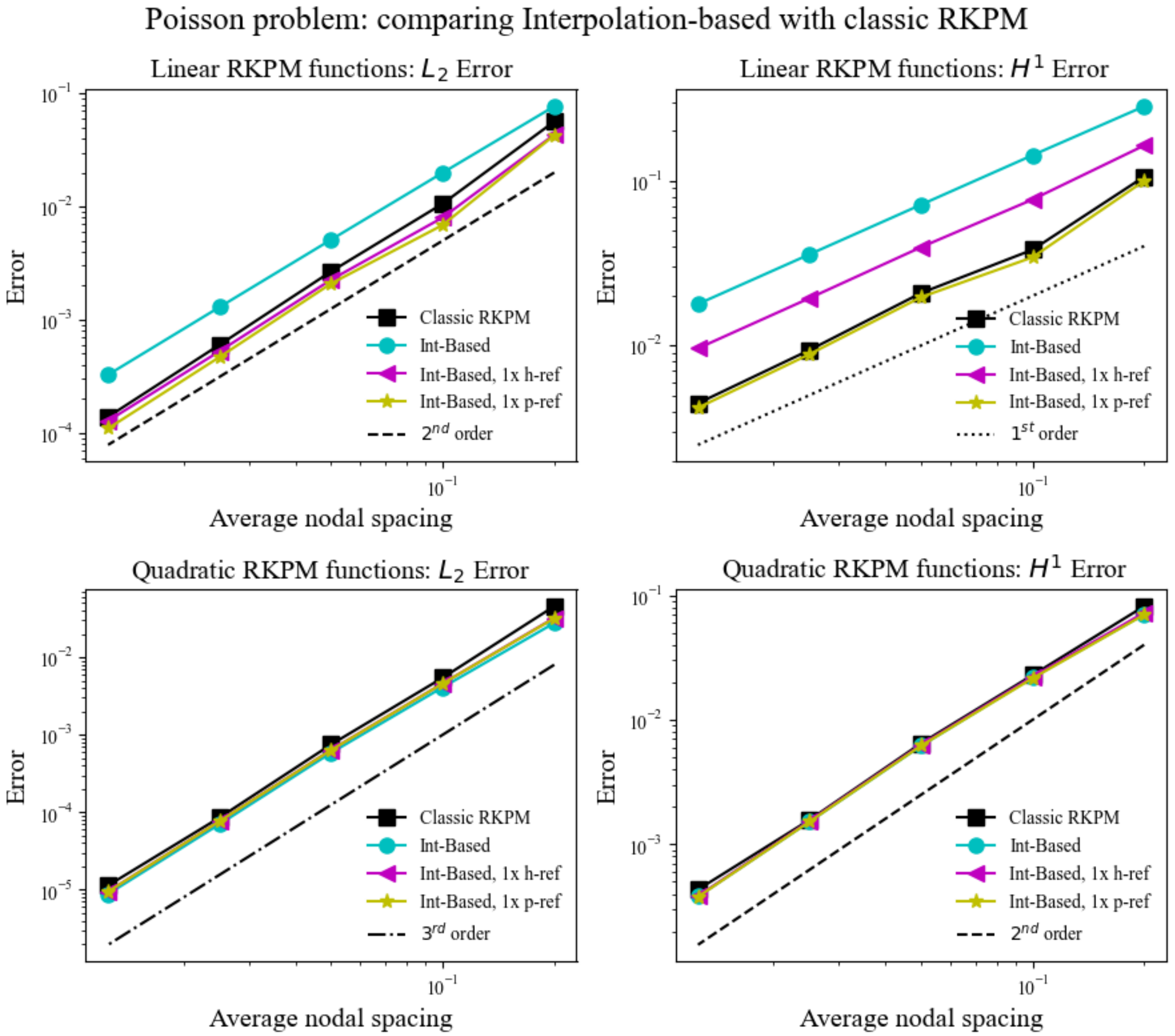


Figure 6.3. Convergence data for the poisson problem, comparing Int-RKPM with classic RKPM.

reduces error magnitude to below that of classic RKPM for the L_2 error norm, and reduces the magnitude marginally for the H_1 error norm. A level of $p-$refinement is sufficient to match the error magnitude of Int-RKPM to classic RKPM for this example.

This behaviour is qualitatively explained by comparing the derivatives of Int-RKPM in Figure 6.1. The second row of Figure 6.1 corresponds to the blue (circle) line in Figure 6.3, where the first derivative is interpolated as the linear combination of relatively coarse piecewise constant foreground functions. When foreground $h-$ refinement is applied for the third row of Figure 6.1, the approximation is improved, resulting in the decreased error magnitude shown by the pink (triangle) line in the plot in Figure 6.3. However, it is not until foreground $p-$refinement is applied to in the fourth row of 6.1, and the interpolation of the derivative is given by pieceswise linear functions, that the interpolation error is decreased sufficiently to reproduce the classic RKPM results. This is shown by the overlap of the yellow (star) line and black (square) line in the plot in Figure 6.3. However, it must be emphasized that no amount or type of local foreground refinement is required for the Int-RKPM method to exhibit optimal convergence rates.

For the quadratic basis, the second row of plots in Figure 6.3, the results produced by Int-RKPM are practically identical to classic RKPM. Additional foreground refinement neither increases the rate of convergence nor decreases the magnitude of the error. These results can be qualitatively understood by analyzing the quadratic Int-RKPM derivatives in Figure 6.2. The second row of the figure shows the Int-RKPM functions without any foreground refinement. The first derivative of the function is interpolated with piecewise linear functions. The improvements to the interpolation of the derivative are minimal with the application of foreground $h-$refinement, seen in the third row, and foreground $p-$refinement, seen in the fourth row.

6.2.2 Interpolation for high-order derivatives

Numerous problems in engineering applications are described by PDEs containing higher order derivatives. Notably, the equations characterizing Kirchhoff love shells contain fourth order derivatives and must be approximated by functions belonging to an H^2-conforming basis.

RKPM bases can easily satisfy this property, but due to the relative difficult in computing high order basis function derivatives, particle methods have not been widely applied to shell problems.

The computation of derivatives of an interpolated RKPM basis can be done with standardized finite element operations, removing this level of difficulty. To demonstrate the efficacy of interpolated-RKPM with non-conforming function spaces, the biharmonic problem is tested, with strong form Find $u : \Omega \to \mathbb{R}$ such that

$$\Delta^2 u = f \; , \tag{6.18}$$

with boundary conditions

$$u = g \text{ on } \partial\Omega \; , \tag{6.19}$$

$$\nabla u \cdot \boldsymbol{n} = h \text{ on } \partial\Omega \; , \tag{6.20}$$

where $f : \Omega \to \mathbb{R}$ is a given source term and $g : \partial\Omega \to \mathbb{R}$ and $h : \partial\Omega \to \mathbb{R}$ are boundary data. Employing the method of manufactured solutions, $f = \Delta^2 u_{ex}$, $g = u_{ex}$, and $h = \nabla u_{ex} \cdot \boldsymbol{n}$. For this numerical example,

$$u_{ex}(x,y) = \sin(0.1y + 0.1)\sin(0.1x + 0.2). \tag{6.21}$$

Boundary conditions are weakly enforced using Nitsche's-like residual terms, such that the weak form can be written as: Find $u^h \in \mathscr{V}^h$ such that, $\forall v^h \in \mathscr{V}^h$,

$$\begin{aligned} &\int_\Omega \Delta u^h \Delta v^h \, d\Omega + \int_{\partial\Omega} \nabla \Delta u^h \cdot \boldsymbol{n} v^h - \Delta u^h \nabla v^h \cdot \boldsymbol{n} \, d\Gamma \\ &+ \int_{\partial\Omega} (\nabla \Delta v^h) \cdot \boldsymbol{n}(u^h - u_{ex}) - \Delta v (\nabla u^h \cdot \boldsymbol{n} - \nabla u_{ex} \cdot \boldsymbol{n}) \, d\Gamma \\ &+ \int_{\partial\Omega} \frac{\alpha}{h^3}(u^h - u_{ex}) v^h + \frac{\beta}{h}(\nabla u^h \cdot \boldsymbol{n} - \nabla u_{ex} \cdot \boldsymbol{n}) \nabla v^h \cdot \boldsymbol{n} \, d\Gamma = \int_\Omega f v^h \, d\Omega \; , \end{aligned} \tag{6.22}$$

where $\alpha > 0$ and $\beta > 0$ are user specified constants. For the computations of this paper,

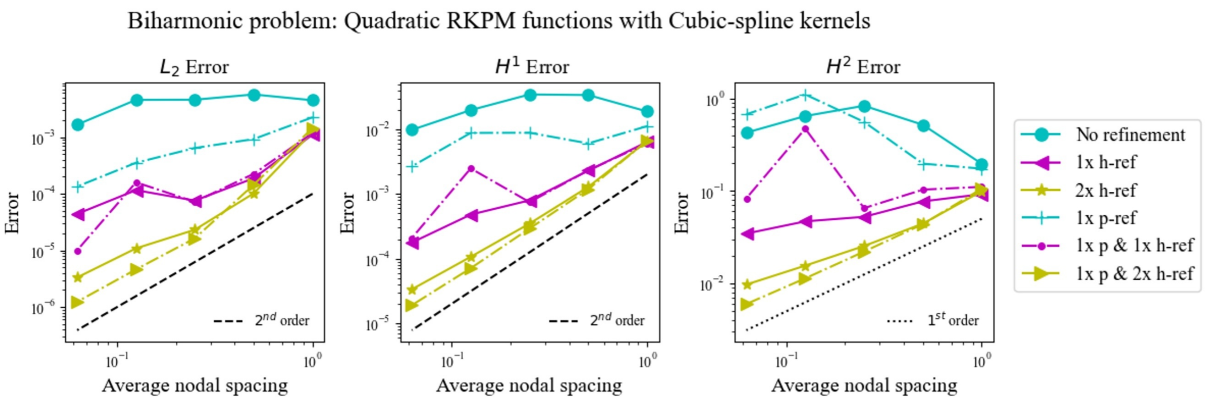


Figure 6.4. Convergence data for the biharmonic problem approximated with Int-RKPM, comparing various levels of foreground $h-$ and $p-$ refinement.

$\alpha = \beta = 10$.

The same suite of background RKPM basis functions as employed in the previous example are used along with uniform quadrilateral foreground meshes. Numerical tests investigated L_2, H_0^1, and H_0^2 error convergence rates with concurrent foreground and background refinement. Results are shown in Figure 6.4.

As in the lower order derivative Poisson problem investigated in subsection 6.2.1, some additional foreground refinement is required to reduce interpolation error. This is because with this example, the method must not only interpolate the first derivatives of the original RKPM function, but also the second derivatives. As seen in the first rows of Figures 6.1 and 6.2, the second derivatives of RKPM functions are difficult to fully resolve. This, in addition to the high computational cost of computing higher order derivatives, is a reason why meshfree methods are under utilized for high order PDEs like the biharmonic problem.

While the results of the previous section showed that $p-$refinement of the foreground mesh improved the approxiation potential of Int-RKPM, $p-$refinement alone is not sufficient to achieve optimal error convergence rates for this PDE, as seen by the solid colored pink blue and green lines. And unlike the Poisson problem, the original Int-RKPM without foreground refinement (the solid pink line) not only has a larger error magnitude, but the errors do not

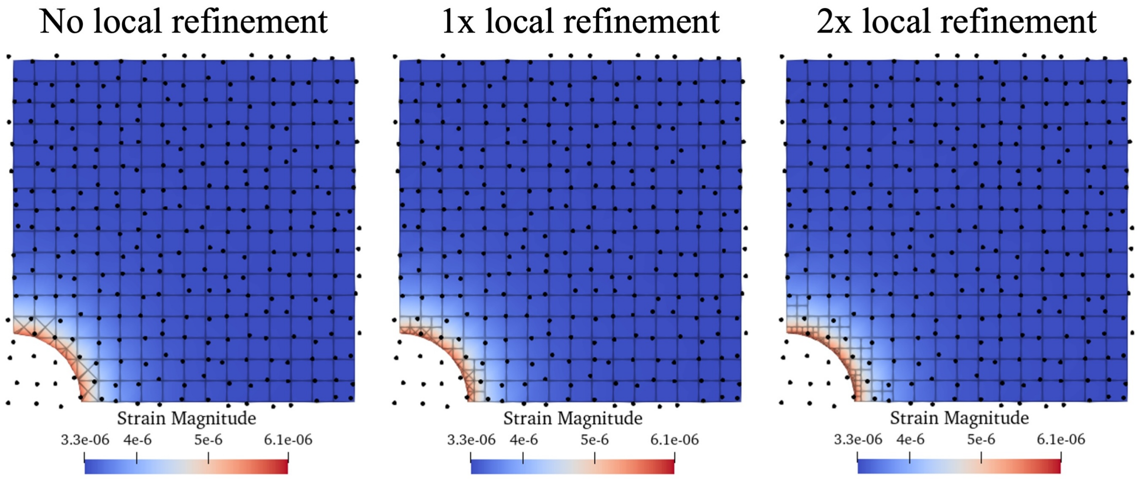


Figure 6.5. Local foreground refinement is performed to increase geometric resolution for problems with approximate interface representation. The foreground meshes are shown along with the RKPM point discretization, which is the same for each level of local refinement.

converge at all. In this case, foreground $h-$refinement is required to acheive optimal rates. that is required. With two levels of $h-$refinement (such that the ratio of foreground element size to background nodal spacing is 4:1) optimal rates are observed. This can be qualitatively understood by examination of the third column of images in Figure 6.2.

