11-16-2016

A Framework for Studying Meshfree Geometry and a Method for Explicit Boundary Determination

Joseph Bradley Alford
University of South Florida, joe.b.alford@gmail.com

A Framework for Studying Meshfree Geometry and a Method for Explicit Boundary Determination

by

Joseph B. Alford

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
Department of Civil and Environmental Engineering
College of Engineering
University of South Florida

Major Professor: Daniel C. Simkins Jr., Ph.D.
Andrés E. Tejada-Martínez, Ph.D.
Les Piegl, Ph.D.
Lennox Hoyte, M.D.
Autar Kaw, Ph.D.

Date of Approval:
November 15, 2016

Keywords: Meshfree Methods, Implicit Geometry, Analysis-Suitable Geometry

Dedication

To my mom, who believed in me when I needed it most. To Coach DeAugustino, who taught me the meaning of perseverance, the value of hard work, and how to win and lose. To Becca, who stood by my side and supported me for this endeavor; without whom I may never have finished. I am forever grateful.

Acknowledgments

There are many people that contributed to my success as a graduate student. I would like to thank my adviser and friend, Dr. Dan Simkins, for providing me with the opportunity, and guiding me along the path to becoming a better academic and person. I would also like to thank Dr. Lennox Hoyte for giving my research the proper motivation and context with which I found the passion to succeed. I would like to thank Art Slowe and Bob Francis for giving me a look into the world of starting a tech company. I would like to acknowledge and thank the CEE departmental staff, I have depended on them many times and they always come through. I would like to especially thank James Olliff, his hard work and moral support were instrumental in helping me finish my research.

Table of Contents

List of Tables

List of Figures

Abstract

Patient-specific biomechanical analysis is an important tool used to understand the complex processes that occur in the body due to physical stimulation. Patient-specific models are generated by processing medical images; once an object from the image is identified via segmentation, a point cloud representation of the object is extracted. Generating an analysis suitable representation from the point cloud has traditionally required generating a finite element mesh, which often requires a well defined surface to accomplish. Point clouds lack a well defined geometry, meaning that the surface definition is incomplete at best. Point clouds that have been generated from images have a fuzzy boundary, meaning that no direct sampling of the boundary is guaranteed to exist and any method for completing the geometry definition requires assumptions on the part of the modeler. The process of generating a finite element mesh of the point cloud is difficult and tedious often requiring manual editing to alleviate poorly constructed elements.

An alternative to generating a finite element mesh is to use meshfree analysis to solve the boundary value problem. The geometric representation of meshfree analysis is a point cloud, thus making it a natural choice for patient-specific analysis. When using meshfree analysis, it is common to skip the geometric reconstruction and use the point cloud directly as an analysis suitable geometry. The lack of a well defined surface can be problematic for a variety of reasons, namely the visualization of results and solving contact driven problems.

The goal of this dissertation is to exploit some characteristics of the meshfree analysis to generate a well defined geometry for point clouds. Meshfree methods are commonly used for the solution of boundary value problems; their lack of a well defined geometry representation is a hindrance that is often remedied by accompanying the meshfree particle distribution with a secondary geometry representation, such as a mesh. The present work outlines a framework that can be used to define and study meshfree geometry representations. A particular meshfree geometry representation called the Meshfree Correction Implicit Geometry is introduced and studied under the guidelines of the framework.

Chapter 1: Introduction

1.1 Motivation

Product design has long benefited from computational engineering analysis to improve and increase efficiency of the design process. Computer-aided engineering (CAE) tools are used to analyze the structural integrity of an object, such as a product or part, during the initial design stage. CAE provides a low cost alternative to the traditional process of prototype generation and destructive testing, and have dramatically decreased the time and cost associated with product design while increasing product life span and consumer safety. Given a set of geometric constraints, computer-aided design (CAD) packages are used for the geometric design of a part. CAD packages use splines, e.g. NURBS, as an exact geometric representation of the design. CAE and CAD tools are used together, often in the same platform, for an iterative design process where CAD is used to refine the geometric model and CAE is used to determine the structural integrity.

CAE tools require a geometric representation that is consistent with the numerical method used in the analysis. A common numerical method used by CAE packages is finite element analysis (FEA). The geometric representation for FEA is a piece-wise polygonal approximation of the object called a mesh. Mesh generation is an automated task in some cases, but many designs have complex features that are not easily meshed. Mesh generation that is not automated requires manual input from the design engineer. Isogeometric analysis (IGA) was developed as a numerical method that requires the same geometric representa-

tion in the analysis stage as is used in the geometric design stage. IGA is actively being researched, and shows potential for reducing the cost of the product design cycle.

The strength of engineering analysis has garnered interest in various scientific fields such as the biology, medicine, and anthropology. Objects of interest in the scientific arena are typically defined by gathering data via imaging devices or laser scanners. Data resulting from these devices are referred to as point clouds, and are generically divided in two classes. Medical imaging modalities, e.g. magnetic resonance imaging (MRI) and computed tomography (CT), result in volumetric point clouds. Laser scanning devices sample the exterior of an object resulting in a surface point cloud. Generating FEA meshes of point clouds presents several problems. Point clouds lack a well defined boundary; before a FEA mesh can be generated, an intermediate step is required to compute the boundary of the point cloud. A variety of methods are used to compute the boundary of point clouds including direct tessellation, polynomial fitting, and distance functions.

Point clouds are the geometric representation for meshfree analysis (MFA); in the purview of meshfree methods point clouds are described as particle distributions. MFA is often used to solve problems that pose difficulties with FEA such as the large deformations associated with metal forming or impact fragment type problems. MFA is a natural fit when modeling point cloud data, yet is affected by the lack of a well defined geometry. The need for a well defined geometry is not evident in all cases; often the analysis is performed directly on the point cloud. Some tasks pose difficulties when dealing directly with the point cloud, such as surface integration and visualization of results.

The goal of this dissertation is to introduce a well defined geometry for particle distributions and provide a framework with which the qualities of the geometry may be assessed.

The time to produce an analysis from point cloud data, with a well defined geometry, is drastically reduced by eliminating the need for an intermediate geometry and the construction of a mesh.

1.2 Overview

The dissertation begins with a discussion on what constitutes a well defined geometry that is suitable for use in analysis. The geometric representations for FEA,MFA, and isogeometric analysis (IGA) is introduced along with characteristics and fundamental difficulties associated with each method of analysis. The difference in the geometric representations for each analysis method is highlighted, while noting that the particle distribution for MFA lacks a well defined geometry.

Following the discussion of the three analysis methods, a framework is introduced for studying the qualities of a well defined meshfree geometry. The framework has three constituents:

1. Definition of three domain representations: Intended, Computational, Approximate.

2. Meshfree geometry function for computing approximate domain given a computational domain.

3. Performance metric for determining the accuracy of the approximate domain in representing the intended domain.

While the domain definitions exist for any geometry used in numerical analysis, the distinction is not always necessary. The need for a distinction is evident when approximating the geometry of a point cloud. Assumptions need to be made when computing a well defined geometry for point clouds, so the approximate geometry will be different for any differences

in the assumption. The meshfree geometry function is defined here abstractly as an implicit function on the computational domain. This allows for any number of particular meshfree geometry functions, but the intent of this dissertation is to use characteristics of MFA to define meshfree geometry functions.

After the introduction of the framework a study is produced where a particular meshfree geometry function is defined. The geometry function is defined based on the consistency correction from the Reproducing Kernel Particle Method (RKPM) and is appropriately called the Meshfree Correction Implicit Geometry (MCIG). It is shown that the framework provides a methodology for explicitly determining the boundary of the particle distribution by minimizing a performance metric. Using the framework, the qualities of the MCIG are studied for two simple cases, a ring and a rectangular object. The following chapter studies the effect of using the MCIG as the contact surface in an elastic impact analysis; comparing the results from a FEA to the results of MFA with a MCIG geometry. The dissertation concludes with a discussion of patient specific modeling technology. The geometry of several patient specific data sets is then reconstructed using the MCIG.

Chapter 2: Analysis Suitable Geometry

A well posed boundary value problem (BVP) is made up of three distinct parts; namely the governing differential equations, the problem domain and a set of boundary conditions. For simplicity, consider the following BVP:

$$\begin{cases} \nabla^2 u = f \text{ for } x \in \Omega \\ u = g \text{ on } \Gamma_g \\ \nabla u \cdot \mathbf{n} = h \text{ on } \Gamma_h \end{cases} \tag{2.1}$$

where the boundary of the domain is decomposed as $\Gamma_g \cup \Gamma_h = \partial\Omega$ with $\Gamma_g \cap \Gamma_h = \varnothing$, and g h are the prescribed boundary conditions often referred to as the essential and natural boundary conditions respectively. For many BVP's, the solution can only be approximated through the machinery of a numerical method. The choice of geometric representation for the problem domain is dependent on the numerical method employed. Grids are the representation for the finite difference method (FDM), meshes for FEA, and particle distributions for MFA; paired with their respective method, the representations are called analysis suitable geometry (ASG). Not all geometry representations are analysis suitable, meshes and particle distributions need to meet constraints in order to be ASG for FEA and MFA respectfully. Of particular interest in this dissertation is the particle distribution associated with MFA. Particle distributions lack a well defined boundary; the intent of this dissertation is to de-

velop a well defined boundary for particle distributions, provide a framework for studying the accuracy and convergence of the geometry representation, and study the application of the new geometry as an ASG.

Due to imperfections in material, tolerances in manufacturing, noise in imaging modalities, human error, and the limitations of machine representation, computer models are always an approximation of reality. In the following sections a distinction is made between the intention of the modeler and the reality of the approximation. The numerical methods focused on in this dissertation are Galerkin methods; the Galerkin approximation and characteristics of commonly used Galerkin methods are discussed in this chapter.

2.1 The Galerkin Approximation

The BVP in Equation 2.1 is stated in what is referred to as the classical, or strong, form. There are methods that directly approximate the solution of the strong form, notably FDM and point collocation. The methods of consequence in this dissertation approximate the solution of the variational, or weak form, of the BVP. Defining the weak form of the BVP requires the introduction of two classes of functions, the trial solution and the variation [26]. The space of trial solutions is defined as:

$$\mathcal{S} = \{u \mid u \in H^1, u|_{\Gamma_g} = g\} \tag{2.2}$$

where H^1 is the space of functions with square integrable derivatives. The space of variations is defined as:

$$\mathcal{V} = \{\delta u \mid \delta u \in H^1, \delta u|_{\Gamma_h} = 0\} \tag{2.3}$$

The weak form of the BVP is then stated as given f, g, h find $u \in \mathcal{S}$ such that for all $\delta u \in \mathcal{V}$:

$$\int_\Omega \nabla(\delta u) \cdot \nabla u \, d\Omega - \int_{\Gamma_g} \delta u \nabla u \cdot \mathbf{n} d\Gamma = \int_\Omega \delta u f \, d\Omega + \int_{\Gamma_h} \delta u h d\Gamma \tag{2.4}$$

The Galerkin procedure for solving the weak form approximates u and δu by expanding them as a linear combination of known basis functions with undetermined coefficients.

$$u(x) = \sum_i^n N_i(x) u_i \qquad \delta u(x) = \sum_i^n N_i(x) \delta u_i \tag{2.5}$$

where n is the size of the function space, or in FEA and MFA the number of nodes in the discretization. It is convenient to express this summation as a vector dot product:

$$u(x) = \mathbf{N}\mathbf{u} \qquad v(x) = \mathbf{N}\boldsymbol{\delta}\mathbf{u} \tag{2.6}$$

where $\mathbf{u} = [u_1, u_2, u_3 ... u_n]^T$ and $\boldsymbol{\delta}\mathbf{u} = [\delta u_1, \delta u_2, \delta u_3 ... \delta u_n]^T$ are vectors of the nodal values and their variations respectively and $\mathbf{N} = [N_1, N_2, N_3 ... N_n]$ is a row vector of the shape functions. Then gradients are approximated in the same fashion:

$$\nabla u = \nabla \mathbf{N}\mathbf{u} \qquad \nabla(\delta u) = \nabla \mathbf{N}\boldsymbol{\delta}\mathbf{u} \tag{2.7}$$

with the strain displacement matrix defined as $\mathbf{B} := \nabla \mathbf{N} = [\mathbf{B}_1, \mathbf{B}_2, \mathbf{B}_3 ... \mathbf{B}_n]$ with $\mathbf{B}_i = N_{i,j}^T$ where j ranges over the number of spatial dimensions. Substituting these into the weak form leads to

$$\boldsymbol{\delta}\mathbf{u}^T \{ \int_\Omega \mathbf{B}^T \mathbf{B}\mathbf{u} \, d\Omega - \int_{\Gamma_g} \mathbf{B}^T \mathbf{B}\mathbf{u} \cdot \mathbf{n} d\Gamma = \int_\Omega \mathbf{N} f(x) \, d\Omega + \int_{\Gamma_h} \mathbf{N} h d\Gamma \} \tag{2.8}$$

The major steps involved in solving a weak form of a BVP with a Galerkin procedure are shown in the flow chart in Figure 2.1. Each one of the steps presented in the flow chart represent obstacles of varying difficulty depending on the choice of numerical method. Aside from the last step, solving for unknown coefficients, which will generally be the same independent of Galerkin method choice. The first step highlighted in the procedure is generating the ASG, which is a representation of the problem domain. FEA has the most difficulties with this step, as mesh generation can be an expensive and tedious task. The goal of using a meshfree geometry is to alleviate the burden of mesh generation when modeling point cloud data. The following section discusses the problem domain from the BVP, along with some formal definitions that are used later in the framework for studying meshfree geometry.

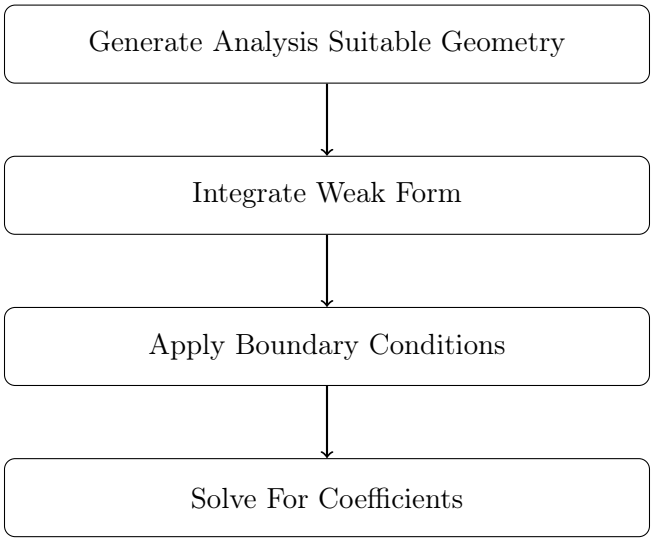


Figure 2.1: Flow chart for solution of BVP.

2.2 The Problem Domain

There are at least three geometric domains in play when approximating the solution of a BVP, the intended domain, the approximate domain, and the computational domain. The intended domain, defined in Definition 2.2.1, is defined exactly by an analytic function, spline representation, or even a physical object depending on the context.

Definition 2.2.1 (Intended Domain)**.** *The* intended domain *of an object, denoted* Ω *is the exact geometric representation of the object or domain to be analyzed.*

The computational domain, defined in Definition 2.2.2, varies in representation depending on the numerical method. For FEA the computational domain is the mesh, for IGA it is the set of control points, and MFA it is the particle distribution with overlapping compact supports. A compact support is a small region of space centered at a particle; the union of compact supports defines the bounds of the MFA computational domain. Constraints on the MFA computational domain will be defined more formally in later sections.

Definition 2.2.2 (Computational Domain)**.** *The* computational domain *of an object, denoted* $\bar{\Omega}_\rho$*, is the discretized computational geometric representation of the object or domain to be analyzed with associated characteristic length* ρ*.*

Finally, the approximate domain, defined in Definition 2.2.3, is dependent on the computational domain and expected to accurately represent the intended domain in some capacity.

Definition 2.2.3 (Approximate Domain)**.** *The* approximate domain *of an object, denoted* $\hat{\Omega}_\rho$*, is the image of the computational domain* $\bar{\Omega}_\rho$ *under approximation by the shape functions of the computational domain.*

It is worth noting here that the distinction between intended and approximate may not be as clear as the definitions provided here portray. As an example take the design process discussed in the introductory section. A CAD model is an approximation of a physical object, though the object may not exist yet, it is intended to be manufactured and the physical object will always be considered the exact representation. In this scope, the CAD

model is the approximate domain and the physical object is the intended domain. Yet in the CAE stage, a FEA mesh is generated to represent the CAD model, so the FEA mesh is the approximate domain and the CAD model is the intended domain. The differentiation of these representations is important, and the practitioner must always be wary of the geometric imperfections that impact analysis solutions and the manufacturing process.

Point clouds are often the most accurate portrayal that can be garnered from physical objects, especially those objects that are only measurable with imaging techniques e.g. internal organs. Point clouds are the best representation of reality that we can hope for, and the rest is just trying to fill in the gaps. A variety of methods have been developed for reconstructing the geometry of point clouds, and any of the methods that result in an ASG typically require multiple representations and a tedious work flow. The introduction of a well defined geometry for MFA allows for a well defined geometry for most point clouds and allows for the direct construction of an ASG for some point clouds.

In some cases $\bar{\Omega}_\rho$ and $\hat{\Omega}_\rho$ are the same, such as with linear FEA. Though this is more the exception than the rule; as with most other representations, including higher order elements, this is not true. In MFA the particle distribution itself is often treated as $\bar{\Omega}_\rho$, though difficulties often arise when evaluating terms in the weak form due to the lack of a well defined boundary. The crux of this dissertation is in developing a well defined geometry for MFA, and outlining a framework by which the geometry can be studied. The distinction between the domains is an integral part of the framework and will be used to define error metrics by which the performance of the meshfree geometry representation can be assessed.

2.3 The Finite Element Method

FEA has been used to successfully to solve a variety of problems, and is known for performing very well in representing complex geometry. The piece-wise polygonal boundary is well suited for representing complex surface features, though it may require a rather large number of small elements. The FEA shape functions are polynomial, so Gauss quadrature is used to integrate the weak form exactly with the exception of round off error. It is beneficial that the FEA shape functions have what is known as the Kronecker-delta property, which means the shape functions are interpolatory, i.e. the nodal coefficients equal the approximated function value at the nodes. This is a nice property because the boundary conditions may be applied directly with no additional constraints.

$$N_i(\mathbf{x}_j) = \delta_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \tag{2.9}$$

The shape function associated with a given mesh node is zero at every other node in the domain, and is only non-zero on elements and element edges to which the node is attached. Figure 2.2 depicts the linear FE shape functions on a one-dimensional domain, these functions are dubbed "hat" functions. Notice the different colors in the shape functions plot, each color represents the shape function associated with a different mesh node.

2.3.1 Fundamental Difficulty

Difficulty arises in FEA when the mesh is pushed to the limits. In problems with really large deformations, the elements may become distorted leading to inaccurate solutions and convergence problems. A possible solution to this problem is local refinement of the problem

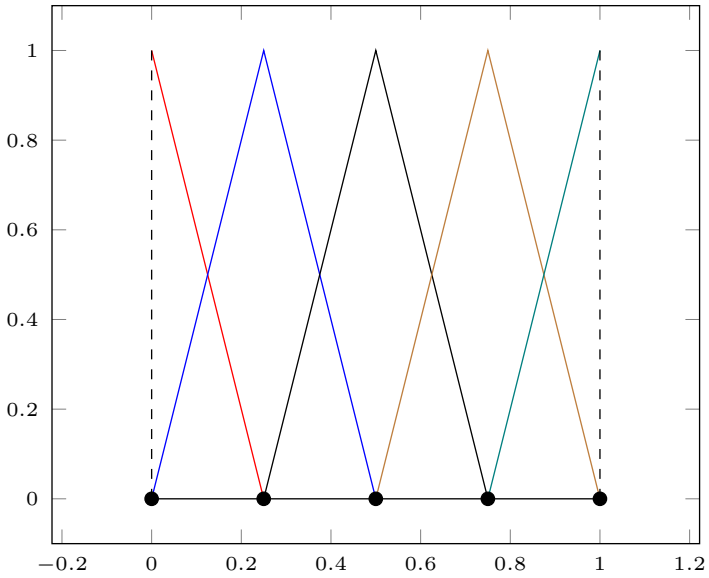


Figure 2.2: Finite element shape functions in one-dimension

area, which may in itself be a difficult problem. Generating constrained mesh representations of complex structures is a difficult prospect in itself. In the best possible circumstances a FEA mesh can be generated automatically, yet in most cases even an automatically generated mesh requires manual editing which can be extremely time consuming and expensive. Mesh generation can take up to 80% of the analysis time in the design process for certain industries [27].

