

UC San Diego

UC San Diego Electronic Theses and Dissertations

Title

A Conforming Reproducing Kernel Framework for Agile Simulation of Problems with Complex Geometries

Permalink

<https://escholarship.org/uc/item/6p36g93r>

Author

Koester, Jacob

Publication Date

2019

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA SAN DIEGO

**A Conforming Reproducing Kernel Framework for Agile Simulation of Problems with
Complex Geometries**

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

in

Structural Engineering (with a specialization in Computational Science)

by

Jacob J. Koester

Committee in charge:

Professor Jiun-Shyan Chen, Chair
Professor Randolph Bank
Professor Yuri Bazilevs
Professor Michael Holst
Professor Petr Krysl

2019

Copyright
Jacob J. Koester, 2019
All rights reserved.

The dissertation of Jacob J. Koester is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

Chair

University of California San Diego

2019

DEDICATION

To Krista, Henry and Hazel

TABLE OF CONTENTS

	Signature Page	iii
	Dedication	iv
	Table of Contents	v
	List of Figures	viii
	List of Tables	xii
	Acknowledgements	xiii
	Vita	xiv
	Abstract of the Dissertation	xvi
Chapter 1	Introduction	1
	1.1 Motivation	1
	1.2 Objectives	2
	1.3 Outline	4
Chapter 2	Literature Review	6
	2.1 Rapid Simulation Model Development	6
	2.2 Meshfree Methods	7
	2.3 Smooth Approximation Spaces	9
Chapter 3	Background	11
	3.1 Reproducing Kernel	11
	3.1.1 Reproducing Kernel Approximation	11
	3.1.2 Reproducing Kernel Particle Method	14
	3.1.3 Essential Boundary Conditions	15
	3.1.4 Non-convex Domains	21
	3.1.5 Material Interfaces	22
	3.2 Spline Functions on Triangulations in Two Dimensions	22
	3.2.1 Smooth Joins	24
	3.2.2 Spaces of Smooth Splines	25
	3.3 Spline Functions on Triangulations in Three Dimensions	29
	3.3.1 Smooth Joins	31
	3.3.2 Spaces of Smooth Splines	31

Chapter 4	Conforming Window Functions	35
	4.1 Window function domains	36
	4.2 Window function space	36
	4.3 Specifying window functions coefficients	39
	4.3.1 Star window functions	40
	4.3.2 Snap window functions	41
	4.3.3 Local window functions	44
	4.4 Modification for conforming to surfaces	46
	4.4.1 Smoothly conforming to boundaries	46
	4.4.2 Sharply conforming to boundaries	47
	4.4.3 Examples of boundary conforming window functions	49
	4.5 Modification for convexity improvement	50
	4.6 Simplifications for implementation	53
	4.7 Three Dimensional Conforming Window Functions	54
	4.8 Additional comments	55
Chapter 5	An $\bar{\mathbf{F}}$ formulation for nearly incompressible materials	56
	5.1 Review of the $\bar{\mathbf{B}}$ and $\bar{\mathbf{F}}$ methods	58
	5.1.1 The $\bar{\mathbf{B}}$ method	58
	5.1.2 The $\bar{\mathbf{F}}$ method	60
	5.2 The $\bar{\mathbf{F}}$ method for conforming window methods	64
Chapter 6	Addressing low quality discretizations	68
	6.1 Example low quality discretization and challenges	69
	6.2 Improving the solution efficiency and accuracy by decimating and aggregating	73
	6.2.1 Example of decimating and aggregating for a two-sphere problem	77
Chapter 7	Applications of Conforming Window Functions	79
	7.1 Two dimensional problem	80
	7.1.1 Linear elastic patch test	80
	7.1.2 Non-convex geometries	82
	7.1.3 Plate with an inclusion	89
	7.1.4 Beam with poorly-shaped triangles	92
	7.1.5 Cook’s membrane	95
	7.2 Three dimensional problems	100
	7.2.1 Taylor bar	100
	7.2.2 Compression of a 3D elastomer billet	105
	7.2.3 Carbon Black Microstructure	105
	7.2.4 Battery microstructure simulation	112

Chapter 8	Conclusions	117
	8.1 Conclusions	117
	8.2 Recommendation for future research	120
Bibliography	121

LIST OF FIGURES

Figure 1.1:	Examples of a complex housing and system	2
Figure 1.2:	Estimation of the relative time costs of each component of the model generation and analysis process	3
Figure 3.1:	Domain and discretization example for a reproducing kernel approximation	12
Figure 3.2:	Two dimensional kernels constructed using the cubic B-spline	14
Figure 3.3:	Point cloud near and on a domain boundary	17
Figure 3.4:	Window and approximation functions near a boundary for the common tensor-product definition	17
Figure 3.5:	Window and approximation functions near a boundary for the Kronecker delta window definition	19
Figure 3.6:	Example window domains for different methods of handling a non-convex boundary.	21
Figure 3.7:	Window and approximation function with visibility criteria near a non-convex boundary.	22
Figure 3.8:	Domain points and Bézier patch for a triangle with polynomial order, $d = 5$.	23
Figure 3.9:	A Bézier patch with C^1 continuity	26
Figure 3.10:	An example minimal determining set for the Argyris space, $\mathcal{S}_5^{1,2}(\Delta)$	28
Figure 3.11:	Domains points and smooth joins for Bernstein-Bézier spline in three dimensions	30
Figure 4.1:	Full domain triangulation used for building conforming window functions .	37
Figure 4.2:	Local triangulations for building conforming window functions	37
Figure 4.3:	Normalized graph distances, \bar{r}_g , at the $\mathcal{S}_5^{1,2}(\Delta_v)$ nodal parameter locations for a second order star in the interior of the domain. The nodal parameter locations are the vertices (circles) and the edge midpoints (triangles)	40
Figure 4.4:	Interior, $star^2(v)$, conforming window and approximation function and their derivatives.	41
Figure 4.5:	Illustration of the snap star window function constructed about a vertex . .	43
Figure 4.6:	Conforming window functions using the triangulation in Figure 4.5 for a snap star with support of $1.2h$ and a $star^2(v)$ window	43
Figure 4.7:	Steps for creating a local triangulation that conforms to the boundary . . .	45
Figure 4.8:	Resulting local window function created using the local triangulation. . . .	46
Figure 4.9:	Normalized graph distances, \bar{r}_g , at the $\mathcal{S}_5^{1,2}(\Delta_v)$ nodal parameter locations for a second order star about points near or on a conforming boundary. The nodal parameter locations are the vertices (circles) and the edge midpoints (triangles)	47
Figure 4.10:	Geometry used for setting coefficients associated with a conforming boundary vertex when using the C^0 window design method. Coefficients are rotated about an axis (purple dashed line)	49
Figure 4.11:	Window and approximation functions for a point near but not on a boundary.	51

Figure 4.12:	Approximation functions derivatives for the smooth and sharp $star^2(v)$ conforming window function definitions plotted along line segments parallel to the boundary and near point 18 (left) and corner point 13 (right).	52
Figure 4.13:	$star^2(v)$ window and approximation functions near a non-convex boundary, with (right) and without (left) using the check to improve convexity.	52
Figure 4.14:	Integration cell examples for assumed strain methods.	54
Figure 5.1:	Dilatational and deviatoric smoothing domains	66
Figure 5.2:	Example of node-based smoothing domains and corresponding kernel triangulation method that aligns integration evaluation points and kernel domain points, allowing efficient evaluations.	67
Figure 6.1:	Lithium-ion battery electrode mesostructure. Example domain and CDFEM mesh.	69
Figure 6.2:	Two sphere example problem. Example domain and meshes.	71
Figure 6.3:	Example predictions of σ_{xx} for the two sphere problem.	72
Figure 6.4:	Cells obtained by aggregating the elements of the CDFEM mesh.	77
Figure 6.5:	Example σ_{xx} predictions using an aggregated and decimated mesh for the two sphere problem.	78
Figure 7.1:	Deformed geometry and magnitude of error in displacement for a linear elastic patch test	81
Figure 7.2:	Additively manufactured notched bar. The region in the blue box is used to create a two dimensional model. The triangulation is created after using image processing to extract the specimen boundaries.	84
Figure 7.3:	σ_{yy} results for the additively manufactured notched bar loaded in tension.	85
Figure 7.4:	Comparison of σ_{yy} results along the a horizontal line starting at the middle of the crack tip for the notched tensile specimen.	86
Figure 7.5:	Close-up view of refined σ_{yy} results near the notch tip for the additively manufactured notched bar loaded in tension.	86
Figure 7.6:	Refined predictions of σ_{yy} results along the a horizontal line starting at the middle of the crack tip for the notched tensile specimen.	87
Figure 7.7:	Plate with an edge crack.	88
Figure 7.8:	Plate with edge crack convergence	90
Figure 7.9:	Error in σ_{yy} near the crack tip for the edge crack problem with a near-tip nodal spacing $h_1 = 0.02$. The black solid line represents the location of the edge for the exact solution.	91
Figure 7.10:	Plate with an inclusion simulated by taking advantage of the symmetry.	91
Figure 7.11:	Convergence plots for the plate with an inclusion example simulated with FEM, RKPM using non-conforming tensor product window functions with $R = 2.0h$ and CRK with $R_g = star^2(v)$ and snap and local stars with $R = 2.0h$ that conform to the material interface.	93

Figure 7.12:	Comparison of ε_{xy} results along the a diagonal line between the origin and upper right corner of the domain for the inclusion problem with a near-inclusion nodal spacing $h = 0.0625$	94
Figure 7.13:	Timoshenko beam with shear load	94
Figure 7.14:	Timoshenko beam convergence plots for FEM, CRK with $star^2(v)$ and snap and local stars with a support of $1.5h$. Boundaries are enforced using static condensation.	96
Figure 7.15:	Cooks membrane. The left edge is fixed and the right edge is vertically displaced.	97
Figure 7.16:	Cook’s membrane. Hydrostatic Stress. Discretization using 4 element on the displaced edge. From left to right: CRK without \bar{F} , RKPM with stabilized conforming nodal integration, CRK with \bar{F}	98
Figure 7.17:	Cook’s membrane. Hydrostatic Stress. Discretization using 16 element on the displaced edge. From left to right: CRK without \bar{F} , RKPM with stabilized conforming nodal integration, CRK with \bar{F}	98
Figure 7.18:	Cook’s membrane. Convergence of tip force, relative to the refined prediction using the selective deviatoric element, versus number of elements along the displaced edge.	99
Figure 7.19:	Example tetrahedral mesh for the Taylor bar with 50 elements along the length.	101
Figure 7.20:	3D Taylor bar impact. Normal stress in vertical directions. Discretization using 25 elements along the length. From left to right: CRK without \bar{F} , RKPM with stabilized conforming nodal integration, CRK with \bar{F} . Side view on top row and end view on bottom row.	102
Figure 7.21:	Taylor bar impact. Normal stress in vertical directions. Discretization using 100 elements along the length. From left to right: CRK without \bar{F} , RKPM with stabilized conforming nodal integration, CRK with \bar{F} . Side view on top row and end view on bottom row.	103
Figure 7.22:	Taylor bar impact. Convergence of radial displacement with mesh refinement	104
Figure 7.23:	Compression billet with the mesh depicting the portion of the domain that is simulated.	106
Figure 7.24:	Compression billet. Hydrostatic Stress. From top to bottom: displacement of 0.50, 1.0 and 1.5 millimeters.	107
Figure 7.25:	Domain and example meshes for the representative volume of an elastomer embedded with carbon particles.	108
Figure 7.26:	Hydrostatic stress predictions for the carbon black microstructure with a shear displacement of 30 nm.	109
Figure 7.27:	Load versus displacement predictions carbon black microstructure with a mesh edge length of 4 nm.	110
Figure 7.28:	Convergence of the load prediction at a displacement of 14 nm for the carbon black microstructure under shear loading.	111
Figure 7.29:	Subdomain and CDFEM mesh for the battery mesostructure	112
Figure 7.30:	Battery mesostructure. Initial state and final state after displacement. . . .	114
Figure 7.31:	Displacement field and stress σ_{zz} (the loading direction) at the final displacement	115

Figure 7.32: Example of clusters 116

LIST OF TABLES

Table 6.1:	Stable time increments and the displacement at which analysis failed for simulations of the two-spheres with electrolyte in compression.	72
Table 7.1:	Error norms for a linear elastic patch test	82
Table 7.2:	Run times for the Taylor bar.	104
Table 7.3:	Stable time increments for simulations of the battery mesostructure.	113

ACKNOWLEDGEMENTS

I would like to thank Steve Attaway, Kim Mish, Eliot Fang, Mike Tupek, John Pott, Justine Johannes, Jim Redmond, Scott Mitchell, Joe Bishop, Kendall Pierson, Pavel Bochev, Nat Trask, Scott Roberts, Kevin Long, and many others from Sandia National Labs. Many have provided support, guidance and challenging exemplar problems.

I thank many fellow group members and other UCSD students that have provided invaluable insight and suggestions along the way. Thank you Marco Pasetto, Hayoan Wei, Qizhi He, Mike Hillman, Frank Beckwith, Edouard Yreux, Jonghyuk Baek, Xialong He, Georgios Moutsanidis and Ivana Escobar.

I also thank my committee members Professors Randolph Bank, Yuri Bazilevs, Michael Holst and Petr Krysl and, in particular, my advisor J.S. Chen for their time and guidance.

Material from Chapters 2, 3, 4 and 7, appears in *Computer Methods in Applied Mathematics and Engineering*, “Conforming window functions for meshfree methods”, 347:588 – 621, 2019, by Jacob Koester and J.S. Chen.

Material from Chapter 5 and some of the examples from Chapter 7 have been submitted to *Computational Particle Mechanics* as “ $\bar{\mathbf{B}}$ and $\bar{\mathbf{F}}$ approaches for the treatment of near-incompressibility in meshfree and immersed-particle methods” with authors Georgios Moutsanidis, Jacob Koester, Michael Tupek, Yuri Bazilevs and J.S. Chen.

Material from Chapter 6 and additional examples from Chapter 8 will be expanded and submitted to a journal in the near future.

The support of this work by Sandia National Laboratories via the Laboratory Directed Research and Development (LDRD) program and the Doctoral Study Program (DSP). Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc. for the U.S. Department of Energy’s National Nuclear Security Administration under contract DE-NA0003525.

VITA

2004	Bachelor of Science in Mechanical Engineering, <i>magna cum laude</i> , South Dakota School of Mines & Technology
2005	Master of Science in Mechanical Engineering, South Dakota School of Mines & Technology
2006-2010	Thermomechanical Analyst, Ball Aerospace & Technologies Corp., Albuquerque, NM
2010-2019	Senior Member of Technical Staff, Sandia National Laboratories, Albuquerque, NM
2016-2017	Graduate Teaching Assistant, University of California, San Diego
2019	Doctor of Philosophy in Structural Engineering, University of California, San Diego

PUBLICATIONS

Journal Papers

Jacob Koester and J.S. Chen, “Conforming window functions for meshfree methods”, *Computer Methods in Applied Mathematics and Engineering*, 347:588 – 621, 2019.

Georgios Moutsanidis, Jacob Koester, Michael Tupek, Yuri Bazilevs and J.S. Chen. “ $\bar{\mathbf{B}}$ and $\bar{\mathbf{F}}$ approaches for the treatment of near-incompressibility in meshfree and immersed-particle methods” *Computational Particle Mechanics*, submitted.

Presentations

Jacob Koester, J.S. Chen, Mohamed Ebeida, Scott Mitchell, Joseph E. Bishop and David J. Littlewood, “Voronoi Cell Based RKPM”, *USACM Conference on Isogeometric Analysis and Meshfree Methods*, San Diego, California, 2016.

Jacob Koester and J.S. Chen “Conforming Reproducing Kernels to Enable Agile Simulation Environments”, *Engineering Mechanics Institute Conference*, La Jolla, California, 2017.

Jacob Koester and J.S. Chen “Conforming Reproducing Kernels to Enable Agile Simulation Environments”, *US National Congress on Computational Mechanics*, Montréal, Canada, 2017.

Jacob Koester, J.S. Chen, Michael Tupek, Scott Mitchell and Joseph Bishop, “The Conforming Reproducing Kernel Method for an Agile Design-to-Simulation Workflow”, *World Congress in Computational Mechanics*, New York, New York, 2018.

Jacob Koester, J.S. Chen, Michael Tupek, Scott Mitchell and Joseph Bishop, “The Conforming Reproducing Kernel Method for an Agile Design-to-Simulation Workflow”, *USACM Conference on Meshfree & Particle Methods: Applications & Theory*, Santa Fe, New Mexico, 2018.

Jacob Koester, J.S. Chen, Michael Tupek, Scott Mitchell and Joseph Bishop, “The Conforming Reproducing Kernel Method for an Agile Design-to-Simulation Process”, *ASME International Mechanical Engineering Congress and Exposition*, Pittsburgh, Pennsylvania, 2018

ABSTRACT OF THE DISSERTATION

A Conforming Reproducing Kernel Framework for Agile Simulation of Problems with Complex Geometries

by

Jacob J. Koester

Doctor of Philosophy in Structural Engineering (with a specialization in Computational Science)

University of California San Diego, 2019

Professor Jiun-Shyan Chen, Chair

Computational mechanics is a growing discipline that is commonly used to provide invaluable insight for engineering analyses, particularly when theoretical or empirical analyses are impractical. Computer simulations can handle complex problems more readily than analytical methods and are often much less expensive and less time consuming than performing experiments. However, developing simulation models for moderate to complex geometries is still a time consuming task. The upfront cost of developing models is an impediment that reduces the effectiveness of simulation in engineering design and analysis.

Meshfree methods provide flexible means of creating approximation functions using only

a collection of points with corresponding window functions. This flexibility makes meshfree methods attractive for problems involving large deformation and failure, where mesh-based methods, such as the finite element method, experience complications due to mesh entanglement. Without the need of generating a quality mesh, meshfree methods also have a greatly simplified simulation development process. However, for problems with complicated geometries such as non-convex or multi-body domains, poor solution accuracy and reduced convergence rates are observed due to inaccurate representation of complex geometry unless the extents of the window functions, and thus approximation functions, are carefully controlled. The required control is not available with the Euclidean metrics that are commonly used to construct the window functions.

In this dissertation, methods of providing more control in window function design are presented. “Conforming” window functions are constructed using local triangulations in two and three dimensions, allowing efficient and systematic handling of complex geometries. Graph distances are used in conjunction with Euclidean metrics to provide adequate information for shaping the window functions. The conforming window functions are demonstrated using the Reproducing Kernel Particle Method showing improved accuracy and convergence rates for problems with challenging geometries. Additional attention is given to addressing the challenges of simulating nearly incompressible material response and to maintaining a higher stable time increment for dynamic problems that are solved using explicit time integration.

Addressing the boundary-related challenges of meshfree methods both opens the methods to a broader class of problems and enables an agile simulation development process.

Chapter 1

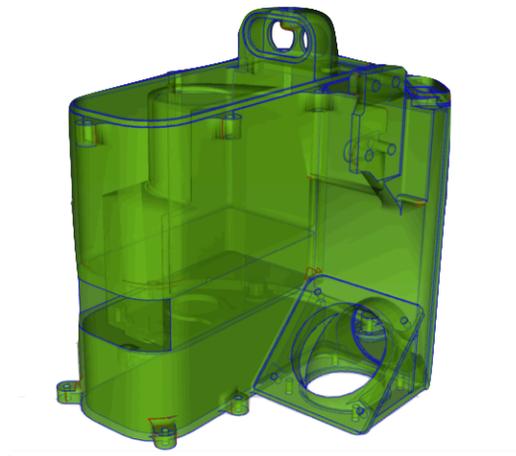
Introduction

1.1 Motivation

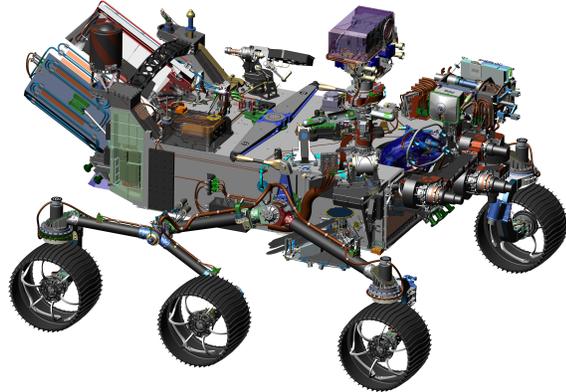
Construction of numerical simulation models still remains a time consuming task for many mechanical analyses. One of the obstacles is the effort required to make the problem geometry amenable to available numerical methods. For decades, the finite element method (FEM) has been the de facto numerical tool for mechanical analysis. A drawback of using FEM for analysis (FEA) is that the method requires the geometry be divided into an organized mesh of elements that have quality and conformity constraints. For simple geometries this is not a problem but as the complexity of a part or system increases it becomes a large challenge, see Figure 1.1.

Plans and budgets for many programs in government and industry set aside significant resources for FEA model development. A study performed by Sandia National Laboratories concluded that model preparation can easily account for 50-75% of an analysis budget depending on the complexity of the system [1], see Figure 1.2. While the upfront cost for FEA model development is clearly high, the largest program expenses occur when guidance is not provided in a timely manner and designs are later determined to be deficient.

Meshfree methods provide alternative means of discretizing a domain. Only a collection



(a) Housing



(b) Mars 2020 Rover, courtesy NASA/JPL-Caltech

Figure 1.1: Examples of a complex housing and system

of points and corresponding window functions are needed to construct an approximation space. This allows simulations models of complex domains to be constructed very efficiently, but other complexities remain. During the construction of the meshfree approximation space, the window functions are used to provide a base that controls the smoothness and extent of the approximation functions. Window functions are commonly defined using Euclidean distances which helps eliminate the need for a mesh. However, for problems with complicated geometries such as non-convex or multi-body domains, poor solution accuracy and convergence can occur unless the extents of the window functions, and thus approximation functions, are carefully controlled. Simple metrics do not provide the requisite control to prevent the issue.

1.2 Objectives

The objective of this work is to develop a framework that utilizes meshfree techniques and provides a means of systematically controlling window functions near domain boundaries and interfaces. Resolving these boundary-related challenges will broaden the application space for meshfree methods and enable an agile simulation development process. The major tasks of this

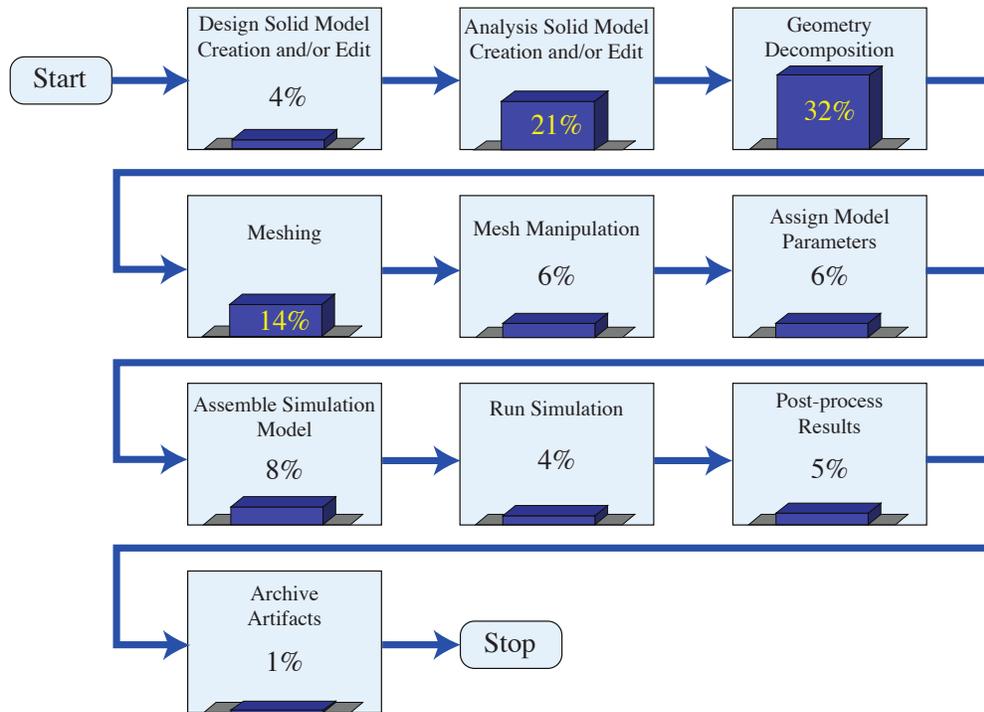


Figure 1.2: Estimation of the relative time costs of each component of the model generation and analysis process at Sandia National Laboratories. [1]

dissertation are summarized as follows:

- Development of methods to conform window functions to domain boundaries and interfaces in two dimensions. Spline functions defined on triangulations of the domain are used to construct smooth window functions with controlled extents. Three different methods are established, the 1) star, 2) snap and 3) local triangulation methods, which provide flexibility in the requirements of the window triangulations and vary in simplicity and discretization sensitivity. Examples window functions and resulting approximations functions are shown for locations near domain boundaries and non-convex geometries.
- Extend the conforming window function method to three dimensions and determine simplifications that allow efficient implementation and computation. Many analyses need to be constructed in three dimensions. The general procedures for building window functions in two dimensions also apply in three dimensions; however, the spline function spaces

required become more complex. Means of dealing with this complexity are presented.

- Address additional complications that arise for applications in non-linear, dynamic, solid mechanics. Building simulation models is a time consuming task for analyses using many different methods that are simulating different physical phenomena. A particularly difficult class are dynamic solid mechanics problems that involve non-linear, nearly incompressible material behavior that are solved using explicit time integration which has a stable time increment limit [1]. A technique of addressing complications due to material incompressibility that is compatible with the three dimensional conforming window procedure is proposed. Also, efficient methods of generating triangulations can result in low quality triangles / tetrahedra. A method of dealing with poor triangulations is developed with maintaining an agreeable stable time increment being a primary goal.
- Demonstrate the performance of the conforming window functions in two and three dimensions. Ideal convergence rates are demonstrated in two dimensions for model problems that have material interfaces and non-convex boundaries. Three dimensional exemplar problems are completed to demonstrate the efficiency of the method and show the performance for nearly incompressible materials and when using triangulations with poorly shaped tetrahedra.

1.3 Outline

The remainder of this dissertation is organized as follows. An overview of the research to date on rapid simulation model development, meshfree methods and smooth approximation spaces is provided in Chapter 2. Chapter 3 includes a review of reproducing kernel (RK) approximation functions and the reproducing kernel particle methods (RKPM), including discussions on enforcing essential boundary conditions, handling material interfaces and non-convex geometries. Also, a background on spline functions defined on triangulations is provided. The different procedures

for constructing window functions that conform to domain boundaries are presented in Chapter 4. Chapter 5 discusses addressing the complications associated with nearly incompressible materials. Chapter 6 presents procedures for handling low quality triangulations with emphasis on increasing the stable time increment for dynamic problems using explicit time integration. The conforming window functions are applied to two dimensional model problems that include material interfaces, non-convex domains and low quality triangulations in Chapter 7. Also, the performance of the conforming window functions are also shown in three dimensions, including problems with nearly incompressible material behavior. Finally, a summary and concluding remarks are given in Chapter 8.

Chapter 2

Literature Review

2.1 Rapid Simulation Model Development

Over the last few decades several efforts have sought to improve simulation agility. Tetrahedral elements were formulated early in FE development and are better suited for efficient mesh generation than hexahedral elements but linear tetrahedral elements are prone to “locking” (low convergence rates) and higher-order elements can have negative nodal masses when using row-summation mass-lumping. Ongoing research to address these issues has focused on composite tetrahedra [2, 3], modified integration [4, 5], pressure averaging [6] and alternative high-order elements [7]. In addition to hexahedral and tetrahedral elements, means of creating approximation functions on arbitrary polyhedral elements are being studied [8, 9, 10].

Meshfree methods avoid meshing all together by generating approximation functions on point clouds. There are many different meshfree methods with two commonly used being the Element Free Galerkin (EFG) method [11] and the Reproducing Kernel Particle Method (RKPM) [12, 13]. The effort needed to discretize complex geometries is much less than the traditional FE method as generating point clouds is much less restrictive than generating meshes of quality elements. More details of meshfree methods are provide in Section 2.2 below.

Immersed boundary and interface methods, also known as fictitious or embedded domain methods, bypass the meshing challenges by choosing to represent the boundary implicitly rather than explicitly [14, 15, 16]. Meshes are typically Cartesian grids with elements that are intersected by part boundaries handled with some specific technique. For example, the Finite Cell Method uses refined quadrature cells near boundaries to better integrate the portion of the element in the physical domain (inside the body) and penalize the portion of the element that is in a “fictitious” domain (outside the body) [17]. Research has continued to focus on developing new methods for improving the accuracy of the near boundary solution in an efficient manner [18, 19].

A prominent goal of IsoGeometric Analysis (IGA) is to reduce the CAD-to-simulation translation effort by using the same basis functions for both the geometry representation and solution field approximation [20]. Some of the advantages include the ability to exactly represent the geometry at different refinement levels, simplified mesh refinement and smooth approximation functions. Volumetric parameterizations of arbitrary and complex geometries is required for many applications and remains an open challenge for IGA as CAD typically provides a Boundary representation (B-rep) rather than a volumetric representation. Many efforts have focused on this issue including research on unstructured Bernstein-Bézier tetrahedra [21, 22] and trivariate splines [23, 24].

2.2 Meshfree Methods

Meshfree methods have proven to be effective simulation techniques for many challenging application areas including large deformation, strain localization, fracture and many more [25]. A common characteristic of meshfree methods is that approximation functions are constructed without the need of a mesh but rather just a collection of points upon which local window functions are defined. The flexibility of constructing approximation functions in meshfree methods does also present new challenges. Approximation functions are typically rational, non-interpolatory

and overlap domain boundaries.

Rational approximation functions pose considerable complexity for integrating the weak form for Galerkin methods. Gauss integration on background cells was commonly used in the early development of RKPM and EFG [11, 12, 13]. Gauss integration can not exactly integrate rational functions; therefore, more expensive, higher-order, integration is used to achieve reasonable accuracy. To improve the accuracy and efficiency of domain integration, stabilized conforming nodal integration [26] and variationally consistent integration methods [27] have been developed.

Many complications arise with applications that have complex geometries. Non-convex boundaries require considerations to prevent the approximation functions from bleeding over and effectively filling in void, which may be addressed using a visibility criteria or the diffraction method [28, 29, 30]. Material interfaces create weak discontinuities in the solution, necessitating modifications to the methods to avoid degrading the accuracy and convergence [31, 32, 33]. In most meshfree methods, approximation functions from non-boundary nodes are non-zero on the boundary and typically don't possess the Kronecker delta property, complicating the imposition of essential boundary conditions. Many methods have been developed that either provide the Kronecker delta property, weak enforcement of essential boundary conditions or couple with other methods [34, 35, 36, 37, 38, 39].

In a practical analysis of a complex system it is common to have many different instances of material interfaces, non-convex geometry and regions where essential boundary conditions are applied. For a meshfree analysis, several of the above techniques may be needed which can complicate model preparation and significantly increase the time required to get a working simulation. In addition, the required algorithms become more complex with many different features needing to work together robustly. The additional efforts have deterred analysts and developers in adopting meshfree methods as more general simulation capabilities.

All mentioned issues originate from the use of window functions that are constructed using only Euclidean metrics. Window functions control the smoothness and extent of the approximation

functions. With Euclidean-based windows, smoothness is easy to obtain but there is insufficient control on the window extent resulting in ineffective handling of complex geometries. Designing window functions to avoid the boundary-related issue was identified as a desirable but elusive solution early in the development of RKPM [40].

Some efforts using non-Euclidean metrics for constructing meshfree window functions have been studied before. The cell-based maximum-entropy method uses smooth R-functions built on triangulations during the construction of approximation functions [41]. The primary goal was to control the extent of the approximation functions to reduce the overall expense of the method. Smooth R-functions were also used in the generalized finite element method to create smooth approximation functions on unstructured meshes [42].

2.3 Smooth Approximation Spaces

The continuity of meshfree approximation functions is controlled by the continuity of the window functions. For regions of problems that have smooth solutions, the continuity of the approximations functions has been demonstrated to reduce the error in predictions [25]. Functions of high continuity can be readily defined about a point when Euclidean metrics are used. For example, one dimensional cardinal B-splines can be defined with arbitrary order of continuity and can be easily extended to multi-dimensions by using radial or tensor product definitions. Unfortunately, Euclidean metrics do not provide enough control to circumvent boundary-related complications for moderate to complex geometries. In order to retain the advantages of higher continuity and avoid boundary-related complications, alternative methods of defining window functions can be explored.

Several methods of constructing smooth local functions have been studied including methods that make use of R-functions [43], approximate signed distance functions [41] and mean value coordinates [44]. Of particular interest are spline functions defined on triangulations. The

weak form of the Kirchhoff-Love plate problem requires an approximation space that is at least C^1 . Interest in solving the plate problem using the finite element method led to development of several elements such as the Argyris triangle, Bell's triangle and the Hsieh-Clough-Tocher triangle, discussed in [45, 46]. To obtain the requisite smoothness, a high polynomial order, and thus many degrees-of-freedom, are needed. This deterred adoption of these elements and researchers largely turned to nonconforming (approximation spaces that are not in C^1) and the Reissner-Mindlin plate theory which only requires a C^0 approximation space. Besides FE-based plate analysis, spline functions on triangulations matured due to their applications in data interpolation and fitting. To handle trivariate data, smooth spaces using spline functions on tetrahedra were created [47, 48, 49]. Lately, they have gained some interest in the IGA community as a means of creating volumetric discretizations [22].

Subdivision surfaces are another class of method for creating smooth functions and are popular in computer graphics and animation [50, 51, 52]. Subdivision surfaces have also been used for shell analysis [53]. For solid representations, subdivision volumes have been explored [54, 55] but the volume subdivision has received far less attention than surfaces.

Material from this Chapter appears in *Computer Methods in Applied Mathematics and Engineering*, "Conforming window functions for meshfree methods", 347:588 – 621, 2019, by Jacob Koester and J.S. Chen.

Chapter 3

Background

3.1 Reproducing Kernel

3.1.1 Reproducing Kernel Approximation

The reproducing kernel (RK) is used herein to provide an example of meshfree approximations. In the RK method, a closed domain $\bar{\Omega}$ is discretized with a set of NP points. An RK approximation $u^h(\mathbf{x})$ is constructed as

$$u^h(\mathbf{x}) = \sum_{I=1}^{NP} \Psi_I(\mathbf{x}) d_I \quad (3.1)$$

where $\Psi_I(\mathbf{x})$ are the approximation functions and d_I are coefficients. Each point I has an associated subdomain Ω_I over which a window function $\phi(\mathbf{x} - \mathbf{x}_I)$, called a *kernel function*, is defined. The set of kernel subdomains $\{\Omega_I\}_{I=1}^{NP}$ covers $\bar{\Omega}$ as in Figure 3.1. The approximation functions are constructed as the product of the kernel functions and correction functions

$$\Psi_I(\mathbf{x}) = C(\mathbf{x}; \mathbf{x} - \mathbf{x}_I) \phi(\mathbf{x} - \mathbf{x}_I) \quad (3.2)$$

The correction functions are used to ensure the RK approximation is able to reproduce a set

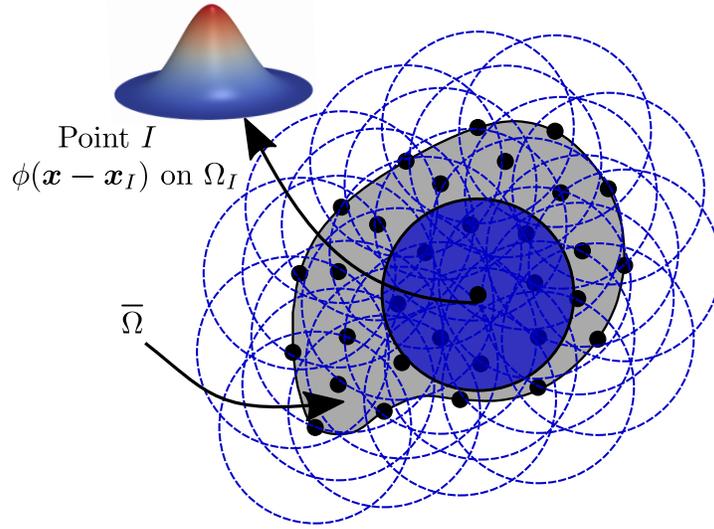


Figure 3.1: Domain and discretization example for a reproducing kernel approximation. A closed domain $\bar{\Omega}$, which is the union of an open domain Ω and its boundary Γ , discretized by a set of points. Each point I has an associated subdomain Ω_I over which a kernel function $\phi(\mathbf{x} - \mathbf{x}_I)$ is defined.

of functions exactly. For example, correction functions for problems in two dimensions that reproduce the set of n^{th} order monomials are defined as

$$C(\mathbf{x}; \mathbf{x} - \mathbf{x}_I) = \sum_{i+j=0}^n b_{ij}(\mathbf{x})(x_1 - x_{1I})^i (x_2 - x_{2I})^j \equiv \mathbf{H}^T(\mathbf{x} - \mathbf{x}_I) \mathbf{b}(\mathbf{x}) \quad (3.3)$$

with

$$\mathbf{H}^T(\mathbf{x} - \mathbf{x}_I) = [1, x_1 - x_{1I}, x_2 - x_{2I}, (x_1 - x_{1I})^2, \dots, (x_2 - x_{2I})^n] \quad (3.4)$$

and

$$\mathbf{b}^T(\mathbf{x}) = [b_{00}(\mathbf{x}), b_{10}(\mathbf{x}), b_{01}(\mathbf{x}), b_{20}(\mathbf{x}), \dots, b_{0n}(\mathbf{x})] \quad (3.5)$$

where $b_{ij}(\mathbf{x})$ are coefficients determined by enforcing the reproducing conditions

$$\sum_{I=1}^{NP} \Psi_I(\mathbf{x}) x_{1I}^i x_{2I}^j = x_1^i x_2^j \quad 0 \leq i + j \leq n. \quad (3.6)$$

Applying (3.3) to (3.6) leads to

$$\mathbf{b}^T(\mathbf{x}) \sum_{I=1}^{NP} \mathbf{H}(\mathbf{x} - \mathbf{x}_I) \mathbf{H}^T(\mathbf{x} - \mathbf{x}_I) \phi(\mathbf{x} - \mathbf{x}_I) = \mathbf{H}^T(\mathbf{0}) \quad (3.7)$$

with

$$\mathbf{H}^T(\mathbf{0}) = [1, 0, \dots, 0]. \quad (3.8)$$

The coefficients can be determined as

$$\mathbf{b}^T(\mathbf{x}) = \mathbf{H}^T(\mathbf{0}) \mathbf{M}^{-1}(\mathbf{x}) \quad (3.9)$$

where the moment matrix $\mathbf{M}(\mathbf{x})$ is

$$\mathbf{M}(\mathbf{x}) = \sum_{I=1}^{NP} \mathbf{H}(\mathbf{x} - \mathbf{x}_I) \mathbf{H}^T(\mathbf{x} - \mathbf{x}_I) \phi(\mathbf{x} - \mathbf{x}_I) \quad (3.10)$$

To solve for $\mathbf{b}(\mathbf{x})$ the moment matrix $\mathbf{M}(\mathbf{x})$ must be invertible. This imposes the requirement that an evaluation location \mathbf{x} is sufficiently covered by kernel functions and is discussed in detail in [56].