6.2.3 Extensions to approximate geometries

Int-RKPM can also be applied to problems of more complicated geometric domains than those illustrated in the previous two sections. RKPM, and other meshfree methods, do not require high quality boundary conforming meshes to approximated curved or otherwise complex geometries, but in general require a boundary conforming geometric discretization to perform integration. This is likewise the case for Int-RKPM where the integration discretization is in fact a mesh used to create a foreground function space.

In this example linear elasticity is modeled in a plate with a hole. Navier's equations for linear elasticity are as follows: Find $\boldsymbol{u} : \Omega \rightarrow \mathbb{R}^2$

$$-\nabla \cdot \boldsymbol{\sigma}(\boldsymbol{u}) = 0 \tag{6.23}$$

with Cauchy stress tensor

$$\sigma_{ij}(\boldsymbol{u}) = 2\mu\varepsilon_{ij}(\boldsymbol{u}) + \lambda\varepsilon_{kk}(\boldsymbol{u}))\boldsymbol{\delta}_{ij}, \tag{6.24}$$

where the $\boldsymbol{\varepsilon}(\boldsymbol{u})_{ij} = \frac{1}{2}(u_{i,j} + u_{j,i})$ is the strain and μ and λ are Lamé parameters. Traction and symmetry boundary conditions are imposed

$$\boldsymbol{u} \cdot \mathbf{n} = 0 \quad \text{on } \Gamma_{\text{sym}} \text{ ,} \tag{6.25}$$

$$(\boldsymbol{I} - \boldsymbol{n} \otimes \boldsymbol{n})(\boldsymbol{\sigma} \cdot \mathbf{n}) = \mathbf{0} \quad \text{on } \Gamma_{\text{sym}} \text{ ,} \tag{6.26}$$

$$\boldsymbol{\sigma} \cdot \mathbf{n} = \boldsymbol{t} \quad \text{on } \Gamma_t. \tag{6.27}$$

Using Nitsche's method to enforce symmetry conditions, Navier's equation is discretized as Find $\boldsymbol{u}^h \in \mathscr{V}^h$ such that, $\forall \boldsymbol{v}^h \in \mathscr{V}^h$,

$$\begin{aligned}
&\int_\Omega \boldsymbol{\sigma}(\boldsymbol{u}^h) : \nabla \boldsymbol{v}^h \, d\Omega \\
&\quad - \int_{\Gamma_{\text{sym}}} (\boldsymbol{u}^h \cdot \mathbf{n})\mathbf{n} \cdot \boldsymbol{\sigma}(\boldsymbol{v}^h) \cdot \mathbf{n} \, d\Gamma - \int_{\Gamma_{\text{sym}}} (\boldsymbol{v}^h \cdot \mathbf{n})\mathbf{n} \cdot \boldsymbol{\sigma}(\boldsymbol{u}^h) \cdot \mathbf{n} \, d\Gamma \\
&\quad + \int_{\Gamma_{\text{sym}}} \frac{\beta\mu}{h} (\boldsymbol{u}^h \cdot \mathbf{n})\mathbf{n} \cdot \boldsymbol{v}^h \, d\Gamma = \int_{\Gamma_t} \boldsymbol{t} \cdot \boldsymbol{v}^h \, d\Gamma,
\end{aligned} \tag{6.28}$$

where $\beta > 0$ is the penalty parameter associated with Nitsche's method, once again set to 10. Exploiting symmetry, a 5×5 square with a quarter circle of radius $R = 1$ was modeled. For an infinite plate under an applied equal-biaxial strain, the resulting stress tensor components can be computed using Kirsch's equations as

$$\sigma_{rr} = \sigma_\infty \left(1 - \left(\frac{r}{R}\right)^2\right), \qquad \text{and} \qquad \sigma_{\theta\theta} = \sigma_\infty \left(1 + \left(\frac{r}{R}\right)^2\right). \tag{6.29}$$

The exact solution for the stress field, converted to Cartesian coordinates is applied as traction $\boldsymbol{t} = \boldsymbol{\sigma}_{ex} \cdot \mathbf{n}$.

Jittered grid RKPM pointsets with $\varepsilon = 0.5$ were generated without regard to the geometry of the hole/inclusion. A series of pointsets with average nodal spacing h were used to generate convergence data, with $h \in 0.625\{1, 0.5, 0.25, 0.125, 0.06125\}$

The boundary conforming foreground meshes shown in Figure 6.5 were generated using the open source software MORIS [120]. MORIS is an immersed finite element code that utilizes level-set geometry descriptions to enrich and integrate both classic lagrange and isogeometric basis functions. It also provides functionality to rapidly generate boundary conforming meshes with its level-set geometry processing sublibrary. In 2D, the algorithm begins with a uniform tensor-product grid, and then triangulates the quadrilateral cells intersected by the level-set isocountour indicating the domain boundary. After triangulation a root finding algorithm is used to locate the intersections of element edges the isocounter. Using the intersection points as new nodes, the cells are further triangulated. The analogous 3D process begins with tetrahedron cells that are decomposed into a boundary conforming mixed hexahedron-tetrahedron mesh.

Local foreground refinement is performed to improve geometric resolution and reduce geometry error. The algorithm to implement local refinement employs quadtree or (or octree in 3D) refinement of cells in the intial background grid prior to their triangulation (or subdivision into hexahedrons). If additional levels of refinement are required, the cells of the newly refined grid intersected by the level-set isocontour can be further identified and subdivided. For this example, three sets of foreground meshes are utilized, with no, one level, and two levels of local refinement, examples of which are shown in Figure 4.9.

The meshes generated for this purpose are not suitable for classical FEM as they are poorly conditioned and may contain elements with arbitarily large aspect ratios. Additionally, the meshes generated with local refinement contain hanging nodes which are not supported by most FEM software. While the poor conditioning of the elements on the foreground mesh presents no challenge for interpolation based methods, the hanging nodes are a different story. Function spaces defined on meshes with hanging nodes are C^{-1} continuous, and are not suitable for approximating PDEs with standard Galerkin's method. While the previous numerical example in Section 6.2.2 utilized a non-conforming function space, it was shown that the interpolation error needed to be minimized with foreground refinement to achieve optimal convergence rates. The problem presented by a C^{-1} foreground mesh is similar, but cannot be overcome with foreground refinement.

Instead, a novel 'double interpolation' strategy is employed. A new C^0 'midground' mesh and function space $\mathscr{V}^{mg} = \text{span}\{N_i^{mg}\}_{i=1}^{\mu}$ is introduced. The midground mesh is selected such that the foreground mesh is 'midground fitted', meaning the domain of each cell on the midground can be exactly represented as the union of cells on the foreground mesh. This minimizes interpolation error between the midground and foreground mesh and insures the continuity of the interpolated basis. In this case, the midground mesh is taken as the initial grid used in the foreground mesh generation process.

The RKPM basis is first interpolated with functions from the midground space, and then interpolated again onto the foreground space

$$\hat{\Psi}_I(\boldsymbol{x}) = \sum_j^{\mu} M_{Ij}^1 N_j^{mg}(\boldsymbol{x}) = \sum_j^{\mu} \sum_k^{\nu} M_{Ij}^1 M_{jk}^2 N_k(\boldsymbol{x}), \tag{6.30}$$

where

$$M_{Ij}^1 = \Psi_I(\boldsymbol{x}_j^{mg}) \qquad \text{and} \qquad M_{ik}^2 = N_i^{mg}(\boldsymbol{x}_j) \tag{6.31}$$

are the new extraction operators.

The impact of this double refinement strategy on an individual shape function is illustrated in Figure 6.6.

The effects of both local refinement and double interpolation on the approximation PDEs is shown in Figure 4.10. The dotted lines depict results from the original interpolation strategy, while solid lines shown result with two levels of interpolation. With the original interpolation strategy, the discontinuities in the locally refined interpolated bases degrade error convergence rates. With the introduction of doubled interpolation, local refinement is sufficient to reduce geometric error and achieve ideal error convergence rates for both linear and quadratic interpolated RKPM functions.

6.2.4 Three-material square model problem

Heaviside enrichment is employed to modify RKPM for multi-material problems, following the enrichment scheme employed for immersed IGA in chapter 5 Section 5.1.2. As with the immersed setting, the material subdomains are described with characteristic functions ψ^m,

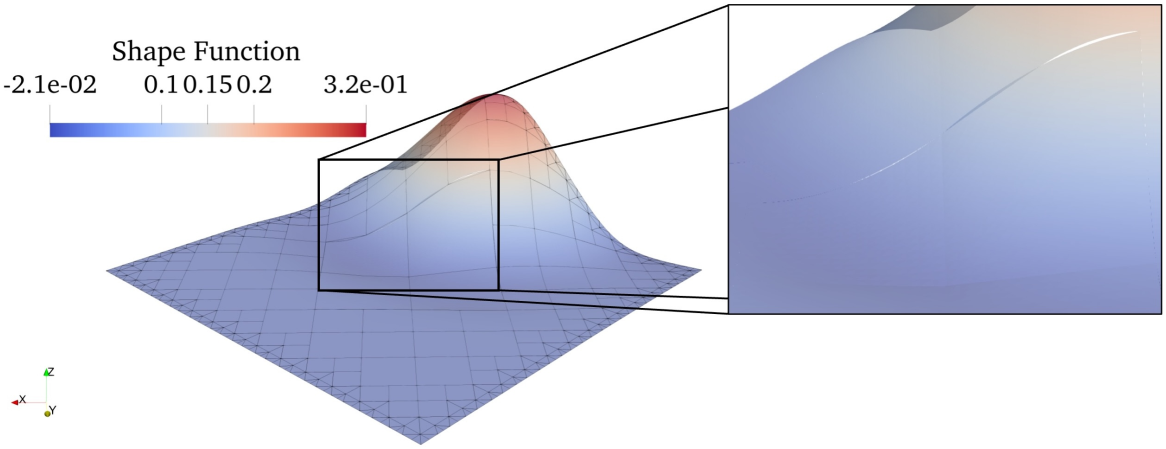
$$\psi^m(\boldsymbol{x}) = \begin{cases} 1, \text{ if } \boldsymbol{x} \in \Omega^m \\ 0, \text{ else.} \end{cases} \tag{6.32}$$

Each basis function is inspected to determine if its support is intersected by a material interface. If so, it is enriched. The enriched basis functions are expressed as

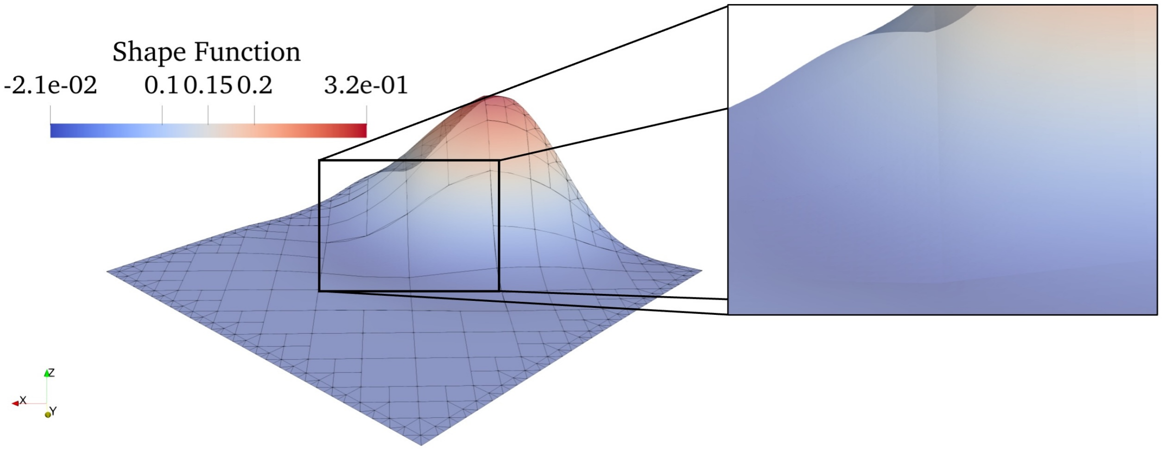
$$\Psi_I^m(\boldsymbol{x}) = \psi^m(\boldsymbol{x})\Psi(\boldsymbol{x}), \;\; \forall\, m \in \{1,...,L\}, \tag{6.33}$$

where L is the number of material subdomains. The original basis function can be recovered by the summation of these enriched functions,

$$\Psi_I(\boldsymbol{x}) = \sum_m \Psi_I^m(\boldsymbol{x}) \tag{6.34}$$



(a) A quadratic shape function is interpolated with a foreground function space defined on a mesh with hanging nodes, resulting in discontinuities.