When FEA is used in the CAE-CAD design process, meshes are generated for every design iteration. Developing the initial mesh can be an expensive task, and during the course of the design process the meshing phase can become the most time consuming manual task requirement. This was the motivation behind the development of IGA, which has the intent of using the same representation for the CAD model as is used in the analysis.

When generating FEA meshes from point clouds, the process becomes even more tedious. Multiple steps, and usually software packages, are the norm when developing a mesh from

a point cloud. Generally the initial data has to be processed into something usable by a meshing platform, such as an analytic surface representation e.g. surface tessellation or NURBS surface.

2.4 Isogeometric Analysis

The difficulty in incorporating FEM analysis in the CAD design process lead to the development of IGA [27]. Details of the method are also discussed in [2, 15]. The most common technology for representing geometry in CAD engines is NURBS (Non-Uniform Rational B-Splines), or more recently T-Splines. Traditionally, a FEA mesh is developed from the spline representation for the analysis stage of design. The meshing process is sometimes done completely automatically, yet often manual editing is required for generating a quality FEA mesh. During the course of the design process this becomes an iterative process where each subsequent design iteration requires meshing. The FEA mesh is a piece-wise representation of the continuous CAD model, so intricate features may require many small elements for an accurate representation leading to longer run times in the analysis phase. The idea behind the development of isogeometric analysis is to generate an analysis suitable spline representation, that is use the same basis for defining the geometry and approximating the solution space. Once the geometry is represented in this fashion, even a coarse representation is exactly the same as every other level of refinement's representation.

2.4.1 Fundamental Difficulty

IGA was devised as a solution to problems associated with the CAD based design process, namely the time required to generate FEA model. Hence the ideal usage for IGA is modeling CAD generated domains. In order to model objects that are not CAD drawings with IGA, one must first develop a spline representation of the data. Generating spline representations

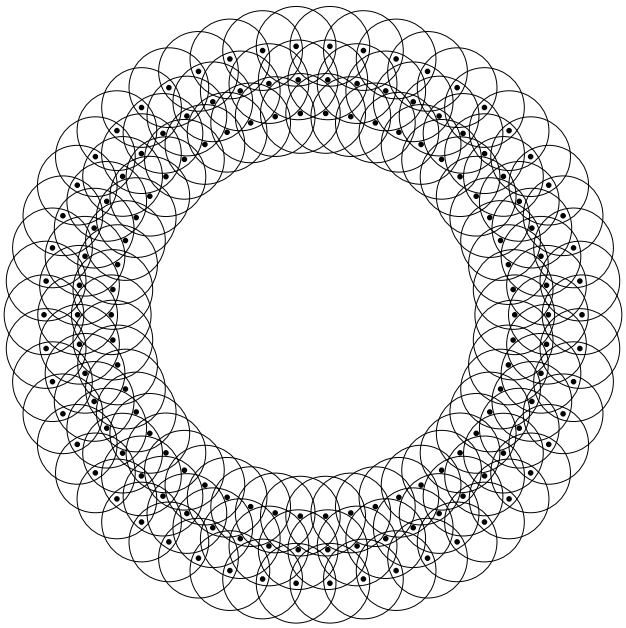


Figure 2.3: An admissible particle distribution with overlapping supports

from generic point clouds is non-trivial and care must be taken to accurately represent some surface features. This will be discussed more in Chapter 6 where the technology for the geometric reconstruction of medical images is talked about in more detail. Another difficulty in using IGA is that the basis functions are rational, not polynomial, so numerical integration error becomes an issue and higher order quadrature routines or special integration methods are required.

2.5 Meshfree Methods

The meshfree approximation of the solution requires a collection of points, commonly referred to as particles or nodes, with associated compact supports that form an admissible particle distribution (APD) [36] as described in Definition 2.5.1. The admissible particle distribution defines the MFA computational domain, in that any point that is "close" to the APD is in the computational domain. Defining what constitutes being "close" to an APD first requires defining the APD itself.

The second property simply means that there can be no regions near the APD that are covered only by compact supports associated with particles that are co-linear in two dimensions or co-planar in three dimension. Now the notion of what is “close" to an APD may be defined by the number of compact supports overlapping at a point, and the MF computational domain is defined by the set of points where $n = dim(\mathbf{P})$ compact supports overlap. Where $\mathbf{P}$ is the polynomial basis for the approximation. Figure 2.3 depicts an admissible particle distribution for a circular domain with overlapping circular supports drawn for each particle.

Definition 2.5.1. *An* admissible particle distribution *is a collection of particles represented by index set* $\mathcal{A} = \{1, 2, ..., N\}$ *with the following properties.*

1. *Each particle is associated with a compact support and the union of the compact supports covers the domain* Ω

2. *The particles must be oriented in a way such that locally they form a non-degenerate simplex.*

The union of the compact supports for an admissible particle distribution bounds the computational, approximate, and intended domains. For example, in two-dimensions with a set of linear monomials with degree 3, there must be at least 3 particle supports covering each spatial point in the domain and the particles corresponding to these supports must be capable of forming a triangle with non-zero area.

An important distinction between the MFA and FEA computational domains is that the MF computational domain is not an accurate portrayal of the intended domain and thus

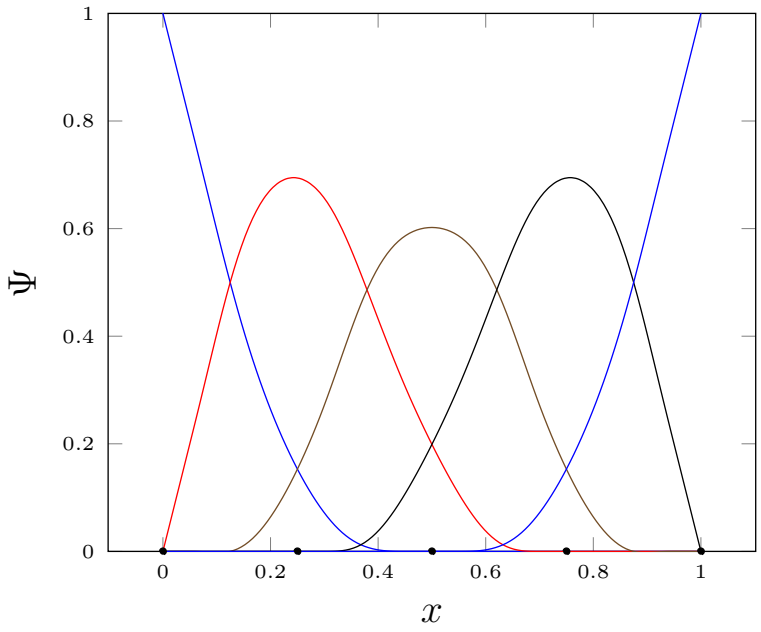


Figure 2.4: Meshfree shape function on one-dimensional domain.

does not serve as an approximate domain. The MFA computational domain varies in size based on the choice of APD and the size of the compact supports, while the FEA mesh conforms to the surface of the intended domain.

2.5.1 Fundamental Difficulty

There are several difficulties associated with meshfree methods, namely the lack of a well defined geometry, application of the boundary conditions, and integration of the weak form. The later two have been researched extensively, and there are various approaches for dealing with the errors associated with each. The lack of a well defined geometric representation is something that has not been tackled directly. The common approach is to use a separate geometric representation along with the MF particle distribution, e.g. FEA mesh with particles at vertices. In many cases though, an ad hoc definition of the surface is used when needed, such as the definition of a surface local to an event such as contact that does not act as a global surface.

The MF shape functions lack the Kronecker-δ property defined in equation 2.9. An example of one-dimensional MF shape functions is depicted in Figure 2.4. This means that the approximation does not pass through the nodes and essential boundary conditions are not automatically satisfied. An excellent overview of the various approaches to enforce essential boundary conditions is given in [19]. Two general approaches have been taken to correctly enforce essential boundary conditions in meshfree Galerkin methods. The first approach is based on a modified weak form, which accounts for the lack of the trial functions to meet the Kronecker-δ property. Methods such as the Lagrange Multiplier method [4], Nitche's method [22], and the penalty method [44] are based on the use of a modified weak form.

The second approach is based on the modification of the meshfree shape functions. A boundary singular kernel was proposed in [28] that would allow the meshfree shape functions to be constructed with the Kronecker-δ property. A method for transforming the MF shape functions on the boundary so that they have the Kronecker-δ property is introduced in [11]. A method coupling meshfree and finite element methods was proposed [6, 25, 43]. A general approach to coupling the meshfree and finite element shape functions coined the continuous blending method was developed by Huerta et. al. [25].

Several techniques have been used for integrating the weak form. Traditionally the most common method has been to develop an underlying grid, or mesh, to define Gauss quadrature routines. There are several difficulties in this alone, namely the MF shape functions are rational functions, since Gauss quadrature exactly integrates polynomial functions there is always error associated with this technique. To accommodate for the error a higher order quadrature rule is typically used leading to higher computational cost. Even with this considered, it was found that the if the underlying grid cells do not align with the compact

supports the method suffers greatly. A background mesh is typically used when a FE model has been developed, but the MF approximation is desired due to large deformations. This of course comes with the additional cost associated with generating the mesh and maintaining a mesh through out the analysis.

Nodal integration was developed in [12], it was found that conforming nodal integration performed particularly well. Conforming nodal integration requires the development of integration cells that align with the Voronoi diagram of the particle distribution. Generating Voronoi diagrams presents the same difficulties as mesh generation; nonconforming nodal integration was developed as a truly meshfree integration technique, where integration cells are arbitrarily chosen for each node. This flexibility comes with great error in the integration, so methods for alleviating the error, called Variationally Consistent Integration (VCI), were developed to correct integration error via strain smoothing [10].

The geometry of point cloud representations is not well defined, and assumptions are made in order to reconstruct the geometry of the intended domain. It is the intention of this dissertation to provide a framework for studying the effects of these assumptions in the scope of MFA. A method is introduced for making geometric assumptions based on properties of the ADP, and will be studied under the guidelines of the framework defined in the next chapter.

Chapter 3: General Framework for Meshfree Geometry

Several conceptual pieces need to be put together to define and formalize MFA geometric representation discussions. In this chapter a framework is described for itemizing these concepts, and some definitions are given make the discussion both clear and precise; further, quantitative metrics for comparing representations are discussed. Within the context of MFA, a larger number of choices of methods, parameters and formulations come into play, each of which may lead to a different geometry. Perhaps in time a particular formulation or set of parameters will become standard, and the geometric description can be simplified. The first step is to lay out the problem, and we do that in three steps

1. Assign clear and concise names to the important pieces of the problem and solution.

2. Explicitly determine of the APD boundary through a set of computable geometry functions.

3. Define a quantifiable metric for assessing performance of a representation.

3.1 Geometry Functions

As noted earlier, a key difficulty in working with APD, and point clouds in general, is the lack of a clear boundary of the domain. To rectify the situation, the work of others is expanded by introducing the meshfree geometry function in Definition 3.1.1. Typically the geometry function, $\mathcal{G}(\mathbf{x}) : \mathbb{R}^3 \mapsto \mathbb{R}$, is such that the level set defined by $\{\mathbf{x}|\mathcal{G}(\mathbf{x}) = 0\}$ is the boundary of the meshfree approximate domain. With the definition of the meshfree geometry

function and the various domain representations, the geometry of particle distribution may be precisely described and discussed.

Definition 3.1.1 (Meshfree Geometry Function)**.** *A* meshfree geometry function*, or simply geometry function, is a function defined on* $\mathbb{R}^n \to \mathbb{R}$ *for an admissible particle distribution that determines whether one is inside or outside the approximate domain,* $\hat{\Omega}_\rho$*.*

3.2 Performance Metric

When discussing various representations, one must be able to quantify how well the representation captures the desired geometry. More precisely, one must be able to quantify how well does $\hat{\Omega}_\rho$ approximate Ω. This is closely related to the problem in Computer Science of shape similarity. In [41] two relevant approaches, the Hamming distance and the Hausdorff distance, are discussed for use on polygons. In the following, the concept of these two distances are generalized for arbitrary domains and spatial dimension. Within the present context, the approximate domain is always a dense point set, but it is necessary to differentiate intended domains that are dense point sets and those that are discrete point sets. The former case arises in intended domains that are described analytically, the second in those coming from imaging systems.

3.2.1 Dense Intended Domains

The distance between dense point sets may be defined in several ways; the measure chosen in this work is how much of the intended domain is not covered by the approximate domain plus how much of the approximate domain is not part of the intended domain. In

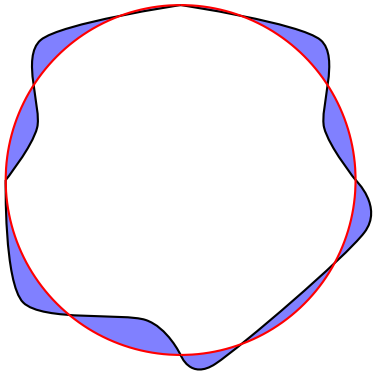


Figure 3.1: The difference set visualized as the shaded region.

set theoretic terms, consider the symmetric difference between the two sets:

$$\overset{\triangle}{\Omega} := \Omega \triangle \hat{\Omega}_\rho = (\Omega \cup \hat{\Omega}_\rho) - (\Omega \cap \hat{\Omega}_\rho)$$

Worth mentioning is that one may find a situation in which only one side of the variation is important, for example one only cares about the portion of the intended domain that is not covered by the approximate domain. Such cases easily fit within the framework with suitable changes to the computation of $\overset{\triangle}{\Omega}$. A difference domain is shown in Figure 3.1 where the red line is the boundary of the intended domain, $\partial\Omega$, the black line is the boundary of the approximate domain, $\partial\hat{\Omega}_\rho$, and the shaded area is the difference domain, $\overset{\triangle}{\Omega}$.

Once the difference set is defined, one can use any number of ways to assign a number to the difference set. In mathematics, this is called a *set function*, [39]. Two set functions of interest are:

- Lebesgue measure, $\mu\left(\overset{\triangle}{\Omega}\right)$, gives the n-dimensional volume of the difference set. Or, one might want the relative volume, $\mu\left(\overset{\triangle}{\Omega}\right)/\mu\left(\Omega\right)$. This is the generalization of the Hamming distance described in [41] to arbitrary shapes and spatial dimension. The

term "Hamming distance" is used in other contexts, such as Information Theory. To avoid confusion, this terminology is avoided and this is referred to as the *difference volume*.

- The Hausdorff distance between two polygons, P_1 and P_2, as defined in [41], is the maximum distance of a point in P_1 from its closest point in P_2. This concept can be generalized to be the maximum distance of a point in $\hat{\Omega}_\rho$ from its closest point in Ω. Since Ω is fixed, this is a set function of $\hat{\Omega}_\rho$. This is not to be confused with the Hausdorff *measure*, and hence will be referred to as the *difference distance*. Unlike problems in character recognition or machine vision, there are not alignment or registration concerns between the two domains in the problems of interest.

3.2.2 Discrete Intended Domains

Point clouds present difficulties when measuring the performance of geometric approximations. By their very nature, point clouds lack a well defined geometry; thus approximating the intended geometry and measuring the representation error requires that assumptions are made. Characteristics of the point cloud and the choice of reconstruction method determine which assumptions are applicable; for instance, the volumetric point clouds derived from medical images are not guaranteed to contain a precise sampling of the actual surface at all, so the boundary is assumed to be near the pixels bounding the region of interest. This is compounded even further when computer vision, or worse yet, human effort, is used to dileanate the boundary of the region of interest. In other cases, such as with point set surfaces, the boundary is either precisely sampled or sampled with noise; often an interpolated or tessellated surface is generated for the point set in the precise case and best fit surface

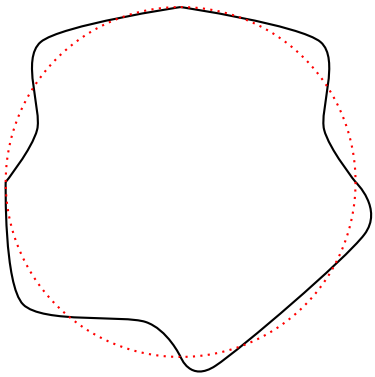


Figure 3.2: Discrete intended domain with potential approximate domain.

is generated for the noisy case. Figure 3.2 shows a discrete point representation of a circle with red points with a potential approximate boundary, the difficulty arises because the area between the solid approximate boundary and the discrete points is not well defined and cannot be directly measured.

In §3.1 the idea of a meshfree geometry function was introduced. The meshfree geometry function is a volumetric, or implicit, representation of the APD. A geometry function that has already been used in conjunction with MFA in the literature is a distance field. In the context of medical image processing, distance fields assume that the representative center point of all pixels bounding the region of interest are surface points. Using this assumption, the distance from any of these surface points is computed and the zero level-set is considered to be the approximate geometry surface. In a method such as this, the accuracy of the approximation can not be determined with the volume difference nor the distance difference; the boundary is exactly fit to the assumed discrete surface leading to the distance difference being zero, and the difference volume is defined thanks to the use of the Lebesgue measure, but it is useless, as demonstrated by Proposition 1.

Thus, the difference volume is the same as the volume of the approximate domain, and hence not useful in assessing how well the approximate domain matches the intended domain. The difference distance may be assessed in more than one way, as the maximum point-wise distance may not provided the best characterization of the error. Instead the boundary points from the intended domain may be classified by the distance from the approximate surface, and the difference distance may be determined by the number of boundary points within a given distance of the approximate surface. This is a common technique used in determining the accuracy of patient specific reconstructions [16].

In the following chapter, a set of geometry functions are introduced that are based on the formulation of MFA. The performance of the method is studied under the guidelines of the framework defined in this chapter, and the methods potential for accurate geometric representation of point clouds and as an ASG are studied.

Proposition 1.

$$\mu(\overset{\triangle}{\hat{\Omega}}) = \mu(\hat{\Omega}_\rho)$$

Proof.

$$\hat{\Omega}_\rho \cup \Omega = (\hat{\Omega}_\rho \triangle \Omega) \cup (\hat{\Omega}_\rho \cap \Omega)$$

We note that we have, for two arbitrary sets, E_1 and E_2 ([39]),

$$\mu(E_1 \cup E_2) + \mu(E_1 \cap E_2) = \mu(E_1) + \mu(E_2),$$

and

$$E_1 \cup E_2 = (E_1 \triangle E_2) \cup (E_1 \cap E_2).$$

Thus,

$$\mu(E_1 \cup E_2) = \mu(E_1 \triangle E_2) + \mu(E_1 \cap E_2), \tag{3.1}$$

since

$$(E_1 \triangle E_2) \cap (E_1 \cap E_2) = \varnothing,$$

by the countable additivity property of measures,[30, 39]. Note that $\hat{\Omega}_\rho \subset \hat{\Omega}_\rho \cup \Omega$, and applying Eqn. 3.1,

$$\mu(\hat{\Omega}_\rho) \leq \mu(\hat{\Omega}_\rho \cup \Omega) = \mu(\hat{\Omega}_\rho \triangle \Omega) + \mu(\hat{\Omega}_\rho \cap \Omega). \tag{3.2}$$

Since, by definition a discrete point set is countable, $\mu(\Omega) = 0$. Therefore,

$$\hat{\Omega}_\rho \cap \Omega \subset \Omega \Rightarrow \mu(\hat{\Omega}_\rho \cap \Omega) \leq \mu(\Omega) = 0.$$

Using this and rearranging around the equal sign in Eqn. 3.2,

$$\mu(\hat{\Omega}_\rho) \leq \mu(\hat{\Omega}_\rho \triangle \Omega) = \mu(\hat{\Omega}_\rho \cup \Omega) \leq \mu(\hat{\Omega}_\rho) + \mu(\Omega) = \mu(\hat{\Omega}_\rho)$$

$$\therefore \mu(\overset{\triangle}{\Omega}) = \mu(\hat{\Omega}_\rho)$$

□

Chapter 4: Meshfree Correction Implicit Geometry

In Chapter 3, a general framework for studying meshfree geometries was presented. Building upon the previous chapters, the MCIG is defined. The MCIG is believed to be the natural way to analytically define the geometry of an admissible particle distribution. Similar to FEA, the MCIG defines the geometry where shape functions exist, except for with FEA this is also coincident with the image of the mesh. With the MCIG defined, the framework for studying meshfree geometry is used to explicitly determine the meshfree boundary and compute the representation error for two simple cases.