The monomial bases are smooth, that is $\mathbf{H}(\mathbf{x} - \mathbf{x}_I) \in C^\infty(\bar{\Omega})$, therefore the smoothness of the approximation functions $\Psi_I(\mathbf{x})$ and thus approximations $u^h(\mathbf{x})$ is controlled by the kernels $\phi(\mathbf{x} - \mathbf{x}_I)$. A common kernel function is the C^2 cubic B-spline

$$\phi(\bar{r}) = \begin{cases} 1 - 6\bar{r}^2 + 6\bar{r}^3 & \text{for } 0 \leq \bar{r} \leq \frac{1}{2} \\ 2 - 6\bar{r} + 6\bar{r}^2 - 2\bar{r}^3 & \text{for } \frac{1}{2} \leq \bar{r} \leq 1 \\ 0 & \text{otherwise,} \end{cases} \quad (3.11)$$

where $\bar{r} = r/R$, the normalized distance at location r for a kernel with extent R (the *support* size). In two dimensions, the cubic B-spline can be used to define a radial kernel using $r = |\mathbf{x} - \mathbf{x}_I|$ and

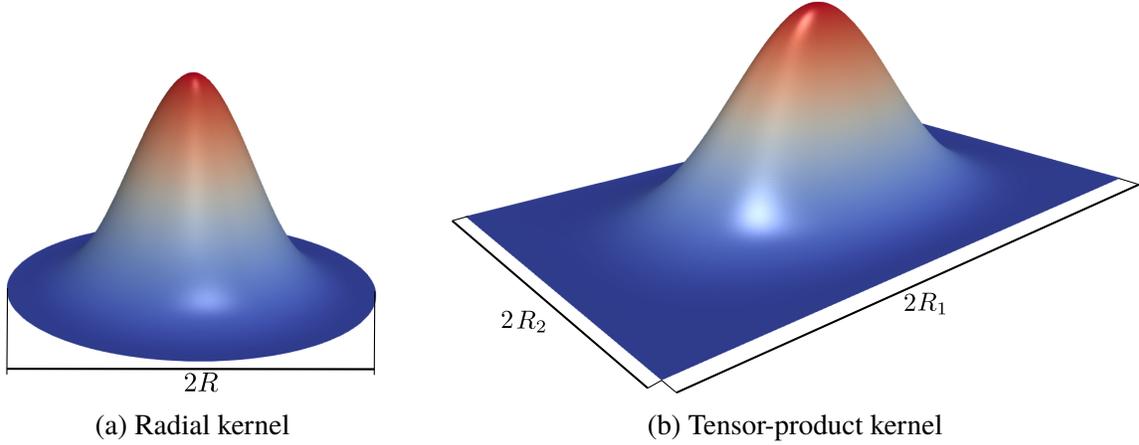


Figure 3.2: Two dimensional kernels constructed using the cubic B-spline

defining R as the support radius. Another common option is to form a tensor product (box) kernel where

$$\phi(\mathbf{x} - \mathbf{x}_I) = \tilde{\phi}\left(\frac{|x_1 - x_{1I}|}{R_1}\right) \tilde{\phi}\left(\frac{|x_2 - x_{2I}|}{R_2}\right) \quad (3.12)$$

with R_1 and R_2 being the support sizes in the two directions and $\tilde{\phi} = \phi$ from (3.11). Examples of the radial and tensor-product kernels are in Figure 3.2

3.1.2 Reproducing Kernel Particle Method

RK approximation functions are used as the test and trial functions in RKPM. For an example application of RKPM we consider an elasticity problem on the domain Ω with boundary $\Gamma = \Gamma_h \cup \Gamma_g$, $\Gamma_h \cap \Gamma_g = \emptyset$ and

$$\nabla \cdot \boldsymbol{\sigma} = \mathbf{b} \quad \text{in } \Omega \quad (3.13)$$

$$\mathbf{n} \cdot \boldsymbol{\sigma} = \mathbf{h} \quad \text{on } \Gamma_h \quad (3.14)$$

$$\mathbf{u} = \mathbf{g} \quad \text{on } \Gamma_g \quad (3.15)$$

where \mathbf{u} is displacement, $\boldsymbol{\sigma} = \boldsymbol{\sigma}(\mathbf{u}) = \mathbf{C} : \boldsymbol{\varepsilon}(\mathbf{u})$ is the Cauchy stress tensor with \mathbf{C} being the fourth-order material tensor and $\boldsymbol{\varepsilon}(\mathbf{u}) = 1/2(\nabla \otimes \mathbf{u} + \mathbf{u} \otimes \nabla)$ is the strain tensor, \mathbf{b} is the body force,

\mathbf{n} the boundary surface normal, \mathbf{h} the prescribed traction on Γ_h and \mathbf{g} the prescribed displacement on Γ_g .

The Galerkin form of the problem becomes: find $\mathbf{u}^h \in U^h \subset U$ such that $\forall \mathbf{v}^h \in V^h \subset V$

$$\int_{\Omega} \boldsymbol{\varepsilon}(\mathbf{v}^h) : \boldsymbol{\sigma}(\mathbf{u}^h) d\Omega = \int_{\Omega} \mathbf{v}^h \cdot \mathbf{b} d\Omega + \int_{\Gamma_h} \mathbf{v}^h \cdot \mathbf{h} d\Gamma \quad (3.16)$$

where $U = H_g^1$, the Sobolev space constrained by the displacement boundary condition and $V = H_0^1$, the Sobolev space of functions with zero trace on the Dirichlet boundary. For the Bubnov-Galerkin case, the same set of approximation functions are used in the trial space, U^h , and test space, V^h .

Using RK approximation functions for V^h and U^h provides great flexibility in the discretization of problem domains but creates challenges because they are typically rational functions and the fields they produce are generally non-interpolatory and not admissible, i.e. $V^h \not\subset H_0^1$ and $U^h \not\subset H_g^1$.

Rational approximation functions pose challenges in integrating the weak form for Galerkin methods. Gauss integration on background cells was commonly used in the early development of RKPM and EFG [11, 12, 13]. Gauss integration can not exactly integrate rational functions and it is difficult to match the support geometry of the approximation functions with the integration cells; therefore, high-order integration is used to achieve reasonable accuracy. To improve the accuracy and efficiency of domain integration, stabilized conforming nodal integration (SCNI) [26] and variationally consistent integration methods [27] have been developed.

3.1.3 Essential Boundary Conditions

An example problem, with domain and discretization shown in Figure 3.3, is constructed to elucidate the general challenges of enforcing boundary conditions in meshfree methods and highlight the characteristics of different techniques. Examples of a window and approximation functions near and on the boundary for traditional RKPM with a tensor-product kernel are shown

in Figure 3.4. Figure 3.4a shows a window function near a boundary that is non-zero on the boundary leading to non-zero values of the point's approximation function at the boundary, Figure 3.4b. Plots of the approximation function values along a boundary segment between two points, 17 and 18, are also shown with Figure 3.4c displaying the values for approximation functions of points on the boundary and Figure 3.4d showing the values for the approximations functions of interior points. The key challenges are shown and include the lack of the Kronecker delta property, i.e.

$$\Psi_I(\mathbf{x}_J) \neq \delta_{IJ}, \quad (3.17)$$

causing the approximation to not be interpolatory, i.e.

$$u^h(\mathbf{x}_I) \neq d_I, \quad (3.18)$$

thus making the coefficients *generalized*; furthermore, the approximation functions do not have the weak Kronecker delta property, where the condition for having the property is,

$$\Psi_I(\mathbf{x}) = 0, \forall \mathbf{x} \in \Gamma_g, I \in \mathcal{V}_o \quad (3.19)$$

where \mathcal{V}_o is the set of all points not on the essential boundary, Γ_g . In other words, weak Kronecker delta is where only nodal points on a given boundary contribute to the approximation on that boundary.

In contrast, the finite element method possesses the Kronecker delta property and thus the solution is interpolatory and, in addition, the weak Kronecker delta condition is present. With these properties boundary conditions can be enforced directly via static condensation for FEM. For meshfree methods, various techniques have been developed to restore all or some of the properties or avoid these issues by enforcing boundary conditions in other ways.

Existing techniques for enforcing boundary conditions can be divided into two classes:

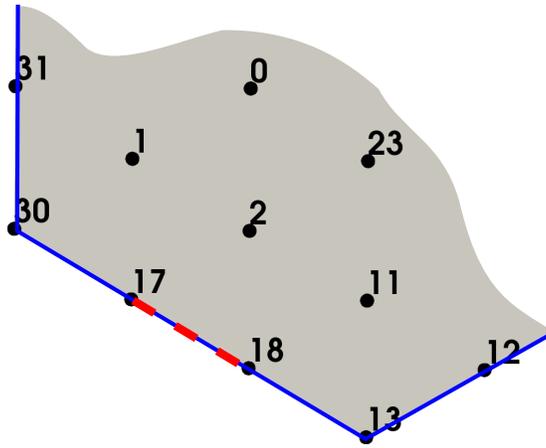
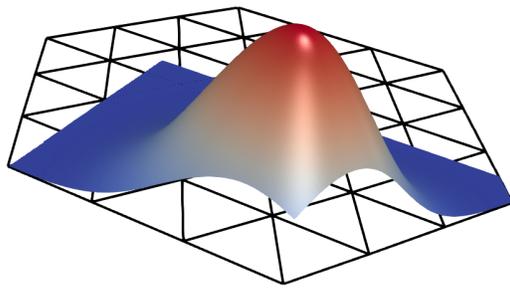
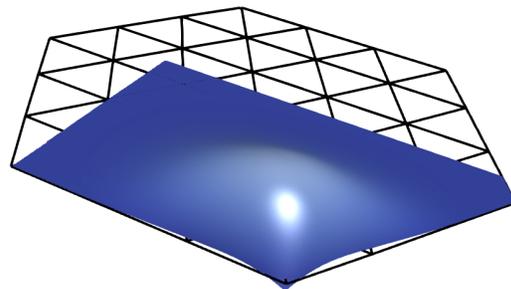


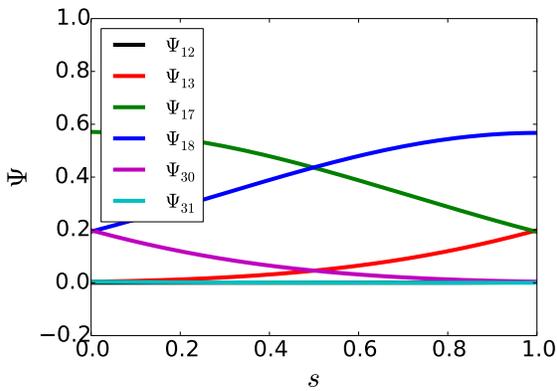
Figure 3.3: Point cloud near and on a domain boundary (blue line). The discretization is used to highlight approximation function behavior on a segment (red dashed line)



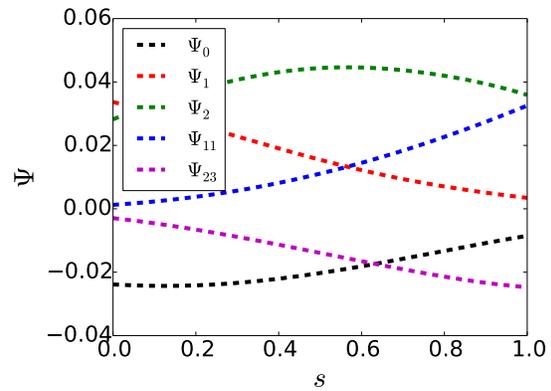
(a) ϕ_{11} , tensor-product



(b) Ψ_{11} , tensor-product



(c) Ψ for points on the boundaries

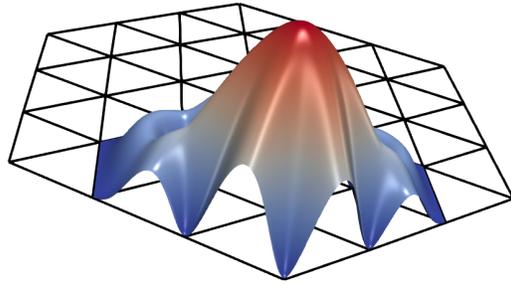


(d) Ψ for points in the interior

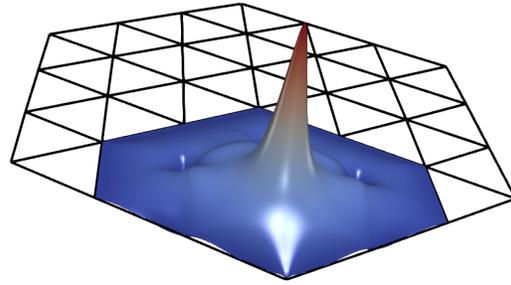
Figure 3.4: Window and approximation functions near a boundary for the common tensor-product definition. The domain and point locations are from Figure 3.3. The top images show the window and approximation functions for a point near but not on the boundary. The bottom plots show approximation functions at the line segment on the boundary.

1) strong enforcement and 2) weak enforcement [25]. With strong enforcement the boundary values are specified and exactly enforced at points or regions of the essential boundary. This has been accomplished by either dealing with the generalized coefficients or modifying the approximations functions near the boundary. A common approach in working with the generalized coefficients is to use Lagrange multipliers [13, 35]. Though the Lagrange multiplier approach is a modified variational approach (thus should be classified as a weak enforcement) it is most commonly implemented where the approximation of the multipliers are delta functions centered at the boundary points, leading to a strong collocation of the solution at these points.

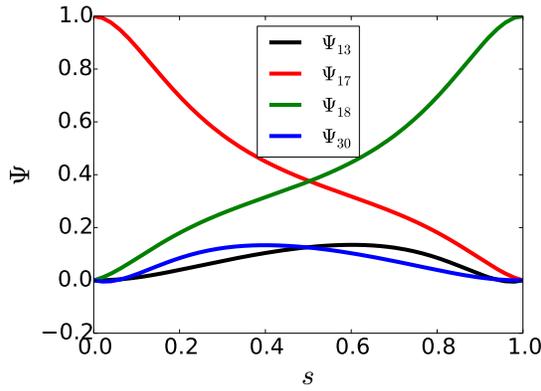
Methods for modifying the meshfree approximation functions to obtain the Kronecker delta property include using singular window functions [35] and modifying the MLS or RK procedure so the interpolation property is achieved [36]. It is important to note that, in general, having the Kronecker delta property in the approximation functions does not imply meeting the weak Kronecker delta condition in (3.19). The aforementioned methods for obtaining the Kronecker delta property, as well as the collocation Lagrange multiplier approach, are collocation methods and do not generally meet the weak Kronecker delta condition and thus the approximation spaces are not admissible. This is emphasized in [34] and is demonstrated in Figure 3.5. In the example, a window function is constructed that has the Kronecker delta property leading to approximation functions with the Kronecker delta property. The means of constructing the window function are deferred to Chapter 4 but the resulting approximation space has the same issues as other collocation methods. As in Figure 3.4, the window and approximation functions for a point near the boundary are shown as well as approximation function values on a boundary section for functions associated with points both on and near the boundary. The lack of the weak Kronecker delta condition is evident by the non-zero values in Figure 3.5d. Due to the lack of the weak Kronecker delta condition, the test functions will not vanish on the essential boundary. In addition, boundary enforcement via collocation only provides values at the collocation points. The values at the collocation points are insufficient to describe the entire boundary. Without



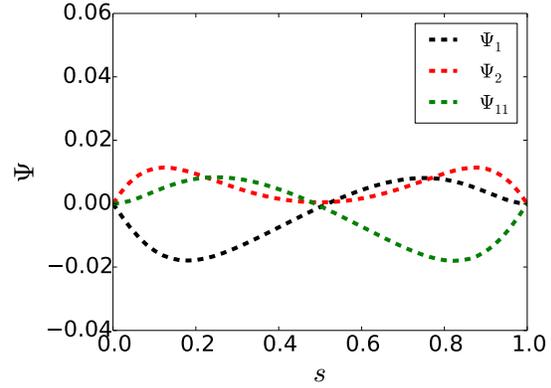
(a) ϕ_{11} , Kronecker delta



(b) Ψ_{11} , Kronecker delta



(c) Ψ for points on the boundaries



(d) Ψ for points in the interior

Figure 3.5: Window and approximation functions near a boundary for the Kronecker delta window definition. The domain and point locations are from Figure 3.3. The top images show the window and approximation functions for a point near but not on the boundary. The bottom plots show approximation functions at the line segment on the boundary.

sufficient information to describe the entire essential boundary displacement field, these methods are not consistent and thus do not pass patch tests. A correction was proposed in [34] where a residual along the boundary was also minimized.

Coupling with FEM [37, 38] provides a means of strongly enforcing boundaries with admissible functions spaces because FEM approximation functions meet the weak Kronecker delta condition. This has proven to be effective, however the approximation continuity returns to C^0 loosing this advantage of meshfree methods. Another method includes modifying the window dilation parameters, i.e. R or R_1 and R_2 , such that the window functions conform to the boundaries [40]. This approach works well for simple geometries but appears very challenging

for arbitrary domains.

Weak enforcement can be accomplished using Nitsche's method [57, 38, 39], the penalty method [58] or the general form of Lagrange multipliers [11]. The penalty method is not consistent in the sense that the weak form does not recover the strong form of the problem. With the Lagrange multiplier method, additional unknowns are introduced and thus the size of the system grows. As it is a mixed method, to guarantee stability the choice of spaces must satisfy the Babuška-Brezzi stability condition. Nitsche's method can be viewed as a combination of the penalty method and the modified variational principle [59], and it offers stable and optimal convergence with properly selected penalty numbers [38, 39].

In Nitsche's method we find $\mathbf{u}^h \in U^h$ such that $\forall \mathbf{v}^h \in U^h$

$$\begin{aligned} \int_{\Omega} \boldsymbol{\varepsilon}(\mathbf{v}^h) : \boldsymbol{\sigma}(\mathbf{u}^h) d\Omega - \int_{\Gamma_g} \mathbf{n} \cdot \boldsymbol{\sigma}(\mathbf{v}^h) \cdot \mathbf{u}^h d\Gamma - \int_{\Gamma_g} \mathbf{v}^h \cdot \mathbf{n} \cdot \boldsymbol{\sigma}(\mathbf{u}^h) d\Gamma + \beta \int_{\Gamma_g} \mathbf{v}^h \cdot \mathbf{u}^h d\Gamma \\ = \int_{\Omega} \mathbf{v}^h \cdot \mathbf{b} d\Omega + \int_{\Gamma_h} \mathbf{v}^h \cdot \mathbf{h} d\Gamma + \int_{\Gamma_g} \mathbf{n} \cdot \boldsymbol{\sigma}(\mathbf{v}^h) \cdot \mathbf{g} d\Gamma + \beta \int_{\Gamma_g} \mathbf{v}^h \cdot \mathbf{g} d\Gamma. \end{aligned} \quad (3.20)$$

The method can be viewed as a combination of the penalty and the Lagrange multiplier methods. To obtain the Lagrange multiplier method, the traction terms, $\mathbf{n} \cdot \boldsymbol{\sigma}(\mathbf{u}^h)$ and $\mathbf{n} \cdot \boldsymbol{\sigma}(\mathbf{v}^h)$, are replaced with Lagrange multipliers $\boldsymbol{\lambda}$ and their variations $\boldsymbol{\gamma}$ respectively and the penalty terms are removed, i.e. $\beta = 0$. To obtain the penalty method from (3.20) the terms with traction are removed. With the penalty method, increasing the values of the penalty parameter β causes better enforcement of the boundary conditions but this reduces the conditioning of the system. The conditioning also decreases with increasing β in Nitsche's method but lower values give agreeable boundary enforcement. Additionally, Nitsche's method is consistent and no additional unknowns are required.

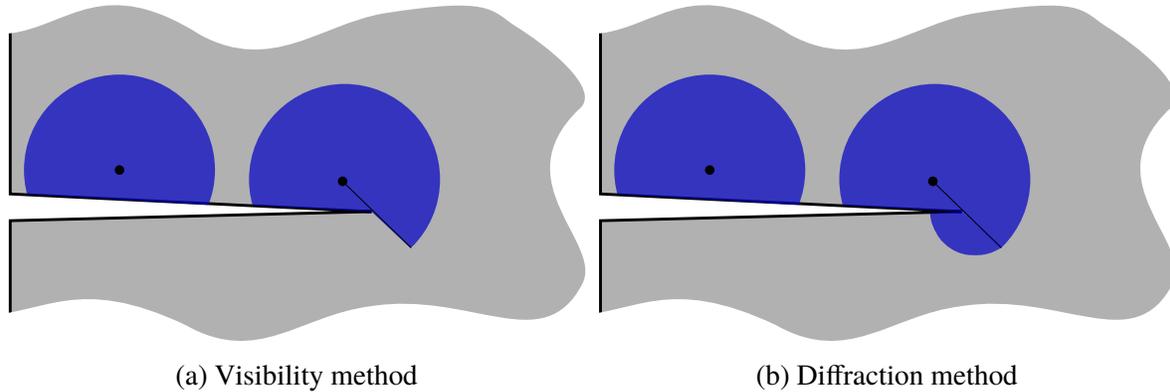
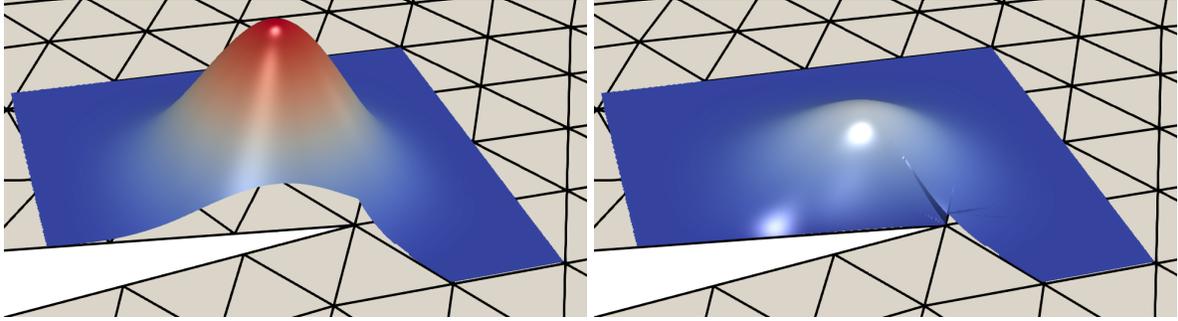


Figure 3.6: Example window domains for different methods of handling a non-convex boundary.

3.1.4 Non-convex Domains

The approach of defining windows using purely Euclidean metrics also causes problems near non-convex boundaries. A point on the opposite side of a re-entrant corner or a thin crack will appear close, causing window and approximation functions to overlap and resulting in stiffness terms between the points when there should be none. Solutions make use of boundary information to modify the window functions. The visibility criteria truncates functions where the line-of-sight between an evaluation location and the function's owning node is obstructed by a domain boundary [28]. An example of the truncated window domain for a point near a non-convex region is shown in Figure 3.6a. The truncation causes the window function, and thus the solution field, to be discontinuous. The discontinuities in the window and approximation function can be seen in Figure 3.7. Discontinuities can be problematic for the continuous Galerkin formulation but it has been proven to not prevent convergence here [29]. The diffraction method was developed to maintain continuous approximations functions [30]. It modifies the visibility truncation by smoothly decaying a window function after visibility is lost. An example of a window domain is in Figure 3.6b.



(a) ϕ , tensor-product, visibility

(b) Ψ , tensor-product, visibility

Figure 3.7: Window and approximation function with visibility criteria near a non-convex boundary.

3.1.5 Material Interfaces

The higher smoothness of meshfree methods is advantageous in most scenarios; however, it is unfavorable in problems where the solution is not as smooth. For example, a weak discontinuity (discontinuous strain) exists at material interfaces. Different enriching techniques have been developed to insert strong and weak discontinuities as needed [31, 32, 33].

3.2 Spline Functions on Triangulations in Two Dimensions

Bernstein-Bézier

The method of constructing conforming window functions described in Chapter 4 utilizes the theory of smooth functions defined on triangulations. A brief introduction to constructing and using such spline functions is provided below. The notation and definitions follow that of Lai and Schumaker [45, 60]. Detailed overviews can be found in their books.

A bivariate polynomial p is represented in Bernstein-Bézier form as

$$p = \sum_{i+j+k=d} c_{ijk} B_{ijk}^d, \quad (3.21)$$

where c_{ijk} are real-valued and known as the Bézier or B-coefficients and B_{ijk}^d are Bernstein

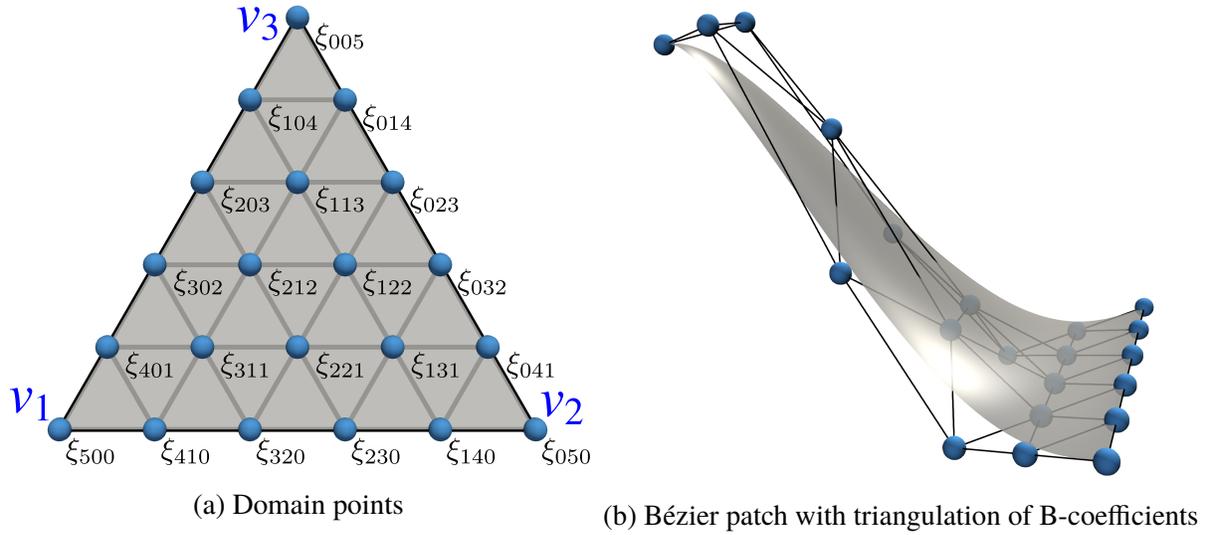


Figure 3.8: Domain points and Bézier patch for a triangle with polynomial order, $d = 5$.

basis polynomials of degree d . Bernstein basis polynomials are expressed using the barycentric coordinates, b_1, b_2, b_3 of a triangle and are defined

$$B_{ijk}^d := \frac{d!}{i!j!k!} b_1^i b_2^j b_3^k, \quad i + j + k = d, \quad (3.22)$$

with i, j, k being non-negative. Some properties of the Bernstein basis polynomials include partition of unity and non-negativity.

An indexing method is used to identify the basis polynomials. The indexing assigns a geometric location, a *domain point*, to the basis functions. The set of domain points for a triangle T of order d are defined:

$$\mathcal{D}_{d,T} := \left\{ \xi_{ijk}^T := \frac{iv_1 + jv_2 + kv_3}{d} \right\}_{i+j+k=d} \quad (3.23)$$

where v_1, v_2 and v_3 are the location of the three vertices of the triangle.

The domain points and an example Bézier surface for a triangle with $d = 5$ are shown in Figure 3.8.

To define functions over a more complex region it is convenient to work with groups of

triangles known as triangulations.

Definition 1. A collection $\Delta := \{T_1, \dots, T_{n_t}\}$ of triangles in a plane is called a *triangulation* of a polygonal domain $\Omega = \cup_{i=1}^{n_t} T_i$ provided that if a pair of triangles in Δ intersect, the intersection is either a common vertex or edge.

The set of all vertices in the triangulation is denoted as \mathcal{V} and, likewise, \mathcal{E} is the set of all edges.

The *domain points* of the triangulation are

$$\mathcal{D}_{d,\Delta} := \bigcup_{T \in \Delta} \mathcal{D}_{d,T}, \quad (3.24)$$

with domain points on the vertices or edges shared by two triangles included just once.

A concept that is commonly used is the *star*.

Definition 2. If v is a vertex of triangulation Δ , then we define the *star* of v , which we denote by $star(v) := star^1(v)$, to be the set of all triangles in Δ which share the vertex v . We define $star^i(v)$ inductively for $i > 1$ to be the set of all triangles in Δ which have a nonempty intersection with some triangle in $star^{i-1}(v)$.

3.2.1 Smooth Joins

A C^0 join of functions described on the triangles is accomplished by asserting that the B-coefficients of the coincident edges are also the same for each triangle. Joins of higher smoothness, i.e. C^r with $r > 0$, can be accomplished by relating directional derivatives between adjacent triangles. To define a directional derivative we first need the concept of a *direction vector* described with directional coordinates.

Definition 3. Given any vector $\mathbf{u} \in \mathbb{R}^2$, let

$$a_i := \alpha_i - \beta_i, \quad i = 1, 2, 3,$$

where $(\alpha_1, \alpha_2, \alpha_3)$ and $(\beta_1, \beta_2, \beta_3)$ are the barycentric coordinates relative to T of two points \mathbf{w} and $\tilde{\mathbf{w}}$ such that $\mathbf{u} = \mathbf{w} - \tilde{\mathbf{w}}$. Then a_1, a_2, a_3 are called the *directional coordinates of \mathbf{u} relative to T* .

The *directional derivative* at a point $\mathbf{v} \in \mathbb{R}^2$ is then defined as,

$$D_{\mathbf{u}}f(\mathbf{v}) := \frac{d}{dt}f(\mathbf{v} + t\mathbf{u})|_{t=0} = \lim_{t \rightarrow 0} \frac{f(\mathbf{v} + t\mathbf{u}) - f(\mathbf{v})}{t} \quad (3.25)$$

with directional derivatives of order n being,

$$D_{\mathbf{u}}^n f(\mathbf{v}) := \frac{d^n}{dt^n} f(\mathbf{v} + t\mathbf{u})|_{t=0}. \quad (3.26)$$

For a C^r smooth join of two triangles, $T := \langle v_1, v_2, v_3 \rangle$ and $\tilde{T} := \langle v_4, v_3, v_2 \rangle$, as in Figure 3.9a, described with polynomials p and \tilde{p} respectively and joined along edge $e := \langle v_2, v_3 \rangle$ we have the relation

$$D_{\mathbf{u}}^n p(\mathbf{v}) = D_{\mathbf{u}}^n \tilde{p}(\mathbf{v}), \quad \forall \mathbf{v} \in e \text{ and } n = 0, \dots, r, \quad (3.27)$$

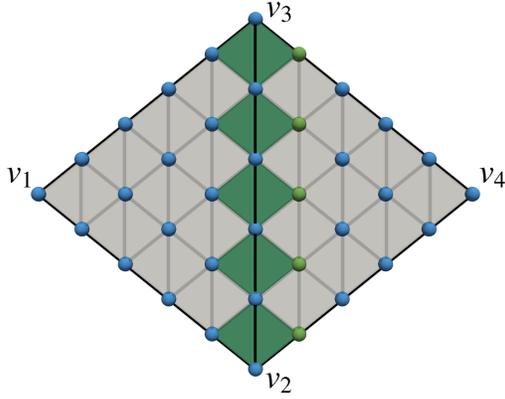
where \mathbf{u} is any direction not parallel with e . This leads to the continuity constraints,

$$\tilde{c}_{njk} = \sum_{\nu+\mu+\kappa=n} c_{\nu k+\mu j+\kappa} B_{\nu\mu\kappa}^n(v_4), \quad j+k = d-n, \quad n = 0, \dots, r. \quad (3.28)$$

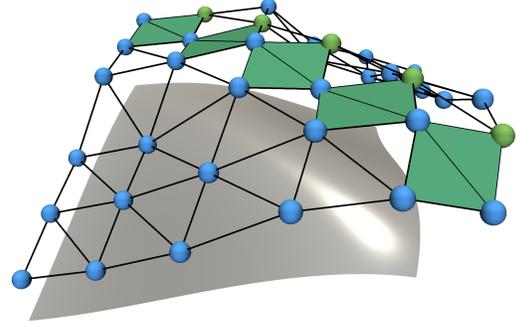
Visually, for a C^1 join, the triangulated B-coefficients of each edge segment are co-planar, see Figure 3.9b.

3.2.2 Spaces of Smooth Splines

A space can be defined using spline functions on triangulations. To efficiently define and communicate the order and smoothness of the spaces we introduce some notation. First, we define C^0 spline spaces.



(a) Domain points



(b) Bézier patch with triangulation of B-coefficients

Figure 3.9: A Bézier patch with C^1 continuity. The coefficients in each green shaded triangle pair are co-planar with dependent control points (green spheres) determined by free points (blue spheres) through continuity constraints. The triangulated B-coefficients are translated up to better show the Bézier patch.

Definition 4. Given a triangulation, $\Delta := \{T_1, \dots, T_{n_i}\}$, of a domain Ω , and d a positive integer, let

$$\mathcal{S}_d^0(\Delta) := \{s \in C^0(\Omega) : s|_{T_i} \in \mathcal{P}_d, i = 1, \dots, n_i\}.$$

where $s|_{T_i}$ is the portion of the spline function s on the triangle T_i and \mathcal{P}_d is the space of polynomials of degree d . We call $\mathcal{S}_d^0(\Delta)$ the *space of C^0 splines of degree d on triangulation Δ* .

Next, we define spaces with higher smoothness.

Definition 5. Given integers r and d with $0 \leq r < d$, let

$$\mathcal{S}_d^r(\Delta) := \{C^r(\Omega) \cap \mathcal{S}_d^0(\Delta)\}.$$

We call $\mathcal{S}_d^r(\Delta)$ the *space of splines of degree d and smoothness r* .

Also, some subspaces have higher smoothness at the vertices.

Definition 6. Suppose r, ρ, d are integers such that $0 \leq r \leq \rho < d$, and let

$$\mathcal{S}_d^{r,\rho}(\Delta) := \{s \in \mathcal{S}_d^r(\Delta) : s \in C^\rho(v) \forall v \in \mathcal{V}\},$$

We call $\mathcal{S}_d^{r,\rho}(\Delta)$ a *superspline space of degree d and smoothness r, ρ* .

Minimal Determining Sets and Hermite Interpolation

Creating smooth joins of the spline functions of two or more triangles is accomplished by constraining some coefficients to be functions of others. That is, a subset of the coefficients fully define the spline function while the remaining coefficients are dependent on their values and the chosen smoothness of the space. The set of defining coefficients are called the *minimal determining set*, \mathcal{M} . For example, in Figure 3.9, the blue coefficients are the minimal determining set. The minimal determining set is not unique for a given triangulation as dependent and defining coefficients can be interchanged. In Figure 3.9, the defining coefficients for the C^1 join could have been chosen to be in the right triangle instead of the left.

Depending on the application, it can be convenient to associate the coefficients to function values and derivatives, i.e. the Hermite data, called *nodal parameters*, at locations in the triangulation. This process creates a set of linear functionals called the *nodal minimal determining set*, \mathcal{N} , which is useful in performing Hermite interpolation.

Example Spline Space

An example bivariate spline space is the Argyris space, $\mathcal{S}_5^{1,2}(\Delta)$. The function is completely defined by six coefficients at each vertex and one coefficient at each edge. An example minimal determining set is shown in Figure 3.10. The filled markers represent the defining coefficients and the open marker values are constrained such that the intended smoothness holds. The Argyris space retains the full approximation power of the broader quintic space from which it is a subset, i.e. it can reproduce fifth order polynomial functions exactly. [45]

The six coefficients at each vertex are responsible for the value and first and second derivatives, i.e. nodal parameters, at that vertex, giving the space C^2 continuity at the vertices. The coefficient for each edge is responsible for the cross derivative, and C^1 continuity, at the edge.

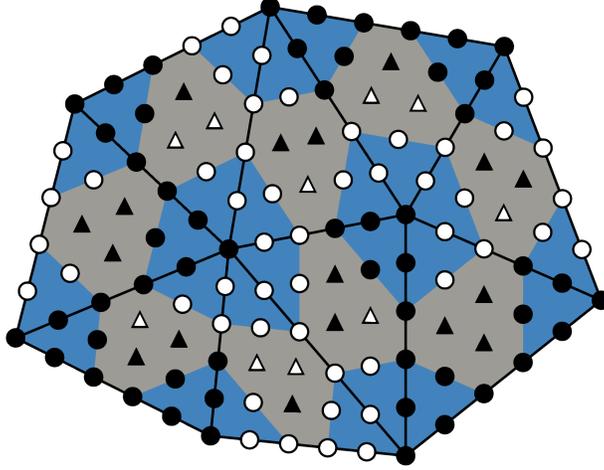


Figure 3.10: An example minimal determining set for the Argyris space, $\mathcal{S}_5^{1,2}(\Delta)$. Circles represent coefficients associated with a vertex (blue region) with solid circles being the defining coefficients and open circles are constrained. Triangles represent coefficients associated with edges with filled triangles being the defining coefficients and open triangles represent the constrained coefficients.

These nodal parameters define the nodal minimal determining set:

$$\mathcal{N} := \bigcup_{v \in \mathcal{V}} \mathcal{N}_v \cup \bigcup_{e \in \mathcal{E}} \mathcal{N}_e \quad (3.29)$$

where

$$\mathcal{N}_v := \left\{ \epsilon_v D_x^\alpha D_y^\beta \right\}, 0 \leq \alpha + \beta \leq 2, \quad (3.30)$$

$$\mathcal{N}_e := \left\{ \epsilon_{\eta_e} D u_e \right\} \quad (3.31)$$

where η_e is the midpoint of edge e and u_e is the unit vector normal to e and ϵ_t is the point evaluation functional defined as $\epsilon_t f = f(t)$.

3.3 Spline Functions on Triangulations in Three Dimensions

Bernstein-Bézier

Spline functions on triangulations can be naturally extend into higher dimensions. Three dimensional spline functions, defined on tetrahedra, are the foundations of three-dimensional conforming window functions. Following the definitions in the previous section and that of Lai and Schumaker [45, 60], a trivariate polynomial p is represented in Bernstein-Bézier form as

$$p = \sum_{i+j+k+l=d} c_{ijkl} B_{ijkl}^d, \quad (3.32)$$

where c_{ijkl} are the Bézier coefficients and B_{ijkl}^d are Bernstein basis polynomials of degree d . Using the barycentric coordinates, b_1, b_2, b_3, b_4 of a tetrahedron, the Bernstein basis polynomials are defined

$$B_{ijkl}^d := \frac{d!}{i!j!k!l!} b_1^i b_2^j b_3^k b_4^l, \quad i + j + k + l = d, \quad (3.33)$$

with i, j, k, l being non-negative. As in two dimensions, the Bernstein basis polynomials form a partition of unity and are non-negative.

The set of domain points for a tetrahedron T of order d are defined:

$$\mathcal{D}_{d,T} := \left\{ \xi_{ijkl}^T := \frac{iv_1 + jv_2 + kv_3 + lv_4}{d} \right\}_{i+j+k+l=d} \quad (3.34)$$

where v_1, v_2, v_3 and v_4 are the location of the four vertices of the tetrahedron. The domain points for a tetrahedron with $d = 3$ are shown in Figure 3.11a.

Similar to triangulations in two dimension, we have *tetrahedral partitions* in three dimensions.

Definition 7. A collection $\Delta := \{T_1, \dots, T_n\}$ of tetrahedra in \mathbb{R}^3 is called a *tetrahedral partition* of a polygonal domain $\Omega = \cup_{i=1}^n T_i$ provided that any pair of tetrahedra in Δ intersect at most a

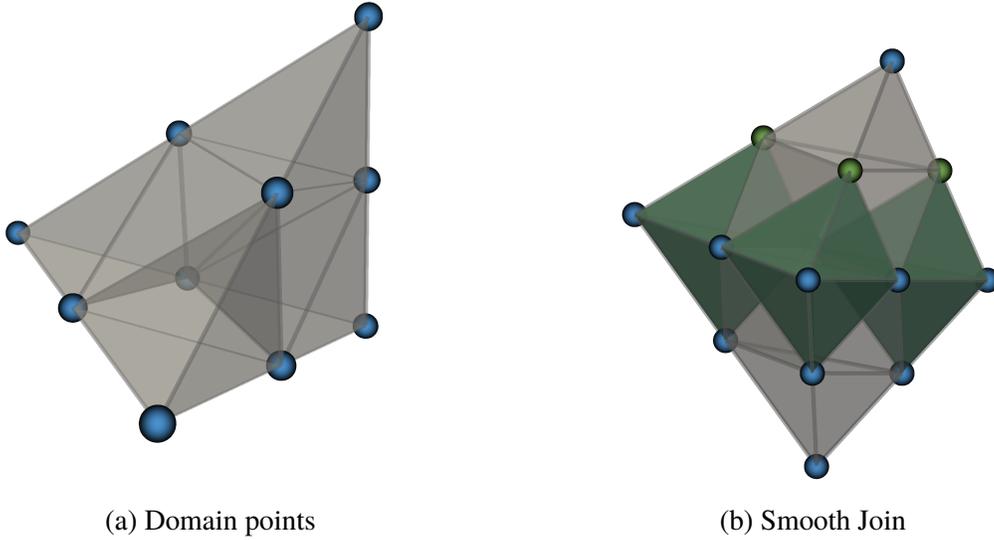


Figure 3.11: Domain points of a tetrahedron and representation of constraints for a C^1 join of for two tetrahedra with polynomial order, $d = 2$. The coefficients of the dependent control points (green spheres) are determined by the free points (blue spheres)

vertex, along a common edge or along a common triangular face.