(b) The same shape function is first interpolated with a C^0 midground function space, and then with the C^{-1} foreground space, eliminating discontinuities

Figure 6.6. Visualizations of an RKPM shape function interpolated with a C^{-1} function space, and in the two step process referred to as double interpolation using an intermediary C^0 midground space.

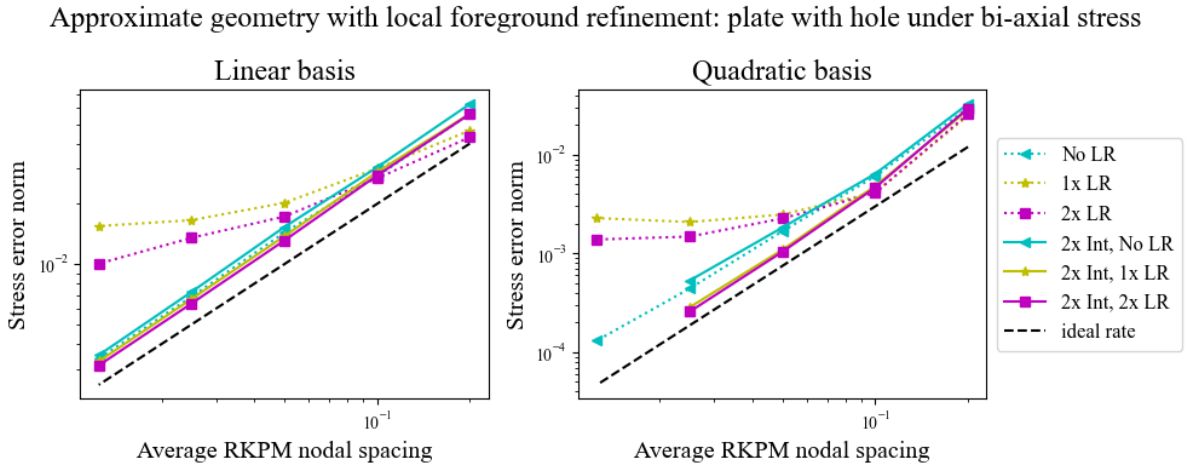


Figure 6.7. Convergence data for Int-RKPM applied to the linear elasticity problem of a plate with hole. Normal interpolation is used for the dotted lines, and double interpolation used for the solid lines

This section implements global enrichment as opposed to function-wise enrichment [146], requiring sufficiently small supports of basis functions in the vicinity of the interface.

For enriched Int-RKPM, the material characteristic functions are discretized with a piece-wise constant function space defined on the foreground integration mesh.

This enrichment scheme is tested on a multi-material domain using Poisson's problem. A 1x1 unit square is divided into 3 subdomains with interfaces at $x = 0.2$ and $x = 0.8$, such that the subdomains are defined

$$\begin{aligned} \boldsymbol{x} &\in \Omega^1, 0.0 \leq x \leq 0.2 \\ \boldsymbol{x} &\in \Omega^2, 0.2 < x \leq 0.8 \\ \boldsymbol{x} &\in \Omega^3, 0.8 < x \leq 1.0. \end{aligned} \tag{6.35}$$

The subdomains are assigned material properties $\kappa(\boldsymbol{x}) = \kappa^m$, $\boldsymbol{x} \in \Omega^m$, with $\kappa^1 = 1.0$, $\kappa^2 = 0.5$, and $\kappa^2 = 1.0$, and are shown in Figure 6.8(a).

A source term $f : \Omega \rightarrow \mathbb{R}$ and Dirichlet boundary data $\overline{T} : \partial\overline{\Omega} \rightarrow \mathbb{R}$ on $\Gamma_{\overline{T}} \subset \partial\overline{\Omega}$ are ascribed. The strong form for the thermal problem then reads as: Find $T : \Omega \rightarrow \mathbb{R}$ such that $\forall$

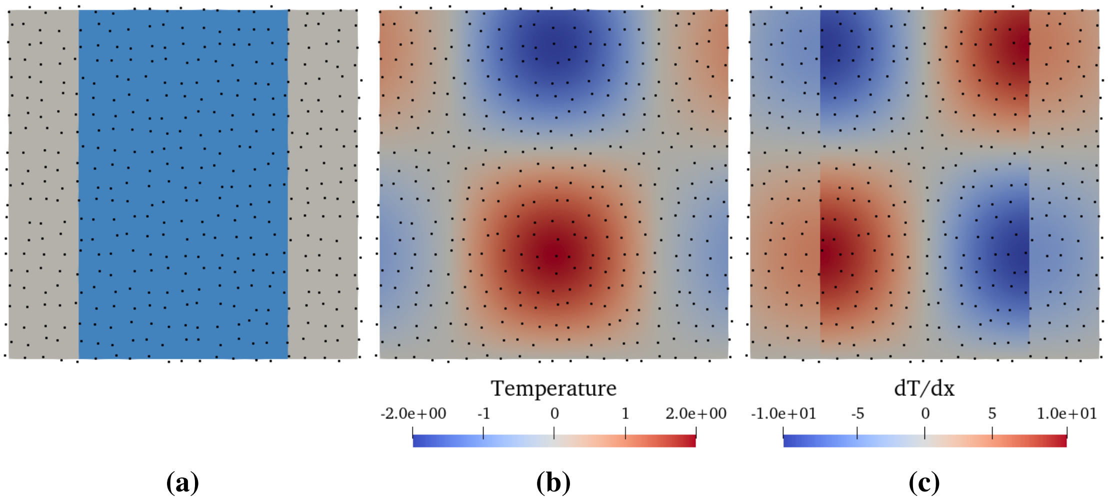


Figure 6.8. The subdomains, exact solution T_{ex}, and the x-derivative of the exact solution. Black dots show the nodal locations of the RKPM basis functions.

$m \in \mathscr{M}$

$$\begin{aligned} -\nabla\cdot(\kappa(\boldsymbol{x})\nabla T) &= f \quad \text{in } \Omega^m, \\ [\![T]\!] &= 0 \quad \text{on all } \Gamma_{km}, \\ [\![\boldsymbol{q}]\!] &= 0 \quad \text{on all } \Gamma_{km}, \\ T &= \overline{T} \quad \text{on } \Gamma_{\overline{T}}^m, \end{aligned} \tag{6.36}$$

where $\Gamma_{km} = \overline{\Omega}^k \cap \overline{\Omega}^m \neq \emptyset$, with $k \in \mathscr{M}$ and $k \neq m$, are the material interfaces, and $[\![\cdot]\!] = (\cdot)^k - (\cdot)^m$ is the jump of a given quantity over an interface Γ_{km}. The material fields are defined $T^m = T(\boldsymbol{x})$, $\boldsymbol{x} \in \Omega^m$, and $\boldsymbol{q}^m = -\kappa^m \nabla T^m$. The domain $\Gamma_{\overline{T}}^m = \Gamma_{\overline{T}} \cup \partial\overline{\Omega}^m$ are the intersections of the domain boundaries with the material subdomain boundaries. $\boldsymbol{n}$ denotes the surface normal.

For this example the method of manufactured solutions is employed using the exact solution

$$T_{ex}(\boldsymbol{x}) = \frac{1}{\kappa} \sin\frac{5\pi(x-0.2)}{3} \sin\frac{5\pi y}{3}, \tag{6.37}$$

which is constructed to be strongly continuous over the domain, but weakly discontinuous at material interfaces. T_{ex} and its x-derivative are shown in Figure 6.8. T_{ex} is ascribed as Dirichlet boundary data on the domain boundary. The source term is thus

$$f = -\nabla \cdot (\kappa(\boldsymbol{x})\nabla T_{ex}) = \frac{25\pi^2}{9}\sin\frac{5\pi(x-0.2)}{3}\sin\frac{5\pi y}{3}. \tag{6.38}$$

The discrete form can be defined as: Find $T^h \in \mathscr{V}^T$ such that $\forall \theta^h \in \mathscr{V}^h$,

$$\sum_{m=1}^{n}\left[\int_{\Omega^m}\kappa\nabla T^h \cdot \nabla\theta^h d\Omega\right] - \int_{\Omega} f\theta^h d\Omega = \mathscr{R}_T^D \ , \tag{6.39}$$

where $\mathscr{R}_T^D$ is the Nitsche's method residual for enforcing Dirichlet boundary conditions,

$$\mathscr{R}_T^D = \sum_{m=1}^{n}\left[\mp\int_{\Gamma_{\overline{T}}^m}\kappa(T^h-\overline{T})(\nabla\theta^h\cdot\boldsymbol{n})\,d\Gamma \quad - \right.$$
$$\left.\int_{\Gamma_{\overline{T}}^m}\kappa\theta^h(\nabla T^h\cdot\boldsymbol{n})\,d\Gamma \quad + \int_{\Gamma_{\overline{T}}^m}\frac{\beta_T^D\kappa}{h}(T^h-\overline{T})\theta^h\,d\Gamma\right], \tag{6.40}$$

h is taken as the foreground mesh size, and β_T^D is a user specified constant. For this example $\beta_T^D = 10$. Heaviside enrichment results in strongly discontinuous basis functions, thus continuity of the solution field must be enforced at the interfaces.This is done through the addition of residual $\mathscr{R}_T^I$ to equation 6.40. The residual is constructed as

$$\mathscr{R}_T^I = \sum_{i=1}^{n}\sum_{j=i+1}^{n}\left[-\int_{\Gamma_{ij}}[\![T^h]\!]\{\kappa\nabla\theta^h\}\cdot\boldsymbol{n})\,d\Gamma\right.$$
$$-\int_{\Gamma_{ij}}[\![\theta^h]\!]\{\kappa\nabla T^h\}\cdot\boldsymbol{n})\,d\Gamma$$
$$\left.+\int_{\Gamma_{ij}}\gamma_T^{ij}[\![T^h]\!][\![\theta^h]\!]\,d\Gamma\right], \tag{6.41}$$

where $\{\cdot\} = w^i(\cdot)^i - w^j(\cdot)^j$ is the weighted average.

The function space used to discretize the temperature field is constructed of RKPM shape functions. The shape functions, of polynomial order $n = 1$ or $n = 2$, have circular supports of normalized radius $a = n + 1$. The support radius is normalized by each domain kernel spacing, computed using the distance between the node and its fourth farthest neighbor. Cubic splines are used as kernel functions ϕ. Interpolation is performed with a boundary conforming quadrilateral mesh, with element sizes equal to nodal spacing and equal order foreground polynomials ($k = n$).