The origin of the method's name lies in its development under the purview of the Reproducing Kernel Particle Method (RKPM) [35], which added a correction function to the Smoothed Particle Hydrodynamics (SPH) approximation kernel [20]. The geometry functions for the MCIG were developed based on the ability to solve for the correction function. The correction perspective is unique to the framework of RKPM, but several other meshfree methods such as element-free Galerkin (EFG) [4], and the h-p cloud method [17] also develop moment equations to compute shape functions. So, while the MCIG was developed under the framework of RKPM, it is valid for other MF methods and the meshfree geometry functions should be valid for an appropriate choice of meshfree method.

Before delving into the MCIG geometry functions, the RKPM formulation is developed to provide an understanding of the role of the correction function and the properties of the moment equations. The approach presented here is that of [32], but the resulting moment

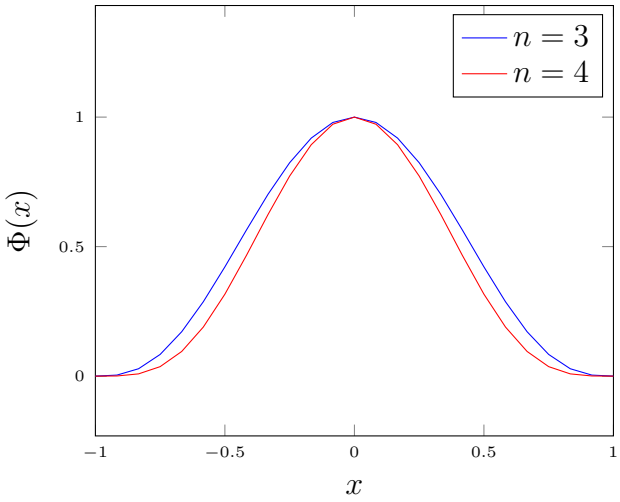


Figure 4.1: Conical kernel functions.

matrix is the same for any of the moment based meshfree methods. Geometry functions based on the moment matrix provide a well defined geometry that is intrinsic to the meshfree approximation and are appropriate for the visualization of MFA results as well as contact modeling.

4.1 Reproducing Kernel Particle Method

RKPM is a Galerkin method that was formulated as a correction to the SPH approximation. The correction alleviated difficulties in SPH such as tensile instability and problems with finite boundaries. The formulation for RKPM was introduced in [34] and is summarized here. Consider the following convolution integral

$$\tilde{u}(x) = \int_{\Omega} \Phi_{\rho}(x - y)\, u(y)\, dy\ , \forall x \in \Omega \tag{4.1}$$

where Φ_{ρ} is a smooth compactly supported kernel functions. This is the approximation used in SPH, which is known to lack completeness. Spline functions, e.g. cubic or quartic splines , are commonly used for the kernel functions. Another type of kernel function that

is commonly used is the conical functions of the form $(1-x^2)^n$, where the function is said to be of order n. Conical function of orders 3,4 are plotted in figure 4.1.

In [34], Liu added a correction term to the approximation to fix consistency:

$$\hat{u}(x) = \int_{\Omega} C_\rho(x-y;x)\Phi_\rho(x-y)u(y)\,dy\ , \forall x \in \Omega \tag{4.2}$$

where the corrected kernel function is defined as

$$\mathcal{K}_\rho(x-y;x) = C_\rho(x-y;x)\Phi_\rho(x-y) \tag{4.3}$$

where C_ρ is the kernel correction function defined as

$$C_\rho(x-y;x) = \mathbf{P}^T(x-y)\,\mathbf{b}(x) \tag{4.4}$$

The vector $\mathbf{P}$ contains the monomial terms of a polynomial basis with degree N, and for the one-dimensional case takes the general form $\mathbf{P}(x) := \{1, x, x^2, x^3, ..., x^N\}$, it is convenient to write the polynomial vector in component form as $\mathbf{P}(\frac{\mathbf{x}-\mathbf{y}}{\rho}) = (\mathbf{x}-\mathbf{y})^i$ for $0 \leq i \leq N$. The vector $\mathbf{b}(x)$, which we call the normalizer, is composed of the coefficient functions which are unknown. To solve for the normalizer, and thus restore consistency to the meshfree approximation,we introduce the Taylor expansion of $u(y)$:

$$u(y) = \sum_{i=0}^{\infty} \frac{(-1)}{i!}(x-y)^i u_{,i}(x) \tag{4.5}$$

where $u_{,i}(x)$ is the i^{th} derivative of u at x. Substituting 4.5 into 4.2 and using the definition of the correction function from 4.4 we obtain:

$$\hat{u}(x) = \bar{m}_0(x)u(x) + \sum_{i=1}^{\infty} \frac{(-1)}{i!} \bar{m}_i u_{,i}(x)\, dy \;, \forall x \in \Omega \tag{4.6}$$

where

$$\bar{m}_i = \int_{\Omega} (x-y)^i \mathbf{P}^T (x-y)\, \mathbf{b}(x) \Phi_\rho(x-y)\, dy \tag{4.7}$$

are referred to as the moments of the kernel function and are constrained to specific values if the equation is to be consistent. Now for the equations to hold we see that $\bar{m}_0 = 1$ and $\bar{m}_i = 0 \; \forall \; i \in \; \{1, N\}$. Rearranging equation 4.7 we get:

$$\int_{\Omega} \mathbf{P}^T (x-y)\, \Phi_\rho(\mathbf{x}-\mathbf{y}) \mathbf{P}\, (x-y)\; dy\, \mathbf{b}(x) = \mathbf{P}(0) \tag{4.8}$$

and the integral term

$$\mathbf{M}(x) = \int_{\Omega} \mathbf{P}^T (x-y)\, \mathbf{P}\, (x-y)\, \Phi_\rho(\mathbf{x}-\mathbf{y})\, dy \tag{4.9}$$

is known as the meshfree moment matrix. In order to ensure a polynomial approximation of degree N, the moment matrix is inverted to solve for the normalizer coefficients. The invertibility of the moment matrix is the defining characteristic used in the MCIG and will be discussed later. The RK approximation functions are defined as

$$\Psi_\rho = \mathbf{P}^T(0)\mathbf{M}^{-1}(x)\mathbf{P}(x-y)\Phi_\rho(x-y) \tag{4.10}$$

and are used in the Galerkin formulation in the same fashion as the FEA shape functions. The formulation can be expanded to higher dimensions by taking a Cartesian product of the kernel function:

$$\Phi_\rho(\mathbf{x}-\mathbf{y}) = \prod_{i=1}^{nsd} \Phi_\rho(x_i - y_i) \tag{4.11}$$

where nsd is the spatial dimension of the domain. It should be mentioned that there exists different methods for extending the kernel function into higher dimensions such as radial kernel functions. The multidimension approximation functions are

$$\Psi_\rho = \mathbf{P}^T(0)\mathbf{M}^{-1}(\mathbf{x})\mathbf{P}(\mathbf{x}-\mathbf{y})\Phi_\rho(\mathbf{x}-\mathbf{y}) \tag{4.12}$$

where the bold font denotes position vectors rather than the scalar coordinates used before.

Following [13] the discrete form of the RKPM approximation is introduced by starting with an admissible particle distribution with nodes $\mathbf{x}_I$ with $I \in \{1, NP\}$ where NP is the number of nodes in the distribution. Now the discrete approximation may be expressed as a summation over the nodes:

$$u(\mathbf{x}) = \sum_{I}^{N} \Phi_I(\mathbf{x}-\mathbf{x}_I) C_\rho(\mathbf{x}-\mathbf{x}_I;\mathbf{x}) w_I u(\mathbf{x}_I) \tag{4.13}$$

with the correction vector defined by the product of the vector of polynomial basis terms and the unknown vector which is computed by inverting the moment matrix.

$$C_\rho(\mathbf{x}-\mathbf{x}_I;\mathbf{x}) = \mathbf{P}^T\left(\frac{\mathbf{x}-\mathbf{x}_I}{\rho}\right)\mathbf{b}(\mathbf{x}) \tag{4.14}$$

where ρ is taken to be the mean value of ρ_I at that location as discussed in [25]. The discrete approximation leads to the discrete moment matrix:

$$\mathbf{M}(\mathbf{x}) = \sum_{I}^{N} \mathbf{P}^T \left(\frac{\mathbf{x} - \mathbf{x}_I}{\rho} \right) \Phi_I(\mathbf{x} - \mathbf{x}_I) \mathbf{P} \left(\frac{\mathbf{x} - \mathbf{x}_I}{\rho} \right) w_I \tag{4.15}$$

which is computed via numerical integration, usually direct nodal integration through a trapezoidal rule where the nodal integration weights are often defined as the representative volume of each particle which may be denoted $w_I = \Delta V_I$. Computing the representative volume of each particle often involves some form of spatial partitioning, such as a Voronoi diagram. Han and Meng [24] showed that the choice of nodal integration weights does not affect the ability to solve for the correction function, as any error in choice will be recovered in the imposition of the consistency conditions.

Going back to Chapter 2, the MF computational domain was defined as all the points "close" to an admissible particle distribution. Here we refine the definition of "close" to be anywhere the moment matrix is invertible. The premise of the MCIG is in using geometry functions based on the invertibility of the moment matrix.

4.2 Candidate Geometry Functions

Of specific interest in this dissertation are geometry functions defined on the moment matrix of the meshfree correction. The moment matrix is, to varying degrees, invertible near the APD and the degree of invertibility can be assigned a value based on matrix properties identified in numerical linear algebra. The following are examples of potential MCIG geometry functions:

$\mathcal{G}1$ Moment matrix determinant

$$\mathcal{G}_{\epsilon}^{det}(\mathbf{x}) := \det(\mathbf{M}(\mathbf{x})) - \epsilon \tag{4.16}$$

for parameter ϵ.

$\mathcal{G}2$ Moment matrix condition number

$$\mathcal{G}_{\epsilon}^{\kappa}(\mathbf{x}) := \kappa^{-1}(\mathbf{M}(\mathbf{x})) - \epsilon \tag{4.17}$$

for parameter ϵ, where κ is the matrix condition number.

$\mathcal{G}3$ Moment matrix singular values/eigenvalues

Let σ_i denote the singular values in the Singular Value Decomposition (SVD) of the moment matrix, $\mathbf{M}$. A class of geometry functions can be defined on the set of singular values. For the L_2 matrix norm the reciprocal condition number can be computed from the maximum and minimum singular values by

$$\kappa^{-1} = \frac{\sigma_{min}}{\sigma_{max}},$$

so the geometry function **$\mathcal{G}2$** can be viewed as a special case of the SVD. Another candidate is

$$\mathcal{G}_{\epsilon}^{SVD}(\mathbf{x}) := \sigma_{min}(\mathbf{M}(\mathbf{x})) - \epsilon \tag{4.18}$$

for parameter ϵ. It is worth mentioning that the moment matrix is symmetric positive definite (SPD) anywhere near an APD, thus the SPD and eigenvalue decomposition are the same.

Now that the meshfree geometry function is defined, the approximate domain for the meshfree particle distribution, which is called the MCIG domain, is introduced in Definition 4.2.1. The MCIG domain fundamentally depends on the MF function space, denoted $\mathcal{F}$, and choice of geometry function, $\mathcal{G}$. The function space depends on choice of MF formulation and various parameters, and the geometry function includes perhaps some other parameters, e.g. ϵ. When it is necessary to be explicit about these choices, a superscript for the function space and subscript for geometry function will be used. If it is unambiguous, no super or subscripts are used. For example, if one were comparing the MCIG for an Element Free Galerkin function space, $\mathcal{E}$, to an RKPM space, $\mathcal{R}$, using the reciprocal condition number geometry function, the MCIG's would be denoted $\tilde{\Omega}^{\mathcal{E}}_{\mathcal{G}^{\kappa}_{\epsilon}}$ and $\tilde{\Omega}^{\mathcal{R}}_{\mathcal{G}^{\kappa}_{\epsilon}}$, respectively.

Definition 4.2.1 (MCIG Domain)**.** *The MCIG domain $\tilde{\Omega}^{\mathcal{F}}_{\mathcal{G}}$ of an admissible particle distribution $\mathcal{A}$, choice of meshfree function space $\mathcal{F}$, and choice of geometry function $\mathcal{G}$, is defined as:*

$$\tilde{\Omega}^{\mathcal{F}}_{\mathcal{G}} := \{\boldsymbol{x} \in \mathbb{R}^n : \mathcal{G}(\boldsymbol{M}) > 0\}.$$

with a boundary defined as the zero level set:

$$\partial\tilde{\Omega}^{\mathcal{F}}_{\mathcal{G}} := \{\boldsymbol{x} \in \mathbb{R}^n : \mathcal{G}(\boldsymbol{M}) = 0\}.$$

4.3 Considerations for Choosing A Geometry Function

Li et.al. [33] used the determinant of the moment matrix to define the MF computational domain in a meshfree contact detection algorithm. The determinant is a natural choice because it is non-zero anywhere the moment matrix is invertible. The authors also showed that the determinant is positive everywhere in the MCIG domain, and as the moment matrix

is evaluated leaving the admissible particle distribution the determinant goes to zero. One possible difficulty in using the determinant is that a nearly zero determinant does not imply a nearly singular matrix. The determinant is zero when a matrix is singular, but when the determinant is close to zero it does not necessarily signal that the matrix is close to being singular, as shown in the Example 4.3.1.

Example 4.3.1. *Consider the identity matrix* $\underline{\underline{\mathbb{I}}}_n$*, the determinant of the identity is* $\det\left(\underline{\underline{\mathbb{I}}}\right) = \prod_i^n \mathbb{I}_{ii} = 1$*. Scaling the identity by some small* ϵ *leads to* $\det\left(\epsilon\underline{\underline{\mathbb{I}}}\right) = \prod_i^n \epsilon\mathbb{I}_{ii} = \epsilon^n$*. Certainly* $\epsilon^n \to 0$ *as* $\epsilon \to 0$*, yet the matrix is still perfectly invertible. For a typical trilinear polynomial basis, the moment matrix has dimension 8. Even a relatively modest value of* $\epsilon = 0.01$ *leads to* $\det\left(\epsilon\underline{\underline{\mathbb{I}}}\right) = 10^{-16}$*, a value on the order of machine epsilon on modern computers.*

For this reason, the use of the determinant is not favored for defining a geometry function. The size of the determinant is based on the size of the elements of a matrix, so for matrices with small elements the determinant becomes small and algorithms involving the determinant may become numerically unstable. A numerically stable measure of singularity for a matrix is the condition number, which is dependent on the relative size of the elements, rather than absolute element size. The condition number is a measure of the stability of the matrix transformation under a small perturbation, and also correlates very closely with how nearly singular a matrix is. The condition number is defined in [21, 40] as

$$\kappa(\mathbf{A}) := ||\mathbf{A}||\,||\mathbf{A}^{-1}||, \qquad \kappa \in [1, +\infty) \tag{4.19}$$

where the best conditioned matrix has $\kappa = 1$, while a singular matrix has $\kappa \to \infty$. It is common to use the reciprocal condition number which ranges from $(0, 1]$ and this is the value

that we choose for the study in this paper. In Example 4.3.2 it is shown that the matrix condition number is invariant when scaling the matrix unlike the matrix determinant.

Example 4.3.2. *Consider the scaled identity matrix* $\epsilon\underline{\underline{\mathbb{I}}}_n$. *Then, using the vector-induced 2-norm,*

$$||\epsilon\underline{\underline{\mathbb{I}}}||_2 = \epsilon,$$

$$||(\epsilon\underline{\underline{\mathbb{I}}})^{-1}||_2 = \frac{1}{\epsilon},$$

and

$$\kappa = 1$$

4.3.1 Analytic Frobenius-norm Condition Number

The reciprocal condition number geometry function can be computed in a number of ways. The condition number itself is based on a matrix norm, of which there are many. Often, the vector-induced 2-norm is chosen, but another good choice is the Frobenius norm which has two nice properties: it is invariant under rotation [21], and it is easily computed. For a matrix $\mathbf{A}$,

$$||\mathbf{A}||_F := \sqrt{\mathbf{A} : \mathbf{A}} = \sqrt{A_{ij}A_{ij}},$$

using this norm the condition number is

$$\kappa_F = ||\mathbf{A}||_F||\mathbf{A}^{-1}||_F = \sqrt{A_{ij}A_{ij}}\sqrt{A^{-1}_{kl}A^{-1}_{kl}}.$$

Since there are explicit expressions for the moment matrix itself, and using the Frobenius norm, an explicit expression for the reciprocal condition number, an explicit formula for the

geometry function and its derivatives can be derived. In Example 4.3.3, the Frobenius norm condition number geometry function is plotted on a one-dimensional domain.

Example 4.3.3. *Particles are uniformly distributed across the domain with spacing* Δx*. To compute the support radius a dilation parameter is applied to the particle spacing* $\rho = \alpha\Delta\boldsymbol{x}$*, in this example we used* $\alpha = 1.4$*. The particle integration weights were assigned for interior particles as* Δx *and the boundary particles as* $\frac{1}{2}\Delta x$*. A linear polynomial basis was used of the form* $\mathbf{P} = \{1, x\}$ *to compute the Frobenius norm reciprocal condition number geometry function,* $\mathcal{G}_{\kappa_F}$*, with* $\epsilon = 0.065$*. The results are plotted in Figure 4.2 with the particle distribution plotted as black points on the x-axis. Notice that the choice of* ϵ *makes the functions go through 0 near the boundary particle.*

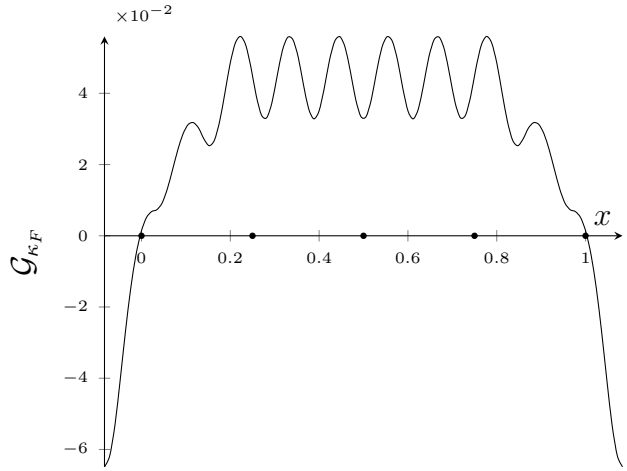


Figure 4.2: Frobenius norm induced geometry function on one-dimensional domain.

4.3.2 Choices in Geometry Function Parameter

The approximate domain, $\hat{\Omega}_\rho$, is defined by a superlevel set of the geometry function. The surface of the approximate domain, $\partial\hat{\Omega}_\rho$, is a level set of the geometry function which is defined by the choice in ϵ. With knowledge of the desired surface location the parameter may be computed by minimizing one of the error metrics discussed in Chapter 3.

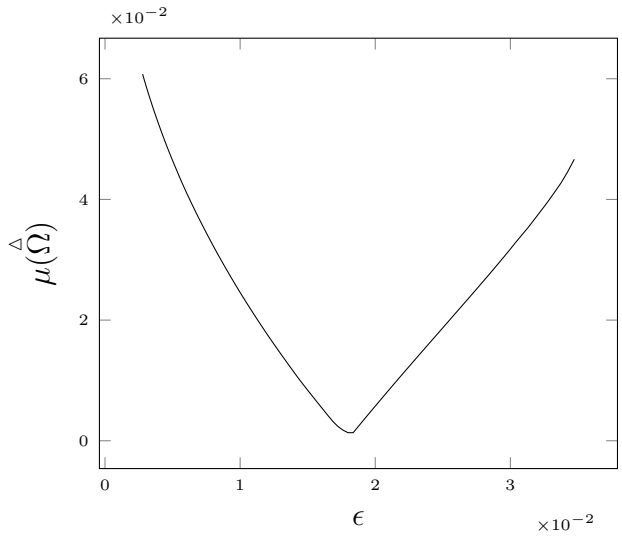


Figure 4.3: Volume difference as a function of ϵ.