Tetrahedral partitions will also be referred to as triangulations herein. The set of all vertices in the tetrahedral partitions is denoted as \mathcal{V} , \mathcal{E} is the set of all edges and \mathcal{F} is the set of all faces.

The *domain points* of the tetrahedral partition are

$$\mathcal{D}_{d,\Delta} := \bigcup_{T \in \Delta} \mathcal{D}_{d,T}, \quad (3.35)$$

with domain points on vertices, edges or faces shared by two tetrahedra included just once.

The concept of a *star* is also extended into three dimensions.

Definition 8. If v is a vertex of tetrahedral partitions Δ , then we define the *star* of v , which we denote by $star(v) := star^1(v)$, to be the set of all tetrahedra in Δ which share the vertex v . We define $star^i(v)$ inductively for $i > 1$ to be the set of all tetrahedra in Δ which have a nonempty intersection with some tetrahedra in $star^{i-1}(v)$.

3.3.1 Smooth Joins

A C^0 join of functions described on the tetrahedra is accomplished by asserting that the B-coefficients of the coincident faces are the same for each tetrahedra. Joins of higher smoothness, i.e. C^r with $r > 0$, can be accomplished by relating directional derivatives between adjacent tetrahedra. In three dimensions the directional derivative is as described in equations (3.25) and (3.26) with the *direction vector* modified to be

Definition 9. Given any vector $\mathbf{u} \in \mathbb{R}^3$, let

$$a_i := \alpha_i - \beta_i, \quad i = 1, 2, 3, 4,$$

where $(\alpha_1, \alpha_2, \alpha_3, \alpha_4)$ and $(\beta_1, \beta_2, \beta_3, \beta_4)$ are the barycentric coordinates relative to T of two points \mathbf{w} and $\tilde{\mathbf{w}}$ such that $\mathbf{u} = \mathbf{w} - \tilde{\mathbf{w}}$. Then a_1, a_2, a_3, a_4 are called the *directional coordinates of \mathbf{u} relative to T* .

For a C^r smooth join of two tetrahedra, $T := \langle v_1, v_2, v_3, v_4 \rangle$ and $\tilde{T} := \langle v_5, v_4, v_2, v_3 \rangle$, as in Figure 3.11b, described with polynomials p and \tilde{p} respectively and joined along face $F := \langle v_2, v_3, v_4 \rangle$ we have the relation

$$D_{\mathbf{u}}^n p(\mathbf{v}) = D_{\mathbf{u}}^n \tilde{p}(\mathbf{v}), \quad \forall \mathbf{v} \in F \text{ and } n = 0, \dots, r, \quad (3.36)$$

where \mathbf{u} is any direction not parallel with F . This leads to the continuity constraints,

$$\tilde{c}_{mijk} = \sum_{\nu+\mu+\kappa+\delta=m} c_{\nu i+\mu k+\kappa j+\delta} B_{\nu\mu\kappa\delta}^m(v_5), \quad i+j+k = d-m, \quad m = 0, \dots, r. \quad (3.37)$$

3.3.2 Spaces of Smooth Splines

As in two dimension, a space can be defined using spline functions on triangulations. For C^0 spline spaces, given a tetrahedral partition, $\Delta := \{T_1, \dots, T_n\}$, of a domain Ω , and d a positive

integer,

$$\mathcal{S}_d^0(\Delta) := \{s \in C^0(\Omega) : s|_{T_i} \in \mathcal{P}_d, i = 1, \dots, n_t\}. \quad (3.38)$$

where $s|_{T_i}$ is the portion of the spline function s on the tetrahedra T_i and \mathcal{P}_d is the space of polynomials of degree d .

For spaces with higher smoothness, given integers r and d with $0 \leq r < d$, we have,

$$\mathcal{S}_d^r(\Delta) := \{C^r(\Omega) \cap \mathcal{S}_d^0(\Delta)\}, \quad (3.39)$$

We also have superspline spaces. Suppose r, ρ, μ, d are integers such that $0 \leq r \leq \mu \leq \rho < d$, we have

$$\mathcal{S}_d^{r,\rho,\mu}(\Delta) := \{s \in \mathcal{S}_d^r(\Delta) : s \in C^\rho(v) \forall v \in \mathcal{V}, s \in C^\mu(e) \forall e \in \mathcal{E}\}. \quad (3.40)$$

with $C^\rho(v)$ and $C^\mu(e)$ implying the polynomial has smoothness of ρ at vertex v and μ and edge e , respectively.

The concepts of the minimal determining set and Hermite interpolation in three dimensions remain unchanged for higher dimensions, except for the additional directional components to the derivatives.

Example Spline Space on a Tetrahedral Partition

The extension of the concepts of the Argyris space into three dimensions leads to $\mathcal{S}_9^{1,4,2}(\Delta)$. The function is completely defined by 35 coefficients at each vertex, 8 coefficients at each edge 7 coefficients at each face and 4 coefficients in each tetrahedron. The space does have full, 9th order, approximation power. The considerable complexity to obtain C^1 smoothness for this space has been motivation to find alternative methods.

Example Spline Space on a the Alfeld Split of a Tetrahedral Partition

A means to obtain the desired smoothness with lower order spaces is to take advantage of subdivisions or *splits* of the input triangulations. Numerous spaces are defined in this manner for both two-dimensional and three-dimensional splines [45]. The additional complexity of three dimensional space $\mathcal{S}_9^{1,4,2}(\Delta)$ makes such using splits a common practice. An example subdivision is the *Alfeld split* [47].

Definition 10. Given a tetrahedron $T := \langle v_1, v_2, v_3, v_4 \rangle$, with the barycenter of T being $v_T := (v_1 + v_2 + v_3 + v_4)/4$. The Alfeld split T_A of T is the four subtetrahedra obtained by connecting v_T to each of the vertices of T .

Δ_A , the Alfeld split of a tetrahedral partition Δ is obtained by applying the Alfeld split to each tetrahedron in Δ .

The C^1 space defined on the *Alfeld split* is

$$\mathcal{S}_5^1(\Delta_A) := \{s \in \mathcal{S}_5^1(\Delta_A) : s \in C^2(v) \forall v \in \mathcal{V}, s \in C^4(v_T) \forall T \in \Delta\}. \quad (3.41)$$

meaning C^2 supersmoothness occurs at each vertex in Δ and C^4 supersmoothness at the added barycenter vertices used to make Δ_A .

The function is completely defined by 10 coefficients at each vertex, 2 coefficients at each edge 3 coefficients at each face and 1 coefficient in each tetrahedron and does have the full, quintic, approximation power. These coefficients relate to the nodal parameters and are used to define the nodal minimal determining set:

$$\mathcal{N} := \bigcup_{v \in \mathcal{V}} \mathcal{N}_v \cup \bigcup_{e \in \mathcal{E}} \mathcal{N}_e \cup \bigcup_{F \in \mathcal{F}} \mathcal{N}_F \cup \bigcup_{T \in \Delta} \mathcal{N}_T \quad (3.42)$$

where

$$\mathcal{N}_v := \left\{ \epsilon_v D_x^\alpha D_y^\beta \right\}, 0 \leq \alpha + \beta \leq 2, \quad (3.43)$$

$$\mathcal{N}_e := \left\{ \epsilon_{\eta_e} D_{\mathbf{u}_e^i}, i = 1, 2 \right\}, \quad (3.44)$$

$$\mathcal{N}_F := \left\{ \epsilon_{\eta_\xi} D_{\mathbf{u}_F}, \xi \in A_F^0 \right\}, \quad (3.45)$$

$$\mathcal{N}_T := \left\{ \epsilon_{v_T} \right\} \quad (3.46)$$

where ϵ_t is the point evaluation functional defined as $\epsilon_t f = f(t)$, η_e is the midpoint of edge e and \mathbf{u}_e^i , for $i = 1, 2$, are orthogonal unit vectors normal to e , ϵ_{η_ξ} are unit vectors normal to F at the three points in A_F^0 , defined for each face $F := \langle v_1, v_2, v_3 \rangle \in \mathcal{F}$ as

$$A_F^0 := \left\{ \xi_{211}^{F,4}, \xi_{121}^{F,4}, \xi_{112}^{F,4} \right\}, \quad (3.47)$$

that is, points that lie on F but are located at the domain points of the quartic space on F and not the quintic space.

Material from this Chapter appears in *Computer Methods in Applied Mathematics and Engineering*, “Conforming window functions for meshfree methods”, 347:588 – 621, 2019, by Jacob Koester and J.S. Chen.

Chapter 4

Conforming Window Functions

Traditional window functions are defined using Euclidean distances and angles making it difficult to conform the functions to arbitrary geometries. The key idea of the conforming window function is to define the functions on subdomains that conform to the parent domain boundaries. The window functions are then used, in place of the traditional window functions, when constructing the meshfree approximation functions. The rest of the meshfree formulation remains the same except the new approximation functions possess the desired properties that simplify handling of non-convex regions, material interfaces and boundary conditions and interactions.

The conceptual steps in creating conforming windows are:

- Choose the subdivision strategy and create subdomains for each window
- Select the function space (on the subdivisions) for building the window function
- Construct the functions by choosing the coefficients of the space

The following sections describe a procedure for constructing conforming window functions in two dimensions followed by a section highlighting the considerations needed for constructing conforming window functions in three dimensions.

4.1 Window function domains

As evident in the finite element method, subdivisions of a domain provide a convenient means of defining functions that conform to the domain. There are many ways to subdivide a domain. Here, we choose to use triangulations as they can be efficiently generated.

A requirement for the window functions used in the construction of meshfree approximation functions, as described in Chapter 3, is that they overlap to cover the entire domain. Thus, the conforming window subdivisions must overlap. A set of overlapping subdomains can be created from a triangulation of the entire domain. For example, given a triangulation Δ of the entire domain Ω , we can use the collection of all vertices, \mathcal{V} , of Δ as the meshfree nodes. The window function domain of vertex $v \in \mathcal{V}$, Δ_v , can then be defined as $star^j(v)$ for $j \geq 1$. Figure 4.1 shows a domain triangulation with two example subdomains constructed using a second order stars ($j = 2$).

In general, a triangulation of the entire domain is not a requirement. Window functions can also be defined on local triangulations created individually about each vertex, Figure 4.2a. In addition, conforming window functions can be used selectively where needed, for example, near an essential boundary, and Euclidean windows used elsewhere, Figure 4.2b.

4.2 Window function space

With the window subdivisions established, we now select a space to use for building the window functions. The Bernstein-Bézier spline spaces described in Chapter 3 are a favorable set of spaces as they are well-suited to constructing functions with Hermite data, i.e. function values and derivatives at specific locations, and there is established theory for building arbitrarily smooth spaces.

Since the window functions control the smoothness of the approximation functions, the desired smoothness will dictate the requirements of the space. Though not required, better than

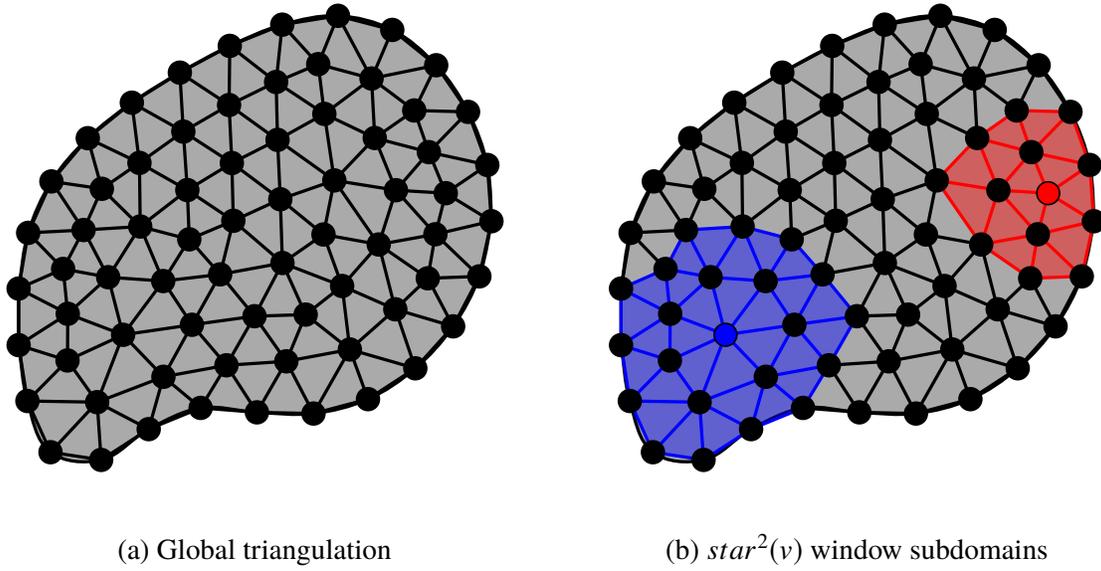


Figure 4.1: Full domain triangulation used for building conforming window functions

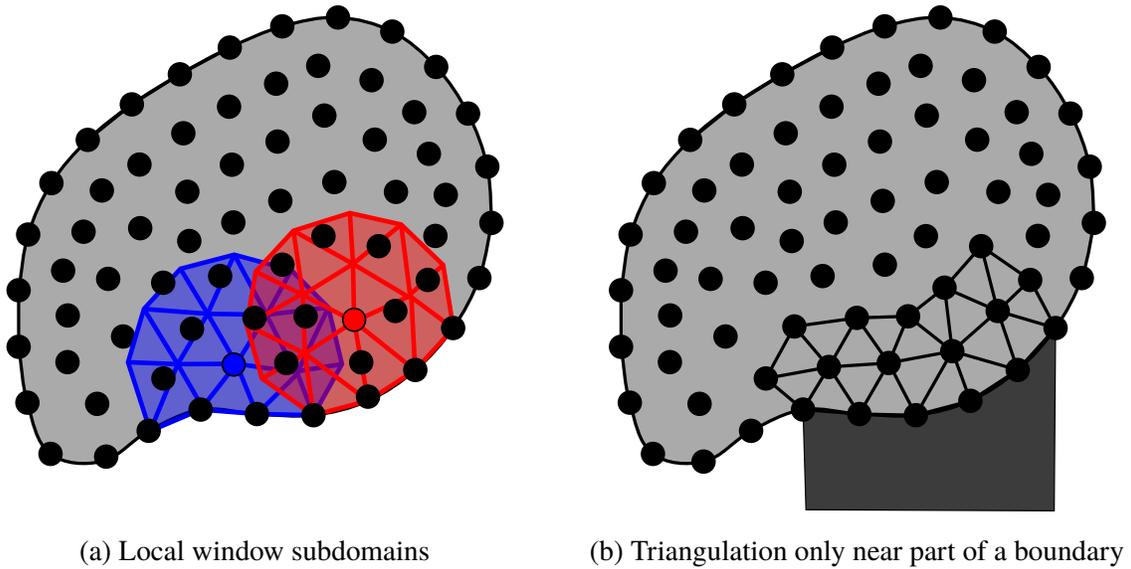


Figure 4.2: Local triangulations for building conforming window functions

C^0 smoothness has been attributed to higher solution accuracy for stress analyses using meshfree methods. To retain this benefit we can choose to use a space that is C^1 . Here we choose the Argyris space, $\mathcal{S}_5^{1,2}(\Delta)$, that was summarized in 3.2.2.

Recall that the Argyris space is completely defined by six coefficients at each vertex and one coefficient at each edge, associated with the nodal parameters at these locations. Given the Bernstein-Bézier form and bases from Equations (3.21) and (3.22), respectively, the Bézier coefficients can be calculated from the nodal parameters. For example, referring to the locations in Figure 3.8, the coefficients of spline s near v_1 of $T := \langle v_1, v_2, v_3 \rangle$ can be determined [45]

$$c_{500} = f_1^{00} \quad (4.1)$$

$$c_{410} = [\tilde{x}_2 f_1^{10} + \tilde{y}_2 f_1^{01}] / 5 + f_1^{00} \quad (4.2)$$

$$c_{401} = [\tilde{x}_3 f_1^{10} + \tilde{y}_3 f_1^{01}] / 5 + f_1^{00} \quad (4.3)$$

$$c_{320} = [\tilde{x}_2^2 f_1^{20} + 2\tilde{x}_2 \tilde{y}_2 f_1^{11} + \tilde{y}_2^2 f_1^{02}] / 20 + 2c_{410} - f_1^{00} \quad (4.4)$$

$$c_{311} = [\tilde{x}_2 \tilde{x}_3 f_1^{20} + (\tilde{x}_2 \tilde{y}_3 + \tilde{x}_3 \tilde{y}_2) f_1^{11} + \tilde{y}_2 \tilde{y}_3 f_1^{02}] / 20 + c_{401} + c_{410} - f_1^{00} \quad (4.5)$$

$$c_{302} = [\tilde{x}_3^2 f_1^{20} + 2\tilde{x}_3 \tilde{y}_3 f_1^{11} + \tilde{y}_3^2 f_1^{02}] / 20 + 2c_{401} - f_1^{00} \quad (4.6)$$

where the nodal parameters $f_i^{\alpha\beta} = D_x^\alpha D_y^\beta s(x_i, y_i)$, with (x_i, y_i) the coordinates of v_i and $\tilde{x}_i := x_i - x_1$ and $\tilde{y}_i := y_i - y_1$ for $i = 2, 3$.

Also, for example, the coefficients near the edge between v_2 and v_3 can be determined from other coefficients and the desired directional derivative, f_e , perpendicular to e [45]

$$c_{122} = \frac{16}{30a_1} f_e - \frac{1}{6} [c_{140} + 4c_{131} + 4c_{113} + c_{104}] \quad (4.7)$$

$$- \frac{a_2}{6a_1} [c_{050} + 4c_{041} + 6c_{032} + 4c_{023} + c_{014}] \quad (4.8)$$

$$- \frac{a_3}{6a_1} [c_{041} + 4c_{032} + 6c_{023} + 4c_{014} + c_{005}] \quad (4.9)$$

where (a_1, a_2, a_3) are the directional coordinates of a unit vector normal to the edge and pointing

into the triangle. Similar relations can be derived for the other vertices and edges of the triangle.

The associated smoothness of the space (C^1 at edges and C^2 at vertices for the Argyris space) is achieved by determining the coefficients for every triangle in the triangulation using the same nodal parameters at the shared vertices and edges. Explicit constraint equations do not have to be constructed when determining the coefficients in this manner.

4.3 Specifying window functions coefficients

With the window triangulations, space and the relationship between coefficients and nodal parameters established, we now chose a systematic way of selecting the nodal parameters for the window functions. The radial cubic B-spline, (3.11), is commonly used for typical, Euclidean window functions. It has the desired properties of its peak value being at the center, smoothly decaying to zero at a set radius and being C^2 continuous. The cubic B-spline takes input of a distance, r , and support size, R , to return its value. The angle, θ , with respect to the x -axis of the radial B-spline's local coordinate system is also needed to determine derivatives in Cartesian coordinates. To get the nodal parameter values needed to build the conforming window functions, the cubic B-spline function is also used but with the distance and support size input modified to be relevant to the window triangulation and to consider the boundary of the domain. The modified input is only needed at the nodal parameter locations with the spline functions on the triangles providing the desired properties elsewhere. The next two sections describe two alternate distance and support inputs and window construction methods using a global triangulation, followed by a short description of constructing window functions using local triangulation that don't come from a global mesh and sections on modifying the triangulations and input to get the desired properties for handling non-convexities, weak discontinuities and essential boundaries.

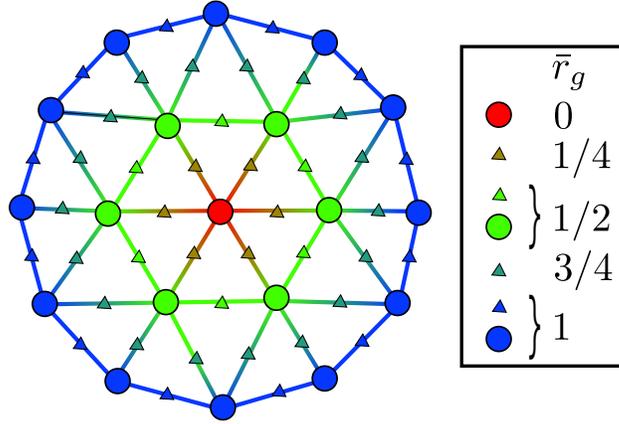


Figure 4.3: Normalized graph distances, \bar{r}_g , at the $\mathcal{S}_5^{1,2}(\Delta_v)$ nodal parameter locations for a second order star in the interior of the domain. The nodal parameter locations are the vertices (circles) and the edge midpoints (triangles)

4.3.1 Star window functions

For the first approach we consider inputs that are defined using graph distances. The required support size is set to be a graph extent, $R := R_g$, which is the order of star in which the window function will have value. That is, for $R_g = 2$, the window function will have value up until the boundary of the $star^2(v)$ sub-triangulation, where v is the center vertex that “owns” the function. The distance to a nodal parameter location, n_I , is then determined to be the graph distance, $d_g(v, n_I)$, resulting in a normalized distance of $\bar{r} := \bar{r}_g = d_g(v, n_I)/R_g$. These modified inputs can then be supplied to the cubic B-spline function which returns the Hermite data to be used to build the window function.

The normalized graph distances of a triangulation for a window function away from a domain boundary are depicted in Figure 4.3. For the Argyris space, graph distances are needed at the vertices and the midpoints of edges. For edges, the distance is taken as a linear interpolation between the endpoints.

To determine the values from the radial cubic B-spline for the requisite derivatives in the Hermite data, values for θ and a Euclidean extent, R , are needed. The values need to only be

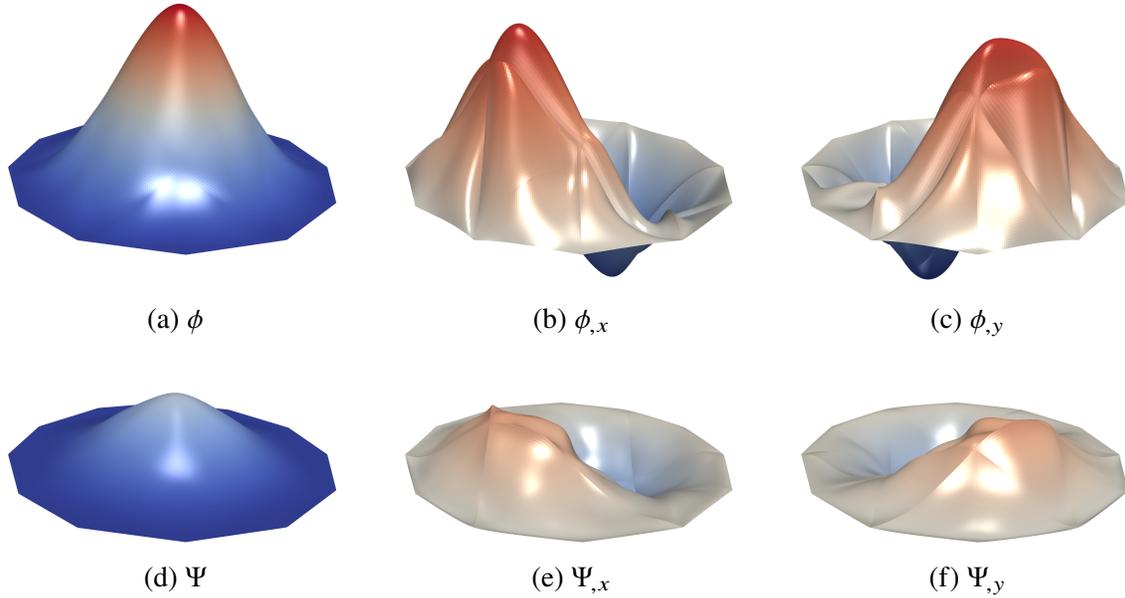


Figure 4.4: Interior, $star^2(v)$, conforming window and approximation function and their derivatives.

estimates as they are only used to set the window function Hermite data and precise data are not required. An appropriate R can be estimated by, for example, taking the radius of a circle that circumscribes the local window triangulation. For θ , the Euclidean angle is used.

An example $star^2(v)$ window function, ϕ , approximation function, Ψ , and their first derivatives for a location away from a conforming boundary are shown in Figure 4.4. The functions decay to zero at the edge of the triangulation as designed and the C^1 smoothness is evident from the derivative plots.

If a global triangulation is used, $R_g = 1$ and linear bases are reproduced, the same approximation functions as a linear triangle element are generated. This can be useful in coupling with finite elements.

4.3.2 Snap window functions

The shapes of the $star^i(v)$, graph-based, window functions have a strong connection to the shapes of the triangles in the input triangulation. Like FEM, the quality of the triangles will affect

the quality of the approximations functions. With $R_g > 1$, the CRK approximation functions will be less local than FEM functions, mitigating some of the approximation function quality issues. Nonetheless, if a global input triangulation is used, it is desirable to further separate the quality of the approximation function from the quality of the triangulation. To accomplish this, alternative support size and distance metrics have been determined for input into the cubic B-spline function.

For the *snap star* window function, a global triangulation is again used to extract a sub-triangulation. The sub-triangulation is defined with two regions, 1) the triangles and vertices completely contained in a Euclidean circle, defined with a support of R and 2) the triangles that intersect the Euclidean circle and their vertices that are outside of the circle, as illustrated in Figure 4.5. For nodal parameter locations inside of radius R , the support size is defined to be R and the distance is simply the Euclidean distance. Thus, the normalized distance at these location will be ≤ 1 . For the vertices outside of R but in triangles intersected by R , the normalized distance is set to 1. The distances at other nodal parameter locations of the intersected triangles are linearly interpolated from the vertices. An example of this technique and how it compares to the $star^2(v)$ approach for a triangulation containing poorly shaped triangles is depicted in Figure 4.6.

The snap star is defined using Euclidean metrics but with the extent of the function “snapping” to the input triangulation. For triangles inside of R , the window function is the projection the radial cubic B-spline function to the spline functions defined on the triangles using collocation of the Hermite data at the nodal parameter locations. Outside of R , the functions smoothly decay to zero.

The Euclidean metric can find vertices that are on the opposite side of a concavity, effectively filling the gap. A simple requirement can be added to the search to prevent this from happening. The requirement is that a vertex of the triangulation is graph connected to the center vertex through a path that is entirely inside the support circle. Essentially, the triangulation needs to be a single contiguous set of triangles.

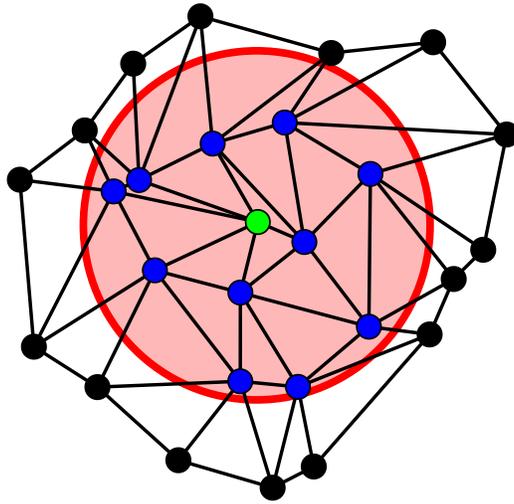
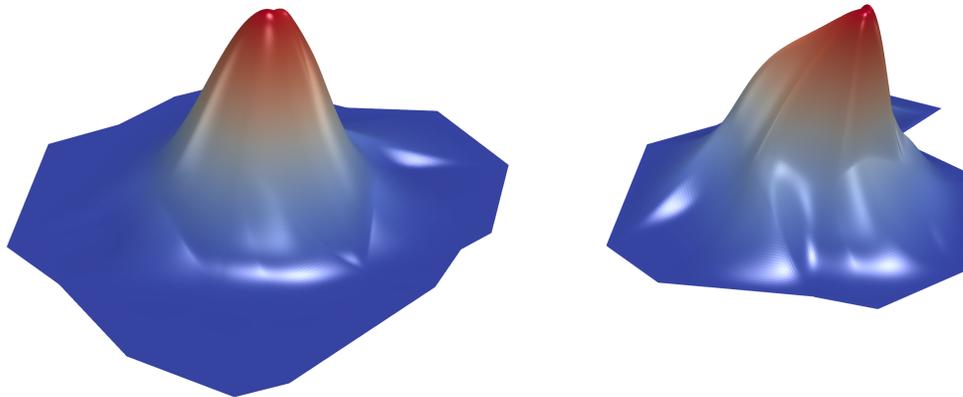


Figure 4.5: Illustration of the snap star window function constructed about the green vertex. The red circle with radius of $1.2h$ is the set extent reference. Vertices inside the radius, in blue, use a Euclidean distance. Vertices of triangles that are intersected by the circle but not inside, in black, have a normalized distance of 1.



(a) ϕ , snap window

(b) ϕ , $star^2(v)$ window

Figure 4.6: Conforming window functions using the triangulation in Figure 4.5 for a snap star with support of $1.2h$ and a $star^2(v)$ window

4.3.3 Local window functions

A global input mesh is not required as the approximation functions are constructed using meshfree techniques. As such, individual window triangulations can be constructed and used instead of being extracted from a global triangulation. The procedure for constructing the local star windows is: 1) choose a support radius, individually for each window, such that all the window functions overlap to cover the entire domain, 2) fill each circle with a triangulation, a first order star about the vertex is sufficient and 3) use the support radius and Euclidean distances to set the Hermite data of each window function. The triangulation is modified to include boundary edges when it is near a domain boundary.

As an example, an outline of a procedure for constructing local triangulations is as follows and depicted in Figure 4.7. First, set the support radii such that the entire domain is covered by overlapping windows. Then, loop over the nodes and:

1. Find if the window overlaps a domain boundary. If it does not overlap the boundary, a tensor product or radial window function can be used. If it does overlap the boundary, continue.
2. Put the boundary vertices and edges the window will overlap in a set. (Figure 4.7b)
3. Check if the line-of-sight between the center vertex and the vertices in the boundary set is obstructed by any of the boundary edges, if so, remove the obstructed vertex and associated edges from the set. (Figures 4.7c and Figure 4.7d)
4. Take the remaining boundary edges and create triangles by connecting the vertices of the edges to the center vertex. (Figure 4.7e)
5. Complete the triangulation: (Figure 4.7f)
 - (a) Start from one end of the boundary edges
 - (b) Choose an increment in angle

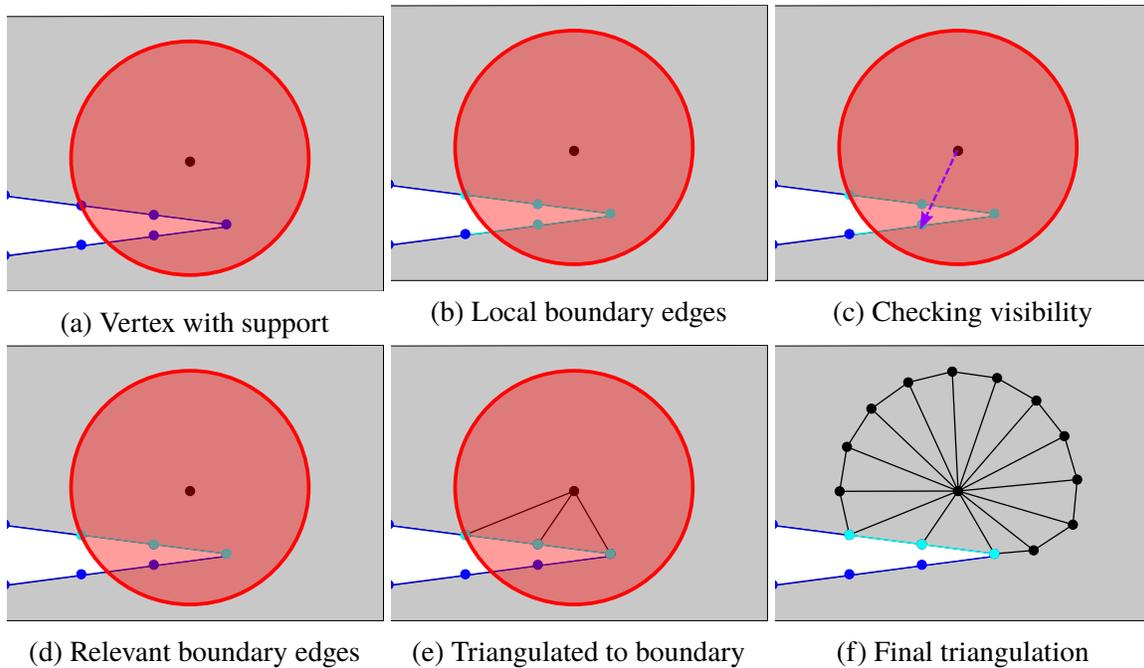


Figure 4.7: Steps for creating a local triangulation that conforms to the boundary

- (c) Increment the angle and place a vertex at that angle and at the support radius. The vertices next to the boundary edges can be placed at a reduce radius to prevent a large change in radius.
- (d) Create a triangle by creating an edge to the previous vertex and an edge to the center vertex
- (e) Continue until the other boundary edge is reached

A first order star that conforms to the boundary is created, is star convex and can be used for the conforming window functions. The window function from the example in Figure 4.7 is shown in Figure 4.8.

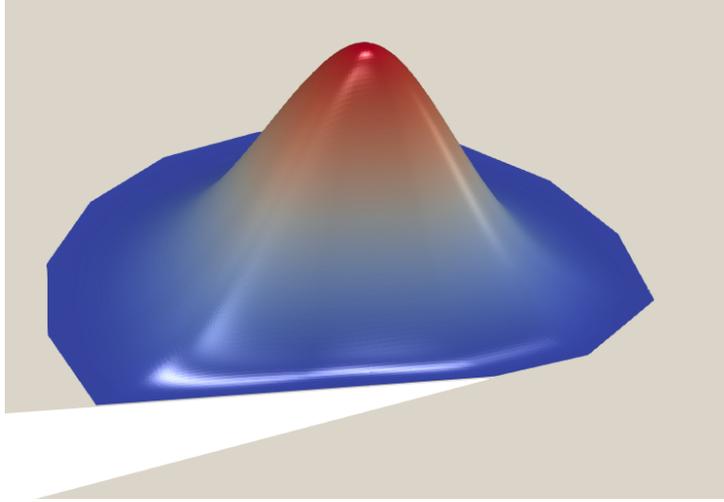


Figure 4.8: Resulting local window function created using the local triangulation.

4.4 Modification for conforming to surfaces

The procedures for the $star^i(v)$ and snap windows mentioned above detail how to construct window functions on triangulations but have not mentioned considerations for the domain boundary. Using those methods, functions near the boundary will not necessarily have zero value on the boundary. Two methods of obtaining the boundary conforming property are summarized here, both are modifications applicable to any of the methods mentioned above, including the local windows).

4.4.1 Smoothly conforming to boundaries

The first approach uses a conditional to override the determined distances for boundary locations,

$$\bar{r} = \begin{cases} 1 & \forall n_I \in \mathcal{V}_b \cup \mathcal{E}_b, v \notin \mathcal{V}_b \\ r/R & \text{otherwise} \end{cases} \quad (4.10)$$

where n_I is a nodal parameter location, v is the window center, r and R represents the distance and support metrics and \mathcal{V}_b and \mathcal{E}_b are the sets of vertices and edges on conforming boundaries, respectively.

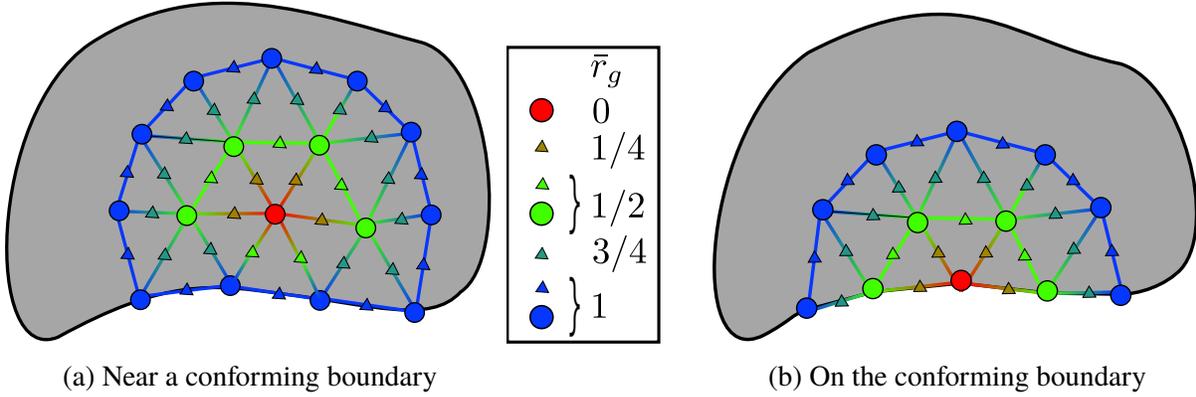


Figure 4.9: Normalized graph distances, \bar{r}_g , at the $\mathcal{S}_5^{1,2}(\Delta_v)$ nodal parameter locations for a second order star about points near or on a conforming boundary. The nodal parameter locations are the vertices (circles) and the edge midpoints (triangles)

Examples of how the conditional modifies distances for a $star^2(v)$ window are shown in Figures 4.9a and 4.9b. Setting the normalized distance to 1 for boundary vertices will set the function values and derivatives to zero, i.e. the function will smoothly conform to the boundary. If the center vertex is on the boundary then no modification will occur.

4.4.2 Sharply conforming to boundaries

Having zero-valued derivatives on the boundary is not a requirement and, with the Argyris space, can cause some undesirable fluctuations near the boundary as first and second derivatives are controlled at vertices but only the first derivative along the edge. A method has been developed to provide “sharper” conformity of the window functions, i.e. conforming to zero on the boundary but with non-zero derivatives. If boundary edges are collinear, the resulting functions will remain C^1 . If the boundary edges are not collinear, the resulting function will be C^0 . This is because triangulations provide continuous but non-smooth, i.e. G^0 , representations of arbitrary surfaces. C^1 functions that are defined on G^0 domains and diminish on the boundary will have zero-valued first derivatives at vertices where two non-collinear boundary edges meet.

To build the functions, a space that is expanded from the Argyris space is defined where

vertices on conforming boundaries and edges between interior and conforming boundary vertices are only required to be C^0 instead of C^2 and C^1 respectively. The key task is then the determination of coefficients at locations near the conforming boundary where now, in general, the nodal parameters vary between triangles that are attached to the same vertex on the conforming boundary. In short, instead of modifying the graph distances near conforming boundaries as in (4.10), sections of the Bézier control surfaces are rotated and coefficients are adjusted so functions are zero on the boundaries. The steps for constructing these “sharp” conforming window functions are:

1. Initial values for the nodal parameters are determined and coefficients associated with vertices are set using assuming that the window function will not conform to any of the boundaries
2. The nodal parameters for vertices on conforming boundaries are adjusted such that the functions are zero at conforming boundaries.
3. The coefficients associated with edges are set using the coefficients near the vertices, as determined in steps 1 and 2, and the nodal parameters from step 1.