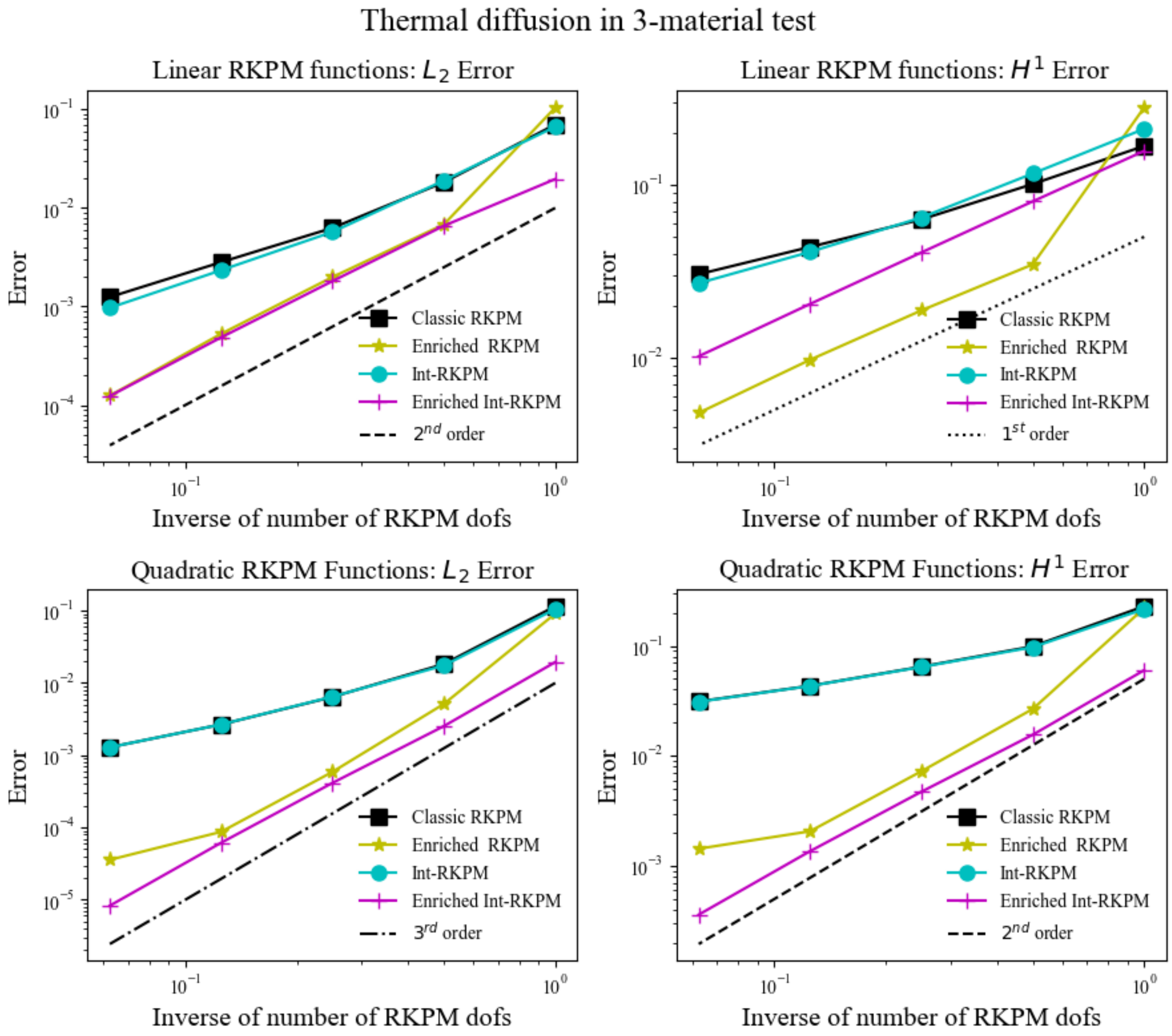
For classic RKPM, integration is performed using Gauss quadrature points defined upon a uniform grid or equal resolution to the RKPM-nodal spacing. A 6×6 Gauss point rule was employed for linear basis functions and an 8×8 Gauss point rule was employed for quadratic basis functions. For the enriched classic RKPM analysis, the exact characteristic function for each material interface was used from the domain description in equation 6.35.

The results of this convergence study are shown in Figure 6.9. Classic RKPM and Int-RKPM behavd almost identically in the presence of solution discontinuities, see the errors plotted along the line $y = 0.3$ in Figure 6.10. Without enrichment, suboptimal error convergence rates are observed due to solution errors near the discontinuities.

With Enriched RKPM and Enriched-Int RKPM, ideal error convergence rates are achieved for $n = 1$, with Enriched-Int RKPM having slightly worse error magnitude in the H^1 error norm, following what was observed in subsection 6.2.1. For $n = 2$, Enriched RKPM demonstrates ideal error convergence rates in the pre-asymptotic regime, but rates increase with refinement due to slight errors near the discontinuities (see the yellow line in Figure 6.10). Enriched Int-RKPM provides a smoother solution space and produces ideal error convergence rates in the asymptotic regime.

6.2.5 Material interfaces with irregular geometries

Employing both the enrichment strategy introduced in Section 6.2.4 and the local refinement techniques from Section 6.2.3, Int-RKPM is extended to multi-material PDEs with irregularly shaped interfaces. This capability is verified with a classic benchmarking problem: a circular inclusion is embedded in a semi-infinite medium and a uniform isotropic eigenstrain is



Thermal diffusion in 3-material test

Linear RKPM functions: L_2 Error

Linear RKPM functions: H^1 Error

Quadratic RKPM Functions: L_2 Error

Quadratic RKPM Functions: H^1 Error

Figure 6.9. Convergence data for the three material heat conduction problem, comparing .

imposed upon the inclusion. The formulation of this problem is given in this thesis in chapter 5 Section 5.3.2 and appendix 5.A.

The weakly discontinuous analytic solution for the radial displacement is given in [161] as

$$u_r = \begin{cases} C_1 r, & r \leq R, \\ C_1 \dfrac{R^2}{r}, & r \geq R, \end{cases} \tag{6.42}$$

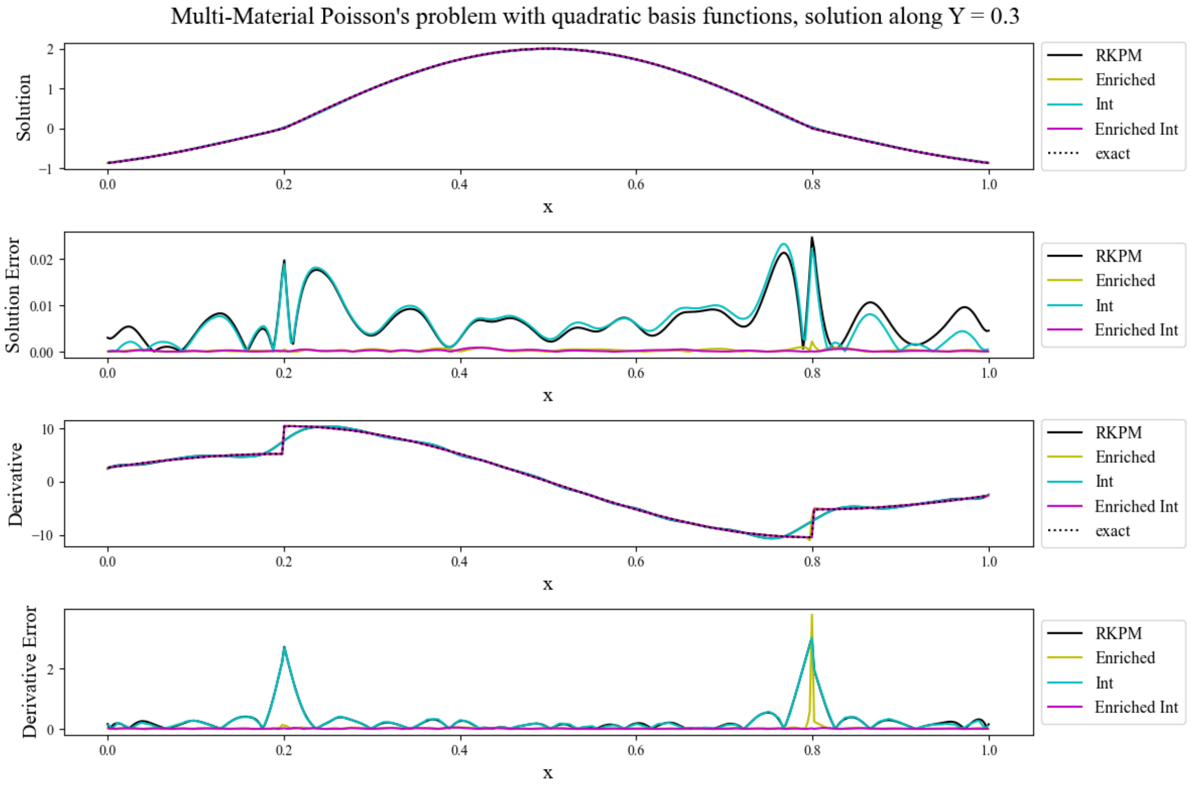


Figure 6.10. Enriched RKPM and classic RKPM are compared with their interpolation counterparts. Enriched RKPM and Enriched Int-RKPM accurately resolves the weakly discontinuous temperature field.

where R is the radius of the inclusion and

$$C_1 = \frac{(\lambda_1 + \mu_1)\varepsilon_0}{\lambda_1 + \mu_1 + \mu_2}. \tag{6.43}$$

For this example, the inclusion is comprised of Material 1 with Lamé constants $\lambda_1 = 497.16$ and $\mu_1 = 390.63$, while the exterior plate is made of Material 2 with $\lambda_2 = 656.79$, and $\mu_2 = 338.35$. The eigenstrain $\varepsilon_0 = 0.1$ is imposed on the inclusion. The same geometric discretizations and RKPM functions used in Section 6.2.3 of this chapter are employed here, such that $R = 1$. To approximate an infinite domain, the exact solution is imposed as Dirichlet boundary data on the top and right of the domain, and symmetry conditions are enforced on the left and bottom.

With enrichment and double interpolation, the midground mesh must also be enriched to accommodate for discontinuities within basis functions. As with the background RKPM functions, midground shape functions with support straddling the material interface are identified. By restricting the midground function space to discontinous Galerkin type spaces, a relationship may be developed associating each cell with a unique set of degrees of freedom. Thus all cells on the midground mesh ω_e^{mg}, covered by triangular elements on the foreground mesh will support basis functions requiring enrichment.

Using the characteristic functions given in Equation 6.32, the midground functions are enriched such that

$$N_i^{mg,m} = \psi^m(\boldsymbol{x}) N_i^{mg}, \qquad \forall m \in \{1,...,L\}. \tag{6.44}$$

As this will add additional degrees of freedom to the midground basis, an additional set of characteristic functions must be defined to facilitate enrichment of the background RKPM basis:

$$\psi_{mg}^m(\boldsymbol{x}) = \begin{cases} 1, \text{ if } \boldsymbol{x} \in \{\omega_e^{mg}\}_m \\ 0, \text{ else,} \end{cases} \tag{6.45}$$

where $\{\omega_e^{mg}\}_m$ is the set of all elements in the midground mesh covered by quadrilateral elements in Ω^m on the foreground mesh or by triangular elements on the foreground mesh.

The foreground and midground characteristic functions are illustrated for the circular inclusion problem in Figure 6.11.

Thus enriched RKPM functions being evaluated for double interpolation are expressed as

$$\Psi_I^m(\boldsymbol{x}) = \psi_{mg}^m(\boldsymbol{x}) \Psi(\boldsymbol{x}), \ \forall \, m \in \{1,...,L\}. \tag{6.46}$$

Note that as the midground material indicator functions may overlap, the summation of enriched RKPM functions is no longer equivalent to the original function.

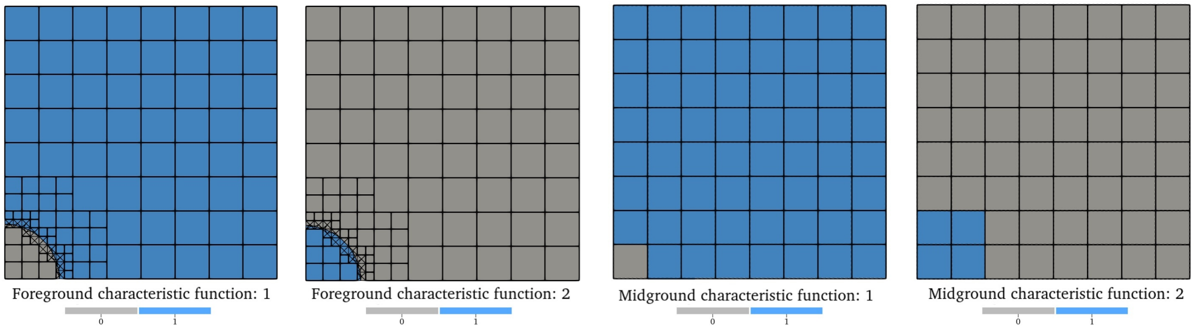

Foreground characteristic function: 1
0
1
Foreground characteristic function: 2
0
1
Midground characteristic function: 1
0
1
Midground characteristic function: 2
0
1

Figure 6.11. The material subdomain indicator functions discretized on both the foreground and midground meshes. Foreground meshes use two levels of local refinement.