Earlier it was determined that when the exact domain is known, an appropriate error measure would be the volume difference $\mu(\overset{\triangle}{\Omega})$. The problem then becomes find ϵ such that $\mu(\overset{\triangle}{\Omega})$ is minimized. An example plot of the volume difference as a function of ϵ is shown in Figure 4.3 for Example 4.3.3. For problems where the exact domain is unknown and the only information comes from a discrete sampling a different approach may be considered. The correction function could be sampled at the surface sample points and a choice of ϵ can be made based these values.

4.4 Numerical Examples

The properties of the MCIG approximation will now be discussed and explored. For all examples, the RKPM function space, denoted $\mathcal{R}$, will be employed which explicitly depends on choice of kernel function, polynomial basis, and support size.

The support size ρ, or smoothing length, is the characteristic length scale of the discretization, it plays a similar role as the element size in a finite element mesh. The smoothing length is closely related the spacing of the particles, but given the overlapping nature of the kernel

functions the smoothing length is arbitrarily larger than the particle spacing. A common approach when discussing convergence is to consider the normalized smoothing length, which takes into account how much larger the smoothing length is than the particle spacing. A larger normalized support size results in more particles acting at a given point, resulting in a smoother solution. The same action applies to the MCIG; larger normalized support sizes result in the smoother surface representations. The MCIG surface with smaller support size may become oscillatory; as the influence of particles changes more rapidly and the condition of the moment matrix changes. This effect is due to the nodal integration of the moment matrix. It was shown that the interior oscillations depicted in Figure 4.2 from Example 4.3.3 would vanish if the moment matrix were exactly integrated [24].

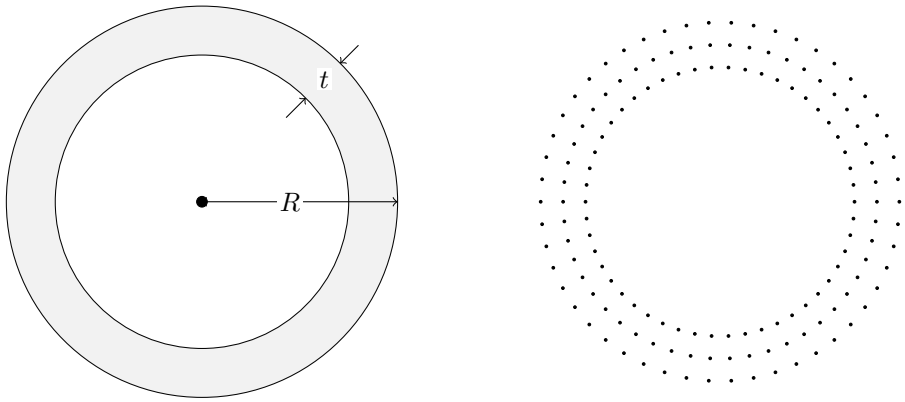
Studying the convergence of the MCIG approximation requires observing two aspects of the particle distribution; the particle density and the support size. For the following work the particle density will not be considered directly, but a parameter that defines the spacing between particles, represented by the symbol $\hat{\rho}$ will be used. The nodal spacing is constant for uniformly spaced particles; for particle distributions that are not uniform a nodal spacing that is characteristic of the distribution may be used. Due to the requirements for an admissible particle distribution the support size is always greater than the nodal spacing. The support size can be decomposed in the form $\rho = \alpha\hat{\rho}$, where α is called the normalized support size.

Error measures were defined in Chapter 3 for both dense and discrete intended domains. The examples in this section are for shapes with analytic expressions; therefore, the volume difference, $\mu(\overset{\triangle}{\Omega})$, is the appropriate error measure. Though there are instances where the distance difference may be used to illustrate a point.

For a given polynomial basis and kernel order, the error in the geometry approximation can be plotted as a function of the particle density ($\hat{\rho}$) and the normalized support size (α). The result is an error surface that depicts the behavior of the geometry as the particle distribution is refined and the normalized support size grows. With the error surface defined the convergence of the approximation is defined by the path through the minimum of the error surface, i.e. for a given nodal spacing the minimum error corresponds to the normalized support size that results in the most accurate geometry.

In this section some of the characteristics of the MCIG are explored via numerical example; the variations of the geometry due to the function space are studied and categorized. The performance of the MCIG approximation is evaluated for two simple cases, a ring shaped domain and a rectangular domain. The ring shape was chosen due to it's smooth circular surfaces; one being the locally convex outer circle and the other being the concave inner circle. The rectangular domain was chosen to study how well the MCIG approximation represents straight lines and sharp corners. The moment matrix is continuous to the same order as the kernel function, in turn the MCIG surface is continuous and can not precisely represent the sharp corners found on the rectangular domain. In the future, it may be possible to define a geometry function that captures sharp features, but for now the best that can be hoped for is that the approximation approaches the shape of the corner under particle refinement.

The moment matrix is composed of a vector of polynomial basis functions and a kernel function. It is common to use linear and quadratic polynomial basis for the approximation; so a linear basis of the form $\mathbf{P} = \{1, x, y\}$ and a quadratic basis of the form $\mathbf{P} = \{1, x, y, xy, x^2\}$ are used in the examples. For the kernel function, order 3 and 4 conical kernel functions are used, the conical kernel function is defined in Section 4.1 and shown in Figure 4.1.



(a) Dimensions (b) Particle distribution with 150 particles.

Figure 4.4: Ring shaped domain.

The MCIG approximation is studied with a single geometry function, the Frobenius norm reciprocal condition number induced geometry function, $\mathcal{G}_{\kappa_F}$.

4.4.1 Ring Shaped Domain

The setup for the ring shaped domain is shown in Figure 4.4; the outer radius of the ring is chosen as $R = 1.0$ and $t = 0.25$ is chosen for the thickness of the ring. The ring was discretized so that particles were generated radially at an increment of Δr and azimuthally at an increment of $\Delta\theta$ so that $\Delta r \approx r\Delta\theta$, resulting in a nodal spacing that is roughly uniform in both directions. The coarsest particle discretization that can be achieved, while having particles through the thickness and following the discretization parameters defined, is shown is 4.4b.

This ring shape was selected in order to study the MCIG representation error for a smooth boundary. Another point of interest is that the ring shape consists of two boundaries; the outer boundary is locally convex with respect to the particle distribution while the inner boundary is concave with respect to the particle distribution. The concave boundary becomes

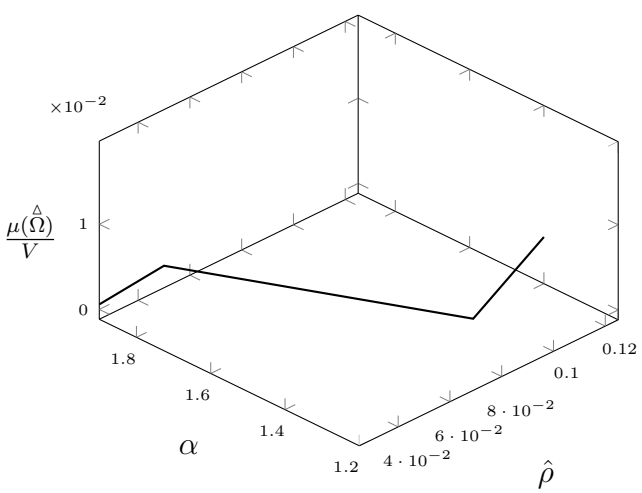
Table 4.1: Minimum error in MCIG approximation for ring shaped domain with coarsest discretization ($\hat{\rho} = 0.125$).

Polynomial Basis	Kernel Order	α	$\mu(\overset{\triangle}{\Omega})$
Linear	3	1.4	0.00043
Linear	4	1.5	0.0035
Quadratic	3	2.5	0.0069
Quadratic	4	2.6	0.0042

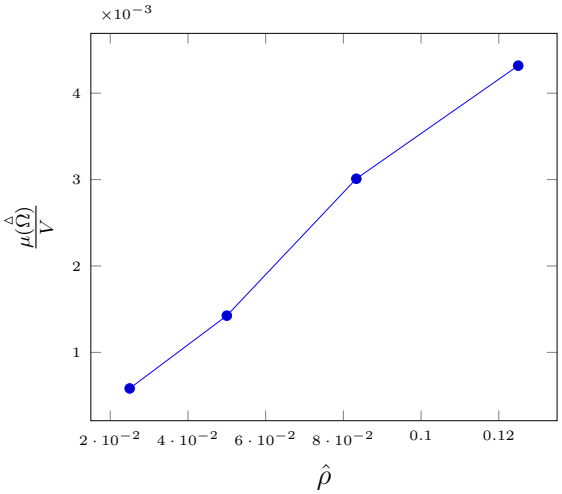
an issue if t were allowed to grow until the particle supports overlapped the empty interior space; methods are discussed later in the chapter for handling just such a case.

The minimum error for the coarsest particle distribution, ($\hat{\rho} = 0.125$), is shown in Table 4.1 for each of the possible combinations of polynomial field and kernel function order. The table also depicts the normalized support size for each of these particle distributions. It is expected that the approximation error is reduced with particle refinement, thus the coarsest distributions are expected to result in the worst case representations, but it is worth noting that the even for the worst case scenario the maximum error was roughly 1.8% of the volume of the intended domain.

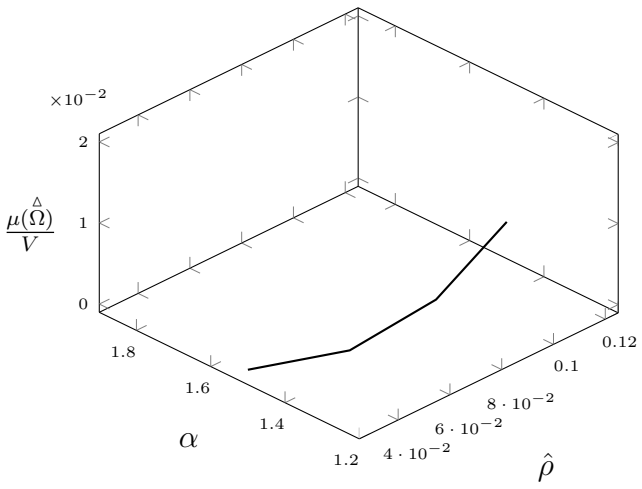
In Figures 4.5 and 4.6 the error surface is plotted next to the minimum error plots. The error surface for the ring shaped domains shows an optimal range for the normalized support. It appears that with a small normalized support the surface is rough, resulting in larger representation error. As the normalized support grows, the interior particles have a greater influence and distorts the ring. Figure 4.7 shows the difference in approximations for a small normalized support versus a large normalized support. The error for the small normalized support is evident in the rough surface as seen in Figure 4.7a. For the large normalized support the error is subtle, but the if one were to closely examine Figure 4.7b



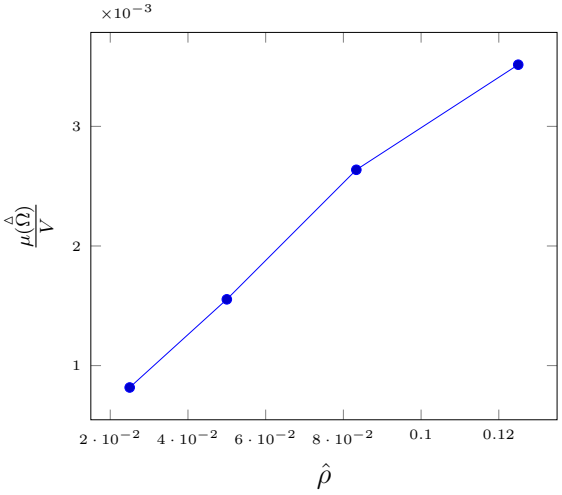
(a) Order 3 kernel



(b) Order 3 kernel

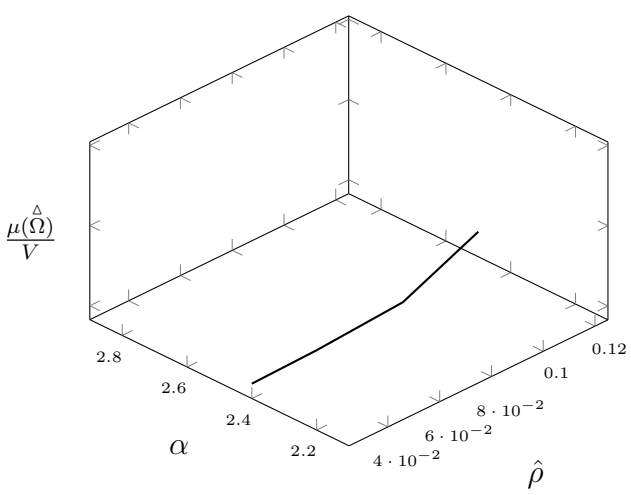


(c) Order 4 kernel

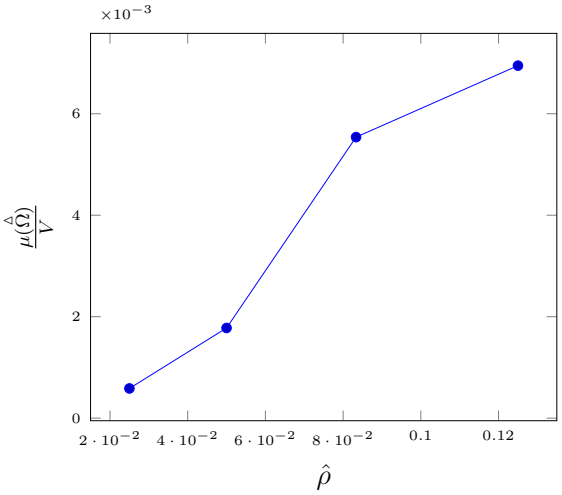


(d) Order 4 kernel

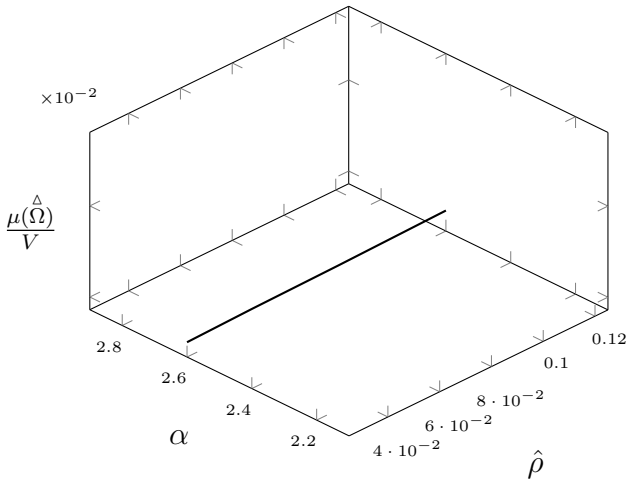
Figure 4.5: MCIG approximation error with linear polynomial basis for ring domain.



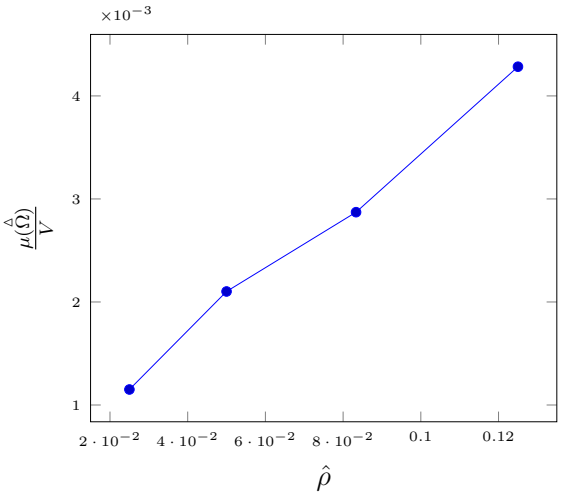
(a) Order 3 kernel



(b) Order 3 kernel



(c) Order 4 kernel



(d) Order 4 kernel

Figure 4.6: MCIG approximation error with quadratic polynomial basis for ring domain.

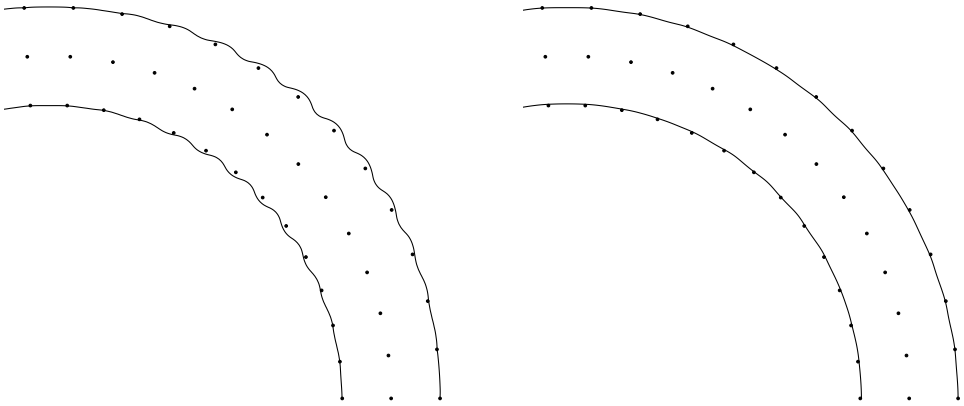
they would see the boundary go through the particle on the x-axis and it pulled very slightly inward towards the origin as it moves away from the x-axis.

For all of the cases so far the particles have been aligned with the surface outward normal through the thickness of the ring. The effects of shifting the particles out of this orientation are shown in Figure 4.8. In Figure 4.8a the interior particles are shifted azimuthally by half the distance to the next node. This result is in contrast to boundary seen in Figure 4.7a. In the next case the particles are shifted randomly along the circle they were generated on by 20% of the nodal spacing. The boundary resulting from the random shift behaves in a much less predictable manner than the uniform or interior shifted distributions.

The error plots for the shifted particle distributions are shown in Figure 4.9. The case with interior particles shifted by a half step shows many similarities to the unshifted case. The randomly shifted particles did not perform as well, which was expected. The error surface depicted is not as well defined as the other cases and the error is a full order of magnitude worse than any other case studied. Still, the random case converges as shown in Figure 4.9.

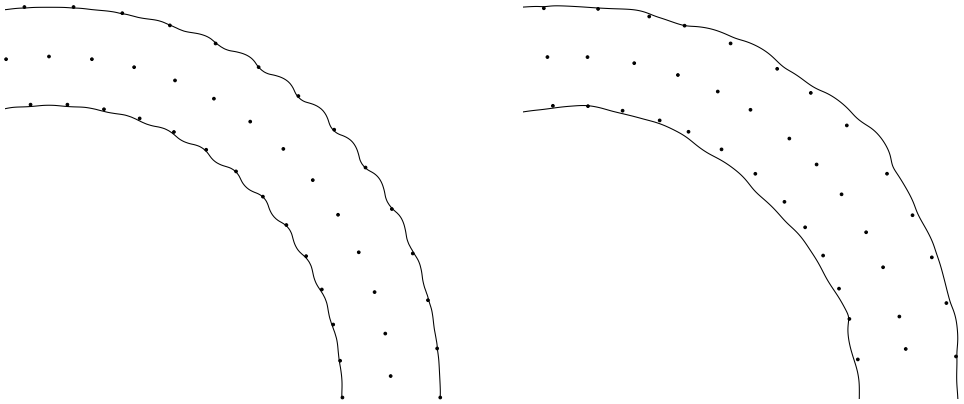
4.4.2 Rectangular Domain

The setup for the rectangular domain is shown in Figure 4.10. The length of the rectangular domain was $L = 1.0$ and $H = 0.5$ was used for the height as shown if Figure 4.10a. The particles were placed uniformly through the domain as shown in Figure 4.10b. The minimum error for the coarsest particle distribution,($\hat{\rho} = 0.3$), is shown in Table 4.2 for each of the possible setups. The table also depicts what the normalized support (α) was for each of the particle distributions. The MCIG approximation for the rectangular domain is best for all case when the normalized support size is at a minimum. This is in contrast to the



(a) $\alpha = 1.2$ (b) $\alpha = 1.8$

Figure 4.7: MCIG approximation for coarsest node spacing for ring domain.