Figure 4.10 shows the geometry that is used in determining the adjustments to coefficients near a conforming boundary vertex in step 2. The logic uses the relationship where triangulated coefficients on C^1 edges are co-planar as articulated in Figure 3.9b. Coefficients are rotated about the axis at the intersection of the plane that bisects the attached conforming edges (e and \tilde{e}) and the plane that intersects the coefficient at the vertex (c_0 at ξ_0) and is parallel to the plane of the triangulation. The angle of rotation is the average of the angles α and $\tilde{\alpha}$, where α is the angle between the plane that intersects c_0 and the plane that containing the rotation axis and the adjacent domain points for edge e (c_1 at ξ_1) and $\tilde{\alpha}$ is the angle between the plane that intersects c_0 and the plane containing the rotation axis and the adjacent domain points for edge \tilde{e} (\tilde{c}_1 at $\tilde{\xi}_1$). After rotation the coefficients are shifted down by c_0 . Finally, all coefficients along the conforming

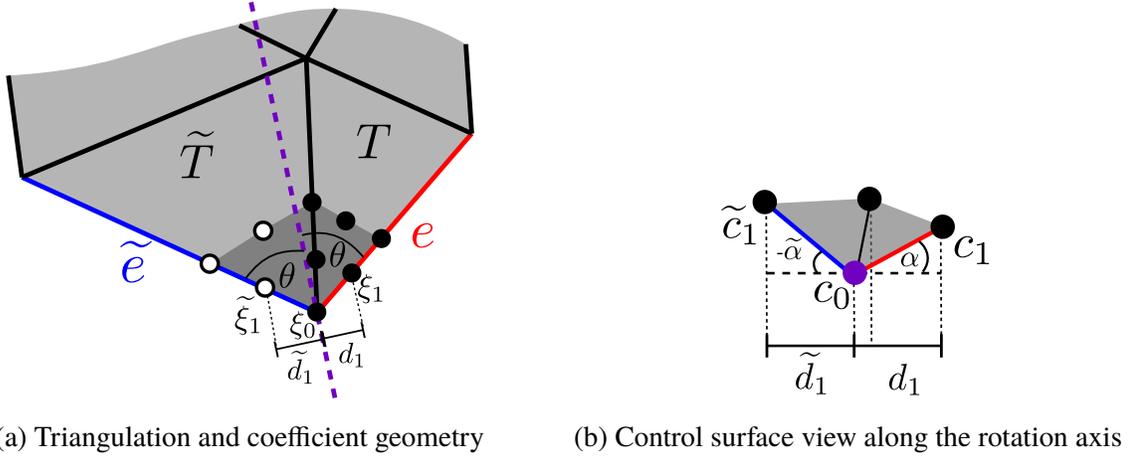


Figure 4.10: Geometry used for setting coefficients associated with a conforming boundary vertex when using the C^0 window design method. Coefficients are rotated about an axis (purple dashed line)

boundary are set to zero. If edges e and \tilde{e} are collinear and, since the edges were C^1 prior to the adjustment, the function will remain C^1 as α will equal $\tilde{\alpha}$. If edges e and \tilde{e} are not collinear, the final step reduces the window function smoothness to C^0 at the conforming boundary vertex and the edges between this vertex and interior vertices.

While this method can reduce the overall smoothness of the approximation space to C^0 , the degraded continuity is only present near non-collinear conforming boundary edges and it only affects window functions centered near but not on the boundary. Also, the jumps in derivatives at these locations will be substantially less than when coupling with linear finite elements. Another option is to work with smoother representations of the boundary, i.e G^r with $r > 0$.

4.4.3 Examples of boundary conforming window functions

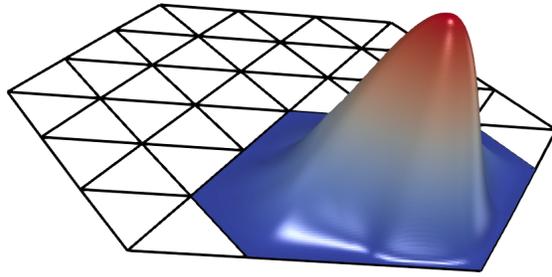
The window and approximation functions of Figure 4.11 show the result of using the two boundary conforming methods for $star^2(v)$ conforming windows near a boundary and can be compared to the regular window example provided in Figure 3.4. Both methods work as designed and the approximations functions for interior points are zero along the boundary (they have

the weak Kronecker delta property) and thus lead to admissible test spaces and, if the essential boundary conditions are spanned by the chosen bases, kinematically admissible approximation spaces, like FEM. The window function using the smooth conforming method show the function smoothly ramping to zero on the boundary whereas the sharp conforming method drops quickly to zero when approaching the boundary. The “pinching” effect of the smooth method causes more fluctuation in the approximation functions near the boundary. Derivatives of the approximations functions plotted along lines near and parallel to the boundary are shown in Figure 4.12. Both methods result in C^1 functions near point 18 (attached to two collinear edges) with the sharp method decreasing the fluctuation in the derivative. Near point 13 (the corner point) the expected jump in derivative is noticed with sharp method.

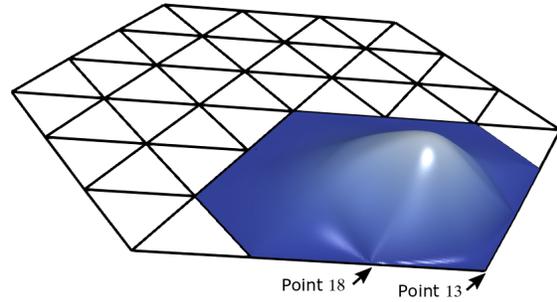
4.5 Modification for convexity improvement

The local triangulation are constructed to be star convex and the $star^i(v)$ and snap windows prevent bleeding across most convex regions; however, some bleeding around corners can occur with the global triangulation-based methods. An additional check can be added to improve the convexity of the window domains. Similar to the visibility method, a line-of-sight between the center and potential vertex is checked when building the triangulation. If the view of the potential vertex is obscured by a domain boundary, the nodal parameters of the owning triangle are set to zero and the triangle can be removed from the domain.

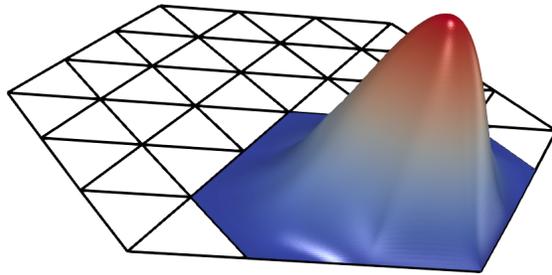
Examples of $star^2(v)$ conforming window functions near a non-convex domain are shown in Figure 4.13 and can be compared to Figure 3.7. The window functions are constructed with and without the convexity improvement check. For the point shown, some vertices in $star^2(v)$ are on the other side of the crack tip causing the window function to wrap. On the right, the convexity improvement criteria is added to prevent any wrapping from occurring. The window function for both methods smoothly decays to zero and the resulting approximation functions are C^1 .



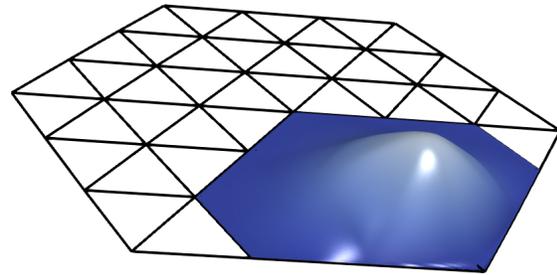
(a) ϕ_{11} , smooth conforming, $star^2(v)$



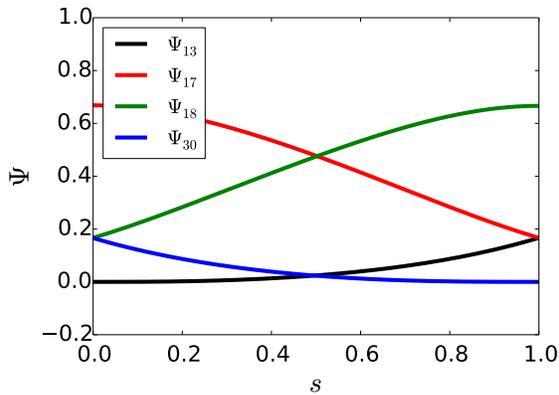
(b) Ψ_{11} , smooth conforming, $star^2(v)$



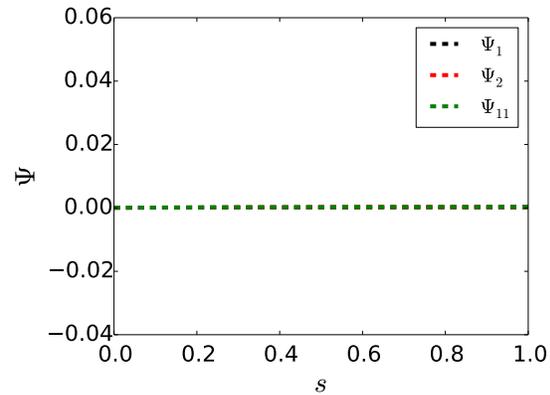
(c) ϕ_{11} , sharp conforming, $star^2(v)$



(d) Ψ_{11} , sharp conforming, $star^2(v)$



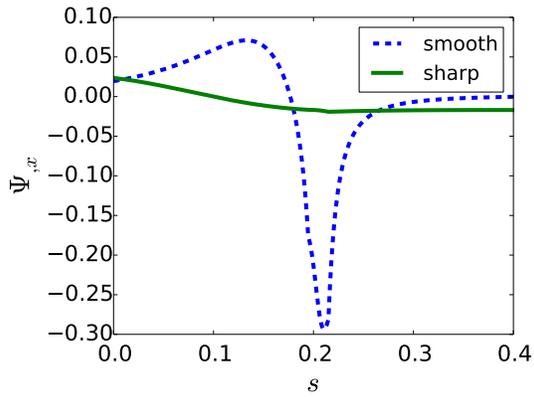
(e) Ψ for points on the boundaries



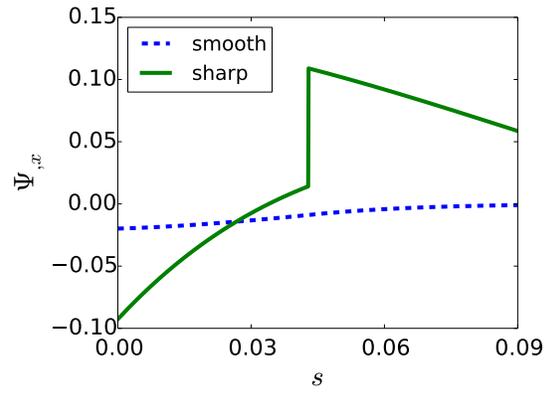
(f) Ψ for points in the interior

Figure 4.11: Window and approximation functions for a point near but not on a boundary for the smooth (top row) and sharp (middle row) $star^2(v)$ conforming window function definitions.

The domain and point locations are from Figure 3.3. Both conforming methods result in the same approximation function values on the boundary (bottom row).

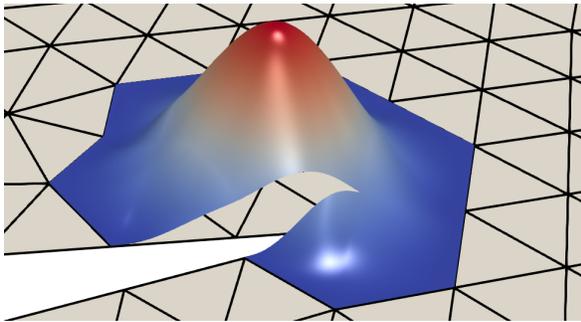


(a) $\Psi_{11,x}$ near point 18

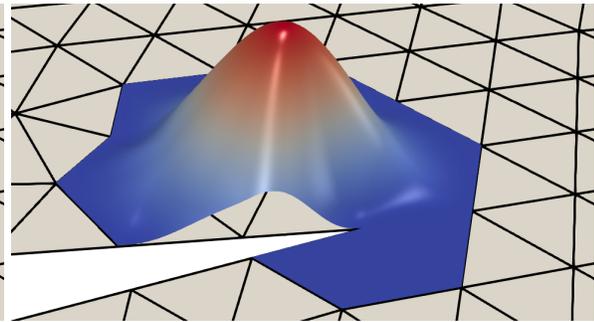


(b) $\Psi_{11,x}$ near point 13

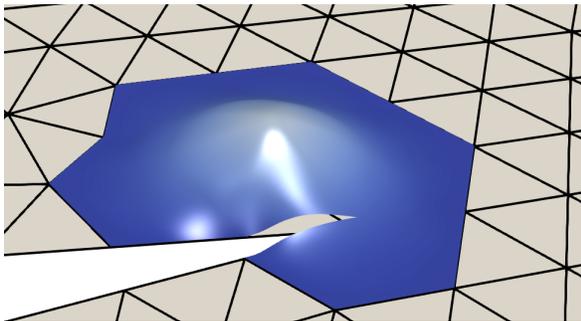
Figure 4.12: Approximation functions derivatives for the smooth and sharp $star^2(v)$ conforming window function definitions plotted along line segments parallel to the boundary and near point 18 (left) and corner point 13 (right).



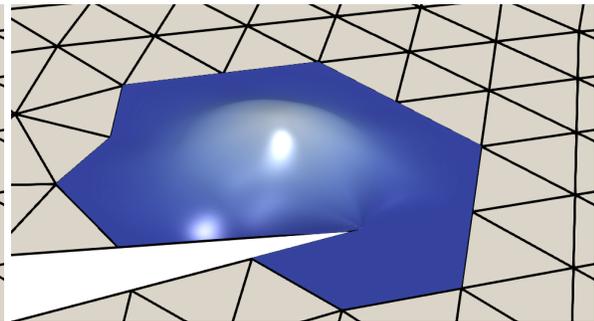
(a) ϕ , no modification



(b) ϕ , improved convexity



(c) Ψ , no modification



(d) Ψ , improved convexity

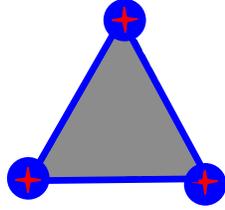
Figure 4.13: $star^2(v)$ window and approximation functions near a non-convex boundary, with (right) and without (left) using the check to improve convexity.

4.6 Simplifications for implementation

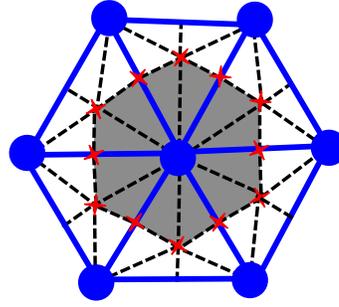
A significant simplification to the process can be made when a global triangulation is used to construct the window functions and provide integration domains. The simplification stems from the Hermite interpolation property of the splines used for the window functions. Values of the window functions are set at the nodal parameter locations. For example, with the Argyris space, values and first and second derivatives at vertices are inputs and thus are known without building or evaluating the window functions. As a consequence, if all the evaluation locations used for domain and boundary integration are at nodal parameter locations, no construction and evaluation of the window function is required. Integration methods exist that are well suited for pairing with this simplification.

The assumed strain technique, as described in [26], uses boundary integration of approximation functions around integration cells to approximate the derivatives of the functions inside the cells. This approach was designed to alleviate the challenges of integrating meshfree approximation functions and has many desirable properties including the ability to pass a linear path test and improve efficiency. One way to pair the assumed strain technique with this simplification of the conforming window function construction is to set the integration cells to be the triangles of the input triangulation and to perform the boundary integration of the cells using the trapezoid rule, as shown in Figure 4.14a. The set of evaluation locations then becomes all of the vertices in the input triangulation. For the $star^i(v)$ and snap star window functions, window values are set at all the vertices and thus the underlying spline functions are not required to setup and solve the problem.

For other integration cells or methods, the input triangulation can be refined for the purpose of constructing the window functions such that vertices exist at the required evaluation locations. For example, the Powell-Sabin split of a triangulation can be used to align the window function's nodal parameter locations to a nodal-based assumed strain cell boundary (SCNI cell),



(a) Triangle integration cell



(b) Nodal integration cell created from edges of a Powell-Sabin split.

Figure 4.14: Integration cell examples for assumed strain methods. Triangles and vertices in blue are present in the input triangulation. Red marks show the evaluation locations when the trapezoid rule is used on each edge of the cell boundary. Black dashed line show the new triangles that are created when refining the input triangulation.

see Figure 4.14b. The new vertices created when refining the triangulation would not be used for degrees-of-freedom and only would assist with defining the window functions.

4.7 Three Dimensional Conforming Window Functions

Most practical applications require three-dimensional simulations. The key concepts of the conforming window function transfer directly into three dimensions. The discussion of the previous sections largely just needs to be modified to refer to the three dimensional counterparts of the two dimensional geometries. That is, instead of triangles and triangulations there are tetrahedra and tetrahedral partitions, the definitions for stars becomes that of definition 8 instead of definition 2, metrics move from \mathbb{R}^2 to \mathbb{R}^3 and boundaries now are faces instead of edges.

There are several toolkits available to assist with generating three-dimensional triangulations, for example, the Cubit Geometry and Meshing Toolkit [61]. The spline function spaces mentioned in section 3.3 are used in place of the two dimensional Argyris space. The procedures for determining the nodal coefficients for the $star^i(v)$, snap and local stars will be the same but will require metrics and Hermite data for the additional dimension. That is, the window coefficients

are specified in accordance with the nodal minimum determining sets of, for example, the C^1 space on the Alfeld split $\mathcal{S}_5^1(\Delta_A)$ as in (3.42). The method for conforming by overriding distance information will also apply in three dimension. Convexity improvement will check line-of-sight using boundary faces instead of edges. Again, to avoid dealing with Bernstein-Bézier functions, the simplification process mentioned in the previous section can be used.

4.8 Additional comments

The Kronecker delta window function used earlier and shown in Figure 3.5a is a $star^2(v)$ window function where the vertices in $star^1(v)$ have adjusted nodal parameters, i.e. their values, first derivatives and the cross derivative is zero ($f_w^{00} = f_w^{10} = f_w^{01} = f_w^{11} = 0 \forall w \in star^1(v)$) and the second derivatives in x and y are positive ($f_w^{20} > 0, f_w^{02} > 0 \forall w \in star^1(v)$). This gives the window and thus approximation functions the Kronecker delta property and can also be made boundary conforming to give the weak Kronecker delta property.

When constructing the approximation functions near a collinear boundary, the conforming window functions can cause the moment matrix to become singular [40]. To avoid this issue, the moment matrix and approximation functions can be constructed in a reduced spatial dimension, ignoring the direction normal to the boundary. Other approaches are to make the function “nearly” conforming by setting the values along the boundary to a small (a few orders above machine precision) positive value instead of zero or to use the quasi-linear reproducing kernel approximation [62]. These methods will improve the conditioning of the moment matrix and allow inversion.

Material from this Chapter appears in *Computer Methods in Applied Mathematics and Engineering*, “Conforming window functions for meshfree methods”, 347:588 – 621, 2019, by Jacob Koester and J.S. Chen.

Chapter 5

An \bar{F} formulation for nearly incompressible materials

Intended applications for the framework created herein include large deformations of systems and components that may be made of metals, elastomers or other materials that exhibit nearly incompressible deformation during all or part of their response. Incompressibility adds an additional constraint on the approximation that has proven to be difficult to meet. Predictions exhibit volumetric locking, i.e. excessive force is required to cause a given displacement and convergence rates are degraded. The FEM community has been faced with this challenge for many decades and has developed several approaches for handling it.

Mixed Galerkin finite element formulations have been developed, using both pressure and displacement degrees of freedom. The methods can mitigate volumetric locking but should be designed to satisfy the so-called LBB condition, setting constraints on the order of approximation for pressure and displacement fields [63, 64, 65]. They can also be designed to include stabilization techniques, giving more flexibility in the choice of approximation spaces and allowing for equal-order interpolations [66, 67, 68]. The incompatible mode technique [69], the assumed strain method [64, 70, 71] and reduced integration with “hourglass” control [72, 73, 74] are other

methods for mitigating volumetric locking.

The $\bar{\mathbf{B}}$ method is another approach that is popular as it is solely displacement-based and does not require stabilization to mitigate hourglass modes [75]. It can be seen as a strain projection technique [75, 76] and it has been proven to be equivalent to mixed methods [77, 78, 79]. In addition to FEM, it has been successfully applied to IsoGeometric analysis (IGA) [80, 81].

Techniques for addressing volumetric locking for meshfree methods have borrowed from the FEM community. A mixed variational principle was used to develop a selective reduced integration procedure [82]. Concepts from the $\bar{\mathbf{B}}$ method were used in a pressure projection approach for Gauss integration performed on a background mesh [83, 84].

Arguably the largest advancement in integration for meshfree methods came with the Stabilized Conforming Nodal Integration (SCNI) method [26, 85]. A mixed variational principle based on the assumed strain method was used to create a displacement-based procedure that satisfies the linear integration constraint. By satisfying the integration constraint, it is able to pass linear patch tests and find the exact solution problems with linear displacement fields. Perhaps the biggest advantage is the computational efficiency. It was designed to have one material point per node whereas Gauss integration on background grids requires high order rules, leading to many more integration points per node, in order to get near ideal convergence rates. Reducing the number of material points is particularly beneficial when expensive constitutive models are used to describe the material response. The performance of SCNI for nearly incompressible materials is analogous to reduced integration finite elements. Using one point per node relaxes the constraints but at the expense of stability. Stabilization techniques are used in conjunction to mitigate the issue [5, 86].

Two main goals are set for the development of a method to address volumetric locking in conforming window function-based methods:

- Be compatible with simplification of the window construction / evaluation procedure

described in Section 4.6

- Avoid the need for stabilization

The strain smoothing technique described with SCNI provides a structure that works well with the simplification of window function construction and evaluation, but SCNI requires stabilization. $\bar{\mathbf{B}}$ methods, by construction, provide stabilization in deviatoric energy and mitigate volumetric locking by lower order approximation in volumetric energy. The strengths of both methods are combined to create an approach that works well with conforming window function methods and doesn't require extra stabilization. Before detailing the new approach, a review of $\bar{\mathbf{B}}$ and $\bar{\mathbf{F}}$ methods are provided below.

5.1 Review of the $\bar{\mathbf{B}}$ and $\bar{\mathbf{F}}$ methods

5.1.1 The $\bar{\mathbf{B}}$ method

The $\bar{\mathbf{B}}$ method, used in the small strain regime, leads to the $\bar{\mathbf{F}}$ method for finite deformations and the $\bar{\mathbf{F}}$ procedure is the foundation for the method developed herein. To begin we review the $\bar{\mathbf{B}}$ method, starting with the definition of the boundary value problem of elastostatics from Section 3.1.2, repeated here for clarity: Given domain Ω with boundary $\Gamma = \Gamma_h \cup \Gamma_g$, $\Gamma_h \cap \Gamma_g = \emptyset$ and

$$\nabla \cdot \boldsymbol{\sigma} = \mathbf{b} \quad \text{in } \Omega \quad (5.1)$$

$$\mathbf{n} \cdot \boldsymbol{\sigma} = \mathbf{h} \quad \text{on } \Gamma_h \quad (5.2)$$

$$\mathbf{u} = \mathbf{g} \quad \text{on } \Gamma_g \quad (5.3)$$

where \mathbf{u} is displacement, $\boldsymbol{\sigma}(\mathbf{u}) = \mathbf{C} : \boldsymbol{\varepsilon}(\mathbf{u})$ is the Cauchy stress tensor with \mathbf{C} being the fourth-order material tensor and $\boldsymbol{\varepsilon}(\mathbf{u}) = 1/2(\nabla \otimes \mathbf{u} + \mathbf{u} \otimes \nabla)$ is the strain tensor, \mathbf{b} is the body force, \mathbf{n} the boundary surface normal, \mathbf{h} the prescribed traction on Γ_h and \mathbf{g} the prescribed displacement on Γ_g .

The material tensor \mathbf{C} can be expressed in index notation in terms of the so-called Lamé parameters λ and μ .

$$C_{ijkl} = \lambda \delta_{ij} \delta_{kl} + \mu (\delta_{ik} \delta_{jl} + \delta_{il} \delta_{jk}) \quad (5.4)$$

where

$$\lambda = \frac{2\mu\nu}{(1-2\nu)} \quad (5.5)$$

$$\mu = \frac{E}{2(1+\nu)} \quad (5.6)$$

and ν is the Poisson's ratio and E is the Young's modulus. The stress in indicial notation is thus

$$\sigma_{ij} = C_{ijkl} \varepsilon_{kl} = \lambda u_{k,k} \delta_{ij} + 2\mu \varepsilon_{ij} \quad (5.7)$$

Problems occur for near incompressibility where $\nu \rightarrow 0.5$ and as a result $\lambda \rightarrow \infty$.

To address this complication, the $\bar{\mathbf{B}}$ method uses a purely displacement based formulation and requires the problem is not fully incompressible but rather **nearly** incompressible where λ / μ is large but not infinite. The strain tensor is split into its deviatoric and volumetric (dilatational) parts as:

$$\boldsymbol{\varepsilon}(\mathbf{u}) = \boldsymbol{\varepsilon}^{dev}(\mathbf{u}) + \boldsymbol{\varepsilon}^{dil}(\mathbf{u}) \quad (5.8)$$

where

$$\boldsymbol{\varepsilon}^{dil}(\mathbf{u}) = \frac{1}{3}(\nabla \cdot \mathbf{u})\mathbf{I} \quad (5.9)$$

where \mathbf{I} is the identity tensor.

To handle near incompressibility, the dilatational part of the strain tensor is replaced by a projected one, giving

$$\bar{\boldsymbol{\varepsilon}}(\mathbf{u}) = \bar{\boldsymbol{\varepsilon}}^{dil}(\mathbf{u}) + \boldsymbol{\varepsilon}^{dev}(\mathbf{u}), \quad (5.10)$$

with

$$\bar{\boldsymbol{\varepsilon}}^{dil}(\mathbf{u}) = \pi(\boldsymbol{\varepsilon}^{dil}(\mathbf{u})), \quad (5.11)$$

where π is a linear projection operator. This leads to

$$\bar{\boldsymbol{\sigma}}(\mathbf{u}) = \mathbf{C} : \bar{\boldsymbol{\varepsilon}}(\mathbf{u}) \quad (5.12)$$

where

$$\bar{\boldsymbol{\varepsilon}}(\mathbf{u}) = \bar{\mathbf{B}}\mathbf{d} \quad (5.13)$$

with \mathbf{d} being the coefficients and $\bar{\mathbf{B}}$ is derived by first splitting the strain-displacement matrix \mathbf{B} into deviatoric and volumetric parts

$$\mathbf{B} = \mathbf{B}^{dev} + \mathbf{B}^{dil} \quad (5.14)$$

and applying the projection operator giving

$$\bar{\mathbf{B}} = \mathbf{B}^{dev} + \bar{\mathbf{B}}^{dil} \quad (5.15)$$

The $\bar{\mathbf{B}}$ variational formulation then becomes: Find $\mathbf{u} \in L$, for all $\mathbf{w} \in V$, where

$$\int_{\Omega} \bar{\boldsymbol{\sigma}}(\mathbf{u}) : \bar{\boldsymbol{\varepsilon}}(\mathbf{w}) d\Omega = \int_{\Omega} \mathbf{w} \cdot \mathbf{b} d\Omega + \int_{\Gamma_h} \mathbf{w} \cdot \mathbf{h} d\Gamma \quad (5.16)$$

and $L = \{\mathbf{u} \mid \mathbf{u} \in H^1(\Omega), \mathbf{u}|_{\Gamma_g} = \mathbf{g}\}$, $V = \{\mathbf{w} \mid \mathbf{w} \in H^1(\Omega), \mathbf{w}|_{\Gamma_g} = 0\}$.

5.1.2 The $\bar{\mathbf{F}}$ method

In finite deformation theory the deformation is measured with the deformation gradient \mathbf{F} as opposed to the strain-displacement matrix \mathbf{B} that is used in small strain theory. We consider a

body occupying a region Ω_X with boundary Γ_X in the initial or reference configuration. The body deforms to its current configuration Ω_x with boundary Γ_x . The current position \mathbf{x} of a particle that was originally at location \mathbf{X} is given by

$$\mathbf{x} = \phi(\mathbf{X}, t) \quad (5.17)$$

where t is the current time. The displacement is then

$$\mathbf{u} = \phi(\mathbf{X}, t) - \mathbf{X} = \mathbf{x} - \mathbf{X} \quad (5.18)$$

The deformation gradient is defined as

$$\mathbf{F} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}} = \frac{\partial \mathbf{X} + \mathbf{u}}{\partial \mathbf{X}} = \frac{\partial \mathbf{X}}{\partial \mathbf{X}} + \frac{\partial \mathbf{u}}{\partial \mathbf{X}} = \mathbf{I} + \frac{\partial \mathbf{u}}{\partial \mathbf{X}} \quad (5.19)$$

A multiplicative split is used to decompose the deformation gradient

$$\mathbf{F} = \mathbf{F}^{dil} \mathbf{F}^{dev} \quad (5.20)$$

where \mathbf{F}^{dil} and \mathbf{F}^{dev} correspond to the dilatational and deviatoric components. Also,

$$\det \mathbf{F} = J = \det \mathbf{F}^{dil} \quad (5.21)$$

$$\det \mathbf{F}^{dev} = 1 \quad (5.22)$$

$$\mathbf{F}^{dev} = J^{-1/3} \mathbf{F} \quad (5.23)$$

$$\mathbf{F}^{dil} = J^{1/3} \mathbf{I} \quad (5.24)$$

Using the deformation gradient we also define the Cauchy-Green tensor

$$\mathbf{C} = \mathbf{F}^T \mathbf{F}. \quad (5.25)$$

and the Green-Lagrange strain tensor

$$\mathbf{E} = \frac{1}{2}(\mathbf{F}^T \mathbf{F} - \mathbf{I}). \quad (5.26)$$

For hyperelastic materials the constitutive response is often defined by relating the second Piola-Kirchhoff \mathbf{S} stress to a strain energy density Ψ

$$\mathbf{S} = 2 \frac{\partial \Psi(\mathbf{C})}{\partial \mathbf{C}} = \frac{\partial \Psi(\mathbf{E})}{\partial \mathbf{E}}, \quad (5.27)$$

The Cauchy stress tensor can be determine by the relation

$$\boldsymbol{\sigma} = \frac{1}{J} \mathbf{F} \mathbf{S} \mathbf{F}^T. \quad (5.28)$$

For elasto-plastic materials, the constitutive response is typically defined in rate form, relating an objective stress rate to its power conjugate strain rate measure.

We can now setup a boundary value problem. Given a body Ω_x in the current configuration with a body force $\mathbf{b}(\mathbf{x})$ on Ω_x , prescribed displacement $\mathbf{g}(\mathbf{x})$ on Γ_x^g and prescribed traction $\mathbf{h}(\mathbf{x})$ on Γ_x^h , find $\mathbf{u}(\mathbf{X}, t) \in \mathcal{L}$ such that

$$\nabla^x \cdot \boldsymbol{\sigma} + \mathbf{b} = 0 \text{ in } \Omega_x, \quad (5.29)$$

$$\mathbf{u} = \mathbf{g} \text{ on } \Gamma_x^g, \quad (5.30)$$

$$\boldsymbol{\sigma} \cdot \mathbf{n} = \mathbf{h} \text{ on } \Gamma_x^h, \quad (5.31)$$

where \mathbf{n} is the unit exterior normal in the current configuration on Γ_x^h , and \mathcal{L} is the space of admissible deformations defined by

$$\mathcal{L} = \{\mathbf{u} | \mathbf{u} \in H^1(\Omega_x) | \det(\nabla^x \mathbf{u}) > 0, \mathbf{u}|_{\Gamma_x^g} = \mathbf{g}\}. \quad (5.32)$$

Where the $\bar{\mathbf{B}}$ method replaced the volumetric component of \mathbf{B} with a projected value, the $\bar{\mathbf{F}}$ method replaces the volumetric component of \mathbf{F} with a projected value, giving

$$\bar{\mathbf{F}} = \bar{\mathbf{F}}^{dil} \mathbf{F}^{dev}, \quad (5.33)$$

where

$$\bar{\mathbf{F}}^{dil} = \pi(\mathbf{F}^{dil}) = \overline{J^{1/3}} \mathbf{I}, \quad (5.34)$$

Using $\bar{\mathbf{F}}$ we arrive at modified definitions of the Green-Lagrange strain tensor and Cauchy stress tensor as

$$\bar{\mathbf{E}} = \frac{1}{2}(\bar{\mathbf{F}}^T \bar{\mathbf{F}} - \mathbf{I}), \quad (5.35)$$

and

$$\bar{\boldsymbol{\sigma}} = \frac{1}{J} \overline{\mathbf{F} \mathbf{S} \mathbf{F}}^T. \quad (5.36)$$

respectively.

For hyperelastic materials, the modified second Piola-Kirchhoff stress is

$$\bar{S}_{ij} = \frac{\partial \Psi(\mathbf{E})}{\partial E_{ij}}(\bar{\mathbf{E}}(\mathbf{u})), \quad (5.37)$$

For elasto-plastic materials, $\bar{\mathbf{F}}$ is used in determining the modified objective stress rate and its

corresponding power conjugate strain rate measure. A detailed review of this process can be found in [80].

A projection of the gradient of the test functions is also needed

$$(\overline{\nabla^x \mathbf{w}})^{sym} = (\nabla^x \mathbf{w})^{sym} + \frac{1}{3} \left(\frac{\pi(J^{1/3} \nabla^x \cdot \mathbf{w})}{J^{1/3}} - \nabla^x \cdot \mathbf{w} \right) \mathbf{I} \quad (5.38)$$

$$= \boldsymbol{\varepsilon}(\mathbf{w}) + \frac{1}{3} (\text{tr}[\overline{\boldsymbol{\varepsilon}}(\mathbf{w})] - \text{tr}[\boldsymbol{\varepsilon}(\mathbf{w})]) \mathbf{I} \quad (5.39)$$

$$= \boldsymbol{\varepsilon}(\mathbf{w}) + \overline{\boldsymbol{\varepsilon}}^{dil}(\mathbf{w}) - \boldsymbol{\varepsilon}^{dil}(\mathbf{w}) \quad (5.40)$$

$$= \overline{\boldsymbol{\varepsilon}}(\mathbf{w}) \quad (5.41)$$

The variational form of the problem is then: Find $\mathbf{u} \in \mathcal{L}$, such that $\forall \mathbf{w} \in \mathcal{V}$

$$\int_{\Omega_x} \overline{\boldsymbol{\sigma}}(\mathbf{u}) : \overline{\boldsymbol{\varepsilon}}(\mathbf{w}) d\Omega, = \int_{\Omega_x} \mathbf{w} \cdot \mathbf{f} d\Omega, + \int_{\Gamma_x^h} \mathbf{w} \cdot \mathbf{h} d\Gamma \quad (5.42)$$

where $\mathcal{V} = \{\mathbf{w} \mid \mathbf{w} \in H^1(\Omega_x), \mathbf{w}|_{\Gamma_g} = 0\}$.

5.2 The $\overline{\mathbf{F}}$ method for conforming window methods

A $\overline{\mathbf{F}}$ -based method is designed to be compatible with the window simplification procedures described in Section 4.6. The main idea of the simplification is to construct the conforming window functions such that the vertices are coincident with the evaluation points used for domain and surface integration. This eliminates the need to invoke Bernstein-Bézier splines in the solution procedure, greatly simplifying the process. The smoothed gradient operator used in the SCNI method is particularly well suited to this approach. The *smoothed gradient* of a field f is defined [26, 85]:

$$\widetilde{\nabla} f(\Omega_L) = \frac{1}{V_L} \int_{\Omega_L} \nabla f d\Omega, \quad (5.43)$$

with the volume,

$$V_L = \int_{\Omega_L} d\Omega, \quad (5.44)$$

where Ω_L is the smoothing volume surrounding corresponding to a material point L . Applying the divergence theorem leads to

$$\widetilde{\nabla} f(\Omega_L) = \frac{1}{V_L} \int_{\Gamma_L} f \mathbf{n} d\Gamma, \quad (5.45)$$

where \mathbf{n} is the outward facing surface normal for the boundary Γ_L of the smoothing volume.

The smoothed gradient operator is applied to the deformation gradient \mathbf{F} around material point L to give [85]

$$\widetilde{F}_{ij}(\Omega_L) = \frac{1}{V_L} \int_{\Omega_L} \left(\frac{\partial u_i^h}{\partial X_j} + \delta_{ij} \right) d\Omega = \frac{1}{V_L} \int_{\Omega_L} \left(\frac{\partial u_i^h}{\partial X_j} \right) d\Omega + \delta_{ij} = \frac{1}{V_L} \int_{\Gamma_L} u_i^h n_j d\Gamma + \delta_{ij} \quad (5.46)$$

For the conforming window $\overline{\mathbf{F}}$ method two different domains are used for each material point, a smaller domain Ω_L^{dev} provides the deviatoric portion and a larger domain Ω_L^{dil} provides the dilatational portion. A depiction in two-dimensions is provided in Figure 5.1.

For $\overline{\mathbf{F}}$ at material point L , per equations 5.33 and 5.46, we have

$$\overline{\mathbf{F}}_L = \overline{\mathbf{F}}_L^{dil} \mathbf{F}_L^{dev}, \quad (5.47)$$

where

$$\mathbf{F}_L^{dev} = \widetilde{\mathbf{F}}(\Omega_L^{dev}), \quad (5.48)$$

$$\overline{\mathbf{F}}_L^{dil} = \overline{J^{1/3}}(\Omega_L^{dil}) \mathbf{I}, \quad (5.49)$$

and

$$\overline{J}(\Omega_L^{dil}) = \det \widetilde{\mathbf{F}}(\Omega_L^{dil}), \quad (5.50)$$

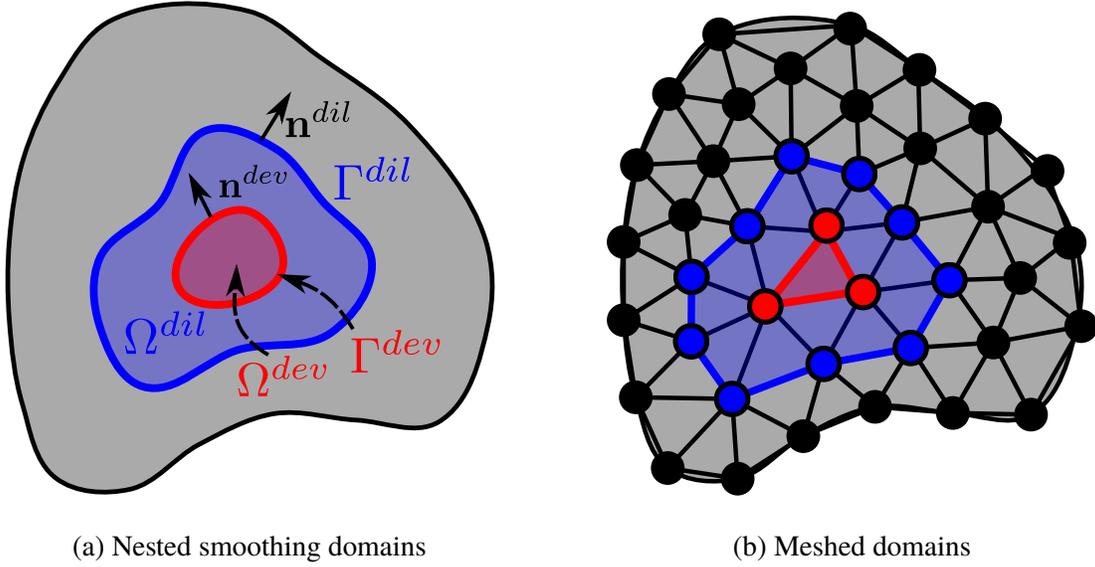


Figure 5.1: Dilatational and deviatoric smoothing domains

The dilatational and deviatoric domains can be constructed from a collection of cells C , as in Figure 5.1b, allowing for \bar{J} to be constructed simply as

$$\bar{J} = \frac{1}{V^{dil}} \sum_{c \in C} V_c J_c, \quad (5.51)$$

with,

$$V^{dil} = \sum_{c \in C} V_c \quad (5.52)$$

This allows J to be compute one time for every cell then aggregated as needed for \bar{J}

The numerical integration of Equation 5.46 is then designed to align with the methodology used in constructing the conforming kernels. There are many ways of accomplishing this. Perhaps the most straight-forward is to use the global triangulation to build the conforming kernels, use the triangles (or tetrahedra in 3D) for the smoothing cells with the trapezoid rule for the surface integral. Thus, the nodes of the triangulation are the vertices used to define the Bernstein-Bézier kernel functions and are thus set and known without actually constructing and evaluating the kernel function. Also, the nodes of the triangulation are the only place where the approximation

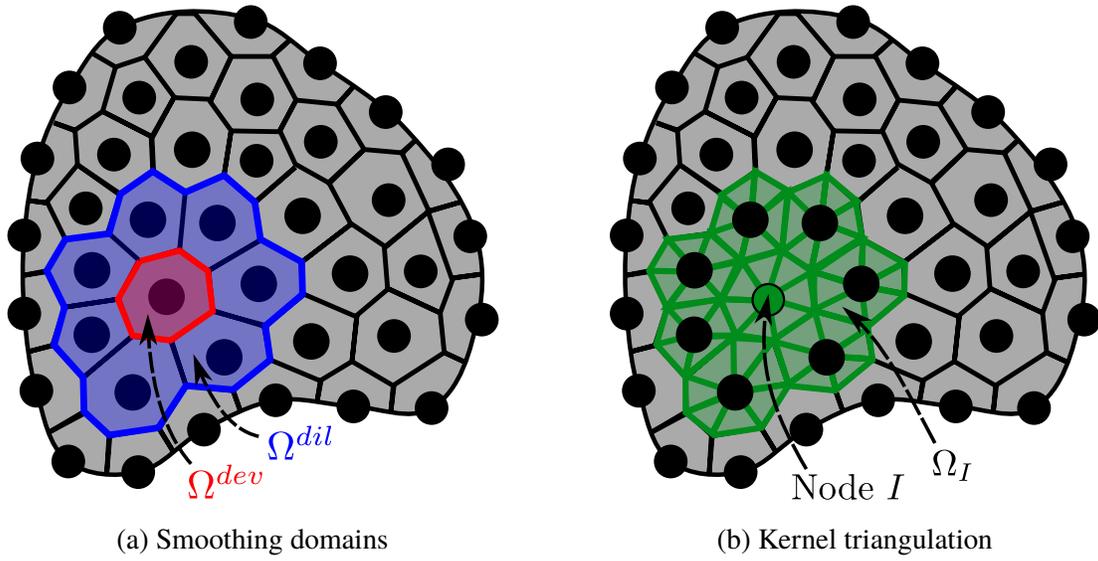


Figure 5.2: Example of node-based smoothing domains and corresponding kernel triangulation method that aligns integration evaluation points and kernel domain points, allowing efficient evaluations.

functions need to be evaluated to integrate Equation 5.46 using the trapezoid rule. Examples using this $\bar{\mathbf{F}}$ method are provided in Chapter 7.