With enrichment and double interpolation, ideal error convergence rates are observed for both linear and quadratic functions. Local refinement is required to reduce geometric error to below method error for quadratic functions. This problem validates the use of Int-RKPM for multi-material PDEs on irregular geometries.

Chapter summary

This chapter introduces a novel interpolation-based reproducing kernel particle method (Int-RKPM). This method is implemented through the open-source finite element software FEniCSx. Classical RKPM shape functions are interpolated as linear combinations of Lagrange polynomial shape functions. As with most meshfree methods, a foreground mesh is required for integration purposes, but it not subject to the typical quality constraints of classic finite element analysis. This foreground mesh is used to create a foreground Lagrange polynomial basis for interpolation of the RKPM basis. The interpolated basis retains the vital properties of the original RKPM basis, and is suitable for the approximation of PDEs, as illustrated in the numerical examples.

The first numerical example demonstrates the error convergence rates of a Poisson problem, comparing classical RKPM (integrated using high-order Gauss quadrature schemes) with the Int-RKPM methods. It is shown that a quadratic foreground basis of equal spatial

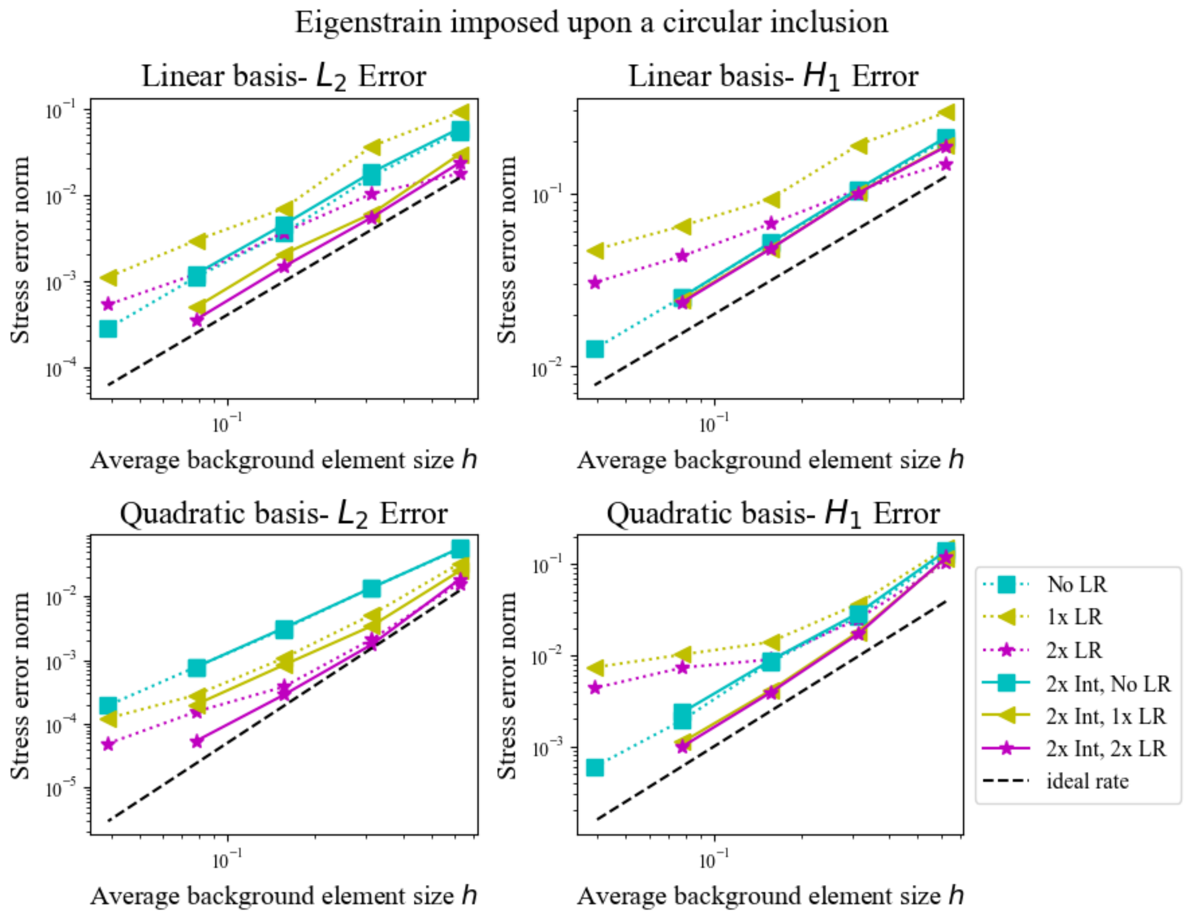


Figure 6.12. Convergence data for eigenstrain imposed upon a circular inclusion, comparing both local refinement and the classic interpolation and double interpolation techniques.

refinement to the background nodal spaceing is sufficient for the Int-RKPM results to match the error magnitude of the classic RKPM. The biharmonic numerical example demonstrate the effective continuity of the Int-RKPM basis is sufficient to solve fourth order PDEs, which is not possible with classic FEA. Int-RPKPM is then tested with foreground meshes possessing hanging nodes, resulting from quadtree local refinement strategies. A novel double interpolation strategy is implemented to ensure C^0 continuity of the interpolated basis, and with this strategy local refinement is implemented to reduce geometric error for problems with irregular domain boundaries. Int-RKPM is implemented with Heaviside enrichment to solve multi-material PDEs, as demonstrated by the fourth numerical example. Enrichment is also implemented with local foreground refinement, using the new double interpolation technique.

Acknowledgements

The above chapter, in full, is a reprint of the material under prepartion by Fromm, Jennifer E; Evans, John A; and Chen, Jiun-Shyan; “Interpolation-based reproducing kernel particle methods” 2024. The dissertation author is the primary researcher and author of this work. The authors of this work, including the dissertation author, were supported by National Science Foundation award number 2104106.

Chapter 7

Conclusions and future work

7.1 Conclusion

This thesis introduces a new paradigm for defining and implementing high-order-accurate immersed-boundary methods, immersogeometric methods, and RKPM. These methods are based on interpolating basis functions defined on a boundary-unfitted background discretization into a function space defined on a foreground mesh that is fitted to the domain boundary but subject to fewer topological or geometrical constraints than a standard FEA mesh.

Compared with existing variational immersed methods that use foreground meshes to define quadrature rules for functions on background meshes, interpolation-based immersed finite element analysis provides more flexibility with respect to the construction of the foreground mesh and more straightforward reuse of existing FE software. The interpolation-based methodology is capable of exactly reproducing quadrature-based immersed methods under certain conditions, but it also enables efficient approximations that reduce computational cost with no significant loss of accuracy.

The extension of the proposed interpolation-based immersed method to multi-material and multi-physics also demonstrates significant advantages to traditional multi-material and multi-physics methods. The more geometrically complex a multi-material interface is, the more expensive traditional immersed boundary methods become, and thus the greater the reduction in computational cost from employing interpolation. The interpolation-based method also employs a single integration mesh that can be used for coupling multiple fields using different background discretizations.

Interpolation-based RKPM is perhaps the most novel contribution within this dissertation. Int-RKPM is able to reproduce the results of classical RKPM for several fundamental benchmarking problems, with considerable computational savings. Viewed as a new integration technique, it has the potential to increase access to meshfree methods through implementation workflows within existing FE software. Taken as a new method for approximating the derivatives of RKPM functions, the potential applications are more numerous.

The interpolation-based methods presented in this thesis require the construction of a foreground mesh and execution of an assembly loop over its elements. An engineer prioritizing simplicity may then ask, why not just directly solve the system resulting from this foreground discretization? The preceding sections contain several different answers to this question, which are now compiled into an itemized summary of the advantages of the various interpolation-based methods over classical body-fitted FE analysis using the foreground mesh:

- The linear system resulting from the foreground discretization would be singular in many problems of interest, e.g., the fourth-order biharmonic (Sections 4.3.2 and 6.2.2) and Kirchhoff–Love (Section 4.4) problems. These problems can only be solved with sufficiently continuous basis functions, such as quadratic B-splines or RKPM functions employing quadratic B-spline kernels.

- Even for second-order problems, one might wish to use a non-conforming approximation of H^1, e.g., a foreground mesh containing hanging nodes, whose construction is subject to fewer topological constraints. This demonstrated in Sections 4.3.3, 5.4, and 6.2.3, which employ mixed element type locally refined foreground meshes with hanging nodes. These types of foreground meshes are not easily discretized by most FEA software, and not at all supported outside the interpolation framework within FEniCS or FEniCSx.

- Even if the domain Ω is easy to mesh in a conforming way, one may still wish to approximate PDE solutions with functions such as B-splines that are only convenient to construct on a tensor-product grid.

- The numerical results of Section 4.3.1 demonstrate that interpolation-based immersed methods consistently require fewer degrees of freedom for the same accuracy when compared to classical body-fitted FE analysis using the foreground mesh (in cases where the latter is well-posed at all).
- Many of the attractive properties of RKPM not available in classical FEA can still be exploited in the interpolated setting. For example local refinement, or the arbiatray addition or subtraction of individual basis functions can be performed without altering the foreground integration mesh. Additionally, the number of degrees of freedom in an RKPM approximation does not increase with increasing polynomial reproducibility, which dramatially reducing the size of the resulting linear system when compared with classical FEA.

The non-invasive implementation of interpolation-based immersed methods discussed in Section 4.2 is of great practical significance, as implementation cost has been a major barrier to widespread adoption and application of immersed methods proposed in the academic literature. The FEniCS-based prototype used for experiments appearing in chapter 4 is available online [69]. The dissertation author is the primary developer of this module. This tool is reccomended for users looking to interface with classic FEniCS modules, including the software library tIGAr. The next gen FEniCSx-based module used for the experiments appearing in chapter 5 is available online [68]. The dissertation author is the sole developer of this module. This tool is recommended for all new users of interpolation-based immersed methods.

The numerical experiments appearing in chapter 6 were also performed within FEniCSx. The python module developed for the implementation of Int-RKPM within FEniCSx will be made available on the dissertation author's github upon publication of the material appearing in chapter 6. This module is made up of two parts, one employing FEniCSx for Int-RKPM, and one with only standard python dependencies for classical RKPM.

7.2 Future work

This thesis has demonstrated the application of interpolation-based methods to several computational mechanics techniques. The dissertation author makes the following recommendations for additional avenues of research.

- Within FEniCSx, additions to the interpolation-based immersed boundary workflow will be the implementation of stabilization techniques to address issues of linear conditioning. Unpublished work has investigated the addition of basis agglomeration (introduced for CutFEM in in [60]) to improve conditioning. The implementation of ghost stabilization [37] could also increase the functionality of the interpolation-based method for unsteady or time dependent problems, where the critical time step can be several decreased by cut cells.

- Further characterization of the effects foreground basis quality on the behaviour of interpolated bases. This work has demonstrated that the behaviour and properties of interpolated bases are primarily dependent upon the properties of the background basis, but with the extension to purely background-unfitted extraction in Chapter 6, users have greater freedom to alter the foreground mesh and basis defined with it.

- Hand in hand with the previous bullet, further characterization of the properties of bases interpolated with C^0 foreground spaces, and their use for higher order PDEs, is still lacking. While this is justified in the case of exact interpolation, it is unclear how this is possible with in the cases of approximate interpolation.