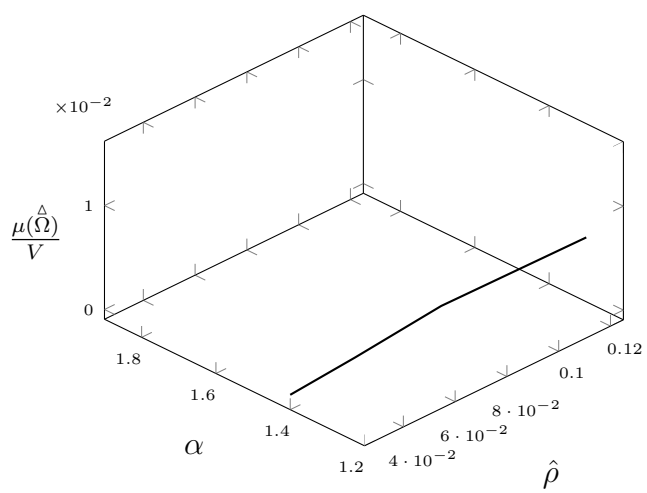


(a) Center particles shifted by half step. (b) All particle randomly shifted 20% of nodal spacing.

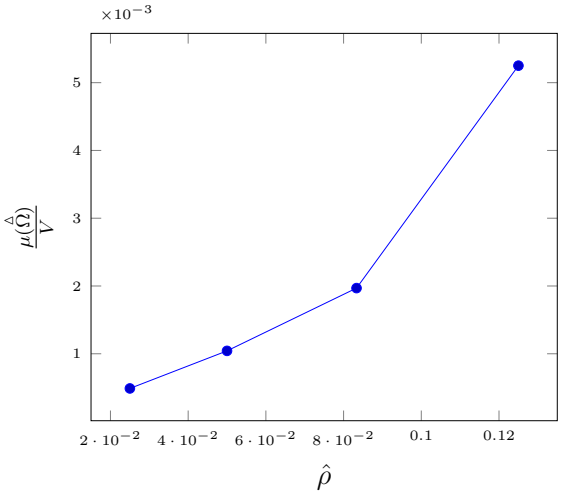
Figure 4.8: MCIG boundary for ring domain with particles shifted.

Table 4.2: Minimum error in MCIG approximation for rectangular domain with coarsest discretization.

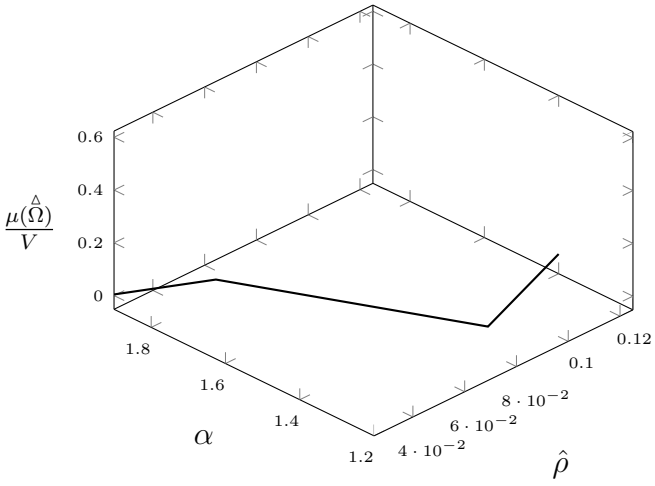
Polynomial Basis	Kernel Order	α	$\mu(\overset{\triangle}{\Omega})$
Linear	3	1.1	0.00418992
Linear	4	1.1	0.00310499
Quadratic	3	2.1	0.00040572
Quadratic	4	2.1	0.000287292



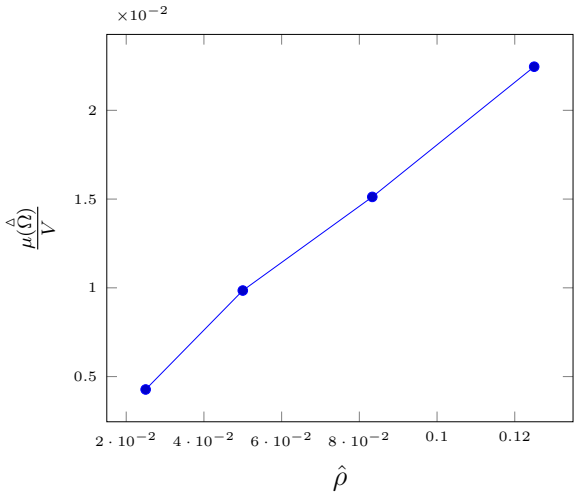
(a) Interior particles shifted by half step.



(b) Interior particles shifted by half step.

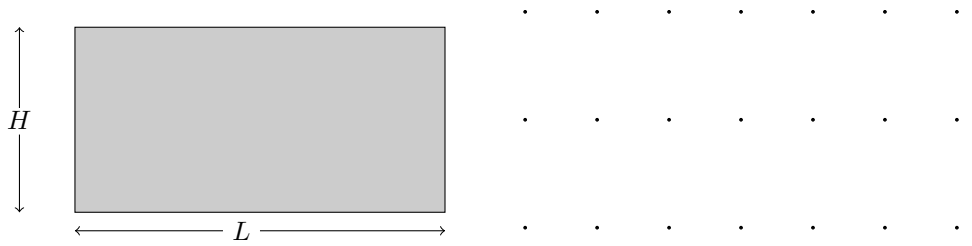


(c) All particles randomly shifted 20%.



(d) All particles randomly shifted 20%.

Figure 4.9: Error convergence for ring domain with non-uniform particle distributions.



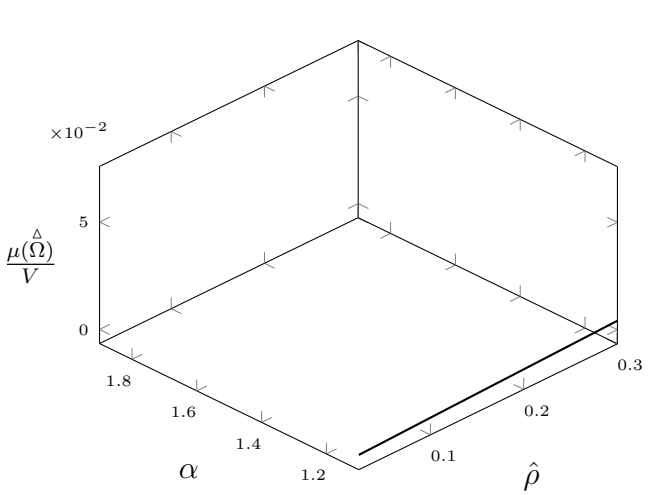
(a) Rectangle dimensions (b) Particle distribution

Figure 4.10: Rectangular domain.

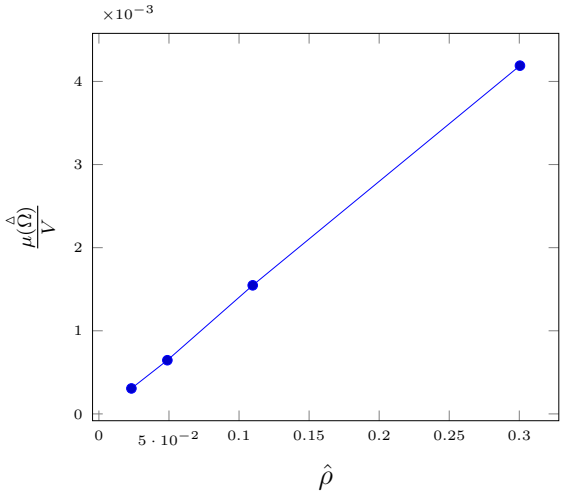
results for the ring shaped domain, which reinforces the notion that larger support sizes are equated with smoother geometry. In Figures 4.11 and 4.12 the error surface is plotted next to the minimum error plots. The error surfaces show that sharp features are best represented with small normalized supports and a highly refined particle distribution.

4.5 Non-Convex Features

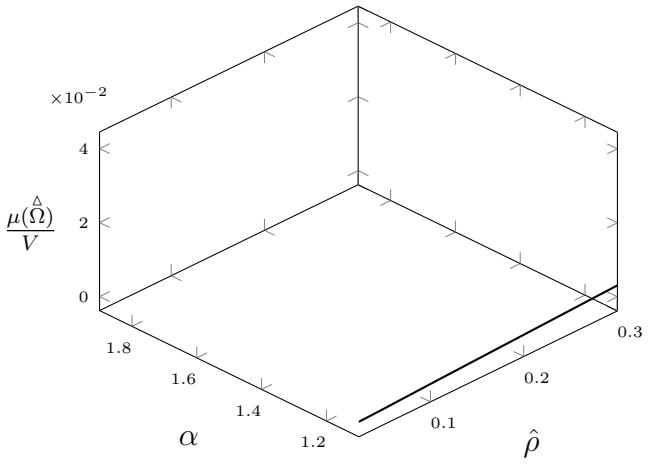
Non-convex surface features which are smaller than the nearby support size pose a problem for the MCIG approximation. The support size allows the particles to have an influence across space that is not covered by the domain. This is a well known problem and was solved with the development of the visibility criterion [5]. The visibility criterion signals a region beyond which a particle can not have an influence, and the typical implementation does a line of sight check to determine if a particle is blocked by a visibility condition. In two-dimensions a visibility condition is typically defined by a line segment, but it could very well be defined by any shape that can be intersected with a line segment. The visibility criterion is known to cause discontinuities because it is strictly based on line of sight. This



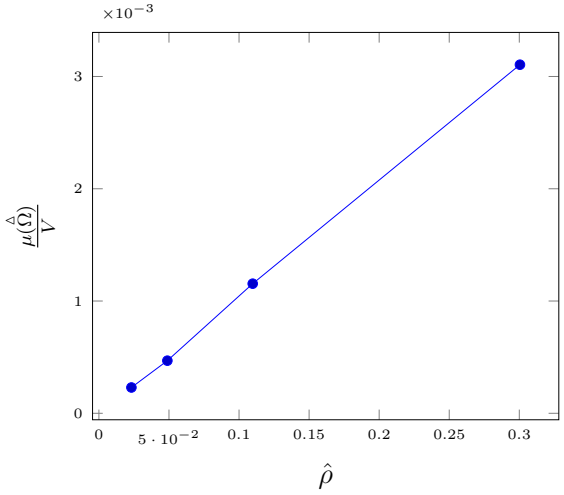
(a) Order 3 kernel



(b) Order 3 kernel

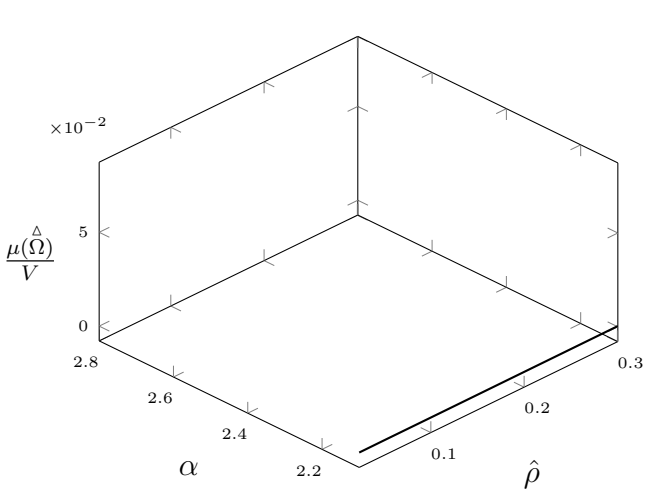


(c) Order 4 kernel

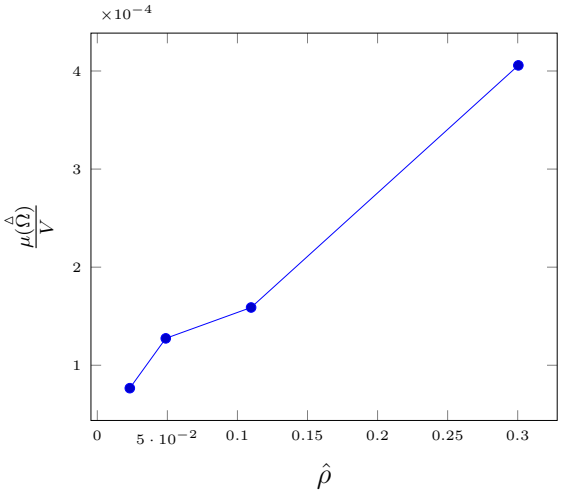


(d) Order 4 kernel

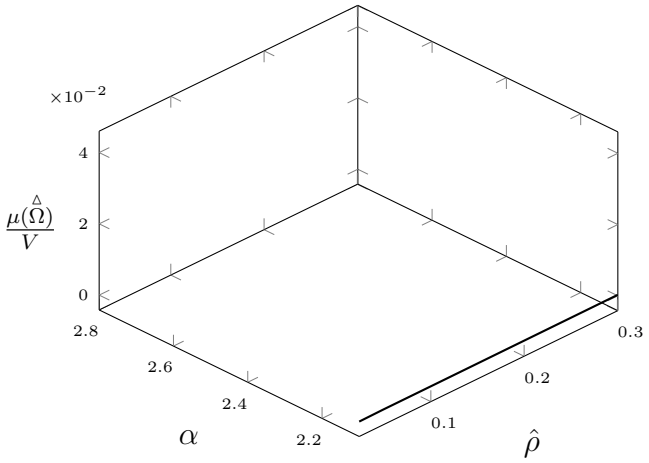
Figure 4.11: MCIG approximation error with linear polynomial basis for rectangular domain.



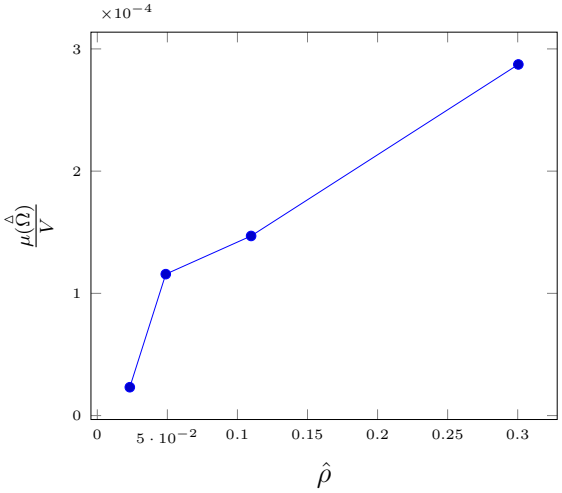
(a) Order 3 kernel



(b) Order 3 kernel

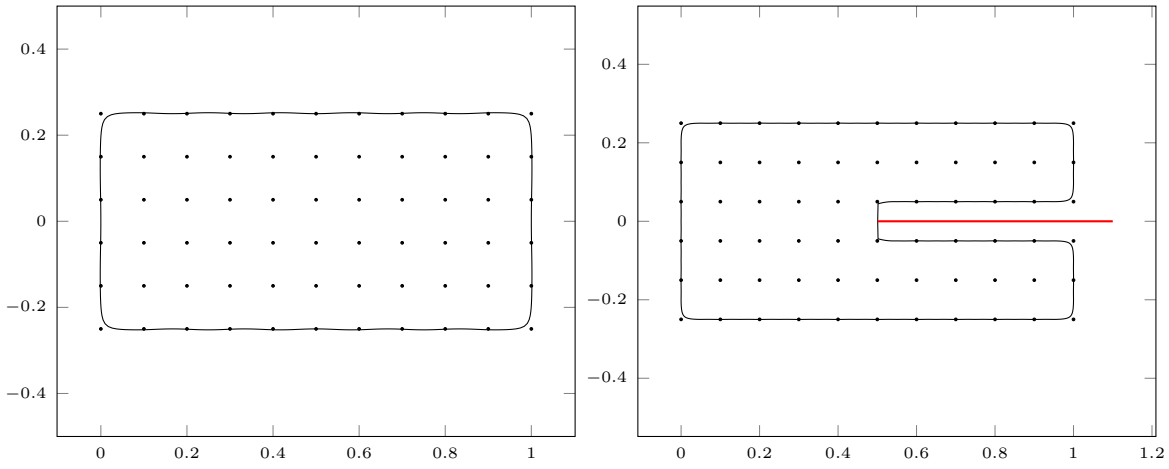


(c) Order 4 kernel



(d) Order 4 kernel

Figure 4.12: MCIG approximation error with quadratic polynomial basis for rectangular domain.



(a) No visibility panel.

(b) Visibility panel placed in slot.

Figure 4.13: Example of visibility panel placed in slot in rectangular domain.

problem was addressed by developing more sophisticated visibility type criterion known as diffraction and transparency conditions [37]; these methods allowed continuous approximations near features such as crack tips by allowing particle interactions to occur if a path could be traced from the particle to any point in it's associated support even if the direct line of sight was blocked by a visibility panel.

As an example take a rectangular domain with a slot cut out of one side that is smaller than the support size of the particles lining the wall of the slot. As shown in Figure 4.13a, the meshfree kernels near the slot will convey information across the slot causing the MCIG approximation to behave as if the slot does not exist. With the proper placement of a visibility panel, as in Figure 4.13b, the MCIG approximation is forced to go around the panel. As mentioned earlier, the visibility criterion is known to cause discontinuities and so it is with the MCIG approximation. The discontinuities express themselves as sharp features in the boundary. The interior corners of the slot depicted in Figure 4.13b show signs of the sharp discontinuity.

Chapter 5: MCIG as an Analysis Suitable Geometry

The lack of an analytic surface definition for meshfree particle distributions has led to the development of ad hoc contact methods such as the natural kernel contact method [14] which defines level-set function based on the interaction of particles from contacting bodies. The condition of the moment matrix has been used in a meshfree contact detection method [33]. The invertibility of the moment matrix was tested to determine when two bodies were close to contacting. In the framework outlined in this dissertation the contact method detected when a body was in the computational domain of another, the method referred to this region outside of the approximate domain as the buffer zone. The buffer zone was used in the method to regularize the action of the contact.

In this chapter the efficacy of the MCIG approximate domain as an analysis suitable geometry is studied by solving contact driven problems of objects impacting a rigid wall. In the first example the object is a steel bar and the second is a rubber ring. In both cases the MCIG approximation is compared to a FE approximation of the problem.

The example problems are solved in the Total Lagrangian (TL) form, which is derived after introducing the BVP. The contact constraints are then introduced along with a penalty contact method that is used to solve both problems. After that a solution strategy is discussed followed by the numerical examples.

5.1 Equations of Motion

The conservation of momentum is stated as:

$$\begin{cases} \nabla \cdot \boldsymbol{\sigma} + \rho \mathbf{b} = \rho \ddot{\mathbf{u}} \\ \boldsymbol{\sigma} \cdot \mathbf{n} = \mathbf{t} \text{ on } \Gamma_t \\ \bar{\mathbf{u}} = \mathbf{g} \text{ on } \Gamma_g \end{cases} \tag{5.1}$$

where $\boldsymbol{\sigma}$ is the Cauchy stress, $\mathbf{b}$ is the body force per unit mass. The boundary conditions are the surface traction vector, $\mathbf{t}$ and the applied displacements $\mathbf{u}$. This governing equation is for deformation of a solids [26].