An example of an alternative alignment between the kernel construction methodology and the integration of Equation 5.46 is to use a nodal-centric domain for integration which is triangulated to provide the subdivisions for kernel construction, as in Figure 5.2

Material from the Chapter are expected to appear in *Computational Particle Mechanics* as “ $\bar{\mathbf{B}}$ and $\bar{\mathbf{F}}$ approaches for the treatment of near-incompressibility in meshfree and immersed-particle methods” with authors Georgios Moutsanidis, Jacob Koester, Michael Tupek, Yuri Bazilevs and J.S. Chen.

Chapter 6

Addressing low quality discretizations

The process of generating subdivisions of complex geometries is often faced with many challenges that make obtaining a discretization of a desired quality very difficult. For example, small features in a design can lead to undesirably short edges and poorly shaped elements if the geometry is not “defeatured” before meshing. Also, there is growing interest in building simulations of “as-built” systems as well as micro, biological and geological structures. Boundary representation are often produced by scanning and processing with computed tomography and segmentation algorithms to generate faceted surfaces that are stored in formats such as STL. These faceted boundaries cause complications in in generating volumetric meshes as one either has to directly use the, potentially low quality, facets on the surface or choose to change the surface to be more agreeable with the set sizing and quality constraints.

Meshfree methods are an attractive option that can be used to avoid the discretization challenges for some applications. However, accurate boundary representations need to be incorporated into the approximation when important to the physics of the problem. Herein, a proof-of-concept is presented for efficiently generating quality simulation models when given low quality surface or volumetric representations of the domain.

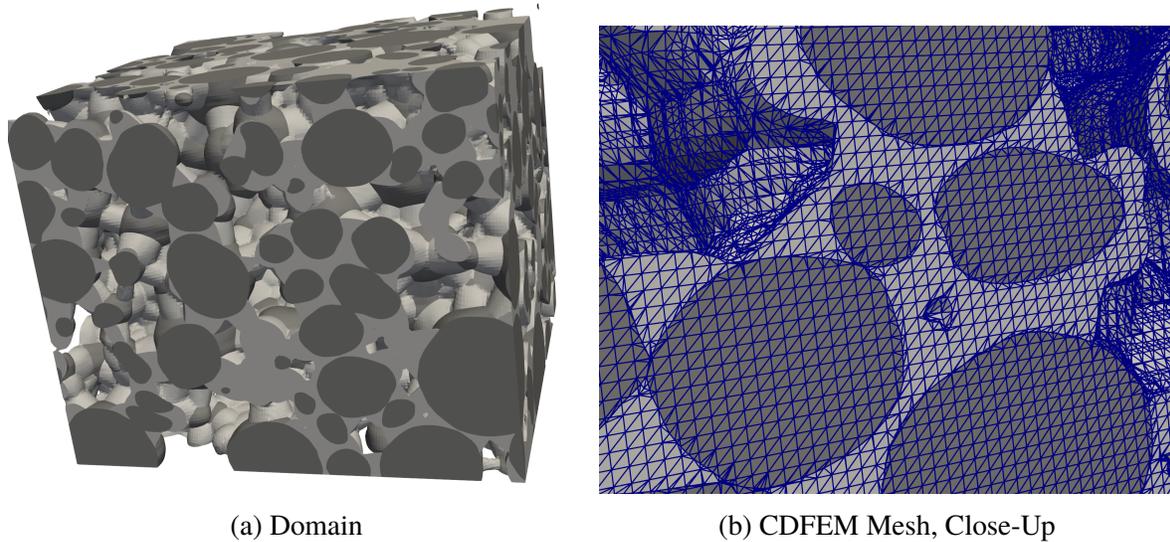


Figure 6.1: Lithium-ion battery electrode mesostructure. Example domain and CDFEM mesh.

6.1 Example low quality discretization and challenges

The Conformal Decomposition Finite Element Method (CDFEM) has been developed to generate simulations of complicated geometries where accurate representations of surfaces are critical to the solution quality [87]. Recently, CDFEM has been used to simulate the electrical conductivity and elastic response of lithium-ion batteries. An example of the battery mesostructure and CDFEM mesh are shown in Figure 6.1. The CDFEM mesh provides detailed interfaces between particles and the surrounding electrolyte. However, accurate interface representation comes at the expense of near-interface element quality. Analysts seek structural simulations that can provide insight into particle-electrolyte debonding and particle failure but the poor element quality limits analyses to very small deformations before problems occur.

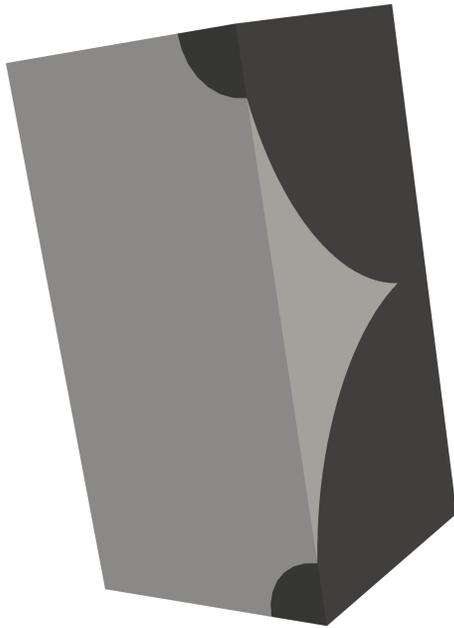
For a simplified look at these issues, an idealized, periodic, mesostructure is considered. The particles are considered to be spheres organized in a regular grid and fully embedded in the electrolyte [87]. An example interaction can be simplified using the periodicity and represented as two quarter spheres in compression as depicted in Figure 6.2a.

The dimensions of the domain are $(r \times r \times 2r)$ with $r = 2.5$. To avoid the complication

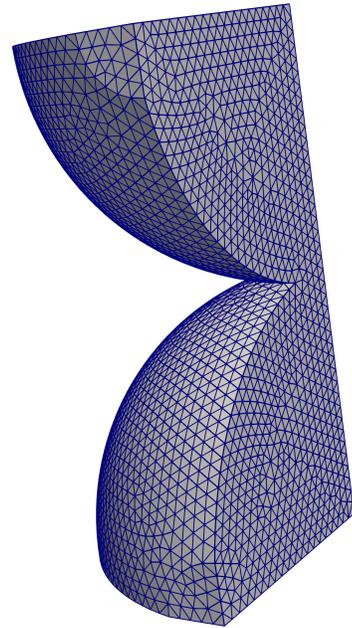
representing a single point of contact, the sphere geometries are created such that they overlap $0.05r$. The CDFEM method is used to generate a mesh, shown in Figure 6.2, with resulting element volumes ranging from $6E-4$ to $1E-12$. The simplified geometry allows a higher quality mesh to be generated where the element volume ranges from $5E-5$ to $7E-4$. The particles and electrolyte are represented as elastic with densities of $\rho = 5000$ and Poisson's ratios $\nu = 0.3$. The particles use a Young's modulus $E = 10E10$ whereas the electrolyte modulus $E = 10E3$. The vertical boundaries are fixed such that displacement can only occur in-plane. The top and bottom surface are fixed so no in-plane motion occurs and are displaced normal to the surface to cause compression.

Four simulation are considered, the higher quality and CDFEM meshes using both FEM and RKPM with interface and boundary conforming window functions. Each tetrahedron is used as an integration domain. The prescribed vertical displacement is smoothly ramped with a target final deformation of 0.1, giving an effective strain of 4% in the loading direction.

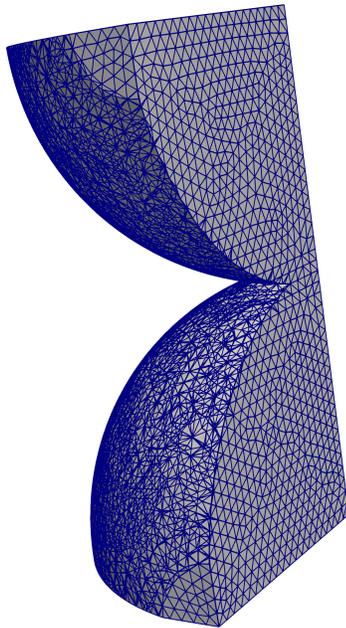
A summary of the stable time increments and maximum displacements achieved are in Table 6.1 and stress fields are in Figure 6.3. The stable time increment is estimated by using the power method to estimate the maximum eigenvalue of the system. Both predictions using the higher quality mesh were able to reach the full displacement without trouble and the prediction of the stress field is smooth. The FEM prediction using the CDFEM mesh failed due to element inversion at less than 10% of the intended displacement and had a stable time increment nearly a factor of 300 less than the FEM analysis on the quality mesh. The RKPM analysis also failed before obtaining the intended final displacement. The stable time increment was only reduced about 6x over the quality mesh as the less local shape functions provide a more favorable mass distribution and mitigate sharp derivatives in the approximation functions. The stress predictions on the CDFEM meshes display much more texture than is expected for the problem.



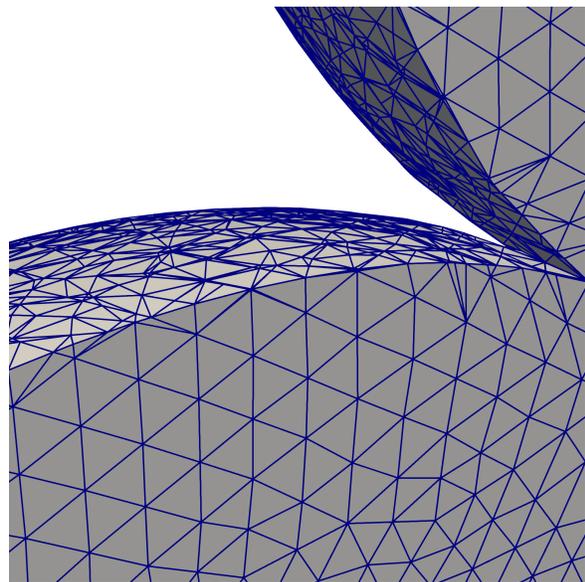
(a) Domain



(b) Spheres with Quality Mesh



(c) Spheres with CDFEM Mesh

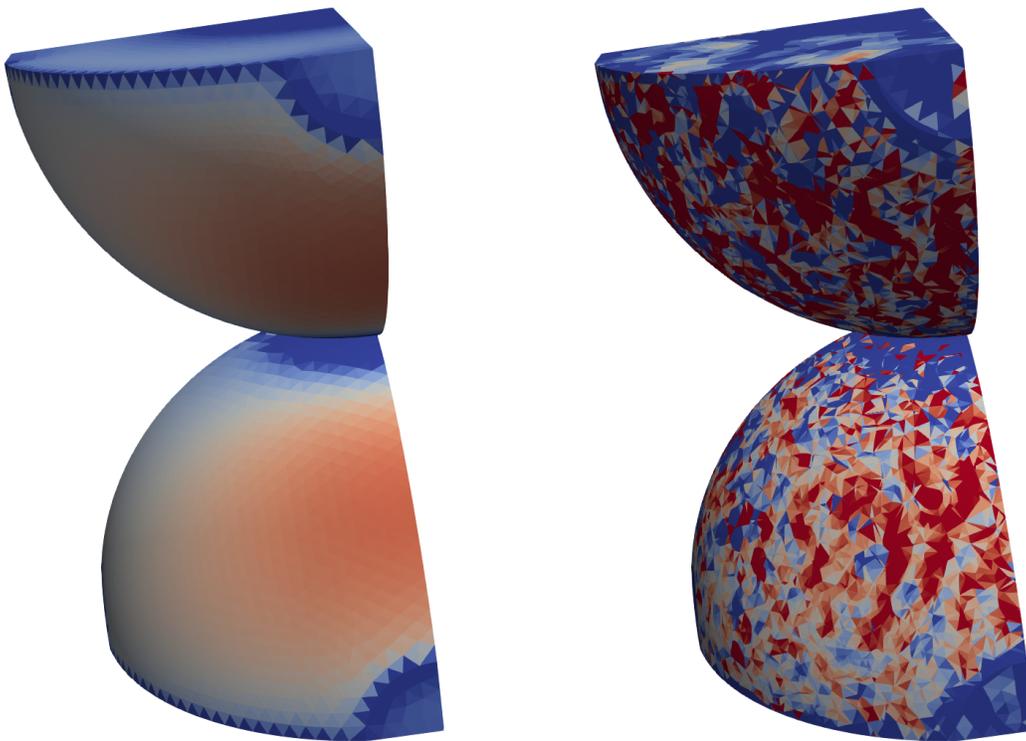


(d) Close-Up, CDFEM Mesh

Figure 6.2: Two sphere example problem. Example domain and meshes.

Table 6.1: Stable time increments and the displacement at which analysis failed for simulations of the two-spheres with electrolyte in compression.

Method	Initial Time Step (seconds)	Displacement at Method Failure
Quality mesh, FEM	2.6E-6	-
Quality mesh, RKPM	7.3E-6	-
CDFEM mesh, FEM	9.48E-9	0.009
CDFEM mesh, RKPM	1.27E-6	0.017



(a) FEM on Quality Mesh

(b) RKPM on CDFEM Mesh

Figure 6.3: Example predictions of σ_{xx} for the two sphere problem.

6.2 Improving the solution efficiency and accuracy by decimating and aggregating

Though a mesh is not required to construct the meshfree approximation function, they can be used to improve domain integration and, as demonstrated in previous sections, used to improve boundary-related challenges. Strictly using a low quality input mesh does, however, place undue constraints on method. Herein, a proof-of-concept is developed to show how a mesh can be used to generate the simulation but without adhering to the specific data structure as is required for FEM.

With FEM, every element represents a single integration cell. For meshfree methods, conforming integration cells are typically generated from an input mesh where, as with FEM, every element represents an integration cell or, as with SCNI, a dual of the mesh such as the Voronoi diagram or barycentric dual are used to construct the integration cells. These approaches create a connection between the quality of the input mesh and the quality of the solution. To reduce this connection, a new look at how to use a provided mesh is presented.

Clustering is one of the fundamental concepts used in machine learning. Centroid-based clustering seeks to group members of a set by proximity. The numbers of clusters sought is typically referred to as k and the process as *k-means clustering*. The clustering process partitions the space into a Voronoi diagram which is similar to the partition that a FEM mesh provides. The idea here is to use a k -means based algorithm to repartition a FEM mesh into integration cells that have more desirable quality, such as volume, than the elements of the input mesh. For example, the low volume elements on the boundaries of CDFEM meshes can be joined with neighboring elements to form a larger polyhedron.

Lloyd's algorithm [88] is the foundation used here to cluster elements into integration cells. In short, Lloyd's algorithm starts by initially selecting the k centroids of the clusters and iterating where each iteration starts by calculating the current centroids of the clusters then reassign items

to the nearest cluster based on these updated centroids. This approach, however, does not seek to balance how many items are in each cluster.

For generating integration cells, cell volume is chosen as a first metric of quality. Lloyd's algorithm is thus modified to consider cell /cluster volumes. An acceptable cluster volume range is set. While updating the cluster members, if a cluster is below the volume range it is not allowed any member, effectively removing the cluster. After the update and before the next iteration, any clusters that are above the volume range are split into two clusters. A outline of process is provided in Algorithm 1.

```

Data: Input Mesh, targetClustersVolumeRange, maxIterations
1 /* grid-based initialization */
2 targetClusterDimension = average(targetClusterVolumeRange)1/3;
3 determine bounding box of the domain;
4 set initial centroids spaced by targetClusterDimension and spanning the domain
   bounding box;
5 foreach element do
6   | join closest cluster;
7 end
8 /* iterations */
9 while not converged do
10  | foreach cluster do
11    | update clusterCentroid;
12    | update clusterVolume;
13    | if clusterVolume > targetClusterVolumeRange then
14      | split in two;
15    | end
16  | end
17  | foreach element do
18    | find closest cluster with clusterVolume > min(targetClusterVolumeRange);
19  | end
20  | if changes == 0 or iteration > maxIterations then
21    | converged = true;
22  | end
23 end

```

Algorithm 1: *k*-means based element clustering to generate integration cells

If the desired number of clusters is on the same order of the number of elements in the provided mesh then the mesh can be subdivided before being input into the algorithm.

The integration of a field f in cell C becomes the volume average value of the elements in the cell:

$$f = \frac{1}{V_{cell}} \sum_{c \in C} V_c f_c \quad (6.1)$$

where c are elements and f_c can be, for example, the smoothed gradient from 5.43.

Low element quality is often accompanied by poor nodal placement. Nodes that are too close can result in unfavorable nodal mass distributions and, if too many nodes are in close proximity, linear dependence in meshfree approximation functions. Since meshfree approximations are used, not all of the vertices in the mesh need to be used as nodes (carrying degrees of freedom) in the analysis. The cell clusters can be used to help decimate the set of nodes. An single vertex from each cluster can be selected and promoted to be a node in the analysis. The other vertices remain to provide the structure for domain integration but do not carry shape functions or coefficients. The algorithm for selecting nodes is outlined in Algorithm 2.

<p>Data: Input Clusters</p> <pre> 1 foreach <i>cluster</i> do 2 if <i>cluster is on one or more boundaries / interfaces</i> then 3 choose the vertex that is closest to the centroid of (and in) the boundary/interface vertices to be a node; 4 end 5 else 6 choose the vertex that is closest to cluster centroid to be a node; 7 end 8 end </pre>
--

Algorithm 2: Cluster-based selection of simulation nodes

Selecting one node per integration cell is another way of getting SCNI integration domains.

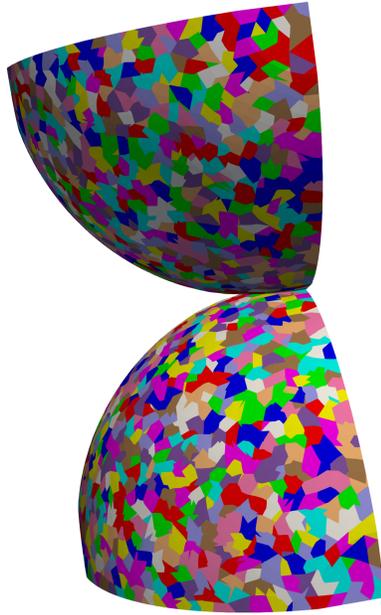


Figure 6.4: Cells obtained by aggregating the elements of the CDFEM mesh.

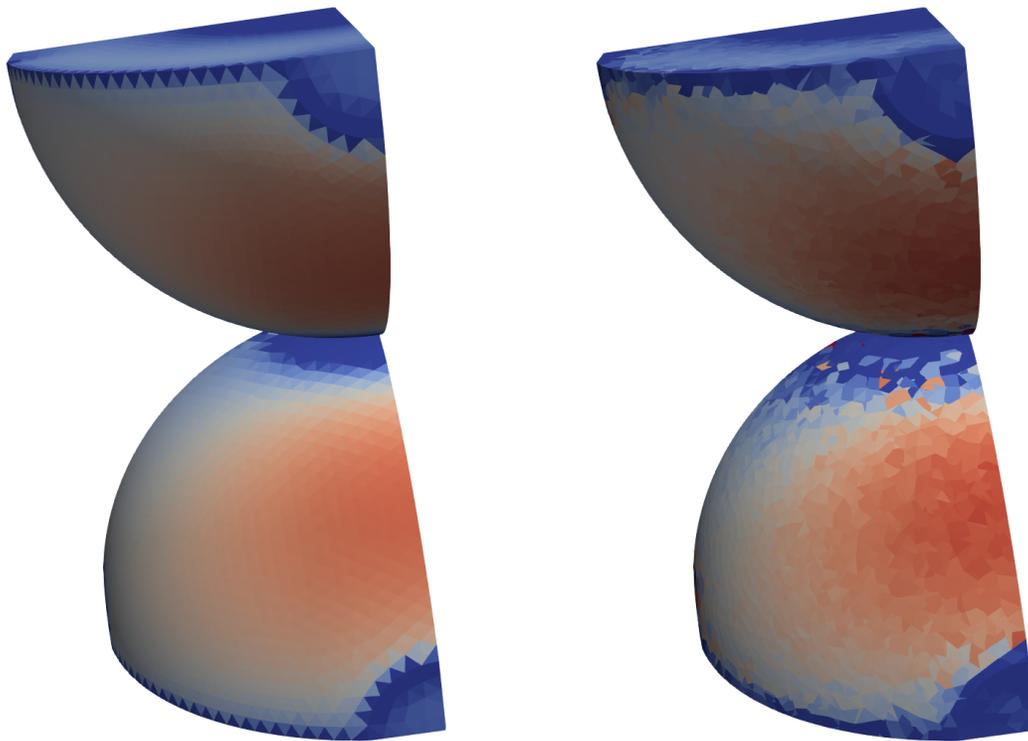
To improve stability and handling of incompressible materials, each cell can be split into subcells and used in the $\bar{\mathbf{F}}$ approach detailed in Chapter 5.

Application of the aggregation and decimation algorithms for the two-sphere example is provided in the following section with another example provided in Chapter 7.

6.2.1 Example of decimating and aggregating for a two-sphere problem

The aggregation and decimation procedure is applied to the two-sphere problem from Figure 6.2c. The input CDFEM mesh has 107,268 elements and 22,929 nodes which are aggregated to give 10,000 cells, each containing one node and two subcells. The resulting cells are shown in Figure 6.4.

The spheres are loaded as done previously. The aggregated and decimated mesh is able to obtain the full displacement without issues. The stable time increment is 7.6E-6, slightly higher than the value when using a quality mesh. The stress field prediction can be seen in Figure 6.5. The field has much less texture than the RKPM analysis performed directly on the CDFEM mesh.



(a) FEM, Quality Mesh

(b) RKPM, Aggregated / Decimated CDFEM Mesh

Figure 6.5: Example σ_{xx} predictions using an aggregated and decimated mesh for the two sphere problem.

Presumably, the stress predictions can be improved further with refinements to the aggregation logic.

Chapter 7

Applications of Conforming Window Functions

A number of examples are provided to demonstrate the performance of RKPM when conforming window functions are used. This subset of RKPM will be referred to as the *Conforming Reproducing Kernel*, or *CRK*, method. Three methods of constructing the conforming window functions, as described in the previous section, are tested throughout the examples. $star^i(v)$ refers to windows constructed using stars of the global triangulation and normalized graph distances. *Snap star* refers to windows constructed using stars of the global triangulation and the normalized Euclidean distance within a set radius with a value of 1 beyond the radius. *Local star* refers to windows constructed using triangulations individually created for each node, using only boundary information from the global triangulation.

Convergence in the normalized energy norm and normalized displacement (L^2) norm are compared in several of the examples. The normalized energy norm is

$$e_{energy} = \left[\frac{\int_{\Omega} (\boldsymbol{\varepsilon}^h - \boldsymbol{\varepsilon}) : (\boldsymbol{\sigma}^h - \boldsymbol{\sigma}) \partial\Omega}{\int_{\Omega} \boldsymbol{\varepsilon} : \boldsymbol{\sigma} \partial\Omega} \right]^{1/2} \quad (7.1)$$

where $\boldsymbol{\varepsilon}^h$ and $\boldsymbol{\sigma}^h$ are the approximated strain and stress and $\boldsymbol{\varepsilon}$ and $\boldsymbol{\sigma}$ are the true strain and stress.

The normalized L^2 norm is

$$e_u = \left[\frac{\int_{\Omega} (\mathbf{u}^h - \mathbf{u}) \cdot (\mathbf{u}^h - \mathbf{u}) \partial\Omega}{\int_{\Omega} \mathbf{u} \cdot \mathbf{u} \partial\Omega} \right]^{1/2} \quad (7.2)$$

where \mathbf{u}^h is the approximated displacement and \mathbf{u} is the true displacement.

The CRK predictions are compared with several other methods. In two dimensions, results are compared to traditional RKPM and FEM with linear triangles. Three dimensional examples include comparisons with RKPM using stabilized conforming nodal integration (SCNI) [26], the modified SCNI (MSCNI) method that includes additional stabilization terms [5], FEM using the selective-deviatoric hexahedral element [89], the composite tetrahedron [2, 3] and the uniform gradient hexahedral element with hourglass control [73]. Sierra/SolidMechanics [89] is used for all FEM results and the modified SCNI results.

7.1 Two dimensional problem

7.1.1 Linear elastic patch test

A linear patch test for elasticity is completed to demonstrate the admissibility of the conforming approximation functions. A square, (1 x 1), domain is used with Young's modulus $E = 10^5$ and Poisson's ratio $\nu = 0.3$. A linear solution is constructed with

$$\mathbf{u} = \begin{Bmatrix} 0.1x + 0.3y \\ 0.2x + 0.4y \end{Bmatrix}. \quad (7.3)$$

In order to pass a linear patch test 1) the approximation space must be complete to at least first order 2) integration must be variationally consistent and 3) the boundary enforcement method must be variationally consistent. To satisfy 1, the RK approximation is constructed with constant

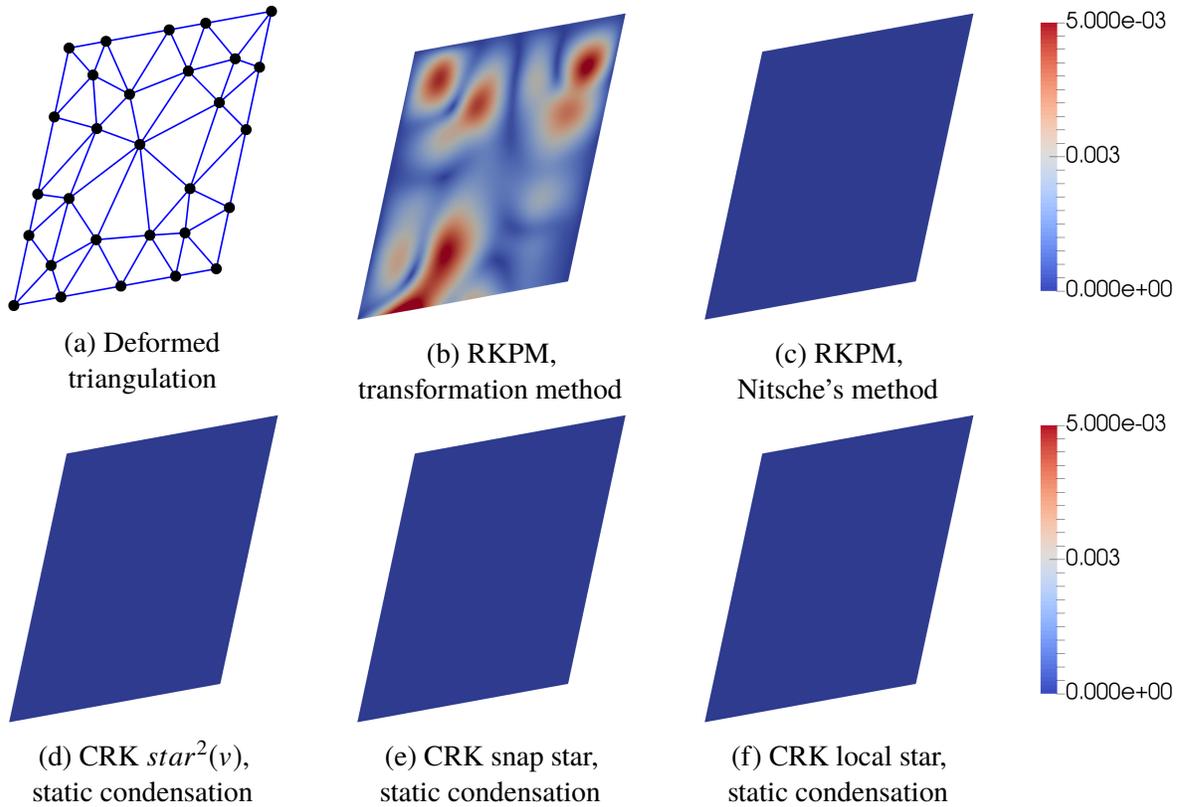


Figure 7.1: Deformed geometry and magnitude of error in displacement for a linear elastic patch test

and linear bases. To satisfy 2, stabilized conforming nodal integration (SCNI) is used [26]. The consistency of different boundary enforcement methods is checked by prescribing the exact displacement on the top, left and right boundaries and traction on the bottom boundary. Figure 7.1 displays the deformed domain with the discretization and Delaunay triangulation and the results using RKPM with the transformation method [13] which is a collocation method, RKPM with Nitsche's method [57] using $\beta = 100E/h$ (h is the local discretization spacing) and CRK with $star^2(v)$, snap and local star boundary conforming windows and direct imposition of boundary conditions at the nodes. Integrated error values are also shown in Table 7.1.

The collocation method does not provide admissible function spaces and specifying boundary values at the nodes alone is insufficient to prescribe the response along the entire boundary. This results in error which is several orders of magnitude above machine precision.

Table 7.1: Error norms for a linear elastic patch test

Method	L^2	H_1
RKPM with transformation method	2.05E-03	2.44E-02
RKPM with Nitsche's method	3.85E-16	4.93E-15
CRK, $star^2(v)$, with static condensation	7.65E-17	1.04E-15
CRK, snap star, with static condensation	8.57E-17	1.78E-15
CRK, local star, with static condensation	6.55E-17	8.63E-15

Nitsche's method has a weak form that is variationally consistent and thus passes the patch test without the need of modifying the test and trial functions. The three different conforming kernel function types provide admissible function spaces and thus are able to pass the patch test. The CRK approximations are interpolatory along the boundary, i.e. $u^h(\mathbf{x}_I) = d_I$ for all $\mathbf{x}_I \in \mathcal{V}_b$ with \mathcal{V}_b being the set of nodes on the boundary. This property is due to the combination of 1) the weak Kronecker delta property, 2) the construction of the approximation functions by enforcing the reproducing conditions in (3.6) and since 3) the boundary condition is in the approximation space. As a result, values at nodes can be specified directly as in FEM.

A variationally consistent integration (VCI) technique that doesn't require a conforming subdivision of the domain has been developed in [27]. Using VCI with the local conforming kernel would allow the patch test to be passed using only a point cloud and boundary edges.

7.1.2 Non-convex geometries

Additively manufactured notched tensile specimen

An example application space where surface geometry can complicate meshfree analyses is additive manufacturing. A notched tensile specimen, manufactured using Direct Metal Laser Sintering (DMLS) for the 2017 Sandia Fracture Challenge [90], is used as an example of an additively manufactured part. An image of the specimen is processed to extract the boundary and the near-notch region is triangulated as shown in Figure 7.2. Plane stress analyses are completed using FEM, RKPM and CRK, all using the same coarse discretization containing 311 nodes.

RKPM is tested both with and without the visibility condition. For CRK, the $star^2(v)$, snap and local stars are used. The $star^2(v)$ and snap stars are created using the convexity improvement approach from section 4.5 and local stars are created to be star convex. A reference solution is created using FEM on a refined mesh containing 272,801 nodes.

The results for σ_{yy} are shown in Figures 7.3 and 7.4. The line used for plotting in Figure 7.4 is depicted in Figure 7.3a. FEM predicts the peak stress to be in the expected area but, at this resolution, the stress profile near the tip is not matched very well. The results for RKPM without the visibility criteria shows the effect of the approximation functions spanning the notch. The stress concentration at the tip is much lower than other locations near the notch, inconsistent with the mechanics of the problem. RKPM with the visibility criteria does show the peak to be in a more agreeable location but several discontinuities can be seen. The discontinuities are affecting the near tip solution, causing a low peak stress prediction. The three CRK methods show similar results. The shape functions do not span the gap, preventing the non-physical behavior observed in the RKPM prediction. Also, the three methods provide reasonable peak stress locations and profiles considering the coarse discretization that is used. The $star^2(v)$ kernel does provide approximation functions with sharper gradients, leading to a solution with slightly higher peaks.

Simulations using refined discretizations are also completed for FEM, RKPM without the visibility condition and CRK using the snap star. A close-up of σ_{yy} near the crack tip is shown in Figure 7.5 with the plots along the center line in Figure 7.6. For RKPM, the effect the approximation functions spanning gaps is greatly reduced but local effects are still present. The displacement at the tip is blunted and the peak stress shifted a small distance. These discrepancies can become problematic for applications involving behaviors such as plastic yielding or fracture. The CRK and FEM results have similar predictions. The CRK results provide a smoother stress profile using an order of magnitude less degrees-of-freedom than FEM and with less than half of the total integration points (542,720 elements for FEM, 237,440 integration points for CRK and RKPM).

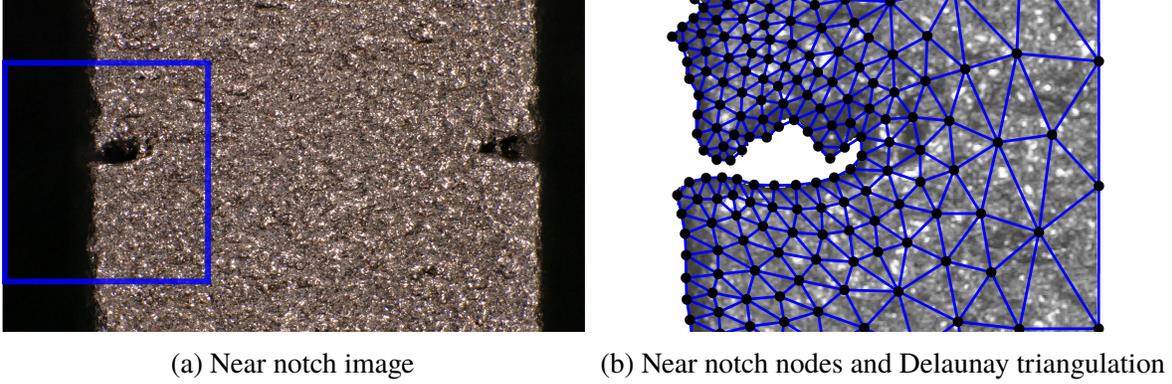


Figure 7.2: Additively manufactured notched bar. The region in the blue box is used to create a two dimensional model. The triangulation is created after using image processing to extract the specimen boundaries.

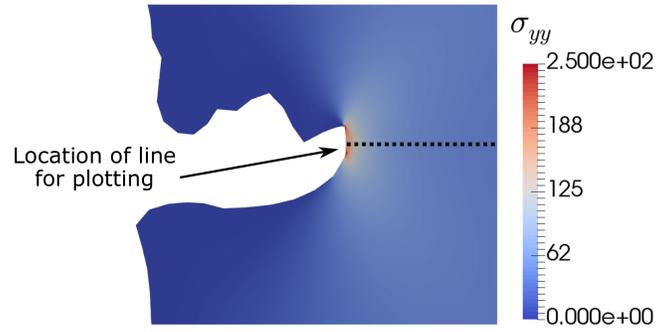
With simple geometries, the truncation of window functions using the visibility criteria is more straight-forward than constructing conforming window functions; however, with complex boundaries, similar geometry querying operations are required. As a result, the visibility methods and CRK methods have similar complexity. The process for creating conforming window functions is conjectured to be more robust and will provide approximation functions of controlled continuity. The discontinuities present for RKPM with the visibility criteria can cause local domain inversion, creating problems for integration or with the deformation gradient. Also, by always including at least one star of elements, CRK guarantees approximation functions can be created.

Plate with an edge crack

A plate with an edge crack, as in Figure 7.7a, is considered to study the convergence of problems with non-convex domains. An analytical solution for this problem is provided in [91]. For mode I loading

$$u_x(x, y) = \frac{1}{2G} r^\lambda [(\kappa - Q(\lambda + 1)) \cos(\lambda\theta) - \lambda \cos((\lambda - 2)\theta)] \quad (7.4)$$

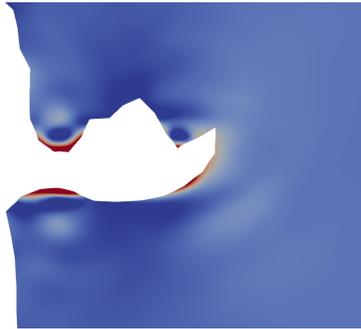
$$u_y(x, y) = \frac{1}{2G} r^\lambda [(\kappa + Q(\lambda + 1)) \sin(\lambda\theta) + \lambda \sin((\lambda - 2)\theta)] \quad (7.5)$$



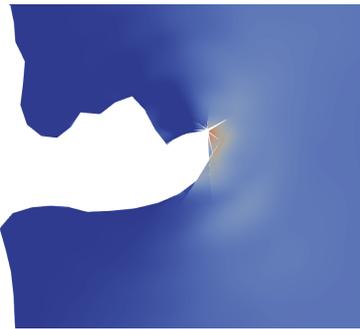
(a) Reference



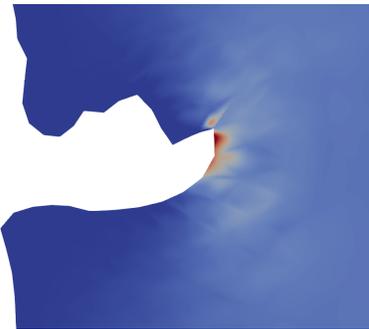
(b) FEM



(c) RKPM



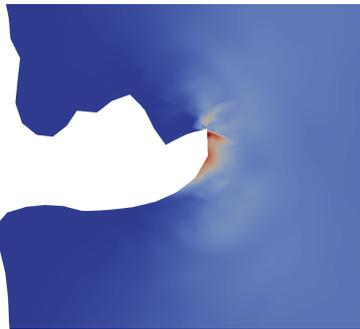
(d) RKPM, visibility



(e) CRK $star^2(v)$



(f) CRK Snap Star



(g) CRK Local Star

Figure 7.3: σ_{yy} results for the additively manufactured notched bar loaded in tension.