- Extension of the discontinuity modeling capabilites of Int-RKPM, in particular investigating the potential of kernel scaling [157, 162] to represent discontinuities as opposed to enrichment. Kernel scaling has the potential to represent either weak or strong discontinuities, avoiding the necessities of Nitsches interface conditions. A ‘purely meshfree’ implementation, kernel scaling also results in smaller linear systems than enrichment as it adds no additional degrees of freedom. This also allows for easier bookkeeping and implementation. Kernel scaling and enrichment also enable Int-RKPM to model strongly discontinuous problems such as fracture. The idea of utilizing Int-RKPM for fracture requires one to grapple with the largest disadvantage of interpolation-based methods: the requirement of a boundary conforming foreground mesh. The dissertation author leaves these questions to future researchers.

Bibliography

[1] Martin Alnæs, Jan Blechta, Johan Hake, August Johansson, Benjamin Kehlet, Anders Logg, Chris Richardson, Johannes Ring, Marie E. Rognes, and Garth N. Wells. The FEniCS Project Version 1.5. *Archive of Numerical Software*, 3(100), December 2015. Number: 100.

[2] Chandrasekhar Annavarapu, Martin Hautefeuille, and John E. Dolbow. A robust Nitsche's formulation for interface problems. *Computer Methods in Applied Mechanics and Engineering*, 225-228:44–54, June 2012.

[3] Nabil M. Atallah, Claudio Canuto, and Guglielmo Scovazzi. The high-order Shifted Boundary Method and its analysis. *Computer Methods in Applied Mechanics and Engineering*, 394:114885, May 2022.

[4] F. Auricchio, F. Calabrò, T. J. R. Hughes, A. Reali, and G. Sangalli. A simple algorithm for obtaining nearly optimal quadrature rules for NURBS-based isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 249-252:15–27, December 2012.

[5] I. Babuška and J. M. Melenk. The Partition of Unity Method. *International Journal for Numerical Methods in Engineering*, 40(4):727–758, 1997.

[6] Ivo Babuška. The finite element method with Lagrangian multipliers. *Numerische Mathematik*, 20(3):179–192, June 1973.

[7] Ivo Babuška. The finite element method with penalty. *Mathematics of Computation*, 27(122):221–228, 1973.

[8] Santiago Badia, Eric Neiva, and Francesc Verdugo. Robust high-order unfitted finite elements by interpolation-based discrete extension, August 2022. arXiv:2201.06632 [cs, math].

[9] Jonghyuk Baek and Jiun-Shyan Chen. A neural network-based enrichment of reproducing kernel approximation for modeling brittle fracture. *Computer Methods in Applied Mechanics and Engineering*, 419:116590, February 2024.

[10] Jonghyuk Baek, Jiun-Shyan Chen, and Kristen Susuki. A neural network-enhanced reproducing kernel particle method for modeling strain localization. *International*

Journal for Numerical Methods in Engineering, 123(18):4422–4454, 2022. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/nme.7040.

[11] Jonghyuk Baek, Yanran Wang, and Jiun-Shyan Chen. N-adaptive ritz method: A neural network enriched partition of unity for boundary value problems. *Computer Methods in Applied Mechanics and Engineering*, 428:117070, August 2024.

[12] S. Balay, S. Abhyankar, M. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, A. Dener, V. Eijkhout, W. Gropp, D. Karpeyev, D. Kaushik, M. Knepley, D. May, L. Curfman McInnes, R. Mills, T. Munson, K. Rupp, P. Sanan, B. Smith, S. Zampini, and H. Zhang. PETSc Users Manual (Rev. 3.11). Technical Report ANL-95/11 Rev 3.11, Argonne National Laboratory (ANL), Argonne, IL (United States), March 2019.

[13] Satish Balay, S. Abhyankar, M. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, V. Eijkhout, W. Gropp, D. Kaushik, M. Knepley, L. C. McInnes, K. Rupp, B. F. Smith, S. Zampini, and H. Zhang. PETSc Web Page, 2015.

[14] Satish Balay, William D. Gropp, Lois Curfman McInnes, and Barry F. Smith. Efficient Management of Parallelism in Object-Oriented Numerical Software Libraries. In Erlend Arge, Are Magnus Bruaset, and Hans Petter Langtangen, editors, *Modern Software Tools for Scientific Computing*, pages 163–202. Birkhäuser, Boston, MA, 1997.

[15] Igor A. Baratta, Joseph P. Dean, Jørgen S. Dokken, Michal Habera, Jack S. Hale, Chris N. Richardson, Marie E. Rognes, Matthew W. Scroggs, Nathan Sime, and Garth N. Wells. DOLFINx: The next generation FEniCS problem solving environment, December 2023.

[16] M. Bauccio and A. S. M. International. *ASM Engineered Materials Reference Book*. ASM International, 1994.

[17] 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):173–201, December 2007.

[18] Y. Bazilevs, V. M. Calo, T. J. R. Hughes, and Y. Zhang. Isogeometric fluid-structure interaction: theory, algorithms, and computations. *Computational Mechanics*, 43(1):3–37, December 2008.

[19] Y. Bazilevs, M.-C. Hsu, J. Kiendl, R. Wüchner, and K.-U. Bletzinger. 3D simulation of wind turbine rotors at full scale. Part II: Fluid–structure interaction modeling with composite blades. *International Journal for Numerical Methods in Fluids*, 65(1-3):236–253, 2011.

[20] Y. Bazilevs, M. C. Hsu, and M. A. Scott. Isogeometric fluid–structure interaction analysis with emphasis on non-matching discretizations, and with application to wind turbines. *Computer Methods in Applied Mechanics and Engineering*, 249-252:28–41, December 2012.

[21] Y. Bazilevs and T. J. R. Hughes. Weak imposition of Dirichlet boundary conditions in fluid mechanics. *Computers & Fluids*, 36(1):12–26, January 2007.

[22] Roland Becker, Erik Burman, and Peter Hansbo. A Nitsche extended finite element method for incompressible elasticity with discontinuous modulus of elasticity. *Computer Methods in Applied Mechanics and Engineering*, 198(41):3352–3360, September 2009.

[23] L. Beirão da Veiga, A. Buffa, J. Rivas, and G. Sangalli. Some estimates for h–p–k-refinement in Isogeometric Analysis. *Numerische Mathematik*, 118(2):271–305, June 2011.

[24] Stephen Beissel and Ted Belytschko. Nodal integration of the element-free Galerkin method. *Computer Methods in Applied Mechanics and Engineering*, 139(1):49–74, December 1996.

[25] T. Belytschko and T. Black. Elastic crack growth in finite elements with minimal remeshing. *International Journal for Numerical Methods in Engineering*, 45(5):601–620, 1999.

[26] T. Belytschko, Y. Y. Lu, and L. Gu. Element-free Galerkin methods. *International Journal for Numerical Methods in Engineering*, 37(2):229–256, 1994. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/nme.1620370205.

[27] T. Belytschko, Y. Y. Lu, and L. Gu. Crack propagation by element-free Galerkin methods. *Engineering Fracture Mechanics*, 51(2):295–315, May 1995.

[28] T. Belytschko, N. Moës, S. Usui, and C. Parimi. Arbitrary discontinuities in finite elements. *International Journal for Numerical Methods in Engineering*, 50(4):993–1013, 2001.

[29] Ted Belytschko, J. S. Chen, and Michael Hillman. *Meshfree and Particle Methods: Fundamentals and Applications*. wiley, January 2023.

[30] Joseph Benzaken, John A. Evans, Stephen F. McCormick, and Rasmus Tamstorf. Nitsche's method for linear Kirchhoff–Love shells: Formulation, error analysis, and verification. *Computer Methods in Applied Mechanics and Engineering*, 374:113544, February 2021.

[31] Paul T. Boggs, Alan Althsuler, Alex R. Larzelere, Edward J. Walsh, Ruuobert L. Clay, and Michael F. Hardwick. DART system analysis. Technical Report SAND2005-4647, Sandia National Laboratories (SNL), Albuquerque, NM, and Livermore, CA (United States), August 2005.

[32] Thomas Boiveau, Erik Burman, and Susanne Claus. Penalty-Free Nitsche Method for Interface Problems. In Stéphane P. A. Bordas, Erik Burman, Mats G. Larson, and Maxim A. Olshanskii, editors, *Geometrically Unfitted Finite Element Methods and Applications*, Lecture Notes in Computational Science and Engineering, pages 183–210, Cham, 2017. Springer International Publishing.

[33] Master Bond. Overview of materials for Epoxy Cure Resin, 2024.

[34] Michael J. Borden, Michael A. Scott, John A. Evans, and Thomas J. R. Hughes. Isogeometric finite element data structures based on Bézier extraction of NURBS. *International Journal for Numerical Methods in Engineering*, 87(1-5):15–47, 2011.

[35] J. Brackbill and H.M. Ruppel. FLIP: A method for adaptively zoned, particle-in-cell calculations of fluid flows in two dimensions. *Journal of Computational Physics*, 65:314–343, August 1986.

[36] F. Brezzi. On the existence, uniqueness and approximation of saddle-point problems arising from lagrangian multipliers. *Revue française d'automatique, informatique, recherche opérationnelle. Analyse numérique*, 8(R2):129–151, 1974. Number: R2 Publisher: EDP Sciences.

[37] Erik Burman. Ghost penalty. *Comptes Rendus Mathematique*, 348(21):1217–1220, November 2010.

[38] Erik Burman, Susanne Claus, Peter Hansbo, Mats G. Larson, and André Massing. CutFEM: Discretizing geometry and partial differential equations. *International Journal for Numerical Methods in Engineering*, 104(7):472–501, 2015.

[39] Erik Burman, Peter Hansbo, Mats G. Larson, and Karl Larsson. Extension operators for trimmed spline spaces. *Computer Methods in Applied Mechanics and Engineering*, 403:115707, January 2023.

[40] Hugo Casquero, Xiaodong Wei, Deepesh Toshniwal, Angran Li, Thomas J. R. Hughes, Josef Kiendl, and Yongjie Jessica Zhang. Seamless integration of design and Kirchhoff–Love shell analysis using analysis-suitable unstructured T-splines. *Computer Methods in Applied Mechanics and Engineering*, 360:112765, March 2020.

[41] Jiun-Shyan Chen, Michael Hillman, and Sheng-Wei Chi. Meshfree Methods: Progress Made after 20 Years. *Journal of Engineering Mechanics*, 143(4):04017001, April 2017. Publisher: American Society of Civil Engineers.

[42] Jiun-Shyan Chen, Michael Hillman, and Marcus Rüter. An arbitrary order variationally consistent integration for Galerkin meshfree methods. *International Journal for Numerical Methods in Engineering*, 95(5):387–418, 2013. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/nme.4512.

[43] Jiun-Shyan Chen, Chunhui Pan, Cheng-Tang Wu, and Wing Kam Liu. Reproducing Kernel Particle Methods for large deformation analysis of non-linear structures. *Computer Methods in Applied Mechanics and Engineering*, 139(1):195–227, December 1996.

[44] Jiun-Shyan Chen, Cheng-Tang Wu, Sangpil Yoon, and Yang You. A stabilized conforming nodal integration for Galerkin mesh-free methods. *International Journal for Numerical Methods in Engineering*, 50(2):435–466, 2001. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/1097-0207%2820010120%2950%3A2%3C435%3A%3AAID-NME32%3E3.0.CO%3B2-A.

[45] Jiun-Shyan Chen, Sangpil Yoon, and Cheng-Tang Wu. Non-linear version of stabilized conforming nodal integration for Galerkin mesh-free methods. *International Journal for Numerical Methods in Engineering*, 53(12):2587–2615, 2002. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/nme.338.

[46] Kwok Wah Cheng and Thomas-Peter Fries. Higher-order XFEM for curved strong and weak discontinuities. *International Journal for Numerical Methods in Engineering*, 82(5):564–590, 2010.

[47] B. Cockburn. Discontinuous Galerkin methods. *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik*, 83(11):731–754, 2003.

[48] Robert L. Cook. Stochastic sampling in computer graphics. *ACM Trans. Graph.*, 5(1):51–72, January 1986.

[49] Luca Coradello, Davide D'Angella, Massimo Carraturo, Josef Kiendl, Stefan Kollmannsberger, Ernst Rank, and Alessandro Reali. Hierarchically refined isogeometric analysis of trimmed shells. *Computational Mechanics*, 66(2):431–447, August 2020.

[50] Luca Coradello, Josef Kiendl, and Annalisa Buffa. Coupling of non-conforming trimmed isogeometric Kirchhoff–Love shells via a projected super-penalty approach. *Computer Methods in Applied Mechanics and Engineering*, 387:114187, December 2021.