5.1.1 Total Lagrangian Weak Form

The Total Lagrangian point of view is one in which Lagrangian measures of stress and strain are used and derivatives and integrals are taken with respect to the material coordinates as opposed to the spatial coordinate [3]. We will denote everything in the reference configuration (material) with a subscript 0, e.g. the domain in the reference configuration is Ω_0. The Cauchy stress is not a Lagrangian stress measure, so we substitute the Cauchy stress with:

$$\boldsymbol{\sigma} = J^{-1} \mathbf{F} \cdot \mathbf{P} \tag{5.2}$$

where $\mathbf{F}$ is the deformation gradient tensor, $J = det(\mathbf{F})$ and $\mathbf{P}$ is the nominal stress tensor. It is convenient at this point to switch to indicial notation with the usual conventions. The strong form is multiplied by an arbitrary test function δu and integration is done over the

reference configuration:

$$\int_{\Omega_0} \delta u_i \left(\frac{\partial P_{ji}}{\partial X_j} + \rho_0 b_i - \rho_0 \ddot{u}_i \right) d\Omega_0 = 0. \tag{5.3}$$

Next the derivative of the nominal stress is eliminated, the derivative product results in:

$$\int_{\Omega_0} \delta u_i \left(\frac{\partial P_{ji}}{\partial X_j} \right) d\Omega_0 = \int_{\Omega_0} \frac{\partial}{\partial X_j} \left(\delta u_i P_{ji} \right) d\Omega_0 - \int_{\Omega_0} \frac{\partial (\delta u_i)}{\partial X_j} P_{ji} \, d\Omega_0. \tag{5.4}$$

The first term of the right hand side can be expressed as a boundary integral by Gauss's theorem:

$$\int_{\Omega_0} \frac{\partial}{\partial X_j} \left(\delta u_i P_{ji} \right) d\Omega_0 = \int_{\Gamma_0} \delta u_i \, n_i^0 \, P_{ji} \, d\Omega_0. \tag{5.5}$$

The variation of the deformation gradient is defined as:

$$\delta F_{ij} = \delta \left(\frac{\partial u_i}{\partial X_j} \right) = \frac{\partial (\delta u_i)}{\partial X_j} \tag{5.6}$$

after substitution:

$$\int_{\Omega_0} \left(\delta F_{ij} P_{ji} - \delta u_i \rho_o b_i + \delta u_i \rho_0 \ddot{u}_i \right) d\Omega_0 = 0 \tag{5.7}$$

or in tensor notation:

$$\int_{\Omega_0} \delta \mathbf{F}^T : \mathbf{P} \, d\Omega - \int_{\Omega_0} \mathbf{b} \cdot \delta \mathbf{u} \, d\Omega + \int_{\Gamma_0} \mathbf{t} \cdot \delta \mathbf{u} \, d\Gamma + \int_{\Omega_0} \rho \, \ddot{\mathbf{u}} \, \delta \mathbf{u} \, d\Omega = 0. \tag{5.8}$$

Equation 5.8, along with the boundary conditions from 5.1, is the principle of virtual work in the total Lagrangian form.

5.1.2 Nodal Forces

Examining the principle of virtual work in Equation 5.8 reveals two terms that represent the work the forces do on the body. The internal virtual work is due to the internal forces (stress) in the body and the external virtual work is due to the traction (external forces) acting on the surface.

The internal virtual work is:

$$\delta w^{int} = \delta u_{iI} f_{iI}^{int} = \int_{\Omega_0} \delta F_{ij} P_{ji} \, d\Omega_0 = \delta u_{iI} \int_{\Omega_0} \frac{\partial N_I}{\partial X_j} P_{ji} \, d\Omega_0 \tag{5.9}$$

where considering the arbitrariness of δu_{iI} leads to the internal forces:

$$f_{iI}^{int} = \int_{\Omega_0} \frac{\partial N_I}{\partial X_j} P_{ji} \, d\Omega_0. \tag{5.10}$$

Next, the external virtual work is:

$$\delta w^{ext} = \delta u_{iI} f_{iI}^{ext} = \int_{\Omega_0} \delta u_i \rho_0 b_i \, d\Omega_0 + \int_{\Gamma_0^t} \delta u_i t_i \, d\Gamma_0^t = \delta u_i \int_{\Omega_0} \rho_0 b_i \, d\Omega_0 + \int_{\Gamma_0^t} t_i \, d\Gamma_0^t \tag{5.11}$$

then again due to the arbitrariness of δu_{iI} the external nodal force is defined as:

$$f_{iI}^{ext} = \int_{\Omega_0} \rho_0 b_i \, d\Omega_0 + \int_{\Gamma_0^t} t_i \, d\Gamma_0^t. \tag{5.12}$$

From the inertial term the lumped mass matrix can be defined as:

$$M_{IJ} = \int_{\Omega_0} \rho_0 N_I N_J \, d\Omega_0. \tag{5.13}$$

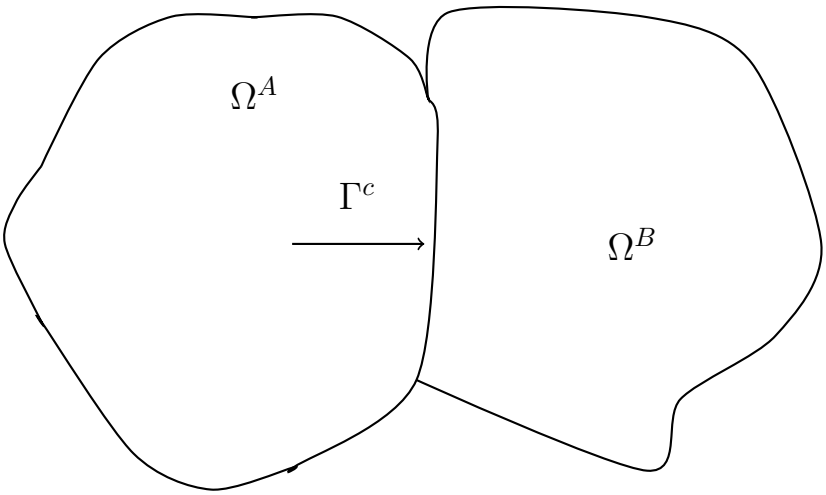


Figure 5.1: Diagram of contacting bodies.

5.1.3 Contact

In contact modeling the impenetrability condition must be observed:

$$\Omega^A \cap \Omega^B = \emptyset \tag{5.14}$$

where the domains Ω^A and Ω^B are shown in Figure 5.1. The impenetrability condition simply states that the contacting bodies can not overlap. This is a necessary condition for contact in reality, yet it is impractical to eliminate all domain intersections in a numerical approximations [3]. Instead it is common to allow some penetration when simulating contact. There are several methods for introducing the contact constraints into the weak form. In this work a penalty contact constraint is added to the weak form:

$$f^c = \int_{\Gamma_c} \beta \theta_n \, d\Gamma \tag{5.15}$$

where f^c is the contact force, β is the penalty parameter, and θ_n is the penetration gap. The contact force is integrated over the contact boundary Γ_c which is shown in Figure 5.1. The

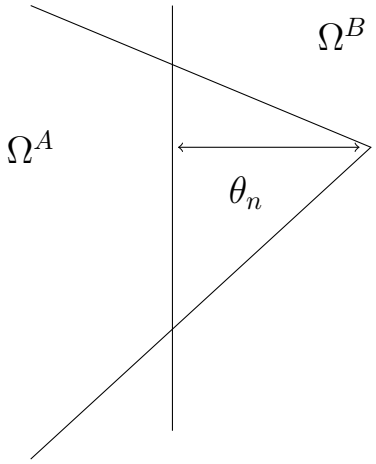


Figure 5.2: The penetration gap for contacting bodies.

use of a penalty method acts as a regularization procedure, meaning the discontinuity in the nodal velocities due to impact is smoothed over a slightly longer period of time.

5.2 Solution Strategy

A central difference method is used for the time integration of all the examples. Explicit time integrators require smaller time steps for stability when compared to implicit time integrators such as Newmark's method, but are much more robust when modeling events such as contact. The stable time step for is given by:

$$\Delta t = \min_e \frac{l_e}{c_e} \tag{5.16}$$

where l_e is the characteristic length of the element and c_e is the element wave speed. The flow chart in Figure 5.3 outlines the strategy for solving the time dependent contact problem.

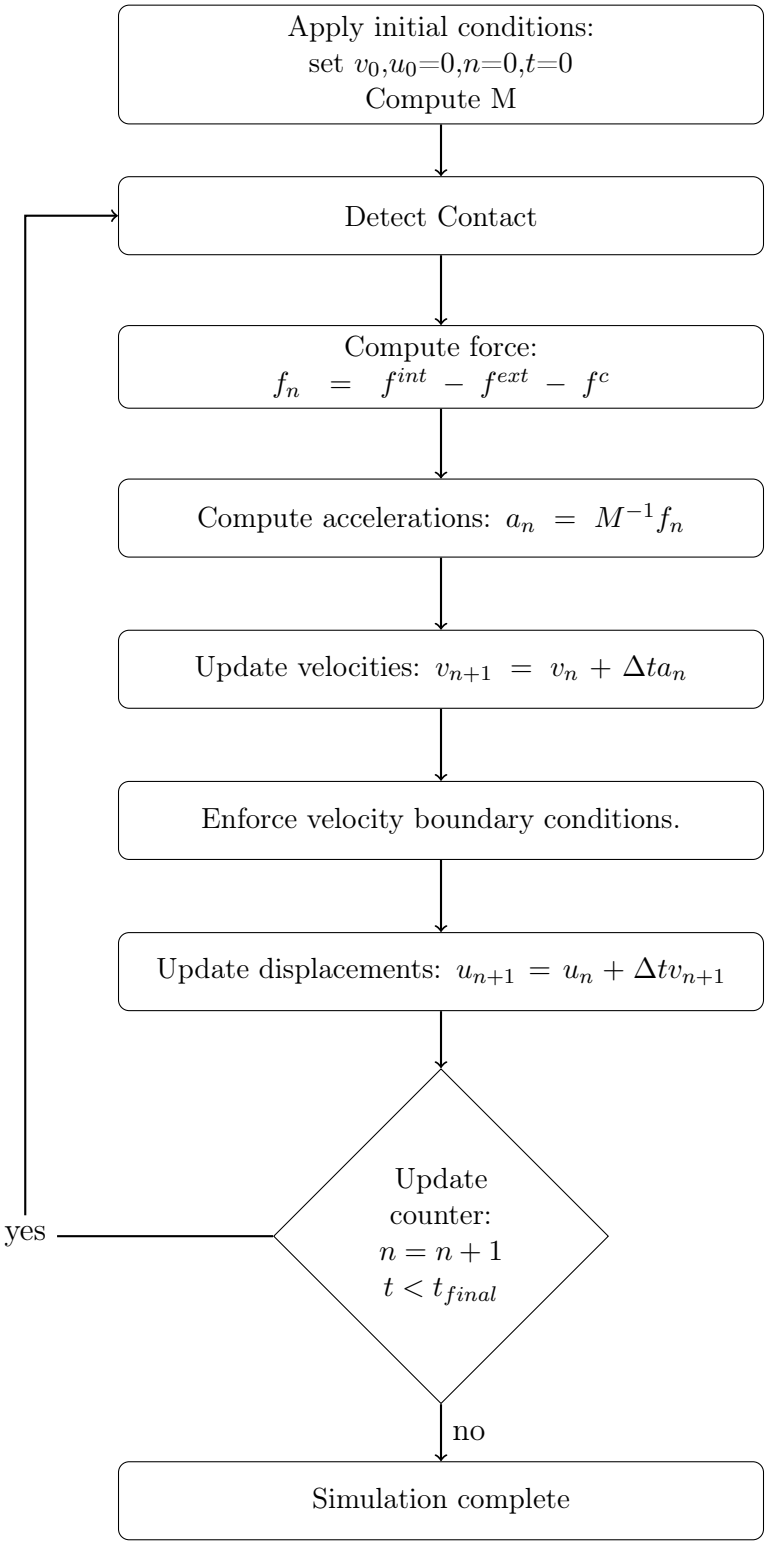


Figure 5.3: Flow chart for solution of contact problem.

5.3 Numerical Examples

In the following two examples, the results for a finite element model are compared to results from an RKPM model with an MCIG approximate domain. The weak form is integrated using Gauss quadrature for both methods. It is not ideal to use Gauss quadrature for the meshfree model because it requires a background mesh, and due to the rational RK shape functions, requires a higher order Gauss rule than the equivalent finite element. For the following study it suffices to show the validity of the MCIG approximate surface for use in contact analysis.

Figure 5.4: Impact of elastic bar with rigid plate.

5.3.1 Impact of Elastic Bar

In this example an elastic bar, 48 inches long and 12 inches tall, with material properties similar to steel is set in motion with an initial velocity of v_0=40in/s. The modulus of elasticity was E=30,000ksi with Poisson's ration ν=0.3 and material density ρ_0=500pcf. The diagram in Figure 5.4 depicts the problem domain and the initial configuration.

The MCIG approximate boundary and background integration mesh are shown in Figure 5.9; the particle distribution was created with a normalized support of α=1.5. A 3^{rd} order conical function was used for the kernel function and a linear polynomial basis for the approximation functions. The resulting MCIG approximation boundary has slightly rounded

corners, with an absolute volume difference error of 7.46119e-04in^2. The volume integration was done on the background quadrilateral mesh with a 4x4 quadrature rule, for the surface integration the MCIG approximate boundary was subdivided into segments and a 4 point quadrature rule was used. A contact penalty of β=1.0e05 was used for the meshfree model. The background mesh was used directly for the finite element model. The volume integration was done with a 2x2 point quadrature rule and the surface was integrated with a 4 point quadrature rule. A contact penalty of β=1.0e04 was used for the finite element model.

Table 5.1: Contact time for elastic bar.

Method	Contact Time	Release Time	Time in Contact
FEM	0.002501	0.012159	0.009658
RKPM	0.002416	0.011989	0.009573

In Table 5.1 the time of the impact and release are shown. The results for both methods were very similar. In Figure 5.6 the displacement of the contact end of the bar is plotted over time. Notice that the bar does not stop immediately when it encounters the rigid wall. This is a consequence of two things, the regularization effect of the penalty method smooths out the interaction over time. The bar does not stop exactly at the wall due to the numerical difficulty in enforcing the impenetration conditions exactly. The finite element model exhibited more penetration than the MCIG approximation, with the former penetrating by 0.017 inches and the later only by 0.002 inches. This is likely due to the choice of penalty parameters, as the meshfree model tolerated a slightly higher penalty parameter before becoming unstable during the impact. In Figure 5.7 the kinetic and strain energy over time are compared between the FE and RKPM model.

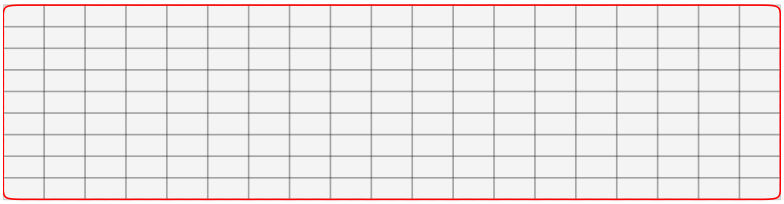


Figure 5.5: MCIG approximate boundary with background integration mesh for elastic bar.

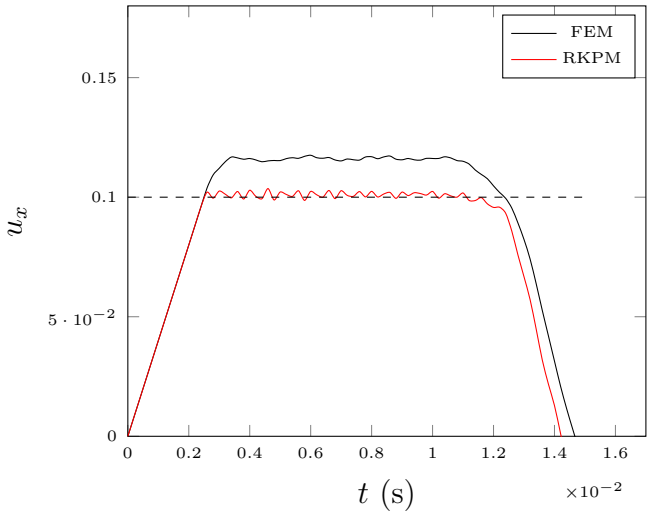


Figure 5.6: Displacement at contact end of elastic bar.

5.3.2 Contact of Rubber Ring

In this example a rubber like ring is impacted on a rigid wall in a similar fashion as the elastic bar in the previous example. The material properties used are E=14,500 psi and ν =0.4 for the modulus of elasticity and Poisson's ratio respectively. The density of the model material was ρ_0=70 pcf. Figure 5.8 depicts the problem and initial conditions. The initial velocity was v_0=15in/s headed perpendicular to the rigid wall. The background integration mesh for the meshfree model is shown in Figure 5.9a along with the MCIG approximate boundary shown as the red dashed line. The MCIG approximate boundary had an error of 0.004 in the representation relative to the actual area of the domain. A 4x4 quadrature rule was used in the quadrilateral elements; the surface integration was done

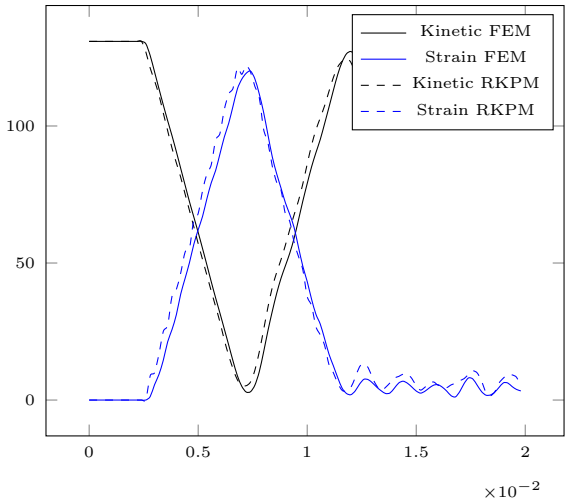


Figure 5.7: Energy profile for impact of elastic bar.

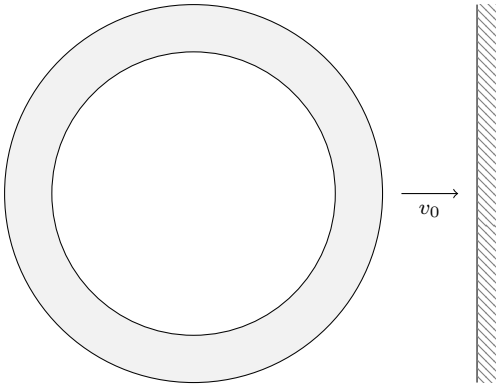
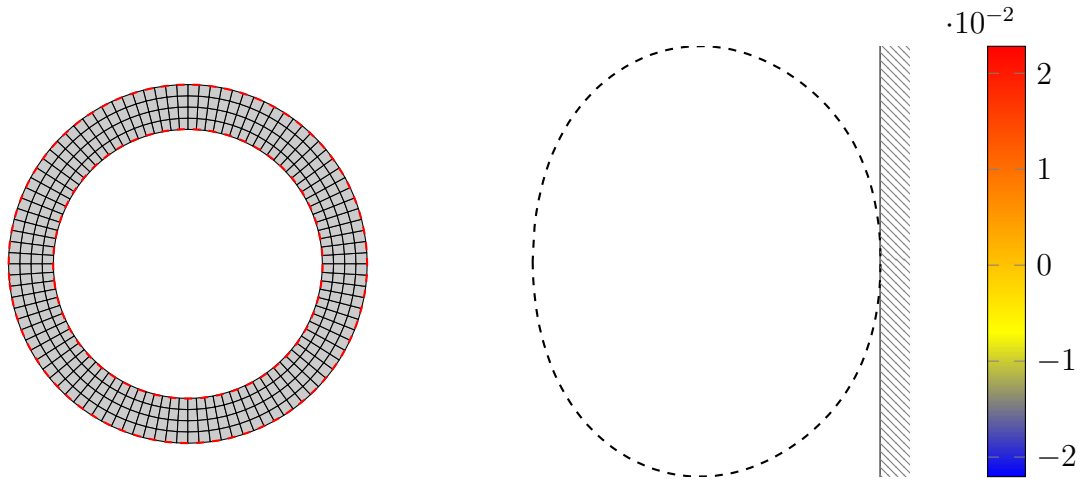


Figure 5.8: Rubber ring impact problem setup.

over the MCIG approximate surface by discretizing the surface into line segments, a 4 point quadrature rule was used in the segments.The background mesh was used directly for the finite element model. The volume integration was done with a 2x2 point quadrature rule and the surface was integrated with a 4 point quadrature rule. A contact penalty of β=1.0e05 was used for the meshfree and finite element models. The strain is shown on the ring during impact in Figure 5.9b along with the deformed MCIG approximate boundary. The deformed states of both the RKPM and FEM models are very similar. The impact event occurred over two distinct periods which will be referred to as the primary contact and secondary



(a) Background mesh with MCIG approximation in red.

(b) Strain in ring during impact.

Figure 5.9: Elastic ring initial and deformed configurations.

Table 5.2: Contact time for rubber ring.

Method	Primary Contact Event(s)	Secondary Contact Event (s)
FEM	0.0167	0.011
RKPM	0.017	0.0107

contact events. When the strain wave that is set in motion from the initial impact makes it all the way around, back to the contact region, the ring temporarily bounces off the wall resulting in a peak in the stored energy. The stored energy then finds a local minimum when the ring comes into contact with the wall a second time before the final bounce. The duration of the events for both methods are shown in Table 5.2; the methods displayed a 1.8% difference in the duration of the primary contact event and a difference of 2.8% for the secondary contact event. The strain energy is plotted for the impact event in Figure 5.10; the MCIG approximation found the strain energy to be slightly less the the finite element model during the impact event and after the contact event it found the strain energy to be slightly greater than the finite element model.