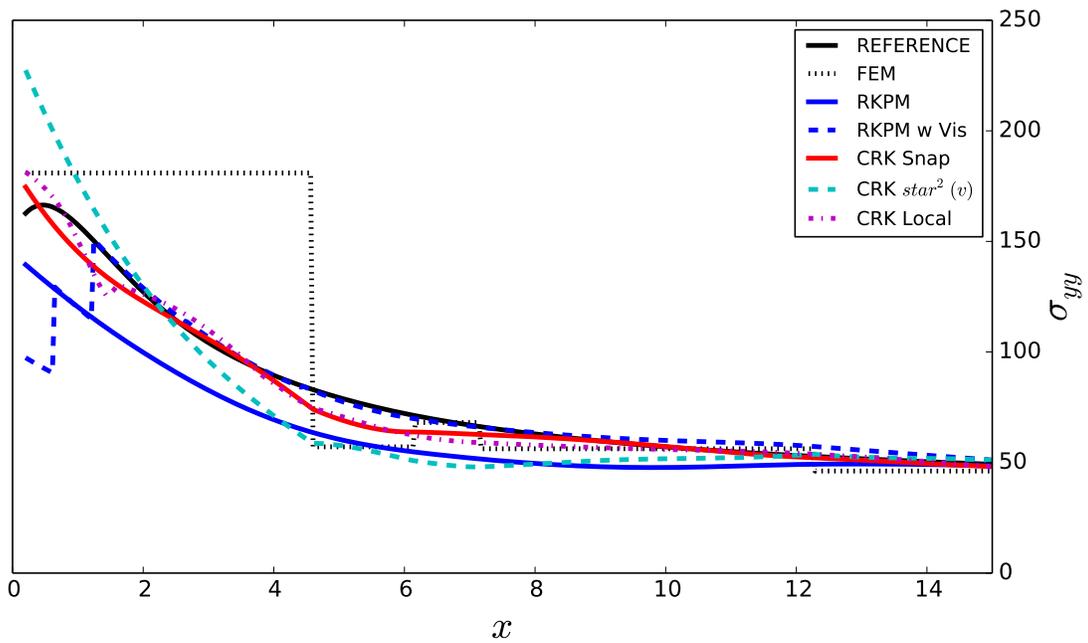


Figure 7.4: Comparison of σ_{yy} results along the a horizontal line starting at the middle of the crack tip for the notched tensile specimen.

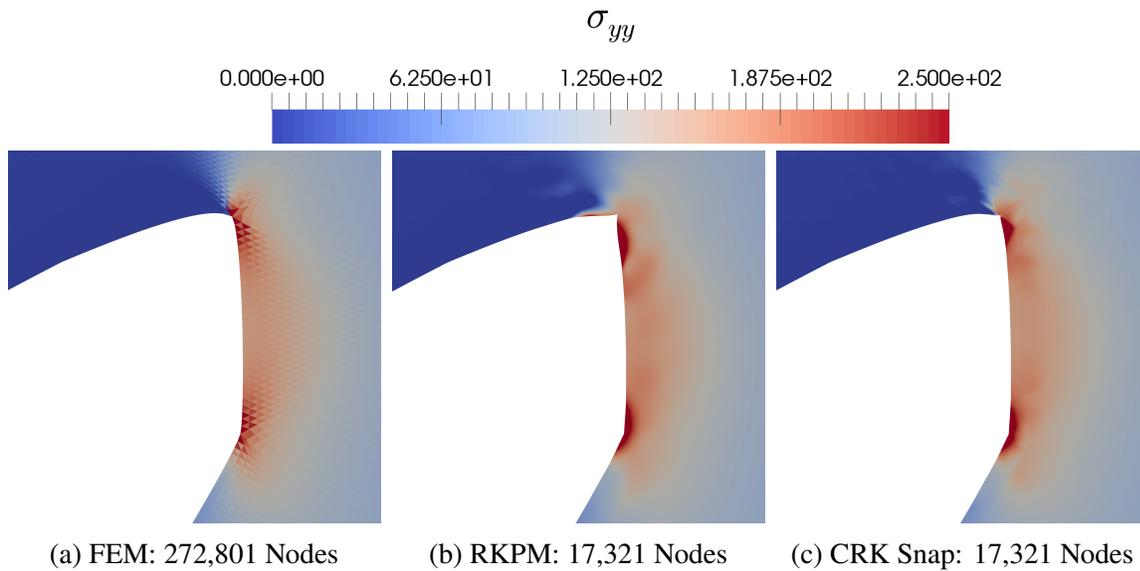


Figure 7.5: Close-up view of refined σ_{yy} results near the notch tip for the additively manufactured notched bar loaded in tension.

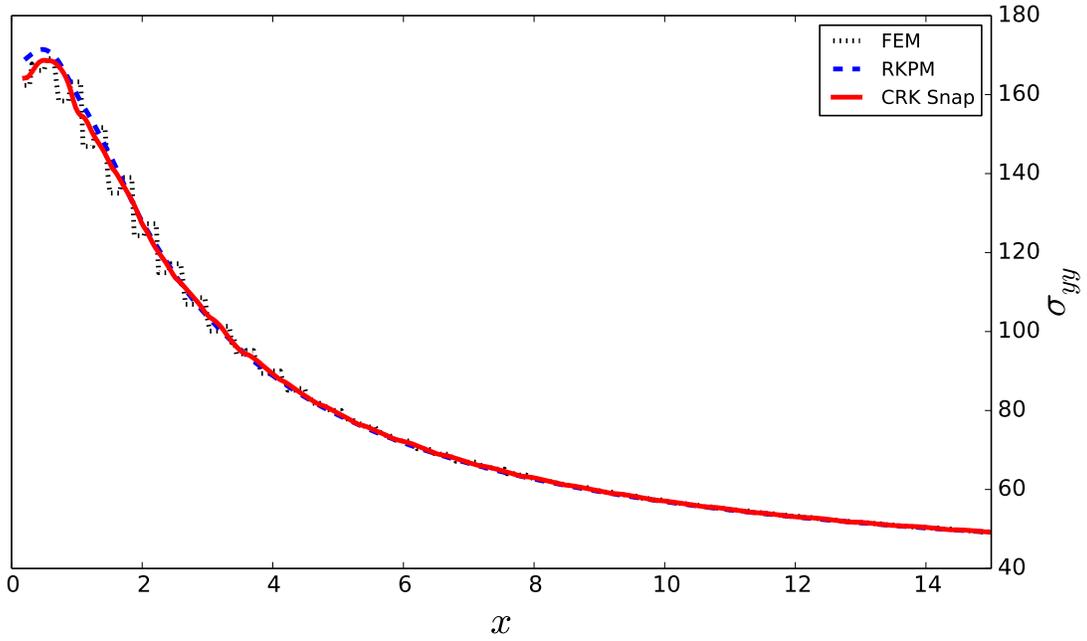


Figure 7.6: Refined predictions of σ_{yy} results along the a horizontal line starting at the middle of the crack tip for the notched tensile specimen.

and

$$\sigma_{xx} = \lambda r^{\lambda-1} [(2 - Q(\lambda + 1)) \cos((\lambda - 1)\theta) - (\lambda - 1) \cos((\lambda - 3)\theta)] \quad (7.6)$$

$$\sigma_{yy} = \lambda r^{\lambda-1} [(2 + Q(\lambda + 1)) \cos((\lambda - 1)\theta) + (\lambda - 1) \cos((\lambda - 3)\theta)] \quad (7.7)$$

$$\sigma_{xy} = \lambda r^{\lambda-1} [(\lambda - 1) \sin((\lambda - 3)\theta) + Q(\lambda + 1) \sin((\lambda - 1)\theta)] \quad (7.8)$$

where $G = E/(2(1 + \nu))$ is the shear modulus, λ and Q are determined so the re-entrant edges are traction free and $\kappa = 3 - 4\nu$ for plane strain. A 360° crack is modeled and thus $\lambda = 1/2$ and $Q = 1/3$. Other chosen values are $E = 30 \cdot 10^6$, $\nu = 0.3$ and the dimension of the plate (1 x 1). The exact displacement is specified on all boundaries except the two re-entrant edges.

An example coarse mesh can be seen in Figure 7.7b. A circular region with consistent nodal spacing, h_1 , is created around the crack tip. The nodal spacing increases from the edge of

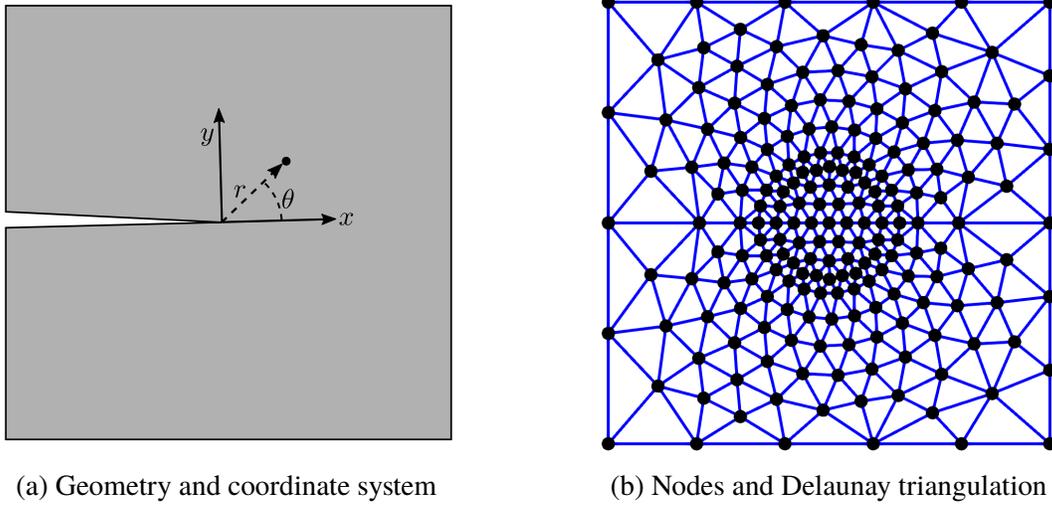


Figure 7.7: Plate with an edge crack.

the circle to the boundary of the domain with the final spacing being approximately $5h_1$. The radius of the circle is chosen to be $4h_1$.

RKPM is modeled using tensor product kernels with a support of $2.5h$ and with and without visibility considerations. Again the three CRK window types are evaluated: $star^2(v)$, snap stars with support of $2.1h$ and local stars with support of $2.5h$. Both the $star^2(v)$ and snap stars are created using the convexity improvement approach and local stars are created to be star convex. For consistency, boundaries are enforced using Nitsche's method with $\beta = 100E/h$. Domain integration is accomplished using Gaussian quadrature with 16 points per triangle, which can exactly integrate polynomials up to eighth order. For FEM, constant strain triangles are used so 1 point per triangle is sufficient for domain integration and essential boundaries conditions are enforced by directly specifying nodal values.

Convergence plots for the different scenarios are shown in Figure 7.8. The error in the prediction of σ_{yy} near the crack tip for a coarse mesh is also shown in Figure 7.9. The issue with RKPM without a visibility condition is clear as no convergence occurs and the final displacement is as if a crack is not present. Due to the lower regularity of the exact solution, the ideal convergence rates for polynomial bases are 1 and 0.5 for e_u and e_{energy} . The ideal rates are achieved for all

CRK window types and for RKPM with the visibility condition. The CRK results give lower error than RKPM with the visibility criteria. The near tip plots in Figure 7.9 show the discontinuities that are present with RKPM and the visibility method. On the other hand, the solutions of CRK are continuous. To further improve the solutions, the enrichment procedure as used in [92] could be implemented for any of the methods.

7.1.3 Plate with an inclusion

The next example is an infinite elastic plate with an elastic circular inclusion. The plate is loaded in tension in the x direction. The analytic solution for the problem can be found in [93]. In the simulation, the problem symmetry is taken advantage of and a (4×4) region is considered with an inclusion of radius $R = 1$. An example of the domain and a coarse discretization are in Figure 7.10. The exact traction is applied on the top and right side corresponding to a far field traction of 10 and the exact displacement is enforced on the symmetry planes. The inclusion is considered to be welded to the surrounding material such that the displacement field is continuous. Both materials have a Poisson's ratio $\nu = 0.3$. The plate has Young's modulus $E = 10^3$ and the inclusion is an order of magnitude stiffer with $E = 10^4$.

The example is simulated using FEM, RKPM with tensor product window functions with $R = 2.0h$ and CRK using $star^2(\nu)$ stars and snap and local stars also using $R = 2.0h$. The CRK windows all use the sharp conforming procedure for the material interface as described in section 4.4. Essential boundaries are enforced using Nitsche's method with $\beta = 100E/h$ for RKPM. For CRK, the windows are also made to conform to the essential boundaries and direct enforcement with static condensation is used. The exact solution is not in the approximation space so the nodal values are determined by performing a least squares fit at the nodes. The assumed strain integration technique, like SCNI [26], is used but using the triangles from the global triangulation as conforming cells instead of node-based cells. For the $star^2(\nu)$ and snap stars, which are constructed using the global triangulation, this integration is compatible with the

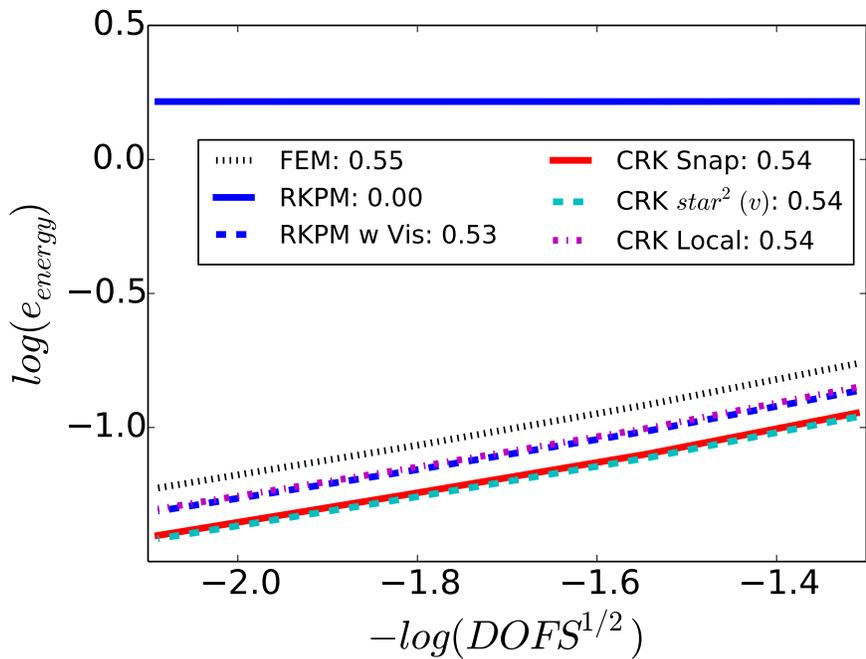
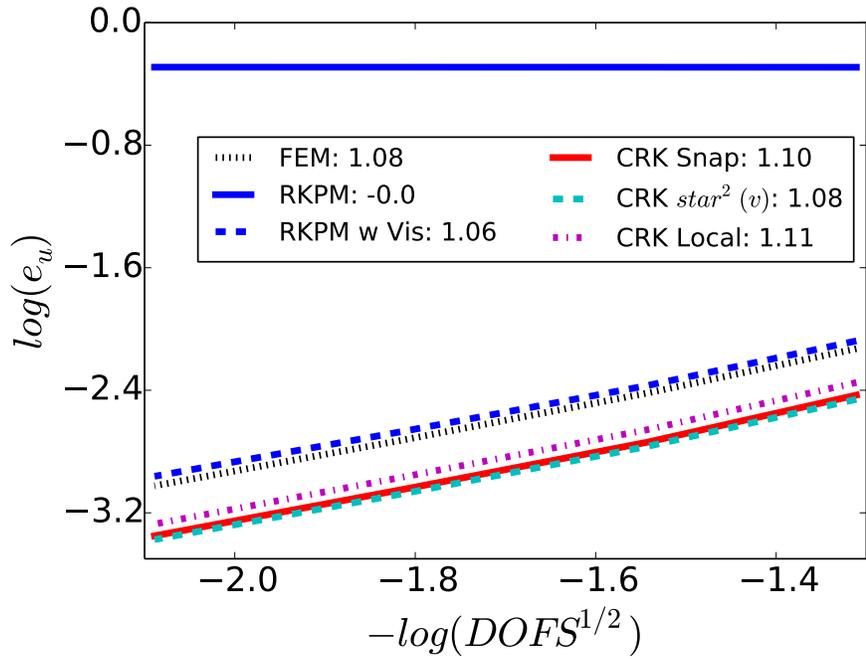


Figure 7.8: Plate with edge crack convergence

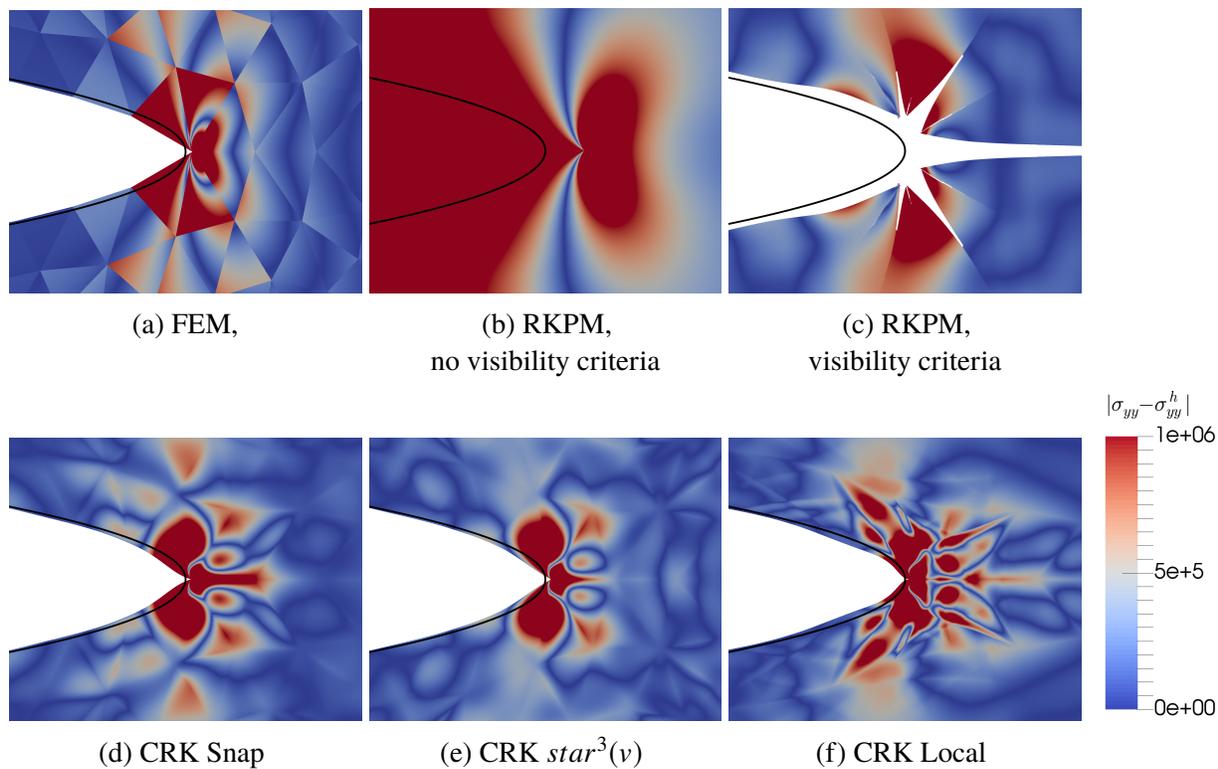


Figure 7.9: Error in σ_{yy} near the crack tip for the edge crack problem with a near-tip nodal spacing $h_1 = 0.02$. The black solid line represents the location of the edge for the exact solution.

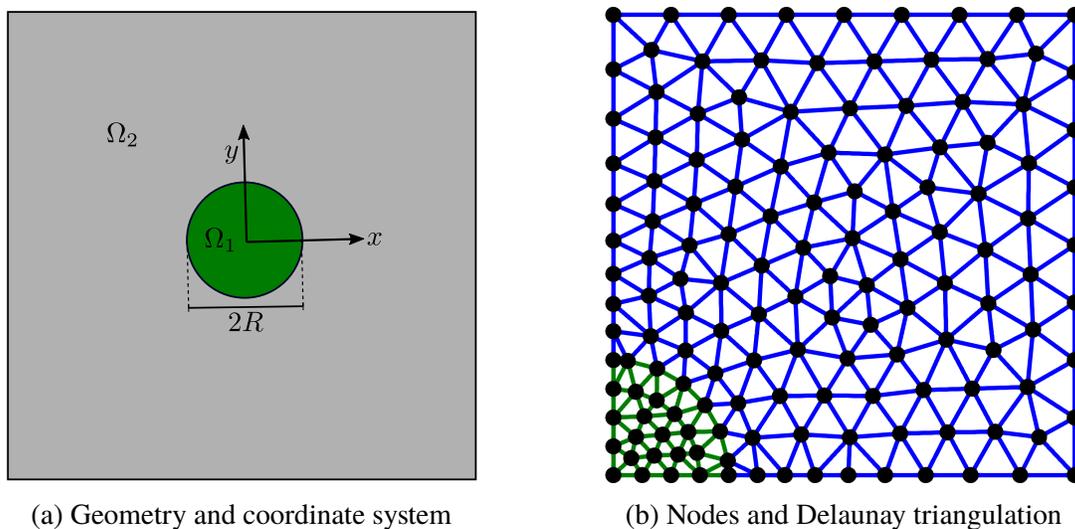


Figure 7.10: Plate with an inclusion simulated by taking advantage of the symmetry.

simplification procedure described in section 4.6, avoiding the need to construct the spline-based window functions as all evaluation locations used in the solution are at the window functions nodal parameter locations. Construction of the full window functions is completed here, however, to allow for error integration and refined visualization.

Convergence plots are provided in Figure 7.11 and ε_{xy} along a diagonal line for a coarse discretization in Figure 7.12. The inability of RKPM to capture the weak discontinuity at the material interface results in sub-optimal convergence and a Gibbs-like phenomenon in the strain field. FEM and all CRK designs have agreeable convergence rates with CRK often giving less error at a given resolution due to the higher smoothness elsewhere in the domain.

7.1.4 Beam with poorly-shaped triangles

Consider a plane stress cantilever beam with a parabolic traction on the free end as shown in Figure 7.13a. The exact solution is as provided in [94]

$$u_x(x, y) = -\frac{Py}{6EI} \left[(6L - 3x)x + (2 + \nu) \left(y^2 - \frac{D^2}{4} \right) \right] \quad (7.9)$$

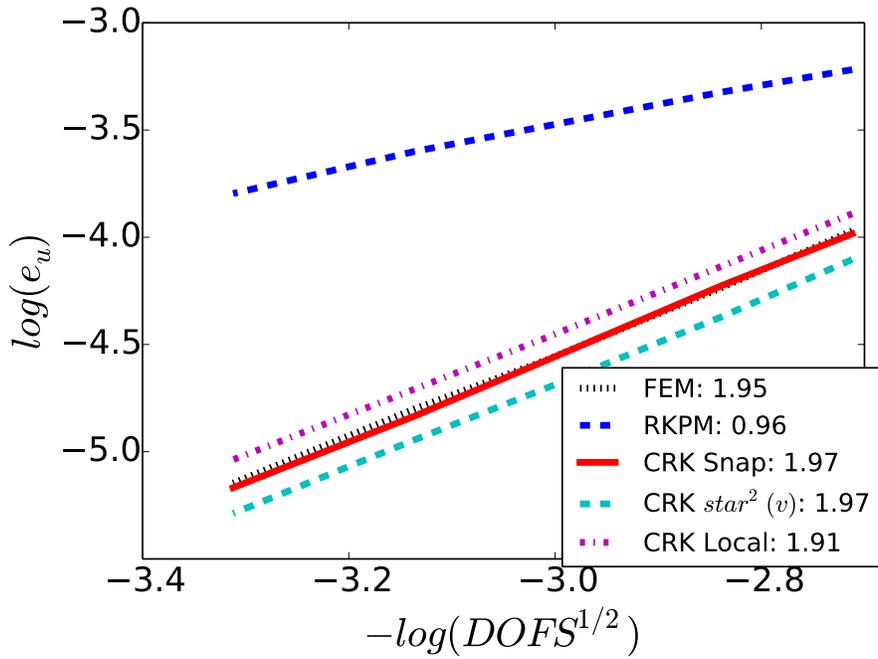
$$u_y(x, y) = \frac{P}{6EI} \left[3\nu y^2(L - x) + (4 + 5\nu) \frac{D^2 x}{4} + (3L - x)x^2 \right] \quad (7.10)$$

$$\sigma_x(x, y) = -\frac{P(L - x)y}{I}, \quad \sigma_{xy}(x, y) = -\frac{P}{2I} \left(\frac{D^2}{4} - y^2 \right), \quad \sigma_y(x, y) = 0 \quad (7.11)$$

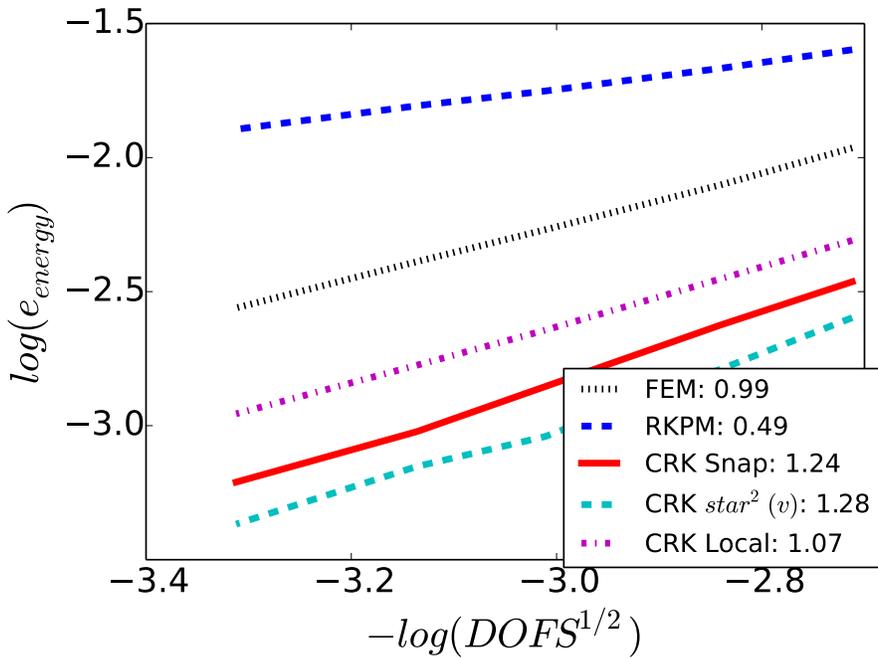
where L is the length, D is the height, $I = D^3/12$ is the moment of inertia and P is the applied shear force. The chosen values are $L = 48$, $D = 12$, $E = 30 \cdot 10^6$ and $\nu = 0.3$.

Low quality triangulations are created to test the performance of the methods when poorly shaped triangles are present. A coarse discretization, as shown in Figure 7.13b, is generated by randomly moving the vertices of an initial Delaunay triangulation. Refined meshes are generated by uniformly subdividing the triangles of the coarse triangulation.

The CRK approximation functions are constructed to be complete to first order. Snap



(a) Convergence in displacement



(b) Convergence in energy

Figure 7.11: Convergence plots for the plate with an inclusion example simulated with FEM, RKPM using non-conforming tensor product window functions with $R = 2.0h$ and CRK with $R_g = star^2(v)$ and snap and local stars with $R = 2.0h$ that conform to the material interface.

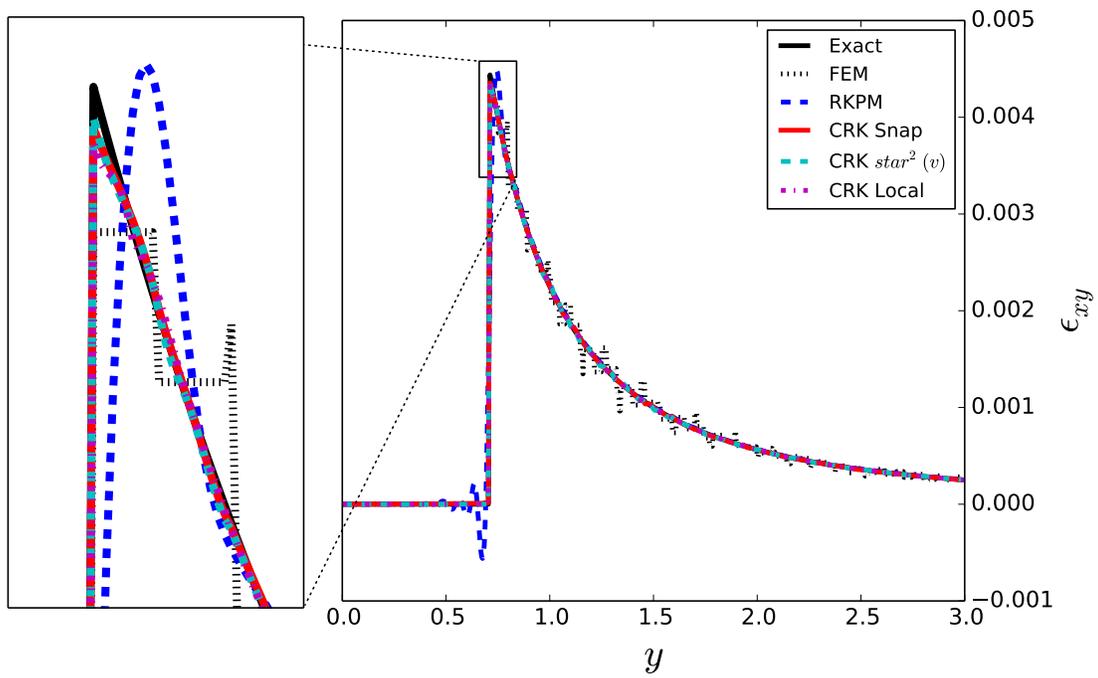


Figure 7.12: Comparison of ε_{xy} results along the a diagonal line between the origin and upper right corner of the domain for the inclusion problem with a near-inclusion nodal spacing $h = 0.0625$.

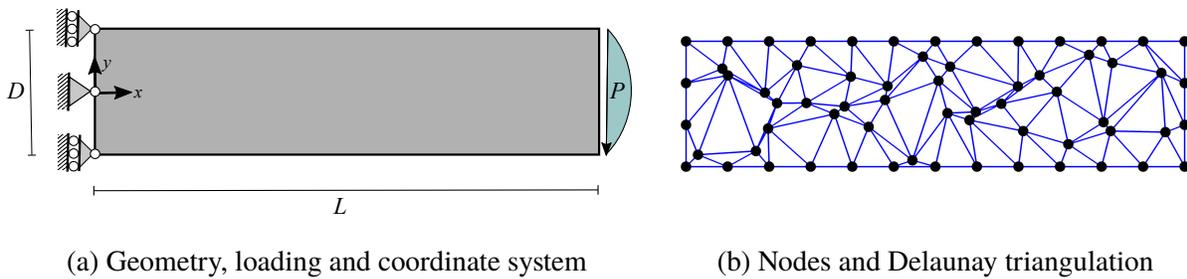


Figure 7.13: Timoshenko beam with shear load

and local windows with supports of $1.5h$ and $star^2(v)$ windows are tested. The assumed strain integration technique is used with the integration cells being the triangles of a uniform refinement of the input triangulation (4 integration triangles per input triangle). The trapezoid rule is used for the cell boundary integration and thus the evaluation points are at the vertices and edge midpoints of the input triangulation. Again, this integration is compatible with the simplification procedure described in section 4.6. The exact traction is applied on the free end and exact displacement approximated on the fixed end. The window functions are conformed to the fixed end giving the weak Kronecker delta property. The essential boundary is enforced directly using static condensation with the boundary coefficients determined by performing a least squares fit to the exact displacement at the nodal locations.

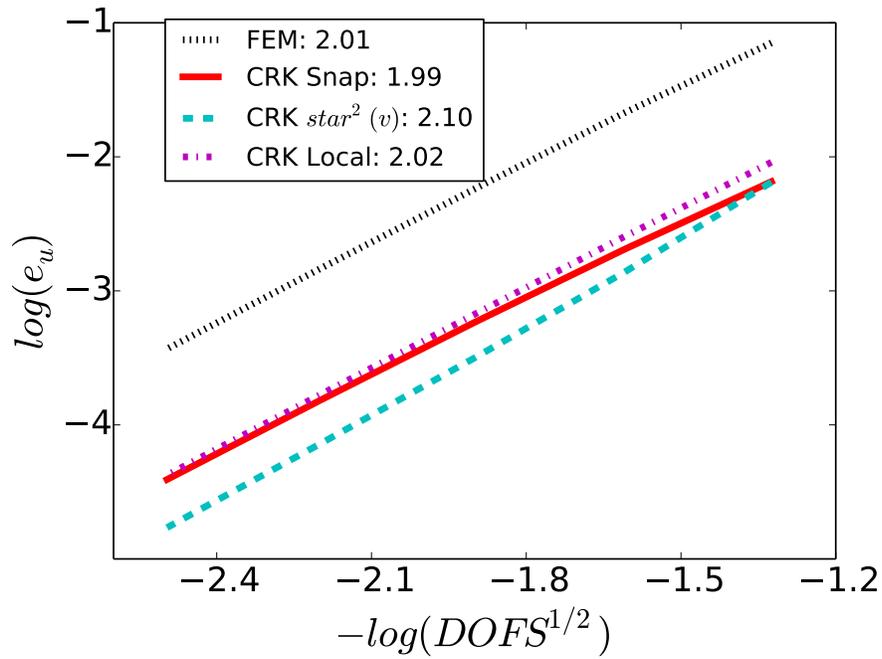
The convergence for the normalized displacement and energy norms are compared in Figure 7.14. The displacement error is similar for the different CRK methods. The energy norm plot shows the snap window procedure giving better solutions than the $star^2(v)$ windows. This is expected as the snap windows are designed to be less sensitive to the quality of the input triangulation. The local star has window shapes that are not connected to the input mesh and, as such, is providing the best solution for the energy norm.

7.1.5 Cook's membrane

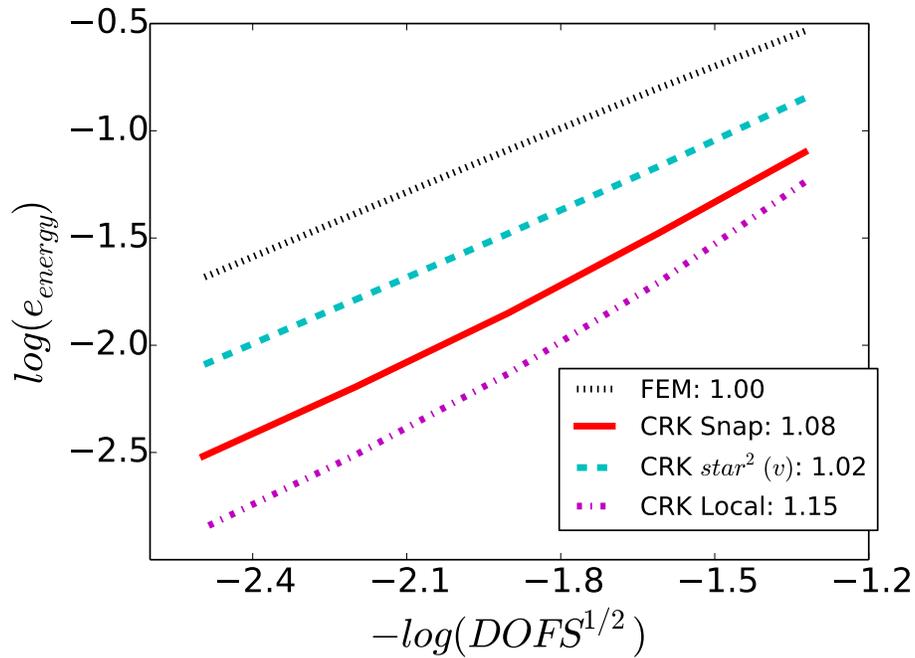
The Cook's membrane is investigated for a first look at the performance of the $\bar{\mathbf{F}}$ method discussed in Chapter 5. A drawing of the problem can be seen in Figure 7.15. A neo-Hookean model is used with $E = 1000$ and $\nu = 0.499$. The problem is simulated in three dimensions but with out-of-plane displacement fixed to represent plane-strain, the left surface is also fixed in the other two dimensions and a vertical displacement of 0.01 is applied on the right surface.

A $star^2(v)$ kernel is used and the CRK- $\bar{\mathbf{F}}$ predictions are setup with \mathbf{F}^{dev} calculated on each tetrahedron and the $star^1$ of elements around and including each tetrahedron for $\bar{\mathbf{F}}$.

A refinement study is completed using meshes with 2, 4, 8, and 16 elements along the



(a) Convergence of e_u



(b) Convergence of e_{energy}

Figure 7.14: Timoshenko beam convergence plots for FEM, CRK with $star^2(v)$ and snap and local stars with a support of $1.5h$. Boundaries are enforced using static condensation.

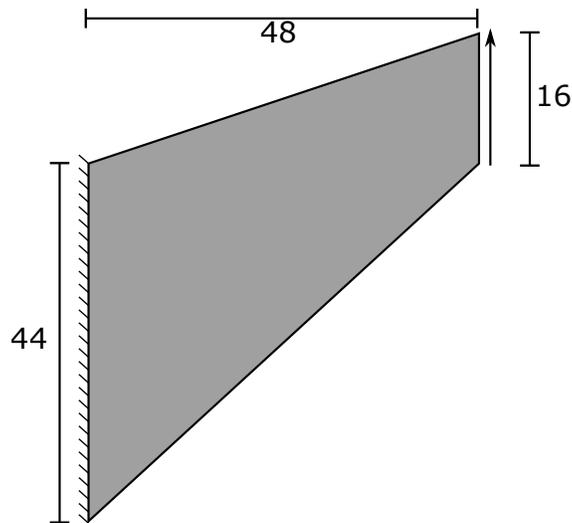


Figure 7.15: Cooks membrane. The left edge is fixed and the right edge is vertically displaced.

displaced edge. Snapshots of the hydrostatic stress field for CRK, RKPM-SCNI and CRK- \bar{F} with discretizations using 4 and 16 elements along the displaced edge are provided in Figures 7.16 and 7.17, respectively. The convergences for the predictions of the final tip load, normalized by the prediction using the selective deviatoric hexahedral element with mesh size of 1, are shown in Figure 7.18. The CRK method is clearly showing volumetric locking and an oscillating stress field. Locking has been mitigated with SCNI and CRK- \bar{F} . Pressure oscillations are eliminated with CRK- \bar{F} but some oscillations can be seen in the SCNI results. SCNI is known to have low energy modes though none are apparent in this problem. Predictions using the MSCNI method with stabilization coefficients of 0.05 and 0.2 are also provided in the convergence plot, showing a degradation in coarse mesh accuracy.