[51] J. A. Cottrell, T.J.R. Hughes, and Y. Bazilevs. *Isogeometric Analysis*. John Wiley & Sons, Ltd, 1 edition, 2009.

[52] Carl de Boor. On calculating with B-splines. *Journal of Approximation Theory*, 6(1):50–62, July 1972.

[53] Frits de Prenter, Clemens V. Verhoosel, E. Harald van Brummelen, Mats G. Larson, and Santiago Badia. Stability and Conditioning of Immersed Finite Element Methods: Analysis and Remedies. *Archives of Computational Methods in Engineering*, 30(6):3617–3656, July 2023.

[54] Sai C. Divi, Clemens V. Verhoosel, Ferdinando Auricchio, Alessandro Reali, and E. Harald van Brummelen. Error-estimate-based adaptive integration for immersed isogeometric analysis. *Computers & Mathematics with Applications*, 80(11):2481–2516, December 2020.

[55] John Dolbow and Ted Belytschko. Numerical integration of the Galerkin weak form in meshfree methods. *Computational Mechanics*, 23(3):219–230, April 1999.

[56] G. Dziuk and C. M. Elliott. Finite elements on evolving surfaces. *IMA Journal of Numerical Analysis*, 27(2):262–292, April 2007.

[57] Gerhard Dziuk and Charles M. Elliott. Finite element methods for surface PDEs. *Acta Numerica*, 22:289–396, May 2013.

[58] 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):3768–3782, August 2008.

[59] Daniel Elfverson, Mats G. Larson, and Karl Larsson. CutIGA with basis function removal. *Advanced Modeling and Simulation in Engineering Sciences*, 5(1):6, March 2018.

[60] Daniel Elfverson, Mats G. Larson, and Karl Larsson. A new least squares stabilized Nitsche method for cut isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 349:1–16, June 2019.

[61] John A. Evans, Yuri Bazilevs, Ivo Babuška, and Thomas J. R. Hughes. n-Widths, sup–infs, and optimality ratios for the k-version of the isogeometric finite element method. *Computer Methods in Applied Mechanics and Engineering*, 198(21):1726–1741, May 2009.

[62] John Andrews Evans. *Divergence-free B-spline discretizations for viscous incompressible flows*. thesis, University of Texas at Austin, December 2011. Accepted: 2012-01-31T21:09:04Z.

[63] Leopoldo P. Franca and Sérgio L. Frey. Stabilized finite element methods: II. The incompressible Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 99(2):209–233, September 1992.

[64] J. Freund and R. Stenberg. On weakly imposed boundary conditions for second order problems. *Finite elements in fluids, Italia, 15-21.10.1995*, pages 327–336, 1995. Publisher: Padovan yliopisto.

[65] Thomas-Peter Fries and Ted Belytschko. The intrinsic partition of unity method. *Computational Mechanics*, 40(4):803–814, September 2007.

[66] Thomas-Peter Fries and Ted Belytschko. The extended/generalized finite element method: An overview of the method and its applications. *International Journal for Numerical Methods in Engineering*, 84(3):253–304, 2010.

[67] Thomas-Peter Fries and Samir Omerović. Higher-order accurate integration of implicit geometries. *International Journal for Numerical Methods in Engineering*, 106(5):323–371, 2016.

[68] Jennifer E. Fromm. jefromm/EXHUME_dolfinx, October 2023. original-date: 2023-10-31T01:39:34Z.

[69] Jennifer E. Fromm. jefromm/interpolation-based-immersed-fea, June 2024. original-date: 2022-08-11T16:53:31Z.

[70] Jennifer E. Fromm, Nils Wunsch, Kurt Maute, John A. Evans, and Jiun-Shyan Chen. Interpolation-based immersogeometric analysis methods for multi-material and multi-physics problems. *Computational Mechanics*, June 2024.

[71] Jennifer E. Fromm, Nils Wunsch, Ru Xiang, Han Zhao, Kurt Maute, John A. Evans, and David Kamensky. Interpolation-based immersed finite element and isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 405:115890, February 2023.

[72] Eduardo M. Garau and Rafael Vázquez. Algorithms for the implementation of adaptive isogeometric methods using hierarchical B-splines. *Applied Numerical Mathematics*, 123:58–87, January 2018.

[73] Carlotta Giannelli, Bert Jüttler, and Hendrik Speleers. THB-splines: The truncated basis for hierarchical splines. *Computer Aided Geometric Design*, 29(7):485–498, October 2012.

[74] P. C. Guan, J. S. Chen, Y. Wu, H. Teng, J. Gaidos, K. Hofstetter, and M. Alsaleh. Semi-Lagrangian reproducing kernel formulation and application to modeling earth moving operations. *Mechanics of Materials*, 41(6):670–683, June 2009.

[75] P. C. Guan, S. W. Chi, J. S. Chen, T. R. Slawson, and M. J. Roth. Semi-Lagrangian reproducing kernel particle method for fragment-impact problems. *International Journal of Impact Engineering*, 38(12):1033–1047, December 2011.

[76] David Gunderman, Kenneth Weiss, and John A. Evans. Spectral mesh-free quadrature for planar regions bounded by rational parametric curves, May 2020.

[77] David Gunderman, Kenneth Weiss, and John A. Evans. High-Accuracy Mesh-Free Quadrature for Trimmed Parametric Surfaces and Volumes. *Computer-Aided Design*, 141:103093, December 2021.

[78] Yujie Guo, Jason Heller, Thomas J. R. Hughes, Martin Ruess, and Dominik Schillinger. Variationally consistent isogeometric analysis of trimmed thin shells at finite deformations, based on the STEP exchange format. *Computer Methods in Applied Mechanics and Engineering*, 336:39–79, July 2018.

[79] Anita Hansbo and Peter Hansbo. An unfitted finite element method, based on Nitsche's method, for elliptic interface problems. *Computer Methods in Applied Mechanics and Engineering*, 191(47):5537–5552, November 2002.

[80] Anita Hansbo and Peter Hansbo. A finite element method for the simulation of strong and weak discontinuities in solid mechanics. *Computer Methods in Applied Mechanics and Engineering*, 193(33):3523–3540, August 2004.

[81] Anita Hansbo, Peter Hansbo, and Mats G. Larson. A finite element method on composite grids based on Nitsche's method. *ESAIM: Mathematical Modelling and Numerical Analysis*, 37(3):495–514, May 2003. Publisher: EDP Sciences.

[82] Austin J. Herrema, Emily L. Johnson, Davide Proserpio, Michael C. H. Wu, Josef Kiendl, and Ming-Chen Hsu. Penalty coupling of non-matching isogeometric Kirchhoff–Love shell patches with application to composite wind turbine blades. *Computer Methods in Applied Mechanics and Engineering*, 346:810–840, April 2019.

[83] Austin J. Herrema, Josef Kiendl, and Ming-Chen Hsu. A framework for isogeometric-analysis-based optimization of wind turbine blade structures. *Wind Energy*, 22(2):153–170, 2019.

[84] Michael Hillman and Jiun-Shyan Chen. An accelerated, convergent, and stable nodal integration in Galerkin meshfree methods for linear and nonlinear mechanics. *International Journal for Numerical Methods in Engineering*, 107(7):603–630, 2016. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/nme.5183.

[85] M. C. Hsu, Y. Bazilevs, V. M. Calo, T. E. Tezduyar, and T. J. R. Hughes. Improving stability of stabilized and multiscale formulations in flow simulations at small time steps. *Computer Methods in Applied Mechanics and Engineering*, 199(13):828–840, February 2010.

[86] Ming-Chen Hsu, David Kamensky, Fei Xu, Josef Kiendl, Chenglong Wang, Michael C. H. Wu, Joshua Mineroff, Alessandro Reali, Yuri Bazilevs, and Michael 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, June 2015.

[87] Tsung-Hui Huang, Jiun-Shyan Chen, Michael R. Tupek, Frank N. Beckwith, and H. Eliot Fang. A variational multiscale immersed meshfree method for fluid structure interactive systems involving shock waves. *Computer Methods in Applied Mechanics and Engineering*, 389:114396, February 2022.

[88] Tsung-Hui Huang, Haoyan Wei, Jiun-Shyan Chen, and Michael C. Hillman. RKPM2D: an open-source implementation of nodally integrated reproducing kernel particle method for solving partial differential equations. *Computational Particle Mechanics*, 7(2):393–433, March 2020.

[89] Simeon Hubrich and Alexander Düster. Adaptive numerical integration of broken finite cells based on moment fitting applied to finite strain problems. *PAMM*, 18(1):e201800089, 2018.

[90] 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):4135–4195, October 2005.

[91] T. J. R. Hughes, A. Reali, and G. Sangalli. Duality and unified analysis of discrete approximations in structural dynamics and wave propagation: Comparison of p-method finite elements with k-method NURBS. *Computer Methods in Applied Mechanics and Engineering*, 197(49):4104–4124, September 2008.

[92] 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):301–313, January 2010.

[93] Thomas J. R. Hughes, John A. Evans, and Alessandro Reali. Finite element and NURBS approximations of eigenvalue, boundary-value, and initial-value problems. *Computer Methods in Applied Mechanics and Engineering*, 272:290–320, April 2014.

[94] August Johansson, Mats G. Larson, and Anders Logg. Multimesh finite elements with flexible mesh sizes. *Computer Methods in Applied Mechanics and Engineering*, 372:113420, December 2020.

[95] Emily L. Johnson and Ming-Chen Hsu. Isogeometric analysis of ice accretion on wind turbine blades. *Computational Mechanics*, 66(2):311–322, August 2020.

[96] 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, October 2019.

[97] David Kamensky. Open-source immersogeometric analysis of fluid–structure interaction using FEniCS and tIGAr. *Computers & Mathematics with Applications*, 81:634–648, January 2021.

[98] David Kamensky. david-kamensky/ShNAPr, May 2024. original-date: 2019-08-23T21:51:23Z.

[99] David Kamensky and Yuri Bazilevs. tIGAr: Automating isogeometric analysis with FEniCS. *Computer Methods in Applied Mechanics and Engineering*, 344:477–498, February 2019.

[100] David Kamensky, John A. Evans, Ming-Chen Hsu, and Yuri Bazilevs. Projection-based stabilization of interface Lagrange multipliers in immersogeometric fluid–thin structure interaction analysis, with application to heart valve modeling. *Computers & Mathematics with Applications*, 74(9):2068–2088, November 2017.

[101] David Kamensky, Ming-Chen Hsu, Dominik Schillinger, John A. Evans, Ankush Aggarwal, Yuri Bazilevs, Michael S. Sacks, and Thomas 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, February 2015.

[102] David Kamensky, Fei Xu, Chung-Hao Lee, Jinhui Yan, Yuri Bazilevs, and Ming-Chen Hsu. A contact formulation based on a volumetric potential: Application to isogeometric simulations of atrioventricular valves. *Computer Methods in Applied Mechanics and Engineering*, 330:522–546, March 2018.

[103] J. Kiendl. *Isogeometric Analysis and Shape Optimal Design of Shell Structures*. PhD thesis, 2011.

[104] J. Kiendl, K. U. Bletzinger, J. Linhard, and R. Wüchner. Isogeometric shell analysis with Kirchhoff–Love elements. *Computer Methods in Applied Mechanics and Engineering*, 198(49):3902–3914, November 2009.

[105] Josef Kiendl, Ming-Chen Hsu, Michael C. H. Wu, and Alessandro Reali. Isogeometric Kirchhoff–Love shell formulations for general hyperelastic materials. *Computer Methods in Applied Mechanics and Engineering*, 291:280–303, July 2015.

[106] Hyun-Jung Kim, Yu-Deok Seo, and Sung-Kie Youn. Isogeometric analysis for trimmed CAD surfaces. *Computer Methods in Applied Mechanics and Engineering*, 198(37):2982–2995, August 2009.

[107] Robert C. Kirby and Anders Logg. A compiler for variational forms. *ACM Trans. Math. Softw.*, 32(3):417–444, September 2006.

[108] P. Knupp. Remarks on Mesh Quality. December 2007.

[109] Florian Kummer. Extended discontinuous Galerkin methods for two-phase flows: the spatial discretization. *International Journal for Numerical Methods in Engineering*, 109(2):259–289, 2017. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/nme.5288.