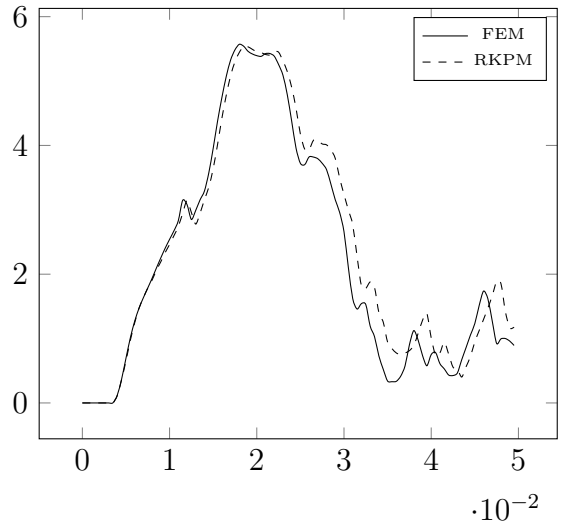


Figure 5.10: Strain energy of rubber ring impact.

5.4 Discussion

The MCIG approximate surface used in conjunction with an integration mesh performed very similar to the finite element models for both the elastic bar and rubber ring impact problems. The need for a large number of quadrature points to integrate the meshfree model led to much longer run times when compared to the finite element model. Using variational consistent nodal integration [14] would potentially lead to a great improvement in the run time of the meshfree model, and obviate the need for a background mesh.

The MCIG approximation paired with a nodal integration scheme has great potential to solve problems with changing geometry or topology. Topology optimization, for instance, is an iterative process in which material is added or subtracted from the model with every analysis iteration. Finite elements requires re-meshing of the domain with every iteration which quickly becomes expensive if the meshing cannot be done automatically. With proper particle placement, the MCIG approximation could provide an explicit surface definition automatically for a domain that changes with every iteration.

Chapter 6: Analysis Suitable Geometry from Medical Images

Patient-specific biomechanical simulations are a non-invasive way to develop an understanding of the physical processes that go on in the body. Modeling the bodies structural components, e.g. musculoskeletal system, for various individuals could reveal why some populations have certain outcomes versus other populations. An example of this is the female pelvic floor, it is known that certain populations of women have much higher rates of prolapse than other populations. This may very well be to differences in the pelvic floor structure and how the load is carried to the connective tissues. Modeling large groups of patients may be the only way to develop enough of an understanding and drive the outcomes for the individual patient. Developing an automatic and efficient method for generating patient-specific analysis suitable models is the only way that these types of studies will ever be possible. In this chapter, the MCIG approximate domain will be studied as a method for two things. The first is taking an image, or three-dimensional stack of images, and generating a smooth continuous analytic geometry. This analytic geometry may then be used to develop patient-specific models in the traditional way, which is to generate a finite element mesh or similar meshfree model. The second use is to directly use the MCIG approximate domain in an analysis.

6.1 Processing the Medical Images

Medical images are two dimensional arrays of grey scale data that are acquired via imaging modalities such as magnetic resonance imaging (MRI), or computed tomography (CT)

as shown in 6.1. Images are composed of elements called pixels, which are associated with a color value. In medical imaging this color value is associated with some property of the tissue being imaged, such as water composition. Pixels are representative of a small region in space, with a size dictated by the imaging machine. The three dimensional counterpart to the pixel is called a voxel, or volume element, which are formed by taking a series of images at uniform increments through the body [42].

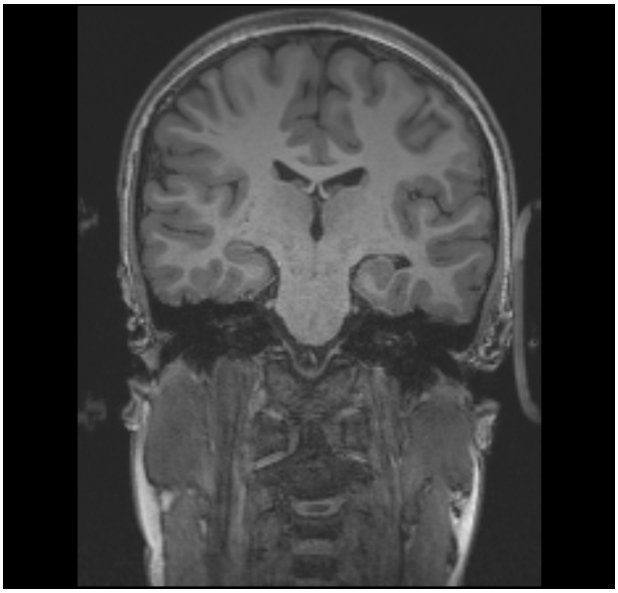


Figure 6.1: Example grey scale medical image

6.1.1 Image Segmentation

The delineation of individual organs in a medical image is called segmentation [9]. The process of segmentation is the assignment of an organ identification value to any pixel associated with a given organ. Segmented images are referred to as labels maps, due to each component in the image being assigned a label. A label map for Figure 6.1 can be seen in Figure 6.2. Manual segmentation consists of drawing the boundary of each component by hand, and can be very time consuming and error prone. In [8] and [9] semi-automated segmentation methods are discussed that rely on segmentation templates. Semi-automated

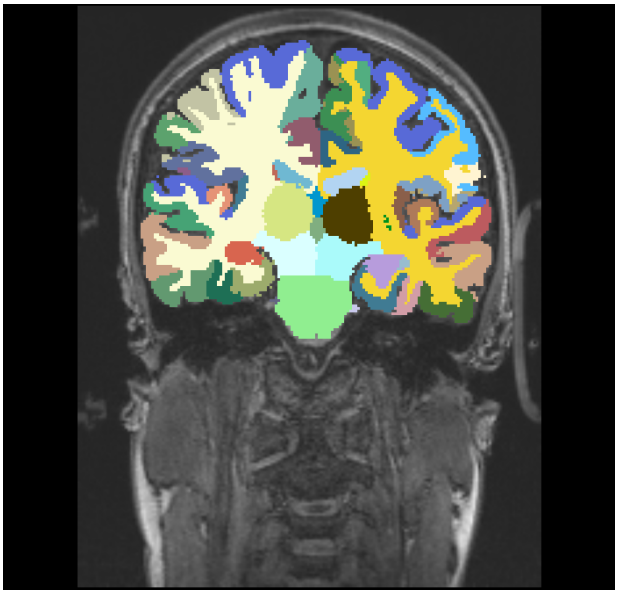


Figure 6.2: Example label map of brain.

methods can help with the tediousness associated with manual segmentation as well as with high frequency noise generated due to hand jitter. Fully automated segmentation still remains an open research problem, yet success in automated segmentation of brain tumors [45].

6.1.2 Transformation to Physical Coordinates

The first step in generating a model from the label map is to transform the pixels from index space,(i, j, k), to the real coordinates, (x, y, z). World space, or anatomical space, is defined as the continuous three-dimensional space in which an image has been sampled. World space is positioned with an origin near the patients body with basis vectors defined in some common anatomical axes such as anterior/posterior,inferior/superior,left/right. Orienting the image in world space is affine, meaning it is a linear transformation **A** with a

translation t. The affine transformation takes the form

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} & A_{13} & t_1 \\ A_{21} & A_{22} & A_{23} & t_2 \\ A_{31} & A_{32} & A_{33} & t_3 \end{bmatrix} \begin{bmatrix} i \\ j \\ k \\ 1 \end{bmatrix} \tag{6.1}$$

There are two perspectives that may be used in generating points from the images, cell centered and node centered. Cell centered simply means generating a point at the center of each pixel or voxel. Node centered refers to treating each pixel or voxel as an element with a node at each corner. A node centered approach will result in slightly more points, but has the advantage of being guaranteed to have a corresponding mesh called a voxel mesh.

6.2 Current State of Technology

There are several approaches for generating analysis suitable models from medical images. The basic approach is to reconstruct a geometric representation from the digitized label map. The geometric representation can the be used to develop an analysis suitable model, such as finite element mesh. This is similar to generating an analysis suitable model from a CAD drawing, so much so that it is common to generate the geometric representation in the form of a NURBS surface.

The simplest method is to treat the voxel model as an accurate geometric representation, and to use it directly, or a smoothed version thereof, as a finite element mesh. When the voxel model is used as the base for the analysis mesh, the mesh is highly dependent on image resolution. The dependence on image resolution generally results in a large number of elements to represent a given object [29]. A more common approach in recent times is

to construct a smooth analytic description of the domain using a contour method. Contour methods are those that attempt to define a smooth continuous approximation of the boundary points.

6.2.1 Voxel Model

Voxel models have a surface with stair-stepped features called aliasing artifacts which result in sharp surface features [7]. An example of a voxel model showing aliasing artifacts can be found in figure 6.3. The voxel models can have a much greater surface area than reality and the sharp features of aliasing artifacts cause stress concentrations, for this reason smoothing of the voxel mesh is usually performed to generate a FEM mesh as in [7],[1]. While smoothing the voxel mesh produces a better looking model, and smoother results from the FEM analysis, the geometric model can be a less accurate representation of the actual data due to mesh shrinking, so effort must be made to retain mesh volume [1].

6.2.2 Contour Methods

Contour methods seek to approximate a surface that fits the set of surface points, either by direct meshing,implicit function, polynomial fitting. Contour methods may be divided into three categories: contour stitching, volumetric methods, and point set surfaces [8]. Contour stitching methods are those that define two-dimensional contours and seek to connect the vertices from neighboring contours to form a three-dimensional surface [23].

The first volumetric approach was defined in [31]. Volumetric methods generally rely on the computation of a discrete distance field in the slice plane. A discrete distance field relays how far a given pixel is from the closest boundary pixel. The idea is to compute a two-dimensional distance field in each image, then interpolate the distance fields between slices. A surface is then defined as the zero level-set of the interpolated distance function.

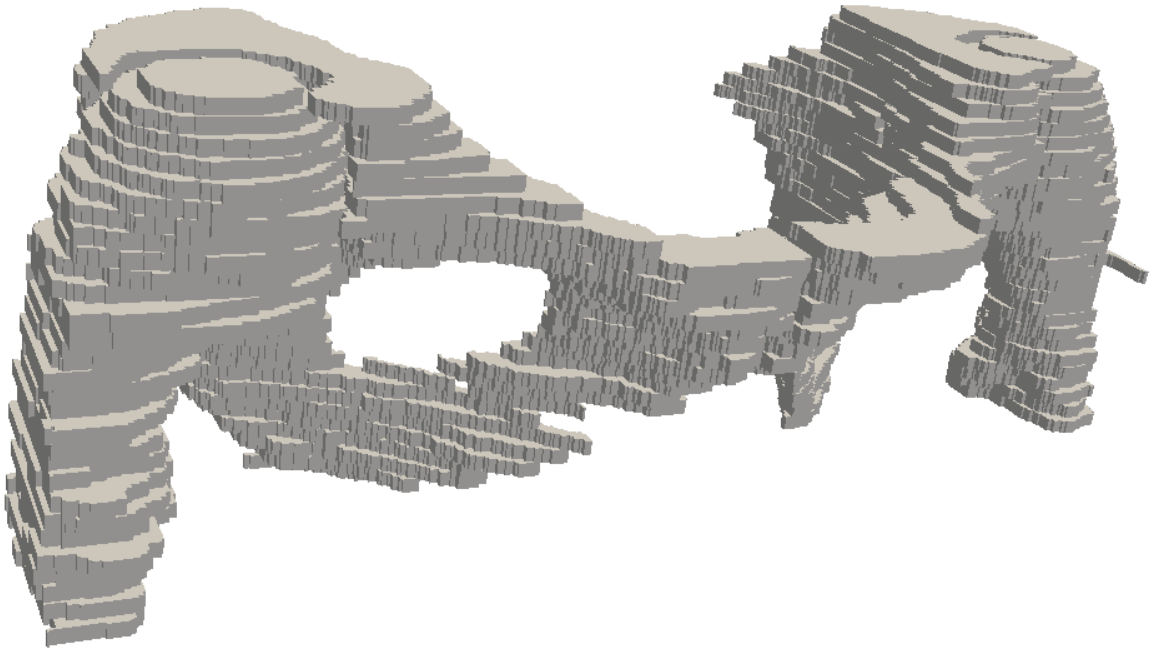


Figure 6.3: Voxel model of human pelvis.

The third approach is that of point set surfaces, which usually rely on the points being oriented, meaning that surface normals are defined at the points. One method of this variety computes a signed distance field from the surface points, by approximating a local tangent plane. Another such method uses moving least squares (MLS) to approximate polynomials through the points. Other methods were derived from the MLS style approximations. Recently, the most popular of this type of reconstruction method has been the multi-partition of unity (MPU) implicit method first described in [8]. The MPU method generates an approximation of the point set surface by computing local overlapping implicit functions that are blended together to form a global implicit function. MPU also requires the point set to be oriented with surface normals.

6.3 The MCIG Approach

The MCIG approximation may be used in two ways. First, the MCIG approximation of the voxel model may be viewed as a volumetric contour method similar to the distance field methods discussed earlier. There are some key differences that, that make the MCIG and attractive alternative. Distance fields are rigid in their definition of the surface, which makes them susceptible to errors due to noise in the data. The segmentation process results in noisy data, especially if there is a manual component to the work flow. A key characteristic of meshfree approximations is their smoothness, and this is a key component of the MCIG approximation.

The second way in which the MCIG approximation can be used is directly as an analysis suitable model. This approach is the most direct, and is very similar to using the voxel model in a finite element analysis with the exception that the MCIG provides a naturally smooth geometric representation. This again ties the size of the analysis model to the image

resolution, which is typically quite large. Using the MCIG representation, the model may be coarsened while retaining the major relevant surface features.

6.3.1 Generating the Admissible Particle Distribution

The structure inherent in the voxel model naturally results in an admissible particle distribution. There are two approaches that can be used to generate the particle distribution, cell centered and node centered. The cell centered approach generates a particle at the center of each voxel, while the node centered approach generates particles at the corners of each voxel. The node centered approach has the benefit of potentially representing thin features where the cell centered approach may fail. This would occur where a single row of pixels juts out in a single image or where a feature is so thin (relative to the image spacing) that it only appears in a single image in the stack. Assigning the support size for each particle is done the same as processing a mesh to generate an ADP. The support size is computed as some small multiple of the side lengths of the voxels.

6.3.2 Non-convex Features

As mentioned in Chapter 4, the MCIG approximation does not automatically represent non-convex features that are smaller than support size of nearby particles. This is due to particle supports overlapping inside the concavity, and the visibility criterion was developed to handle such cases. The visibility criterion is generally implemented as a simple shape used to block the interaction of adjacent particles, i.e. if the line of sight between a point and a particle is blocked by the shape, the particle does not contribute to the geometry at that point. Due to the size of medical images, it is prohibitive to manually dictate the location of all concavities. A simple method for automating the generation of visibility conditions was developed out of necessity. The visibility conditions are generated by first marking every

pixel 1 if inside domain and 0 if outside domain. Then a Marching Cubes procedure is used to extract a surface just outside the domain by selecting a value between 0 and 1. This value will effectively be an offset based on the pixel spacing of the image. The resulting set of visibility conditions will form a covering of the domain, which will be a much large set than is necessary. The majority of the visibility panels may be jettisoned after determining they are not blocking the interaction of any particle.

6.4 Numerical Examples

In this section the versatility of the MCIG procedure for constructing analysis suitable models from medical images is displayed. The examples are three-dimensional reconstructions of two skeletal structures, the scapula and the pelvis; followed by reconstructinos of the heart, brain, and female pelvic floor. For the bony structures, the accuracy of the MCIG reconstruction is studied and compared to the accuracy of a method from the literature. The heart and brain examples display the strength of the MCIG reconstruction when modeling domains with multiple components with complex topological relationships. The female pelvic floor example displays the potential of using the MCIG reconstruction directly as an ASG in a simulation of the first stage of childbirth.

6.4.1 Bones

The bony pelvis and scapula were reconstructed from two different data sets. The image datasets for the scapula are publicly available from the Laboratory of Human Anatomy and Embryology, University of Brussels (ULB), Belgium. The segmentation was done using the Slicer 3D [18] software platform using a semi-automatic threshold algorithm. There were 158 images taken in 1mm increments; each image has a resolution of 511x513 with pixel size of 0.35mm. The particle distribution had 686,531 particles with a normalized support size

of $\alpha = 2.2$. The MCIG approximation used a Quadratic polynomial basis and a 3rd order conical kernel function. A three-dimensional rendering of the resulting geometry is shown from three different view points in Figure 6.4. The pelvis data set is available from the open source software platform OsiriX [38]. The segmentation was done using the Osirix platform with a semi-automatic threshold algorithm. The images have a resolution of 512x512 with a pixel size of 0.74mm. The 288 images in the set were taken incrementally at a spacing of 1.5mm. A rendering of the MCIG approximation is shown in Figure 6.5. A method for determining the accuracy of a reconstruction is to count the number of boundary voxels that the approximated surface cuts through. This method is an extension of the difference distance described in Chapter 3. Instead of computing the maximum point-wise distance between the intended domain and the approximate domain, the accuracy is categorized by the number of boundary points in the intended domain that fall within a given distance of the approximate surface. The accuracy of a MPU reconstruction was found to be in the range of 36%-48% for models of the lumbar spine, femur, and mandible in [16]. Initial studies of the accuracy of the MCIG reconstruction find an accuracy of 88%-98% for the scapula and pelvis respectivly. Though this is not a direct comparison, it suggests that the MCIG is a very accurate portrayal of the segmented medical image.

6.4.2 Heart

The data set for the heart was donated by a private donor, and was segmented by a group of researchers in the Department of Radiology at the University of South Florida's Morsani College of Medicine. The heart image set was segmented into seven distinct parts including the heart muscle, aortic wall and blood, arterial wall and blood, and venous wall and blood. The resulting label maps had an image resolution of 384x384 with a pixel spacing of 1mm.

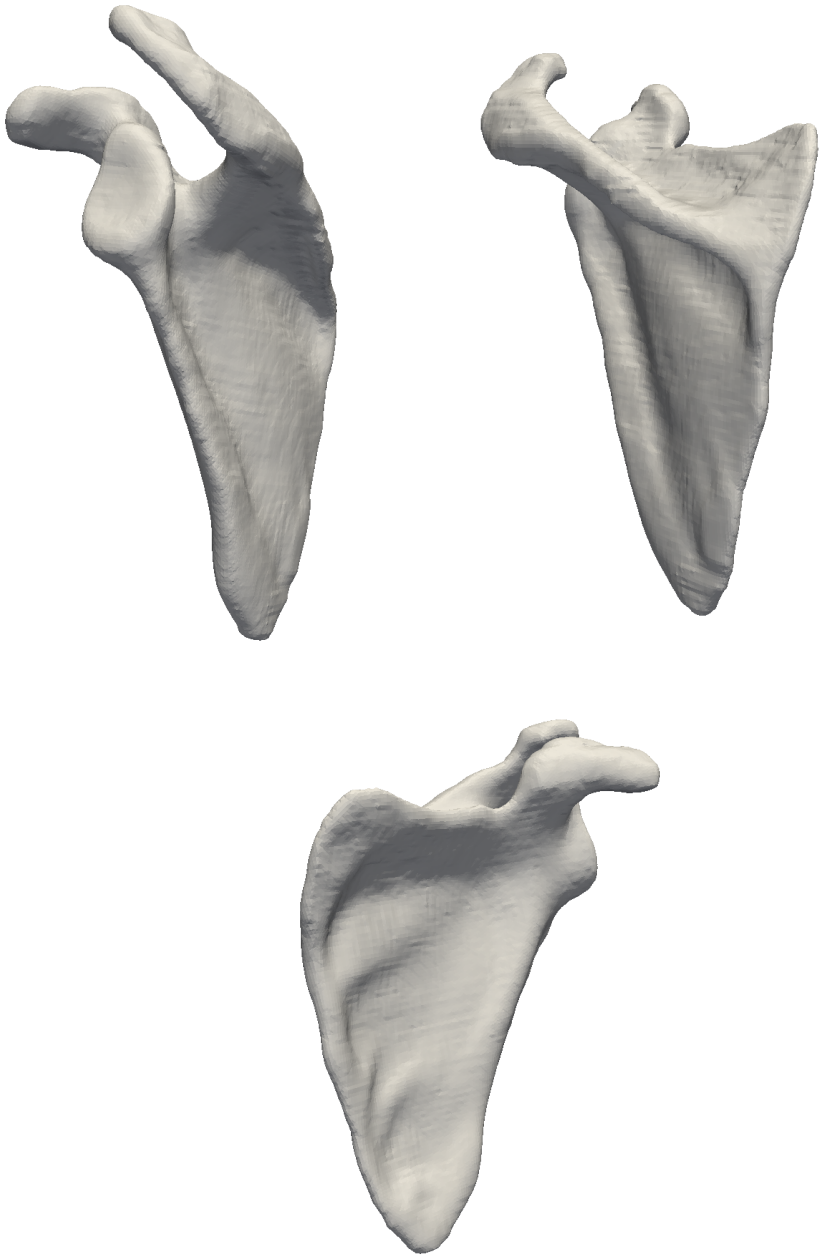


Figure 6.4: Three views of the MCIG approximation of the scapula.