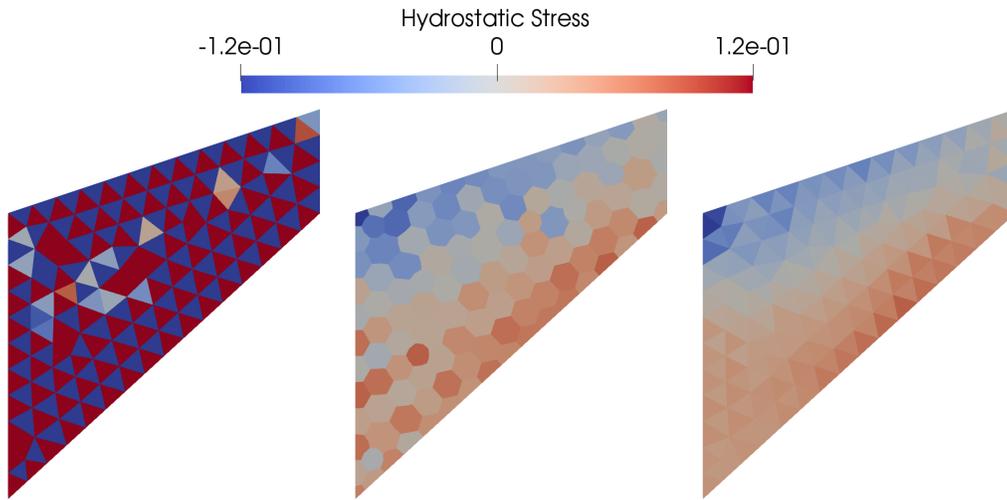


Figure 7.16: Cook's membrane. Hydrostatic Stress. Discretization using 4 element on the displaced edge. From left to right: CRK without $\bar{\mathbf{F}}$, RKPM with stabilized conforming nodal integration, CRK with $\bar{\mathbf{F}}$

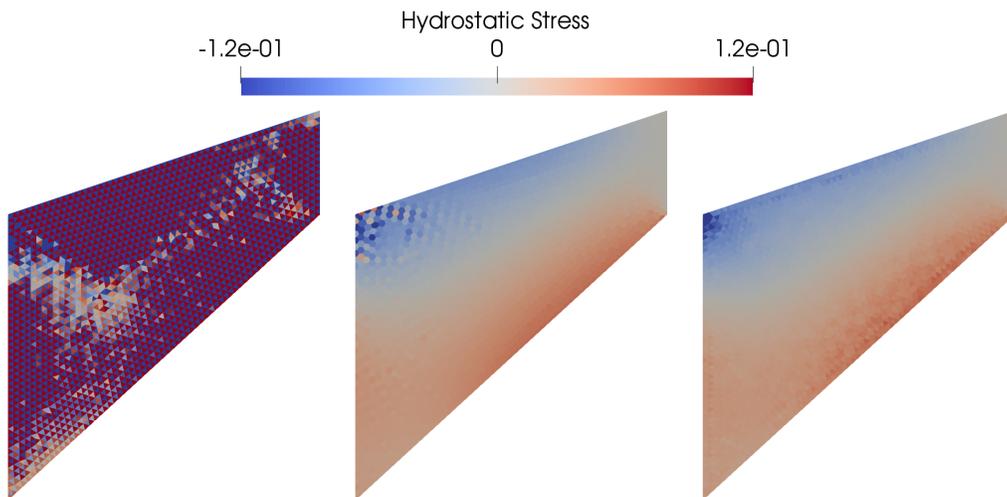


Figure 7.17: Cook's membrane. Hydrostatic Stress. Discretization using 16 element on the displaced edge. From left to right: CRK without $\bar{\mathbf{F}}$, RKPM with stabilized conforming nodal integration, CRK with $\bar{\mathbf{F}}$

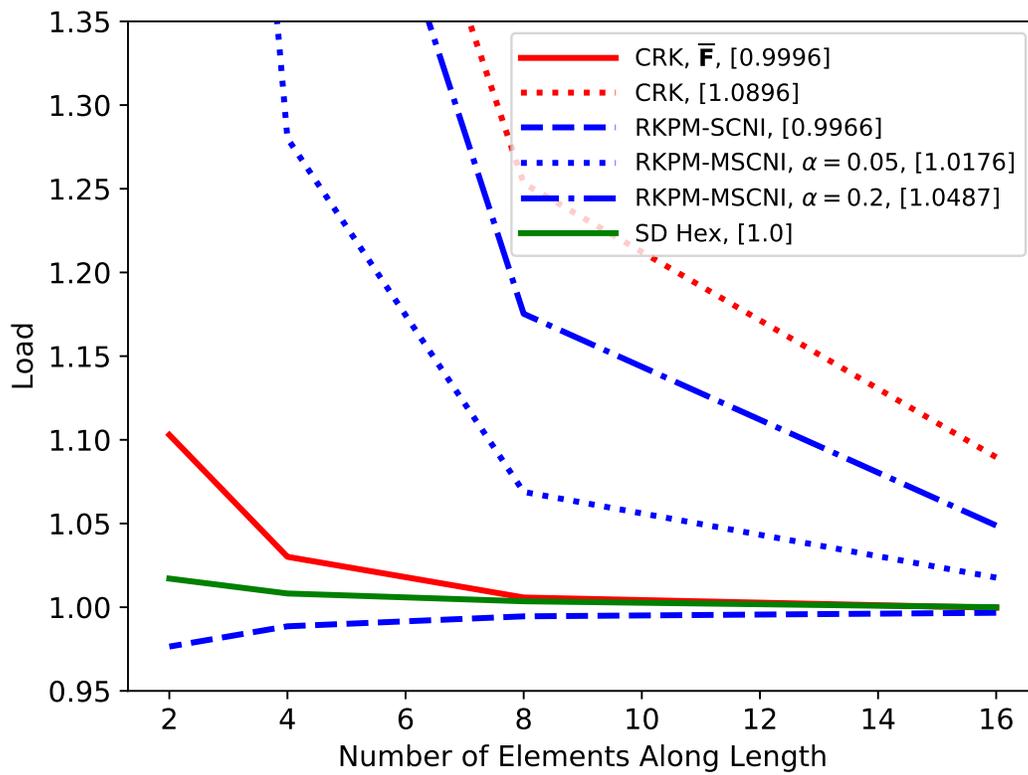


Figure 7.18: Cook's membrane. Convergence of tip force, relative to the refined prediction using the selective deviatoric element, versus number of elements along the displaced edge.

7.2 Three dimensional problems

7.2.1 Taylor bar

A Taylor bar impact problem is investigated to explore the performance of CRK. The setup, similar to what is used in [95, 86], is a cylindrical aluminum bar with an initial height of 2.346 cm and radius of 0.391 cm, that impacts a rigid wall with an initial velocity of 373 m/s. The wall impact is approximated by fixing the deformation to be in-plane on the impacting surface. J_2 plasticity with isotropic hardening is considered, and the material properties are Young's modulus $E = 78.2$ GPa, Poisson's ratio $\nu = 0.30$, density $\rho = 2700$ kg/m³. The yield function is taken as

$$f(\boldsymbol{\sigma}^{dev}, \bar{e}_p) = \|\boldsymbol{\sigma}^{dev}\| - \sqrt{\frac{2}{3}}K(\bar{e}_p) \quad (7.12)$$

where

$$K(\bar{e}_p) = 0.6\sigma_Y(\bar{e}_p)^{0.27} \quad (7.13)$$

and \bar{e}_p is the equivalent plastic strain and $\sigma_Y = 0.29$ GPa.

A refinement study is completed using 25, 50, 100, and 200 elements along the length of the bar. An example mesh is provided in Figure 7.19. A hexahedral mesh was constructed for use with the corresponding FEM analysis. The tetrahedral mesh was generated from the hexahedral mesh using the *htet* command in Cubit [61], splitting each hexahedron into six tetrahedrons.

As with the Cook's membrane example, the CRK predictions use a *star*² kernel. The CRK- $\bar{\mathbf{F}}$ predictions, use each tetrahedron of the mesh as a material point providing \mathbf{F}^{dev} and using the star of elements around each tetrahedron for $\bar{\mathbf{F}}^{dil}$. The CRK method is also used without the $\bar{\mathbf{F}}$ method where both the deviatoric and dilatational response is calculate by applying the strain smoothing operator over each tetrahedron. The RKPM results use a spherical support with size of $1.6x$ the local nodal spacing. The MSCNI predictions use a stabilization coefficient $\alpha = 0.2$.

Figures 7.20 and 7.21 show snapshots of the axial stress field predictions for the CRK- $\bar{\mathbf{F}}$,

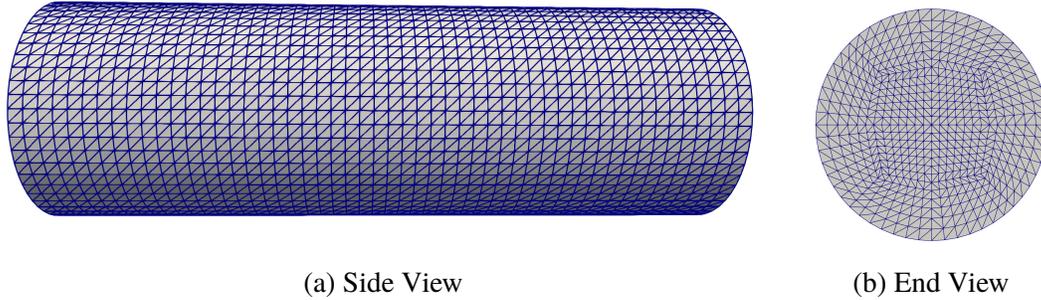


Figure 7.19: Example tetrahedral mesh for the Taylor bar with 50 elements along the length.

CRK and SCNI-based simulations for the discretizations using 25 and 100 elements along the length. The convergence for the prediction of the final radius of the impacting face is shown in Figure 7.22. The CRK method is clearly showing volumetric locking and an oscillating stress field whereas both SCNI and CRK- $\bar{\mathbf{F}}$ have smooth stress fields and much better coarse discretization accuracy. SCNI is known to have low energy modes though none are apparent in this problem. Predictions using the MSCNI method with a stabilization coefficient of 0.2 are also provided in the convergence plot, showing a degradation in coarse mesh accuracy.

A quick-look at the performance is provided in Table 7.2. The three dimensional CRK implementation uses the Sierra ToolKit (STK) [96]. In particular, it leverages the STK Mesh utilities for storing and manipulating meshed-based data. The mesh-based foundation of CRK exposes means of using FEM toolkits such as STK. As such, an efficient implementation is more readily available.

CRK, like RKPM and other methods that use meshfree shape functions, has approximation functions that are less local than FEM. Computing local values is more expensive than FEM as it involves accumulating data from more neighbors. On the other hand, meshfree-based methods often have larger stable time increments than mesh-based methods. These competing effects result in the CRK method being competitive with a linear tetrahedron element. Using a $\bar{\mathbf{F}}$ formulation does increase the expense over 4x for this problem but provides the same quality of solution as CRK, and several other methods, at much coarser discretizations.

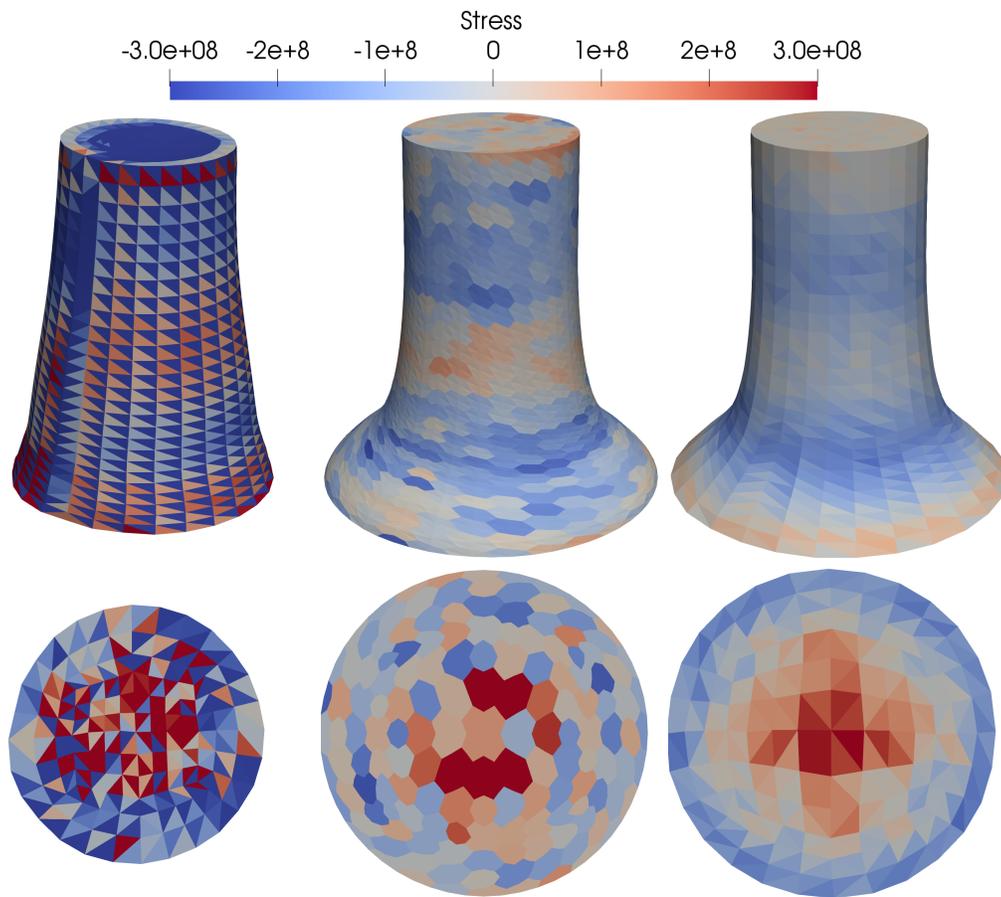


Figure 7.20: 3D Taylor bar impact. Normal stress in vertical directions. Discretization using 25 elements along the length. From left to right: CRK without $\bar{\mathbf{F}}$, RKPM with stabilized conforming nodal integration, CRK with $\bar{\mathbf{F}}$. Side view on top row and end view on bottom row.

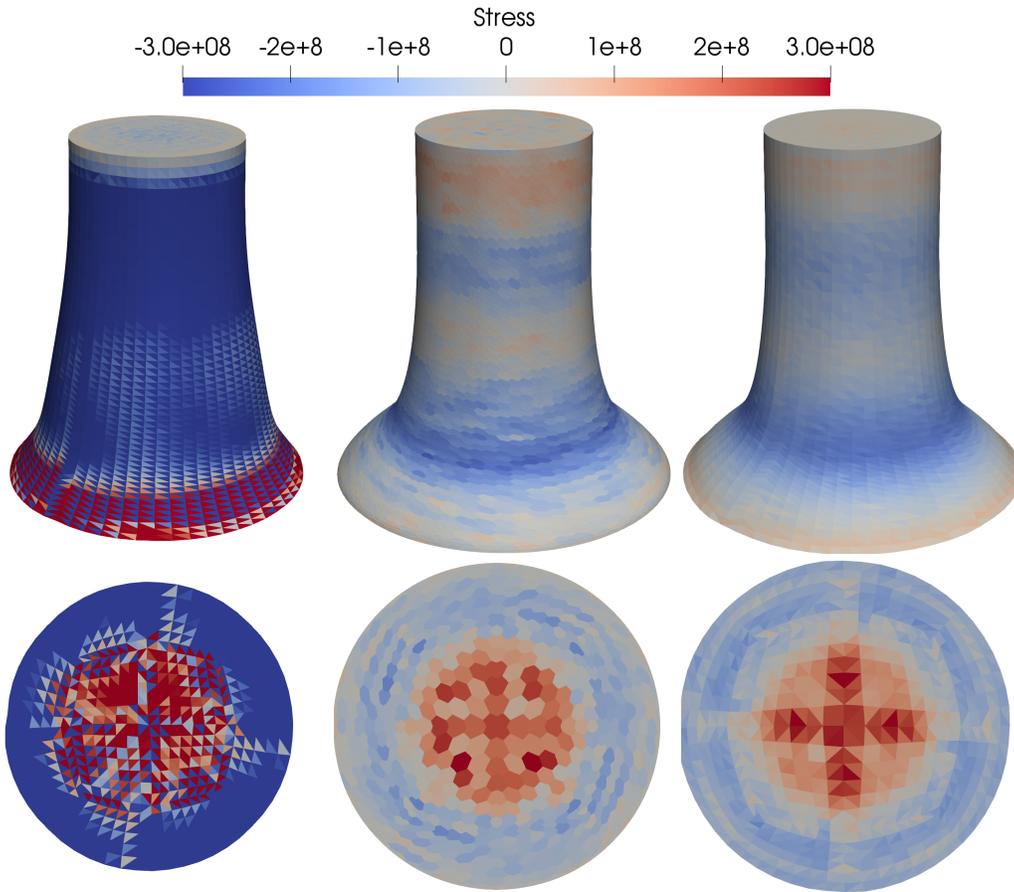


Figure 7.21: Taylor bar impact. Normal stress in vertical directions. Discretization using 100 elements along the length. From left to right: CRK without \bar{F} , RKPM with stabilized conforming nodal integration, CRK with \bar{F} . Side view on top row and end view on bottom row.

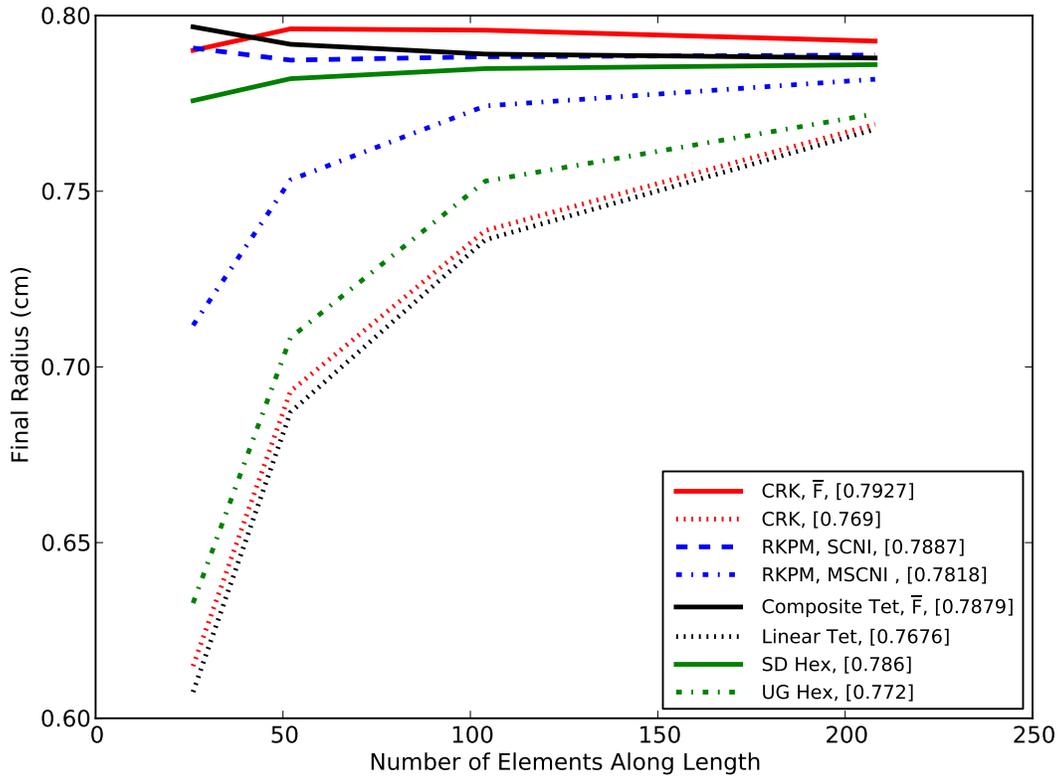


Figure 7.22: Taylor bar impact. Convergence of radial displacement with mesh refinement

Table 7.2: Run times for the Taylor bar using 50 elements along the length. The meshes for the hexahedral elements, linear tetrahedral elements and CRK have 29,733 nodes. The hexahedral mesh has 27,644 elements and the tetrahedral mesh has 165,984 elements. The composite tetrahedron uses a higher order element with 10 nodes. Two discretizations are provide, a coarser mesh that has 29,521 nodes and 20,592 elements and a finer mesh with 229,425 nodes and 165,984 elements. The CRK analysis using the snap window has a relative support size of 1.6x the nodal spacing. The global time step is estimated using the power method to estimate the largest eigenvalue. The problems are solved in parallel using 4 CPUs.

Method	Initial Time Step (seconds)	Run Time (seconds)
CRK	4.29E-8	158
CRK- \bar{F}	5.72E-8	678
CRK-Snap, SCNI	6.37E-8	240
Linear Tet	1.53E-8	130
Composite Tet (coarse)	1.90E-8	147
Composite Tet (fine)	9.12E-9	2242
Uniform Gradient Hex	2.37E-8	42
Selective Deviatoric Hex	2.77E-8	279

7.2.2 Compression of a 3D elastomer billet

The compression of a hyperelastic billet is also studied. A schematic of the problem is provided in Figure 7.23. A billet with height of 2 mm and radius of 1 mm is compressed from both ends. The end displacements are constrained so no in-plane movement occurs. The symmetry of the problem is used such that 1/8 of the domain is simulated. The Gent model is used with a bulk modulus K of 100 MPa and Young's modulus E of 1 MPa. An element size of 0.1 mm is used.

CRK- $\bar{\mathbf{F}}$ predictions are setup as in the previous sections with \mathbf{F}^{dev} on each tetrahedron, a star of elements around and including each tetrahedron for $\bar{\mathbf{F}}^{dil}$ and a *star*² of tetrahedra for the conforming kernel domains. The method is compared to CRK without $\bar{\mathbf{F}}$ and RKPM using SCNI and MSCNI using a stabilization coefficient of 0.05.

Snapshots of the deformed billet with hydrostatic stress field at three displacements are provided in Figure 7.24. As the billet is compressed the material bulges radially and eventually wraps around toward the loaded faces. Overly stiff responses are observed with CRK without $\bar{\mathbf{F}}$ and MSCNI as the wrapping behavior is restricted. SCNI shows instabilities and hydrostatic stress oscillations starting early in the analysis and growing with time. MSCNI does remove the instabilities at lower displacements but eventually a few nodes begin to protrude. CRK with $\bar{\mathbf{F}}$ does well at predicting the response through large deformation while providing a smoother hydrostatic stress prediction.

7.2.3 Carbon Black Microstructure

The previous three dimensional examples show the performance of CRK and the need for the $\bar{\mathbf{F}}$ formulation but the geometries are simple and can be readily meshed with hexahedral elements. Microstructures provide geometries that can present significant challenges in meshing. The microstructure of an elastomer reinforced with carbon particles is used in this example. The representative cubic domain, (29 x 29 x 29 nm) shown in Figure 7.25 is simulated. The geometry

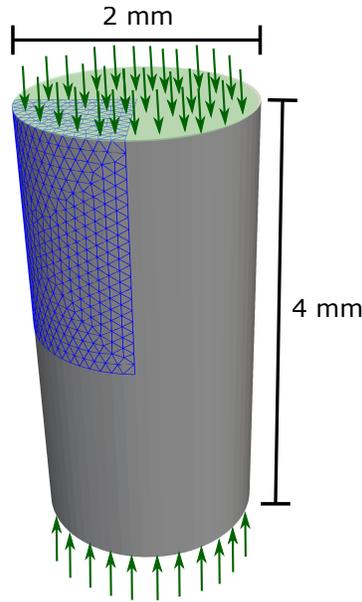


Figure 7.23: Compression billet with the mesh depicting the portion of the domain that is simulated.

can be easily meshed with tetrahedrons with only a few minutes worth of user input. Traditional hexahedral meshing techniques would require the domain to be decomposed into subdomains that are amenable to paving and sweeping algorithms. This has not been completed here and is estimated to take 10 hours of user input. However, the *Sculpt* algorithm in Cubit [97, 61] was used to create a hexahedral mesh. *Sculpt* uses a volume fraction-based approach to eliminate the need for domain decomposition. Balancing element quality and geometric accuracy is still a challenge and the resulting mesh in Figure 7.25c took approximately two hours of user time to generate.

The carbon was simulated as elastic with a density of 1200 kg/m^3 , Young's modulus $E = 25 \text{ GPa}$ and Poisson's ratio $\nu = 0.25$. The elastomer was simulated with the Gent model with a density of 980 kg/m^3 , shear modulus $G = 950 \text{ KPa}$ and bulk modulus $K = 920 \text{ MPa}$. The displacement was fixed to be in-plane on the top and bottom of the domain and a shearing displacement was increased until the analysis failed due to issues with the element Jacobian (FEM) or deformation gradient.

CRK- $\bar{\mathbf{F}}$ predictions are setup as in the previous sections with \mathbf{F}^{dev} on each tetrahedron, a

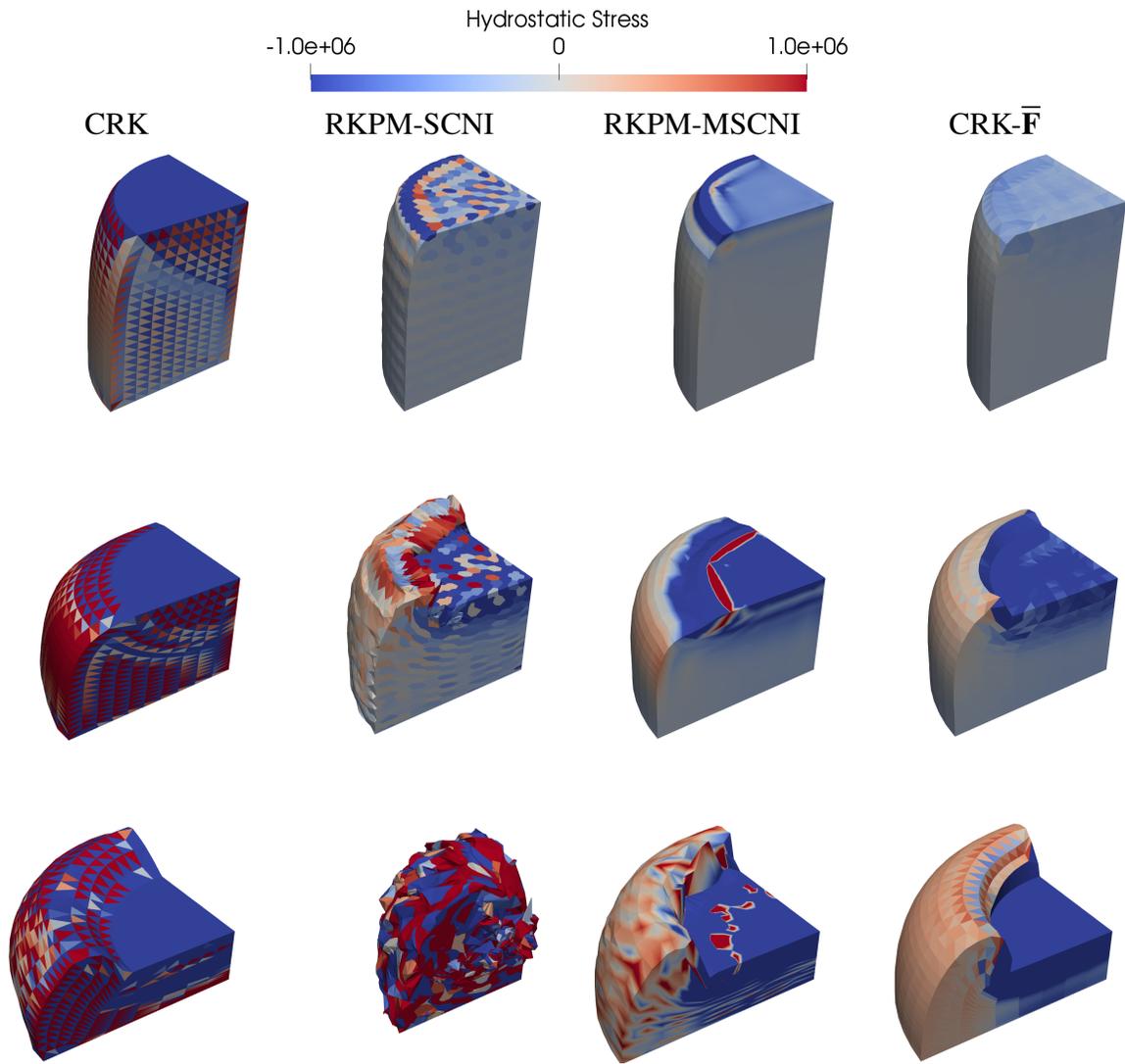


Figure 7.24: Compression billet. Hydrostatic Stress. From top to bottom: displacement of 0.50, 1.0 and 1.5 millimeters.

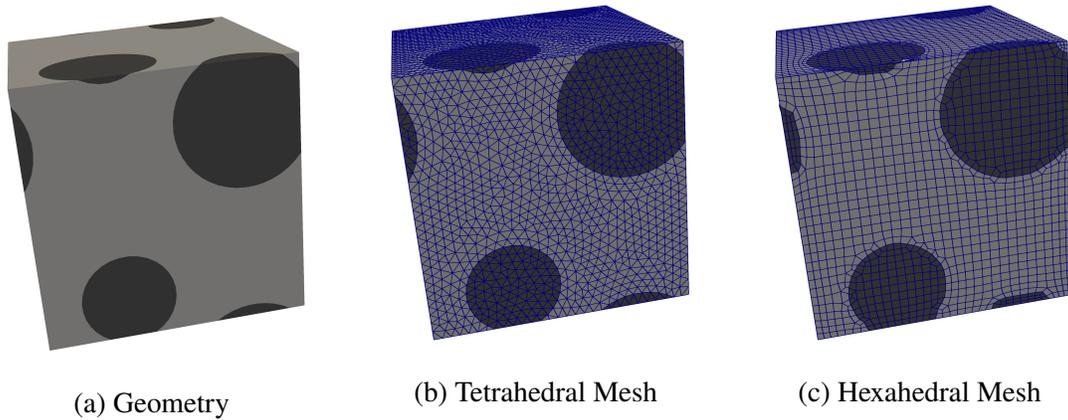


Figure 7.25: Domain and example meshes for the representative volume of an elastomer embedded with carbon particles.

star of elements around and including each tetrahedron for $\bar{\mathbf{F}}^{dil}$ and a *star*² of tetrahedra for the conforming kernel domains. The method is compared to RKPM where approximation functions do not conform to the material interfaces. The integration for the RKPM example is the same as for the CRK example. Additionally, simulations using the composite tetrahedron, uniform gradient and selective deviatoric hexahedrons were completed.

Snapshots with the hydrostatic stress field are provided in Figure 7.26. At moderate deformation, all methods provide similar stress predictions. The load versus displacement predictions are provided in Figure 7.27. The end of the curves marks where the analysis failed due to complications. The CRK and RKPM predictions reach much higher loads than the FEM results.

Three levels of mesh refinement were also completed. The predicted load versus mesh edge length at a displacement of 14 nm is shown in Figure 7.28. All methods appear to be approaching the same load. The uniform gradient hexahedron and composite tetrahedron were unable to reach 14 nm of displacement. The RKPM is showing less coarse mesh accuracy than CRK, likely due to a degraded approximation near the material interface.

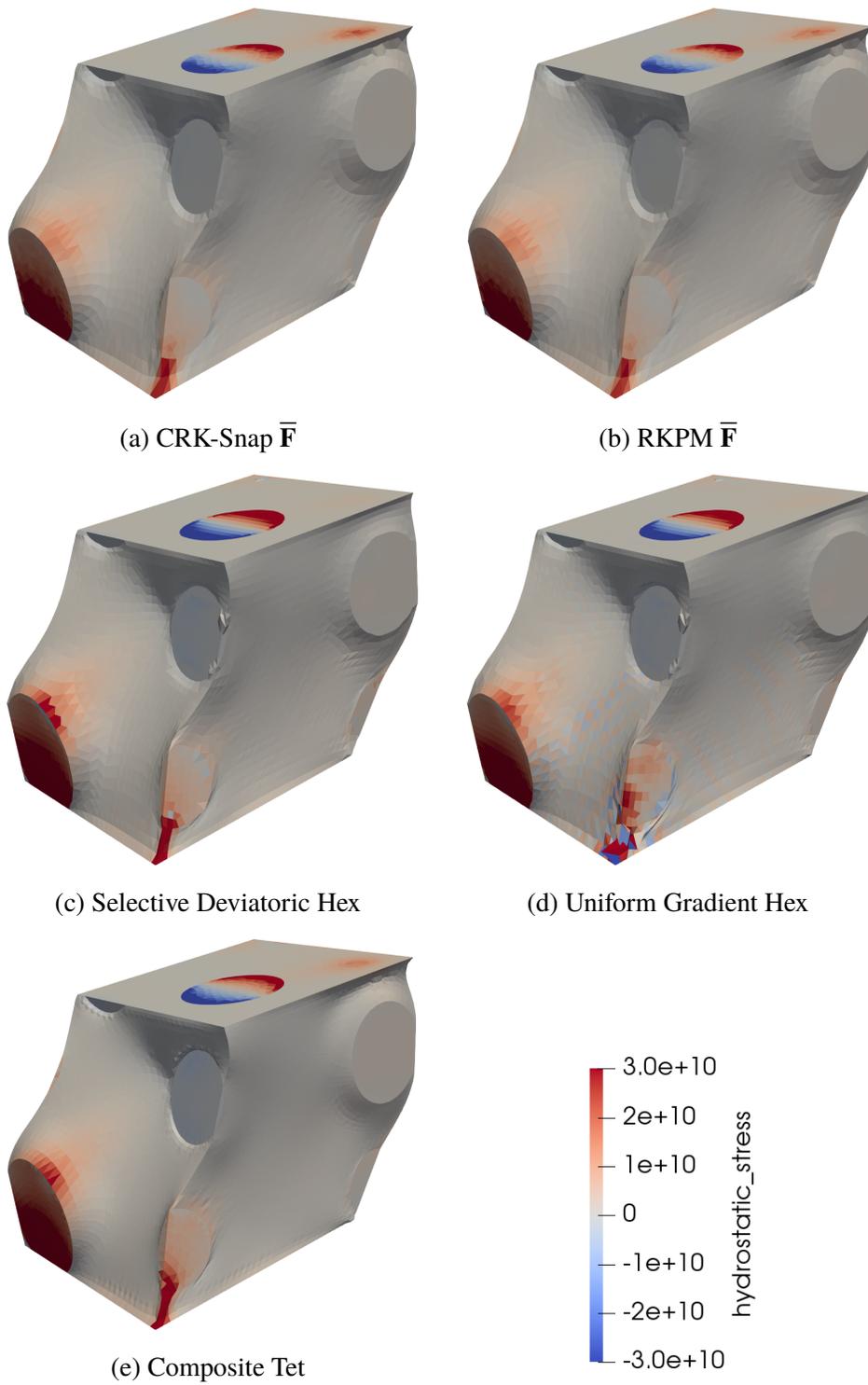


Figure 7.26: Hydrostatic stress predictions for the carbon black microstructure with a shear displacement of 30 nm.

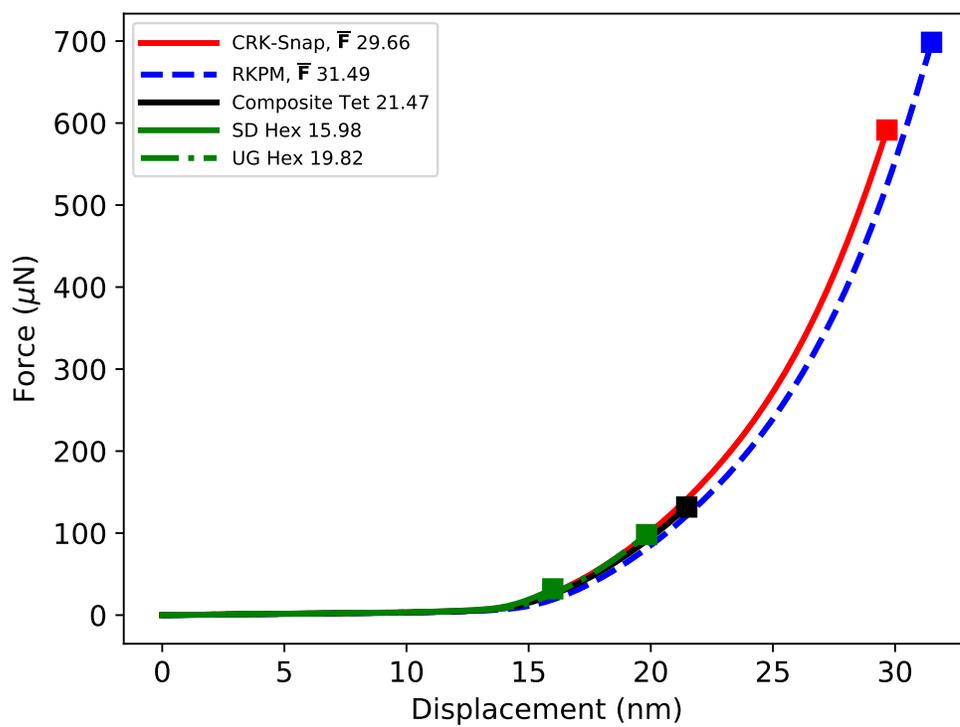


Figure 7.27: Load versus displacement predictions carbon black microstructure with a mesh edge length of 4 nm.

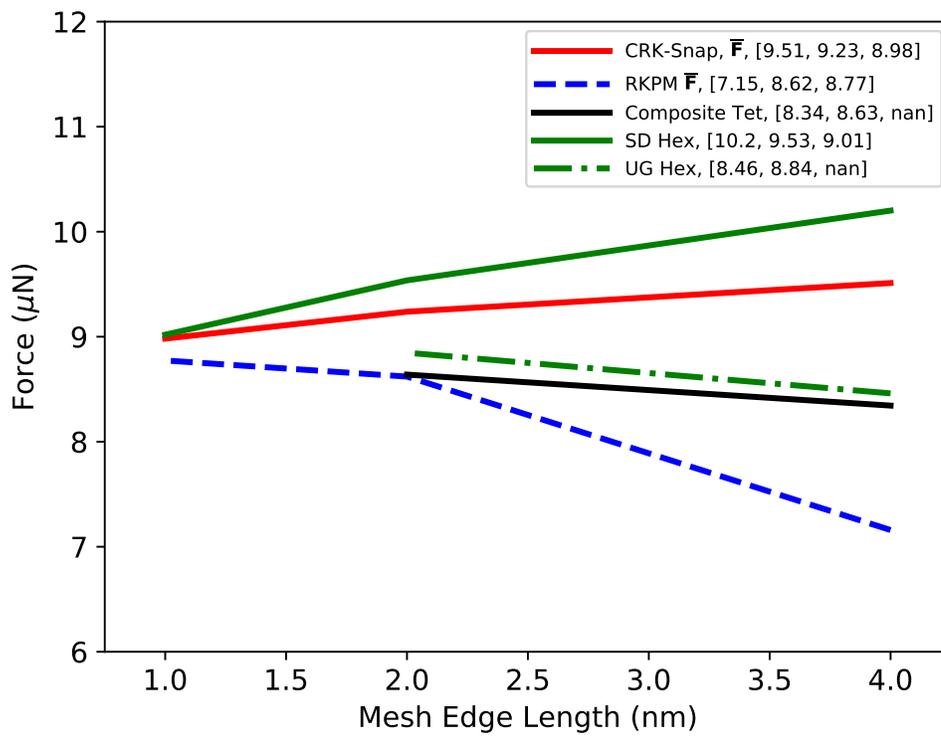


Figure 7.28: Convergence of the load prediction at a displacement of 14 nm for the carbon black microstructure under shear loading.

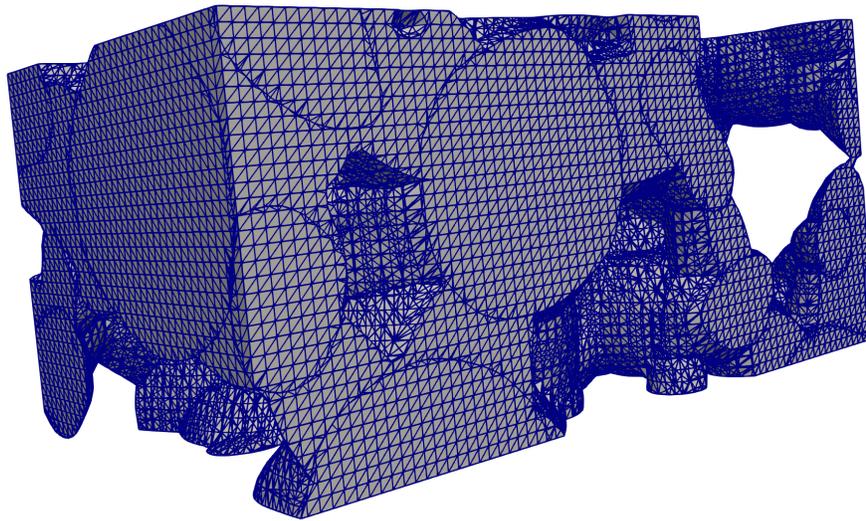


Figure 7.29: Subdomain and CDFEM mesh for the battery mesostructure

7.2.4 Battery microstructure simulation

As a final example, the a (3 x 6 x 3) subsection of the battery mesostructure from Chapter 7 is simulated. The domain and mesh are displayed in Figure 7.29 The CDFEM mesh contained 821,437 elements and 173,917 which were aggregated and decimated to give 50,000 nodes each with two integration cells giving a total of 100,000 cells. The materials were represented as elastic with the same properties as the two-sphere example in Chapter 7. Displacement was fixed to be in-plane on the sides of the domain. No in-plane displacement was allowed on the top and bottom where a compressive displacement was smoothly ramped to 0.5.