[110] X. Li and Michael Scott. On the Nesting Behavior of T-splines. page 22, May 2011.

[111] Xin Li, Jianmin Zheng, Thomas W. Sederberg, Thomas J. R. Hughes, and Michael A. Scott. On linear independence of T-spline blending functions. *Computer Aided Geometric Design*, 29(1):63–76, January 2012.

[112] Ning Liu, Emily L. Johnson, Manoj R. Rajanna, Jim Lua, Nam Phan, and Ming-Chen Hsu. Blended isogeometric Kirchhoff–Love and continuum shells. *Computer Methods in Applied Mechanics and Engineering*, 385:114005, November 2021.

[113] Wing Kam Liu, Sukky Jun, and Yi Fei Zhang. Reproducing kernel particle methods. *International Journal for Numerical Methods in Fluids*, 20(8-9):1081–1106, 1995. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/fld.1650200824.

[114] Anders Logg, Kent-Andre Mardal, and Garth Wells, editors. *Automated Solution of Differential Equations by the Finite Element Method: The FEniCS Book*, volume 84 of *Lecture Notes in Computational Science and Engineering*. Springer, Berlin, Heidelberg, 2012.

[115] Y. Y. Lu, T. Belytschko, and L. Gu. A new implementation of the element free Galerkin method. *Computer Methods in Applied Mechanics and Engineering*, 113(3):397–414, March 1994.

[116] M.V. Lubarda and V.A. Lubarda. *Intermediate Solid Mechanics*. Cambridge : University Press, 2020.

[117] A. Main and G. Scovazzi. The shifted boundary method for embedded domain computations. Part I: Poisson and Stokes problems. *Journal of Computational Physics*, 372:972–995, November 2018.

[118] A. Main and G. Scovazzi. The shifted boundary method for embedded domain computations. Part II: Linear advection–diffusion and incompressible Navier–Stokes equations. *Journal of Computational Physics*, 372:996–1026, November 2018.

[119] Benjamin Marussig and Thomas 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, November 2018.

[120] Kurt Maute. MORIS, August 2023. original-date: 2022-08-28T19:52:17Z.

[121] Chohong Min and Frédéric Gibou. Geometric integration over irregular domains with application to level-set methods. *Journal of Computational Physics*, 226(2):1432–1443, October 2007.

[122] Rajat Mittal and Gianluca Iaccarino. IMMERSED BOUNDARY METHODS. *Annual Review of Fluid Mechanics*, 37(Volume 37, 2005):239–261, January 2005. Publisher: Annual Reviews.

[123] 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, February 2015.

[124] Nicolas Moës, John Dolbow, and Ted Belytschko. A finite element method for crack growth without remeshing. *International Journal for Numerical Methods in Engineering*, 46(1):131–150, 1999.

[125] B. Müller, F. Kummer, and M. Oberlack. Highly accurate surface and volume integration on implicit domains by means of moment-fitting. *International Journal for Numerical Methods in Engineering*, 96(8):512–528, 2013.

[126] B. Nayroles, G. Touzot, and P. Villon. Generalizing the finite element method: Diffuse approximation and diffuse elements. *Computational Mechanics*, 10(5):307–318, September 1992.

[127] J. Nitsche. Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, 36(1):9–15, July 1971.

[128] Charles R. Noble, Andrew T. Anderson, Nathan R. Barton, Jamie A. Bramwell, Arlie Capps, Michael H. Chang, Jin J. Chou, David M. Dawson, Emily R. Diana, Timothy A. Dunn, Douglas R. Faux, Aaron C. Fisher, Patrick T. Greene, Ines Heinz, Yuliya Kanarska, Saad A. Khairallah, Benjamin T. Liu, Jon D. Margraf, Albert L. Nichols, Robert N. Nourgaliev, Michael A. Puso, James F. Reus, Peter B. Robinson, Alek I. Shestakov,

Jerome M. Solberg, Daniel Taller, Paul H. Tsuji, Christopher A. White, and Jeremy L. White. ALE3D: An Arbitrary Lagrangian-Eulerian Multi-Physics Code. Technical Report LLNL-TR-732040, Lawrence Livermore National Lab. (LLNL), Livermore, CA (United States), May 2017.

[129] L. Noël, M. Schmidt, K. Doble, J. A. Evans, and K. Maute. XIGA: An eXtended IsoGeometric analysis approach for multi-material problems. *Computational Mechanics*, 70(6):1281–1308, December 2022.

[130] Maxim A. Olshanskii, Annalisa Quaini, Arnold Reusken, and Vladimir Yushutin. A Finite Element Method for the Surface Stokes Problem. *SIAM Journal on Scientific Computing*, 40(4):A2492–A2518, January 2018. Publisher: Society for Industrial and Applied Mathematics.

[131] Maxim A. Olshanskii, Arnold Reusken, and Jörg Grande. A Finite Element Method for Elliptic Equations on Surfaces. *SIAM Journal on Numerical Analysis*, 47(5):3339–3358, January 2009. Publisher: Society for Industrial and Applied Mathematics.

[132] Stanley Osher and James A Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79(1):12–49, November 1988.

[133] Jamshid Parvizian, Alexander Düster, and Ernst Rank. Finite cell method. *Computational Mechanics*, 41(1):121–133, December 2007.

[134] Charles S Peskin. Flow patterns around heart valves: A numerical method. *Journal of Computational Physics*, 10(2):252–271, October 1972.

[135] M. A. Puso, J. S. Chen, E. Zywicz, and W. Elmer. Meshfree and finite element nodal integration methods. *International Journal for Numerical Methods in Engineering*, 74(3):416–446, 2008. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/nme.2181.

[136] Prabhu Ramachandran, Aditya Bhosale, Kunal Puri, Pawan Negi, Abhinav Muta, A. Dinesh, Dileep Menon, Rahul Govind, Suraj Sanka, Amal S. Sebastian, Ananyo Sen, Rohan Kaushik, Anshuman Kumar, Vikas Kurapati, Mrinalgouda Patil, Deep Tavker, Pankaj Pandey, Chandrashekhar Kaushik, Arkopal Dutt, and Arpit Agarwal. PySPH: A Python-based Framework for Smoothed Particle Hydrodynamics. *ACM Transactions on Mathematical Software*, 47(4):34:1–34:38, September 2021.

[137] 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*, 249-252:104–115, December 2012.

[138] David F. Rogers. *An Introduction to NURBS: With Historical Perspective*. Morgan Kaufmann, 2001. Google-Books-ID: DiyxPUiKvB8C.

[139] 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.

[140] T. Rüberg and F. Cirak. Subdivision-stabilised immersed b-spline finite elements for moving boundary flows. *Computer Methods in Applied Mechanics and Engineering*, 209-212:266–283, February 2012.

[141] Dominik Schillinger, Luca Dedè, Michael A. Scott, John A. Evans, Michael J. Borden, Ernst Rank, and Thomas 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*, 249-252:116–150, December 2012.

[142] Dominik Schillinger, Isaac Harari, Ming-Chen Hsu, David Kamensky, Stein K. F. Stoter, Yue Yu, and Ying Zhao. The non-symmetric Nitsche method for the parameter-free imposition of weak boundary and coupling conditions in immersed finite elements. *Computer Methods in Applied Mechanics and Engineering*, 309:625–652, September 2016.

[143] Dominik Schillinger and Martin 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, July 2015.

[144] Dominik Schillinger, Praneeth K. Ruthala, and Lam H. Nguyen. Lagrange extraction and projection for NURBS basis functions: A direct link between isogeometric and standard nodal finite element formulations. *International Journal for Numerical Methods in Engineering*, 108(6):515–534, November 2016.

[145] Ryan T. Schlinkman, Jonghyuk Baek, Frank N. Beckwith, Stacy M. Nelson, and J. S. Chen. A Quasi-Conforming Embedded Reproducing Kernel Particle Method for Heterogeneous Materials, April 2023. arXiv:2304.06150 [cs, math].

[146] Mathias Schmidt, Lise Noël, Keenan Doble, John A. Evans, and Kurt Maute. Extended isogeometric analysis of multi-material and multi-physics problems using hierarchical B-splines. *Computational Mechanics*, 71(6):1179–1203, June 2023.

[147] 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*, 213-216:206–222, March 2012.

[148] Michael A. Scott, Michael J. Borden, Clemens V. Verhoosel, Thomas W. Sederberg, and Thomas 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.

[149] Thomas W. Sederberg, Jianmin Zheng, Almaz Bakenov, and Ahmad Nasri. T-splines and T-NURCCs. *ACM Transactions on Graphics*, 22(3):477–484, July 2003.

[150] Soheil Soghrati. Hierarchical interface-enriched finite element method: An automated technique for mesh-independent simulations. *Journal of Computational Physics*, 275:41–52, October 2014.

[151] Alvise Sommariva and Marco Vianello. Gauss–Green cubature and moment computation over arbitrary geometries. *Journal of Computational and Applied Mathematics*, 231(2):886–896, September 2009.

[152] Gilbert Strang and George Fix. *An Analysis of the Finite Element Method*. Wellesley-Cambridge Press, Wellesley, MA, 2008.

[153] T. Strouboulis, I. Babuška, and K. Copps. The design and analysis of the Generalized Finite Element Method. *Computer Methods in Applied Mechanics and Engineering*, 181(1):43–69, January 2000.

[154] T. Strouboulis, K. Copps, and I. Babuška. The generalized finite element method: an example of its implementation and illustration of its performance. *International Journal for Numerical Methods in Engineering*, 47(8):1401–1417, 2000.

[155] Y. Sudhakar and Wolfgang A. Wall. Quadrature schemes for arbitrary convex/concave volumes and integration of weak form in enriched partition of unity methods. *Computer Methods in Applied Mechanics and Engineering*, 258:39–54, May 2013.

[156] Deborah Sulsky, Shi-Jian Zhou, and Howard L. Schreyer. Application of a particle-in-cell method to solid mechanics. *Computer Physics Communications*, 87(1):236–252, May 1995.

[157] Kristen Susuki, Jeffery Allen, and Jiun-Shyan Chen. Image-based modeling of coupled electro-chemo-mechanical behavior of Li-ion battery cathode using an interface-modified reproducing kernel particle method. *Engineering with Computers*, August 2024.

[158] Kenjiro Terada, Mitsuteru Asai, and Michihiro Yamagishi. Finite cover method for linear and non-linear analyses of heterogeneous solids. *International Journal for Numerical Methods in Engineering*, 58(9):1321–1346, November 2003.

[159] Marie Tirvaudey, Robin Bouclier, Jean-Charles Passieux, and Ludovic Chamoin. Non-invasive implementation of nonlinear isogeometric analysis in an industrial FE software. *Engineering Computations*, 37(1):237–261, January 2020. Publisher: Emerald.

[160] Luminita A. Vese and Tony F. Chan. A Multiphase Level Set Framework for Image Segmentation Using the Mumford and Shah Model. *International Journal of Computer Vision*, 50(3):271–293, December 2002.

[161] Dongdong Wang, Jiun-Shyan Chen, and Lizhi Sun. Homogenization of magnetostrictive particle-filled elastomers using an interface-enriched reproducing kernel particle method. *Finite Elements in Analysis and Design*, 39(8):765–782, May 2003.

[162] Yanran Wang, Jonghyuk Baek, Yichun Tang, Jing Du, Mike Hillman, and Jiun-Shyan Chen. Support vector machine guided reproducing kernel particle method for image-based modeling of microstructures. *Computational Mechanics*, 73(4):907–942, April 2024.

[163] Y. You, J.-S. Chen, and H. Lu. Filters, reproducing kernel, and adaptive meshfree method. *Computational Mechanics*, 31(3):316–326, July 2003.

[164] Han Zhao, David Kamensky, John T. Hwang, and Jiun-Shyan Chen. Automated shape and thickness optimization for non-matching isogeometric shells using free-form deformation. *Engineering with Computers*, March 2024.

[165] Han Zhao, Xiangbei Liu, Andrew H. Fletcher, Ru Xiang, John T. Hwang, and David Kamensky. An open-source framework for coupling non-matching isogeometric shells with application to aerospace structures. *Computers & Mathematics with Applications*, 111:109–123, April 2022.

[166] T. Zhu and S. N. Atluri. A modified collocation method and a penalty formulation for enforcing the essential boundary conditions in the element free Galerkin method. *Computational Mechanics*, 21(3):211–222, April 1998.