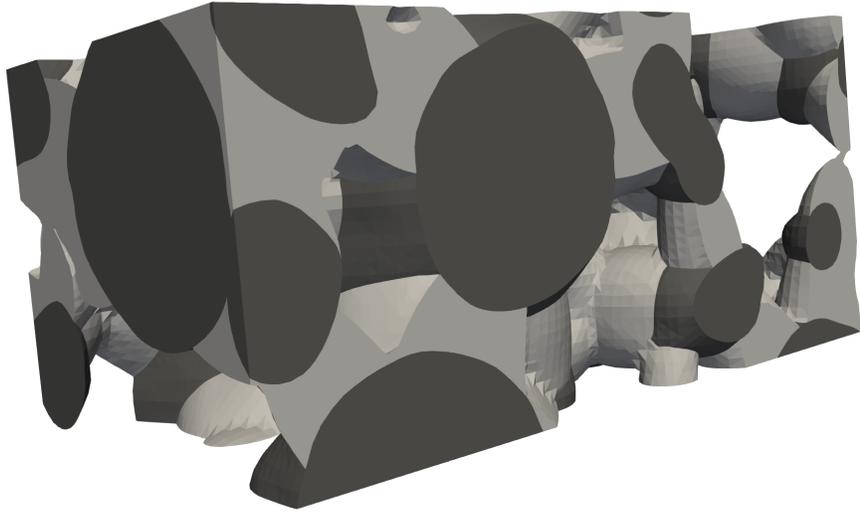
FEM and CRK analyses were attempted by directly using the CDFEM mesh but the simulations failed after a few iterations. The CRK analyses that used the aggregated and decimated mesh successfully completed. The initial stable time increments are listed in Table 7.3. The stable time increment of the aggregated CRK analysis was more than 3x larger than the CRK analysis without aggregation and 200x larger than FEM. Displacements and stresses are represented in Figures 7.30 and 7.31. The stress profile shows more texture than would be expected for this

Table 7.3: Stable time increments for simulations of the battery mesostructure.

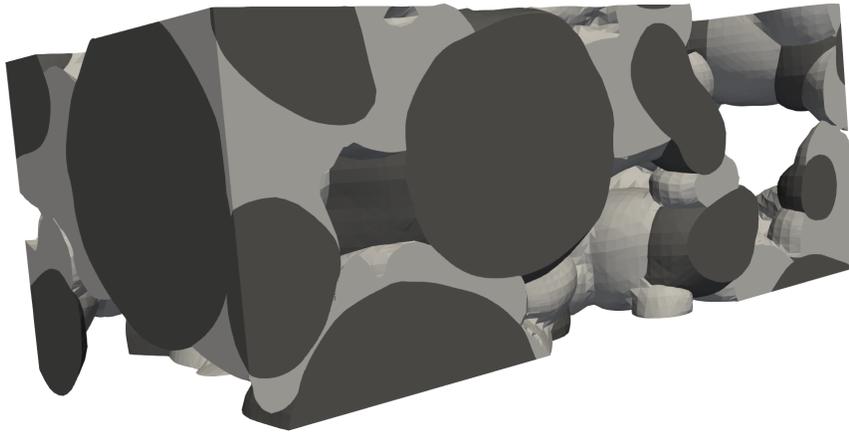
Method	Initial Time Step (seconds)
FEM	1.83E-8
CRK, no aggregation	9.86E-7
CRK, with aggregation	3.68E-6

problem, likely indicating a need for improved clustering. An close up view of some integration cells is provided in Figure 7.32. The volume are well-balanced but many of the cell boundaries are jagged. An additional quality metric to control the cluster shape may help improve the solution quality and will be explored in the future.

Material from this Chapter appears in *Computer Methods in Applied Mathematics and Engineering*, “Conforming window functions for meshfree methods”, 347:588 – 621, 2019, by Jacob Koester and J.S. Chen. and is expected to appear in *Computational Particle Mechanics* as “ $\bar{\mathbf{B}}$ and $\bar{\mathbf{F}}$ approaches for the treatment of near-incompressibility in meshfree and immersed-particle methods” with authors Georgios Moutsanidis, Jacob Koester, Michael Tupek, Yuri Bazilevs and J.S. Chen.

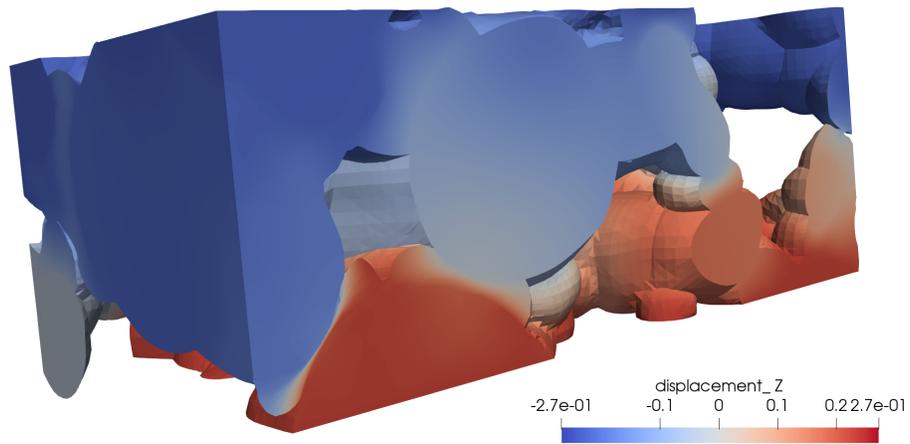


(a) Initial State

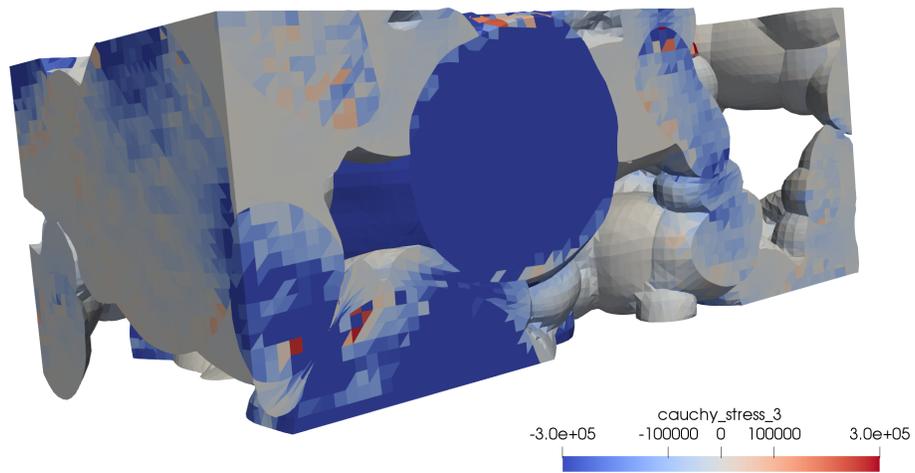


(b) Final State

Figure 7.30: Battery mesostructure. Initial state and final state after displacement.



(a) Final Displacement



(b) Final Stress

Figure 7.31: Displacement field and stress σ_{zz} (the loading direction) at the final displacement

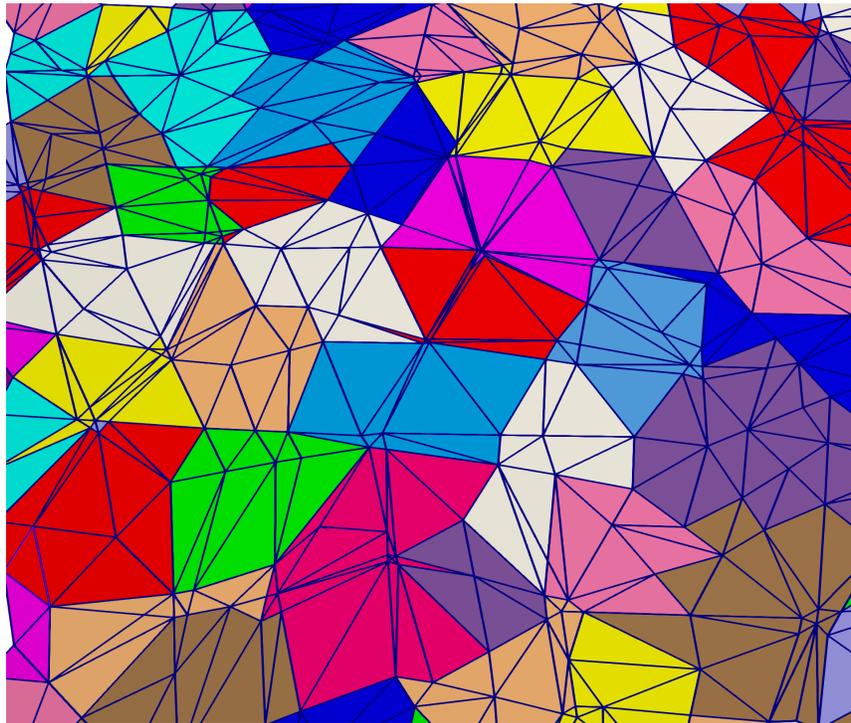


Figure 7.32: Example of clusters

Chapter 8

Conclusions

8.1 Conclusions

An efficient means of generating simulation models for complex geometries while retaining numerical accuracy and efficiency has been elusive. Unfortunately, simulating problems with complex geometries by meshfree methods has also been challenging. Meshfree methods provide great flexibility in the construction of approximation functions but generally result in degraded representations of complex geometries and domain boundaries.

The root issue for common meshfree methods such as RKPM and EFG is the difficulty in making approximation functions aware of, or conform to, accurate representations of domain boundaries. The extent of an approximation function is controlled by the extent of its window / kernel function. Window functions are typically defined using Euclidean metrics and cannot be easily made to conform to arbitrary geometries.

This work developed and demonstrated an alternate method of defining window functions that greatly improves control of the function support. The proposed method for constructing conforming window functions uses the Hermite interpolation properties of Bernstein-Bézier splines defined on local triangulations. The choice of spline space controls the attainable

smoothness of window and thus approximation functions. The local triangulations can be constructed separately for individual functions or extracted from a global triangulation. This method centralizes the solution to several boundary related complications by

- systematically handling non-convex regions,
- providing a means of directly creating weak discontinuities in the approximation space,
- generating admissible approximation spaces for enforcement of essential boundary conditions.

This can reduce the complexity of the simulation framework by eliminating the need of having several interacting methods handle each challenge separately. If a global triangulation is used, data structures and proximity searching can be simplified when compared to traditional meshfree methods, helping with the efficiency of the implementation.

An efficient implementation was also highlighted, demonstrating how the conforming window functions can be used while circumventing building or evaluating functions on the Bernstein-Bézier splines spaces. This was accomplished by collocating the nodal parameter locations with evaluation points needed for domain and boundary integration.

Although this method does require subdivisions of the geometry, some typical restrictions are not present, providing flexibility in the process of subdividing. The flexibility is realized as

- triangulations are used instead quadrilaterals (hexahedrals in three dimensions), simplifying the subdivision process
- local overlapping triangulations can be used and a global triangulation is not required.

Also, the snap and local window functions utilized a Euclidean metric in their construction and were demonstrated to have improved accuracy in the presence of low quality triangles.

With the conforming window functions, higher smoothness is attainable without increasing the size of the system. That is, for a given triangulation where vertices correspond to the unknown

coefficients of the system, the method is capable of keeping the smoothness of the spline spaces used to define the window functions while maintaining completeness of any order. Furthermore, the method can be used to recover linear triangle or tetrahedral finite elements if a global triangulation, linear bases and $star^1(v)$ window functions are selected, simplifying coupling with finite elements.

A method to mitigate volumetric locking that is compatible with efficient conforming window construction was also developed. The method describes a projection that is used in the established $\bar{\mathbf{F}}$ procedure. The projection provides a smoothed volumetric strain from a larger volume than is used in determining the smoothed deviatoric strain, reducing the near-incompressibility constraint.

A proof-of-concept for utilizing poor quality triangulations was presented and demonstrated. The procedure used k-means clustering to aggregate tetrahedra into larger integration cells, eliminating the issues associated with small tetrahedra. The clusters were also used to decimate the degrees-of-freedom. Instead of using every vertex of the input triangulation, a single vertex per integration cell was selected to carry coefficients with the other vertices remaining strictly to provide the geometries of integration cells. Recursive clustering was also highlighted as a means for splitting each integration cell into subcells to be used in the $\bar{\mathbf{F}}$ procedure.

The performance of the conforming window functions, $\bar{\mathbf{F}}$ and aggregation / decimation methods was demonstrated on several relevant problems. The conforming window functions improved solution accuracy and convergence for problems with non-convex domains and material interfaces. The $\bar{\mathbf{F}}$ method effectively mitigated volumetric locking for nearly incompressible hyperelastic and elasto-plastic materials. Clustering was used on poor quality meshes, demonstrating that the stable times increment can be two to three orders of magnitude larger than a FEM analysis on the same mesh. Solution quality and robustness was also greatly improved by aggregating and decimating.

8.2 Recommendation for future research

The work presented herein can be further developed and extended in several directions. First, aggregation and decimation can be further explored and improved. The clustering procedure presented effectively aggregated elements based on proximity and volume but additional quality metrics can be developed and used to improve cluster shape and thus improve solution quality. Clustering can also be explored as a means of generating reduced-order models.

Several applications involve material separation or very large deformations and would benefit from an ability to periodically update the discretization. A procedure for reconstructing conforming window function to account for newly formed surfaces could be explored. Also, the method could leverage existing re-meshing algorithms to generate updated discretizations or adopt the semi-Lagrangian framework [98] when deformation becomes sufficiently large.

Some applications would benefit from a more accurate or smooth surface representation. Rational Bernstein-Bézier splines could be used to construct conforming window functions that represent the exact surface instead of a faceted approximation.

Bibliography

- [1] M. F. Hardwick, R. L. Clay, P. T. Boggs, E. J. Walsh, A. R. Larzelere, and A. Altshuler, “DART system analysis,” Sandia National Laboratories, Tech. Rep. SAND2005-4647, 2005. [Online]. Available: <https://prod.sandia.gov/techlib-noauth/access-control.cgi/2005/054647.pdf>
- [2] P. Thoutireddy, J. F. Molinari, E. A. Repetto, and M. Ortiz, “Tetrahedral composite finite elements,” *International Journal for Numerical Methods in Engineering*, vol. 53, no. 6, pp. 1337–1351, 2002. [Online]. Available: <http://dx.doi.org/10.1002/nme.337>
- [3] J. T. Ostien, J. W. Foulk, A. Mota, and M. G. Veilleux, “A 10-node composite tetrahedral finite element for solid mechanics,” *International Journal for Numerical Methods in Engineering*, 2016. [Online]. Available: <http://dx.doi.org/10.1002/nme.5218>
- [4] M. A. Puso and J. Solberg, “A stabilized nodally integrated tetrahedral,” *International Journal for Numerical Methods in Engineering*, vol. 67, no. 6, pp. 841–867, 2006. [Online]. Available: <http://dx.doi.org/10.1002/nme.1651>
- [5] 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*, vol. 74, no. 3, pp. 416–446, 2008. [Online]. Available: <http://dx.doi.org/10.1002/nme.2181>
- [6] C. Dohrmann, M. W. Heinstejn, J. Jung, S. W. Key, and W. R. Witkowski, “Node-based uniform strain elements for three-node triangular and four-node tetrahedral meshes,” *International Journal for Numerical Methods in Engineering*, vol. 47, no. 9, pp. 1549–1568, 2000. [Online]. Available: [http://dx.doi.org/10.1002/\(SICI\)1097-0207\(20000330\)47:9<1549::AID-NME842>3.0.CO;2-K](http://dx.doi.org/10.1002/(SICI)1097-0207(20000330)47:9<1549::AID-NME842>3.0.CO;2-K)
- [7] K. T. Danielson, “Fifteen node tetrahedral elements for explicit methods in nonlinear solid dynamics,” *Computer Methods in Applied Mechanics and Engineering*, vol. 272, pp. 160 – 180, 2014. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0045782514000255>
- [8] S. Martin, P. Kaufmann, M. Botsch, M. Wicke, and M. Gross, “Polyhedral finite elements using harmonic basis functions,” *Computer Graphics Forum*, vol. 27, no. 5, pp. 1521–1529, 2008. [Online]. Available: <http://dx.doi.org/10.1111/j.1467-8659.2008.01293.x>

- [9] J. E. Bishop, “A displacement-based finite element formulation for general polyhedra using harmonic shape functions,” *International Journal for Numerical Methods in Engineering*, vol. 97, no. 1, pp. 1–31, 2014. [Online]. Available: <http://dx.doi.org/10.1002/nme.4562>
- [10] G. Manzini, A. Russo, and N. Sukumar, “New perspectives on polygonal and polyhedral finite element methods,” *Mathematical Models and Methods in Applied Sciences*, vol. 24, no. 08, pp. 1665–1699, 2016/04/07 2014. [Online]. Available: <http://dx.doi.org/10.1142/S0218202514400065>
- [11] T. Belytschko, Y. Y. Lu, and L. Gu, “Element-free Galerkin methods,” *International Journal for Numerical Methods in Engineering*, vol. 37, no. 2, pp. 229–256, 1994. [Online]. Available: <http://dx.doi.org/10.1002/nme.1620370205>
- [12] W. K. Liu, S. Jun, and Y. F. Zhang, “Reproducing kernel particle methods,” *International Journal for Numerical Methods in Fluids*, vol. 20, no. 8-9, pp. 1081–1106, 1995. [Online]. Available: <http://dx.doi.org/10.1002/flid.1650200824>
- [13] J.-S. Chen, C. Pan, C.-T. Wu, and W. K. Liu, “Reproducing kernel particle methods for large deformation analysis of non-linear structures,” *Computer Methods in Applied Mechanics and Engineering*, vol. 139, no. 1–4, pp. 195 – 227, 1996. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0045782596010833>
- [14] Z. Li and K. Ito, *The Immersed Interface Method*. Society for Industrial and Applied Mathematics, 2006. [Online]. Available: <http://epubs.siam.org/doi/abs/10.1137/1.9780898717464>
- [15] D. Schillinger, L. Dedè, M. A. Scott, J. A. Evans, M. J. Borden, E. Rank, and T. J. 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*, vol. 249–252, no. 0, pp. 116 – 150, 2012, higher Order Finite Element and Isogeometric Methods. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S004578251200093X>
- [16] J. Bishop, “Rapid stress analysis of geometrically complex domains using implicit meshing,” *Computational Mechanics*, vol. 30, no. 5-6, pp. 460–478, 2003. [Online]. Available: <http://dx.doi.org/10.1007/s00466-003-0424-5>
- [17] J. Parvizian, A. Düster, and E. Rank, “Finite cell method,” *Computational Mechanics*, vol. 41, no. 1, pp. 121–133, 2007. [Online]. Available: <http://dx.doi.org/10.1007/s00466-007-0173-y>
- [18] T.-P. Fries and S. Omerović, “Higher-order accurate integration of implicit geometries,” *International Journal for Numerical Methods in Engineering*, vol. 106, no. 5, pp. 323–371, 2016. [Online]. Available: <http://dx.doi.org/10.1002/nme.5121>
- [19] L. Kudela, N. Zander, S. Kollmannsberger, and E. Rank, “Smart octrees: Accurately integrating discontinuous functions in 3d,” *Computer Methods in Applied Mechanics*

- and Engineering*, vol. 306, no. Supplement C, pp. 406–426, 2016. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0045782516301578>
- [20] T. Hughes, J. Cottrell, and Y. Bazilevs, “Isogeometric analysis: CAD, finite elements, nurbs, exact geometry and mesh refinement,” *Computer Methods in Applied Mechanics and Engineering*, vol. 194, no. 39–41, pp. 4135 – 4195, 2005. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0045782504005171>
- [21] L. Engvall and J. A. Evans, “Isogeometric triangular Bernstein–Bézier discretizations: Automatic mesh generation and geometrically exact finite element analysis,” *Computer Methods in Applied Mechanics and Engineering*, vol. 304, pp. 378–407, 6 2016. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0045782516300342>
- [22] S. Xia and X. Qian, “Isogeometric analysis with bézier tetrahedra,” *Computer Methods in Applied Mechanics and Engineering*, vol. 316, pp. 782–816, 2017. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0045782516312774>
- [23] G. Xu, B. Mourrain, R. Duvigneau, and A. Galligo, “Analysis-suitable volume parameterization of multi-block computational domain in isogeometric applications,” *Computer-Aided Design*, vol. 45, no. 2, pp. 395–404, 2013. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0010448512002278>
- [24] L. Liu, Y. Zhang, T. J. R. Hughes, M. A. Scott, and T. W. Sederberg, “Volumetric t-spline construction using boolean operations,” *Engineering with Computers*, vol. 30, no. 4, pp. 425–439, 2014. [Online]. Available: <https://doi.org/10.1007/s00366-013-0346-6>
- [25] J.-S. Chen, M. Hillman, and S.-W. Chi, “Meshfree methods: Progress made after 20 years,” *Journal of Engineering Mechanics*, vol. 143, no. 4, p. 04017001, 2017/07/28 2017. [Online]. Available: [https://doi.org/10.1061/\(ASCE\)EM.1943-7889.0001176](https://doi.org/10.1061/(ASCE)EM.1943-7889.0001176)
- [26] J.-S. Chen, C.-T. Wu, S. Yoon, and Y. You, “A stabilized conforming nodal integration for Galerkin mesh-free methods,” *International Journal for Numerical Methods in Engineering*, vol. 50, no. 2, pp. 435–466, 2001. [Online]. Available: [http://dx.doi.org/10.1002/1097-0207\(20010120\)50:2<435::AID-NME32>3.0.CO;2-A](http://dx.doi.org/10.1002/1097-0207(20010120)50:2<435::AID-NME32>3.0.CO;2-A)
- [27] J.-S. Chen, M. Hillman, and M. Rüter, “An arbitrary order variationally consistent integration for Galerkin meshfree methods,” *International Journal for Numerical Methods in Engineering*, vol. 95, no. 5, pp. 387–418, 2013. [Online]. Available: <http://dx.doi.org/10.1002/nme.4512>
- [28] T. Belytschko, Y. Krongauz, D. Organ, M. Fleming, and P. Krysl, “Meshless methods: An overview and recent developments,” *Computer Methods in Applied Mechanics and Engineering*, vol. 139, no. 1–4, pp. 3 – 47, 1996. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S004578259601078X>

- [29] P. Krysl and T. Belytschko, “Element-free Galerkin method: Convergence of the continuous and discontinuous shape functions,” *Computer Methods in Applied Mechanics and Engineering*, vol. 148, no. 3–4, pp. 257 – 277, 1997. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0045782596000072>
- [30] D. Organ, M. Fleming, T. Terry, and T. Belytschko, “Continuous meshless approximations for nonconvex bodies by diffraction and transparency,” *Computational Mechanics*, vol. 18, no. 3, pp. 225–235, 1996. [Online]. Available: <http://dx.doi.org/10.1007/BF00369940>
- [31] D. Wang, J.-S. Chen, and L. Sun, “Homogenization of magnetostrictive particle-filled elastomers using an interface-enriched reproducing kernel particle method,” *Finite Elements in Analysis and Design*, vol. 39, no. 8, pp. 765 – 782, 2003, 14th Robert J. Melosh Competition. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0168874X03000581>
- [32] Y. Krongauz and T. Belytschko, “EFG approximation with discontinuous derivatives,” *International Journal for Numerical Methods in Engineering*, vol. 41, no. 7, pp. 1215–1233, 1998. [Online]. Available: [http://dx.doi.org/10.1002/\(SICI\)1097-0207\(19980415\)41:7<1215::AID-NME330>3.0.CO;2-#](http://dx.doi.org/10.1002/(SICI)1097-0207(19980415)41:7<1215::AID-NME330>3.0.CO;2-#)
- [33] L. Cordes and B. Moran, “Treatment of material discontinuity in the element-free Galerkin method,” *Computer Methods in Applied Mechanics and Engineering*, vol. 139, no. 1–4, pp. 75 – 89, 1996. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0045782596010808>
- [34] G. J. Wagner and W. K. Liu, “Application of essential boundary conditions in mesh-free methods: a corrected collocation method,” *International Journal for Numerical Methods in Engineering*, vol. 47, no. 8, pp. 1367–1379, 2000. [Online]. Available: [https://doi.org/10.1002/\(SICI\)1097-0207\(20000320\)47:8<1367::AID-NME822>3.0.CO;2-Y](https://doi.org/10.1002/(SICI)1097-0207(20000320)47:8<1367::AID-NME822>3.0.CO;2-Y)
- [35] J.-S. Chen and H.-P. Wang, “New boundary condition treatments in meshfree computation of contact problems,” *Computer Methods in Applied Mechanics and Engineering*, vol. 187, no. 3–4, pp. 441 – 468, 2000. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0045782500800043>
- [36] J.-S. Chen, W. Han, Y. You, and X. Meng, “A reproducing kernel method with nodal interpolation property,” *International Journal for Numerical Methods in Engineering*, vol. 56, no. 7, pp. 935–960, 2003. [Online]. Available: <http://dx.doi.org/10.1002/nme.592>
- [37] Y. Krongauz and T. Belytschko, “Enforcement of essential boundary conditions in meshless approximations using finite elements,” *Computer Methods in Applied Mechanics and Engineering*, vol. 131, no. 1, pp. 133–145, 1996. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/004578259500954X>
- [38] S. Fernández-Méndez and A. Huerta, “Imposing essential boundary conditions in mesh-free methods,” *Computer Methods in Applied Mechanics and Engineering*, vol. 193, no. 12–14, pp. 1257–1275, 3 2004. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0045782504000222>

- [39] M. Griebel and M. A. Schweitzer, *A Particle-Partition of Unity Method Part V: Boundary Conditions*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 519–542. [Online]. Available: https://doi.org/10.1007/978-3-642-55627-2_27
- [40] J. Gosz and W. K. Liu, “Admissible approximations for essential boundary conditions in the reproducing kernel particle method,” *Computational Mechanics*, vol. 19, no. 1, pp. 120–135, 1996. [Online]. Available: <http://dx.doi.org/10.1007/BF02824850>
- [41] D. Millán, N. Sukumar, and M. Arroyo, “Cell-based maximum-entropy approximants,” *Computer Methods in Applied Mechanics and Engineering*, vol. 284, pp. 712–731, 2 2015. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S004578251400382X>
- [42] C. A. Duarte, D. J. Kim, and D. M. Quaresma, “Arbitrarily smooth generalized finite element approximations,” *Computer Methods in Applied Mechanics and Engineering*, vol. 196, no. 1–3, pp. 33–56, 12 2006. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0045782506001101>
- [43] V. L. Rvachev and T. I. Sheiko, “R-functions in boundary value problems in mechanics,” *Applied Mechanics Reviews*, vol. 48, no. 4, pp. 151–188, 04 1995. [Online]. Available: <http://dx.doi.org/10.1115/1.3005099>
- [44] M. S. Floater, “Mean value coordinates,” *Computer Aided Geometric Design*, vol. 20, no. 1, pp. 19 – 27, 2003. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0167839603000025>
- [45] M.-J. Lai and L. L. Schumaker, *Spline Functions on Triangulations*. Cambridge University Press, 2007.
- [46] P. Ciarlet, *The Finite Element Method for Elliptic Problems*. Society for Industrial and Applied Mathematics, 2002. [Online]. Available: <https://doi.org/10.1137/1.9780898719208>
- [47] P. Alfeld, “A trivariate Clough-Tocher scheme for tetrahedral data,” *Computer Aided Geometric Design*, vol. 1, no. 2, pp. 169–181, 1984. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/0167839684900293>
- [48] A. J. Worsey and G. Farin, “An n -dimensional Clough-Tocher interpolant,” *Constructive Approximation*, vol. 3, no. 1, pp. 99–110, 1987. [Online]. Available: <https://doi.org/10.1007/BF01890556>
- [49] A. J. Worsey and B. Piper, “A trivariate Powell-Sabin interpolant,” *Computer Aided Geometric Design*, vol. 5, no. 3, pp. 177–186, 1988. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/0167839688900015>
- [50] E. Catmull and J. Clark, “Recursively generated b-spline surfaces on arbitrary topological meshes,” *Computer-Aided Design*, vol. 10, no. 6, pp. 350 – 355, 1978. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/0010448578901100>

- [51] D. Doo and M. Sabin, “Behaviour of recursive division surfaces near extraordinary points,” *Computer-Aided Design*, vol. 10, no. 6, pp. 356 – 360, 1978. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/0010448578901112>
- [52] J. Stam, “Exact evaluation of catmull-clark subdivision surfaces at arbitrary parameter values,” in *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH ’98. New York, NY, USA: ACM, 1998, pp. 395–404. [Online]. Available: <http://doi.acm.org/10.1145/280814.280945>
- [53] F. Cirak, M. Ortiz, and P. Schröder, “Subdivision surfaces: a new paradigm for thin-shell finite-element analysis,” *International Journal for Numerical Methods in Engineering*, vol. 47, no. 12, pp. 2039–2072, 2000. [Online]. Available: [http://dx.doi.org/10.1002/\(SICI\)1097-0207\(20000430\)47:12<2039::AID-NME872>3.0.CO;2-1](http://dx.doi.org/10.1002/(SICI)1097-0207(20000430)47:12<2039::AID-NME872>3.0.CO;2-1)
- [54] D. Burkhart, B. Hamann, and G. Umlauf, “Iso-geometric finite element analysis based on catmull-clark subdivision solids,” *Computer Graphics Forum*, vol. 29, no. 5, pp. 1575–1584, 2010. [Online]. Available: <http://dx.doi.org/10.1111/j.1467-8659.2010.01766.x>
- [55] Y.-S. Chang, K. T. McDonnell, and H. Qin, “A new solid subdivision scheme based on box splines,” in *Proceedings of the Seventh ACM Symposium on Solid Modeling and Applications*, ser. SMA ’02. New York, NY, USA: ACM, 2002, pp. 226–233. [Online]. Available: <http://doi.acm.org/10.1145/566282.566316>
- [56] W.-K. Liu, S. Li, and T. Belytschko, “Moving least-square reproducing kernel methods (i) methodology and convergence,” *Computer Methods in Applied Mechanics and Engineering*, vol. 143, no. 1, pp. 113–154, 1997. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0045782596011322>
- [57] J. Nitsche, “Über ein variationsprinzip zur lösung von dirichlet-problemen bei verwendung von teilräumen, die keinen randbedingungen unterworfen sind,” in *Abhandlungen aus dem mathematischen Seminar der Universität Hamburg*, vol. 36, no. 1. Springer, 1971, pp. 9–15.
- [58] 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*, vol. 21, no. 3, pp. 211–222, 1998. [Online]. Available: <https://doi.org/10.1007/s004660050296>
- [59] Y. Lu, T. Belytschko, and L. Gu, “A new implementation of the element free Galerkin method,” *Computer Methods in Applied Mechanics and Engineering*, vol. 113, no. 3–4, pp. 397 – 414, 1994. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/0045782594900566>
- [60] L. L. Schumaker, *Spline Functions*. Society for Industrial and Applied Mathematics, 2015.
- [61] “Cubit, geometry and meshing toolkit,” 10 2018. [Online]. Available: <https://cubit.sandia.gov>

- [62] E. Yreux and J.-S. Chen, “A quasi-linear reproducing kernel particle method,” *International Journal for Numerical Methods in Engineering*, vol. 109, no. 7, pp. 1045–1064, 2017, nme.5319. [Online]. Available: <http://dx.doi.org/10.1002/nme.5319>
- [63] F. Brezzi and M. Fortin, *Mixed and hybrid finite element methods*. Springer Science & Business Media, 2012, vol. 15.
- [64] J. C. Simo and M. Rifai, “A class of mixed assumed strain methods and the method of incompatible modes,” *International journal for numerical methods in engineering*, vol. 29, no. 8, pp. 1595–1638, 1990.
- [65] U. Brink and E. Stein, “On some mixed finite element methods for incompressible and nearly incompressible finite elasticity,” *Computational Mechanics*, vol. 19, no. 1, pp. 105–119, 1996.
- [66] O. Klaas, A. Maniatty, and M. S. Shephard, “A stabilized mixed finite element method for finite elasticity.: Formulation for linear displacement and pressure interpolation,” *Computer Methods in Applied Mechanics and Engineering*, vol. 180, no. 1-2, pp. 65–79, 1999.
- [67] A. M. Maniatty, Y. Liu, O. Klaas, and M. S. Shephard, “Higher order stabilized finite element method for hyperelastic finite deformation,” *Computer methods in applied mechanics and engineering*, vol. 191, no. 13-14, pp. 1491–1503, 2002.
- [68] B. Ramesh and A. M. Maniatty, “Stabilized finite element formulation for elastic–plastic finite deformations,” *Computer Methods in Applied Mechanics and Engineering*, vol. 194, no. 6-8, pp. 775–800, 2005.
- [69] E. Wilson, R. Taylor, W. Doherty, and J. Ghaboussi, “Incompatible displacement models,” in *Numerical and computer methods in structural mechanics*. Elsevier, 1973, pp. 43–57.
- [70] E. P. Kasper and R. L. Taylor, “A mixed-enhanced strain method: Part I: Geometrically linear problems,” *Computers & Structures*, vol. 75, no. 3, pp. 237–250, 2000.
- [71] E. P. Kasper and R. L. Taylor, “A mixed-enhanced strain method: Part ii: Geometrically nonlinear problems,” *Computers & Structures*, vol. 75, no. 3, pp. 251–260, 2000. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0045794999001352>
- [72] T. Belytschko, J. S.-J. Ong, W. K. Liu, and J. M. Kennedy, “Hourglass control in linear and nonlinear problems,” *Computer Methods in Applied Mechanics and Engineering*, vol. 43, no. 3, pp. 251–276, 1984.
- [73] D. P. Flanagan and T. Belytschko, “A uniform strain hexahedron and quadrilateral with orthogonal hourglass control,” *International Journal for Numerical Methods in Engineering*, vol. 17, no. 5, pp. 679–706, 1981. [Online]. Available: <https://doi.org/10.1002/nme.1620170504>

- [74] S. Reese, P. Wriggers, and B. Reddy, “A new locking-free brick element technique for large deformation problems in elasticity,” *Computers & Structures*, vol. 75, no. 3, pp. 291–304, 2000.
- [75] T. J. Hughes, “Generalization of selective integration procedures to anisotropic and nonlinear media,” *International Journal for Numerical Methods in Engineering*, vol. 15, no. 9, pp. 1413–1418, 1980.
- [76] T. Hughes, *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*, ser. Dover Civil and Mechanical Engineering. Dover Publications, 2012. [Online]. Available: https://books.google.com/books?id=cHH2n_qBK0IC
- [77] T. J. Hughes, “Equivalence of finite elements for nearly incompressible elasticity,” *J Appl Mech ASME, E*, vol. 44, pp. 181–183, 1977.
- [78] T. Hughes and D. Malkus, “A general penalty/mixed equivalence theorem for anisotropic, incompressible finite elements,” in *International Symposium on Hybrid and Mixed Finite Element Methods, Atlanta, GA*, 1981, p. 1981.
- [79] D. S. Malkus and T. J. Hughes, “Mixed finite element methods—reduced and selective integration techniques: a unification of concepts,” *Computer Methods in Applied Mechanics and Engineering*, vol. 15, no. 1, pp. 63–81, 1978.
- [80] T. Elguedj, Y. Bazilevs, V. M. Calo, and T. J. R. Hughes, “ \bar{B} and \bar{F} projection methods for nearly incompressible linear and non-linear elasticity and plasticity using higher-order NURBS elements,” *Computer Methods in Applied Mechanics and Engineering*, vol. 197, no. 33, pp. 2732–2762, 2008. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0045782508000248>
- [81] T. Elguedj and T. J. Hughes, “Isogeometric analysis of nearly incompressible large strain plasticity,” *Computer Methods in Applied Mechanics and Engineering*, vol. 268, pp. 388–416, 2014.
- [82] J. Dolbow and T. Belytschko, “Volumetric locking in the element free galerkin method,” *International Journal for Numerical Methods in Engineering*, vol. 46, no. 6, pp. 925–942, 2019/02/06 1999. [Online]. Available: [https://doi.org/10.1002/\(SICI\)1097-0207\(19991030\)46:6<925::AID-NME729>3.0.CO;2-Y](https://doi.org/10.1002/(SICI)1097-0207(19991030)46:6<925::AID-NME729>3.0.CO;2-Y)
- [83] J.-S. Chen, S. Yoon, H.-P. Wang, and W. K. Liu, “An improved reproducing kernel particle method for nearly incompressible finite elasticity,” *Computer Methods in Applied Mechanics and Engineering*, vol. 181, no. 1, pp. 117–145, 2000. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0045782599000675>
- [84] J. S. Chen, H. P. Wang, S. Yoon, and Y. You, “Some recent improvements in meshfree methods for incompressible finite elasticity boundary value problems with contact,” *Computational Mechanics*, vol. 25, no. 2-3, pp. 137–156, 2000. [Online]. Available: <http://dx.doi.org/10.1007/s004660050465>

- [85] J.-S. Chen, S. Yoon, and C.-T. Wu, “Non-linear version of stabilized conforming nodal integration for Galerkin mesh-free methods,” *International Journal for Numerical Methods in Engineering*, vol. 53, no. 12, pp. 2587–2615, 2002. [Online]. Available: <http://dx.doi.org/10.1002/nme.338>
- [86] M. Hillman and J.-S. Chen, “An accelerated, convergent, and stable nodal integration in Galerkin meshfree methods for linear and nonlinear mechanics,” *International Journal for Numerical Methods in Engineering*, pp. n/a–n/a, 2015. [Online]. Available: <http://dx.doi.org/10.1002/nme.5183>
- [87] S. A. Roberts, H. Mendoza, V. E. Brunini, and D. R. Noble, “A verified conformal decomposition finite element method for implicit, many-material geometries,” *Journal of Computational Physics*, vol. 375, pp. 352–367, 2018. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S002199911830545X>
- [88] S. Lloyd, “Least squares quantization in PCM,” *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.
- [89] SIERRA Solid Mechanics Team, “Sierra/SolidMechanics 4.50 user’s guide,” Sandia National Laboratories, Tech. Rep. SAND2018-10673, 2018.
- [90] S. L. Kramer and B. L. Boyce, “2017 Sandia fracture challenge,” September 2017. [Online]. Available: <http://imechanica.org/node/20707>
- [91] B. A. Szabó and I. Babuška, “Computation of the amplitude of stress singular terms for cracks and reentrant corners,” in *Fracture Mechanics: Nineteenth Symposium*. ASTM International, 1988.
- [92] M. Fleming, Y. A. Chu, B. Moran, and T. Belytschko, “Enriched element-free Galerkin methods for crack tip fields,” *International Journal for Numerical Methods in Engineering*, vol. 40, no. 8, pp. 1483–1504, 1997. [Online]. Available: [http://dx.doi.org/10.1002/\(SICI\)1097-0207\(19970430\)40:8<1483::AID-NME123>3.0.CO;2-6](http://dx.doi.org/10.1002/(SICI)1097-0207(19970430)40:8<1483::AID-NME123>3.0.CO;2-6)
- [93] M. L. Kachanov, B. Shafiro, and I. Tsukrov, *Handbook of elasticity solutions*. Dordrecht, The Netherlands: Kluwer Academic Publishers, 2003.
- [94] S. Timoshenko and J. N. Goodier, *Theory of elasticity*, 3rd ed. New York: McGraw-Hill, 1970.
- [95] M. L. Wilkins and M. W. Guinan, “Impact of cylinders on a rigid boundary,” *Journal of Applied Physics*, vol. 44, no. 3, pp. 1200–1206, 2018/05/15 1973. [Online]. Available: <https://doi.org/10.1063/1.1662328>
- [96] SIERRA Toolkit Development Team, “Sierra Toolkit manual version,” Sandia National Laboratories, Tech. Rep. SAND2018-2856, 2018.

- [97] S. Owen and T. Shelton, “Validation of grid-based hex meshes with computational solid mechanics,” in *Proceedings of the 22nd International Meshing Roundtable*, J. Sarrate and M. Staten, Eds. Springer International Publishing, 2014, pp. 39–56. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-02335-9_3
- [98] P. 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*, vol. 41, no. 6, pp. 670 – 683, 2009, advances in the Dynamics of Granular Materials. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0167663609000404>