Figure 6.5: MCIG approximation of the pelvis.

There were 295 images sampled at a spacing of 1mm. The point cloud extracted from the label maps contained approximatley 1.7 million points for all seven objects combined; Figure 6.6 shows the recontructed model. The heart model is topologically interesting, there are many vessels extending from the heart muscle that grow into intricate branching structures.

6.4.3 Human Brain

The brain was segmented into 134 individual structures and exhibits many complex geometric features that highlight the capabilities of the MCIG approximation. There are many concavities in the brain that present a challenge for geometry reconstruction. The MCIG contours are shown in Figure 6.7, they appear to very closely follow the boundary pixels of the segmentation though no quantitative measure of accuracy was taken. Figure 6.8 depicts the three-dimensional reconstruction of the brain with the various structures colored

by their label map value, on the right side of the model the top layer of structures composed of gray matter have been removed to display the reconstruction of the underlying white matter. The voxel size of the brain data set was 1mm in all directions, and the extracted point cloud consisted of 1.2 million points.

6.4.4 Female Pelvic Floor

The data set for the pelvic floor consisted of 135 images with a resolution of 200 by 200, and a voxel size of 1mm in all directions. The reconstruction of the female pelvic floor consist of five components; Figure 6.9 is a rendering of the models shown from two points of view. The components are:

1. Bony pelvis (white)
2. Obturator internus (beige)
3. Levator ani (orange)
4. Vagina (blue)
5. Rectum (green)

The levator ani muscles are primarily responsible for regulating bodily functions and providing support for the reproductive organs. Childbirth simulations seek to determine the amount that the levator ani stretches due to the fetal head moving through the birth canal. Figure 6.10 shows the initial geometry for the levator ani along with the deformed levator ani during the childbirth simulation using the MCIG reconstruction for the ASG.

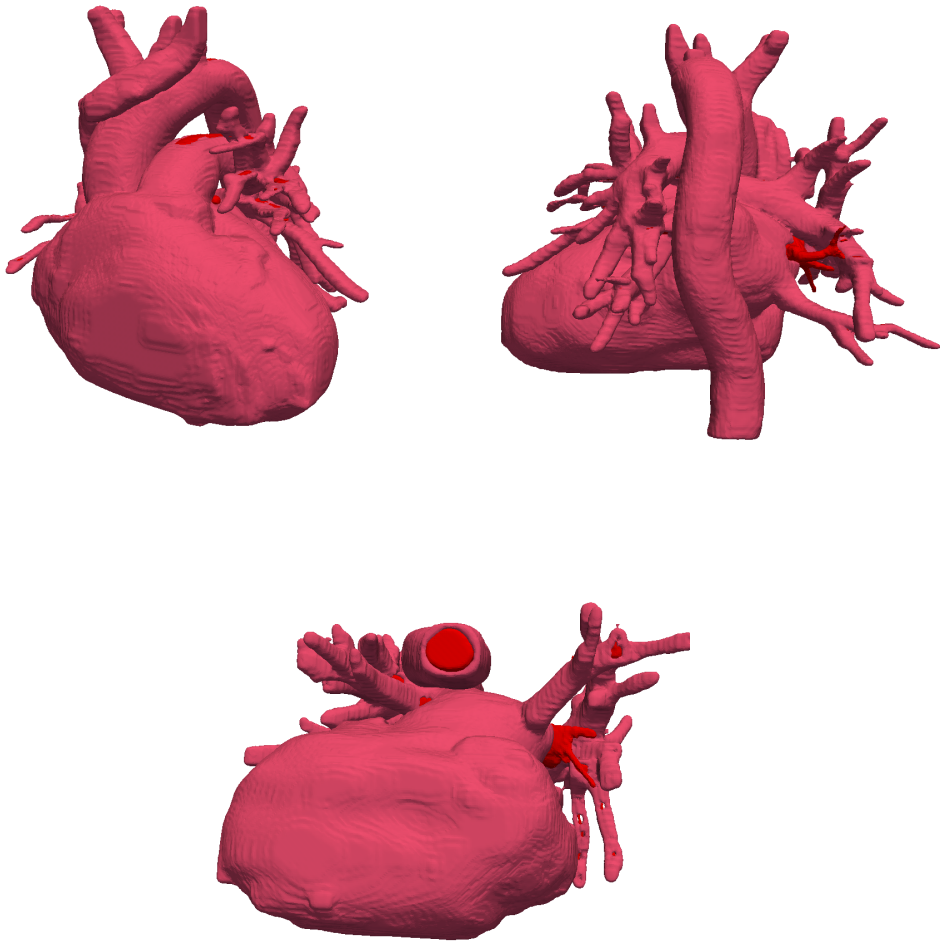


Figure 6.6: Three views of the MCIG approximation of the heart.

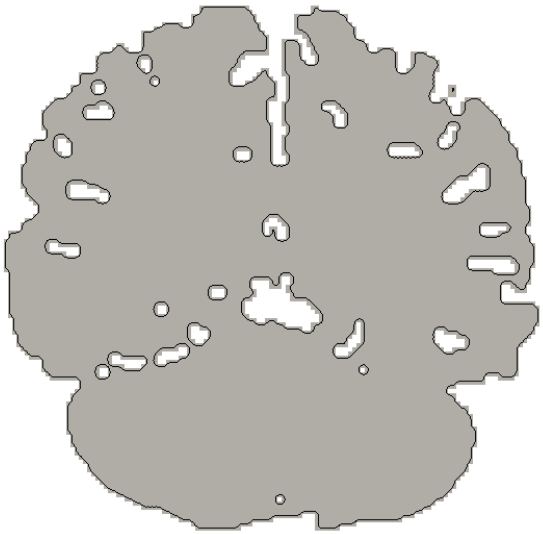


Figure 6.7: MCIG approximation of brain image

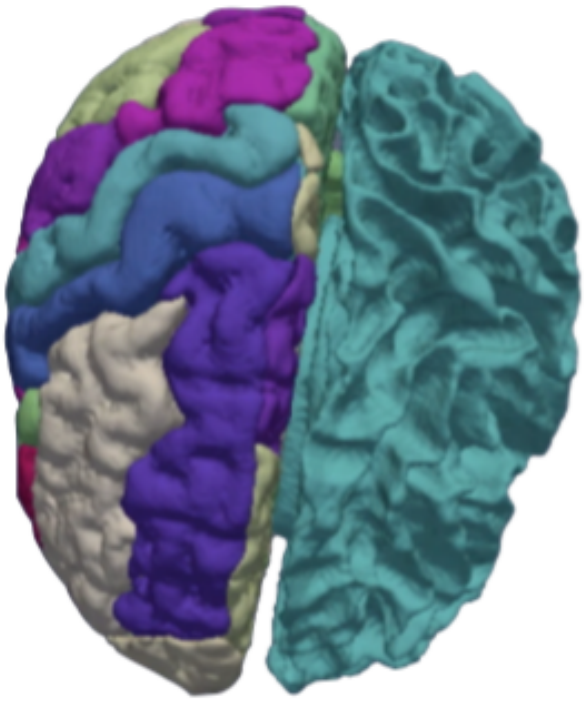


Figure 6.8: Three-dimensional reconstruction of brain

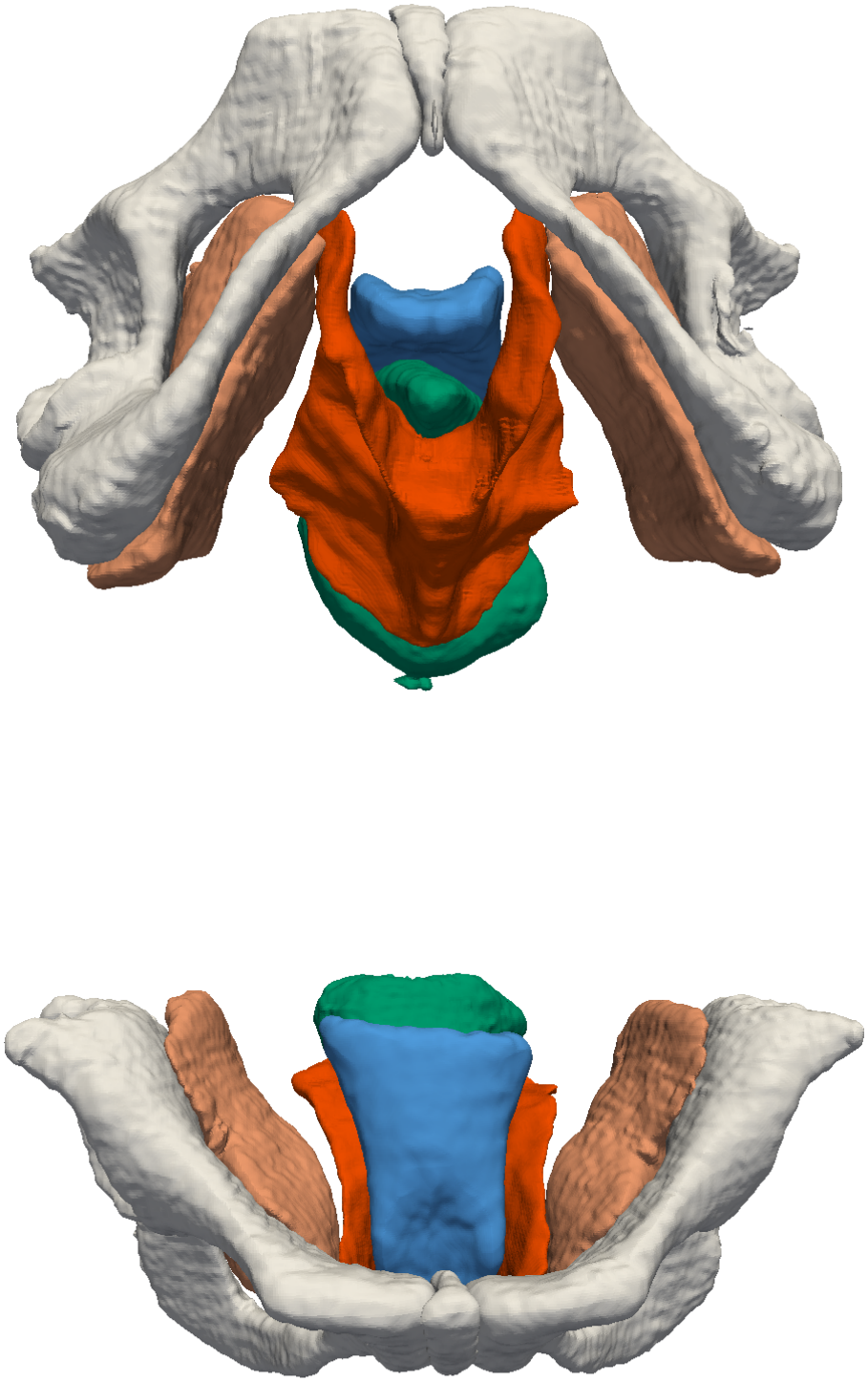


Figure 6.9: MCIG approximation of the female pelvic floor.

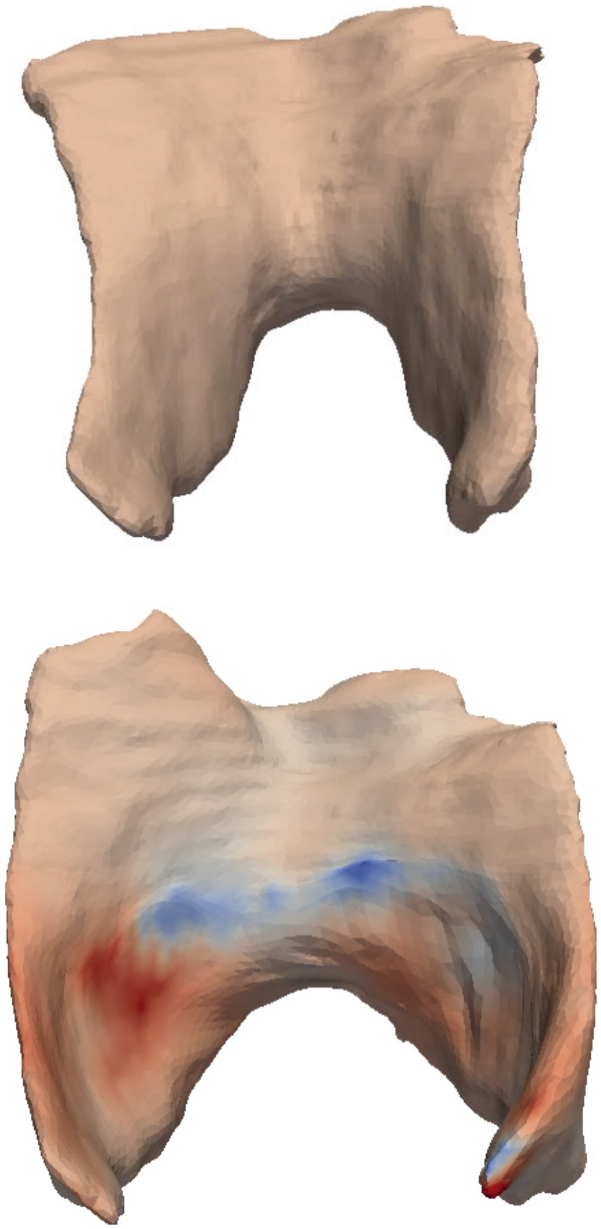


Figure 6.10: First stage of child birth simulation.

Chapter 7: Conclusions

This dissertation opened with a discussion on the various geometry representations that constitute analysis suitable geometry. The particle distributions used in meshfree analysis were shown to be unique due to their lack of a continuous geometry representation. Often meshfree particle distributions are paired with a secondary representation, such as a mesh or NURBS surface, that is used in the solution of boundary value problems. This lead to the definition of the meshfree geometry function; a continuous approximation of the particle distribution. Error metrics were introduced that allowed for the explicit determination of a boundary given a particle distribution and a geometry function.

A potential meshfree geometry, called the Meshfree Correction Implicit Geometry, was introduced and studied given the guidelines in the framework. It was shown that the MCIG has potential as an accurate analysis suitable geometry that is intrinsic to the meshfree particle distribution. The dissertation concluded with a discussion on potential applications for the MCIG approximation. It was shown that given the uniform orientation of medical image data, the MCIG can be used to both generate analysis suitable models and used as an analysis suitable geometry directly. Several examples were shown to highlight the robustness of the method for generating models.

7.1 Future Work

There remains a lot of work to be done in the study of meshfree geometry in general and specifically patient specific modeling. Further study is warranted in characterizing the

meshfree geometry functions and the effect of the function space parameters on the resultant surface. The following list highlights some ideas that could bring improvements in the near future.

1. The MCIG approximation was shown to be accurate for particles with a specific orientation in reference to the intended surface. Understanding the effect of particle arrangement on the meshfree geometry could lead to further refinement of the geometry function definition or constraints on particle distribution routines.

2. The meshfree boundary is defined as the level-set of an implicit function. In the current work this boundary was discretized with elements so that quadrature routines could be used to integrate the weak form. The possibility of directly integrating on the implicit surface exists and should be explored.

3. The MCIG approximation shows promise as tool in the generation of patient-specific biomechanics simulations. Using the MCIG approximation on the voxel model may have an effect on the stability and convergence properties of the method, this requires further study to determine what these effects are.

References

[1] Thibaut Bardyn, Mauricio Reyes, Xabier Larrea, and Philippe Bächler. Influence of smoothing on voxel-based mesh accuracy in micro-finite element. In Karol Miller and Poul M.F. Nielsen, editors, *Computational Biomechanics for Medicine*, pages 85–93. Springer New York, 2010.

[2] Y. Bazilevs, L. Beira Da Veiga, J. A. Cottrell, T. J. R. Hughes, and G. Sangalli. Isogeometric analysis: Approximation, stability and error estimates for h-refined meshes. *Mathematical Models and Methods in Applied Sciences*, 16(07):1031–1090, 2006.

[3] T. Belytschko, W.K. Liu, and B. Moran. *Nonlinear Finite Elements for Continua and Structures*. Nonlinear Finite Elements for Continua and Structures. Wiley, 2000.

[4] T Belytschko, Y.Y. Liu, and L. Gu. Element-free galerkin methods. *International Journal for Numerical Methods in Engineering*, 37:229–256, 1994.

[5] T. Belytschko, YY Lu, and L. Gu. Fracture and crack growth by element-free Galerkin methods. *Model. Simul. Sci. Comput. Engrg.*, 2:519–534, 1994.

[6] T. Belytschko, Y.Y. Lu, L. Gu, and M. Tabbara. Element-free galerkin methods for static and dynamic fracture. *International Journal of Solids and Structures*, 32(1718):2547 – 2570, 1995. A Symposium on the Dynamic Failure Mechanics of Modern Materials In Memory of Professor J. Duffy.

[7] Steven K Boyd and Ralph Müller. Smooth surface meshing for automated finite element model generation from 3d image data. *Journal of Biomechanics*, 39(7):1287–1295, 2006.

[8] Ilya Braude, Jeffrey Marker, Ken Museth, Jonathan Nissanov, and David Breen. Contour-based surface reconstruction using mpu implicit models. *Graphical Models*, 69(2):139 – 157, 2007.

[9] Martin Caon. Voxel-based computational models of real human anatomy: a review. *Radiation and environmental biophysics*, 42(4):229–235, 2004.

[10] Jiun-Shyan Chen, Michael Hillman, and Marcus Rüter. An arbitrary order variationally consistent integration for galerkin meshfree methods. *International Journal for Numerical Methods in Engineering*, 95(5):387–418, 2013.

[11] Jiun-Shyan Chen and Hui-Ping Wang. New boundary condition treatments in meshfree computation of contact problems. *Computer Methods in Applied Mechanics and Engineering*, 187(34):441 – 468, 2000.

[12] Jiun-Shyan Chen, Cheng-Tang Wu, Sangpil Yoon, and Yang You. A stabilized conforming nodal integration for galerkin mesh-free methods. *International Journal for Numerical Methods in Engineering*, 50:435–466, 2001.

[13] J.S. Chen, C. Pan, and C.T. Wu. Large deformation analysis of rubber based on a reproducing kernel particle method. *Computational Mechanics*, 19(3):211–227, 1997.

[14] Sheng-Wei Chi, Chung-Hao Lee, Jiun-Shyan Chen, and Pai-Chen Guan. A level set enhanced natural kernel contact algorithm for impact and penetration modeling. *International Journal for Numerical Methods in Engineering*, 102(3-4):839–866, 2015.

[15] J.A. Cottrell, T.J.R. Hughes, and A. Reali. Studies of refinement and continuity in isogeometric structural analysis. *Computer Methods in Applied Mechanics and Engineering*, 196(4144):4160 – 4183, 2007.

[16] Vinciane d'Otreppe, Romain Boman, and Jean-Philippe Ponthot. Generating smooth surface meshes from multi-region medical images. *International Journal for Numerical Methods in Biomedical Engineering*, 28(6-7):642–660, 2012.

[17] C.Armando Duarte and J.Tinsley Oden. An h-p adaptive method using clouds. *Computer Methods in Applied Mechanics and Engineering*, 139(14):237 – 262, 1996.

[18] Andriy Fedorov, Reinhard Beichel, Jayashree Kalpathy-Cramer, Julien Finet, Jean-CristopheC. Fillion-Robin, Sonia Pujol, Christian Bauer, Dominique Jennings, FionaM Fennessy, Milan Sonka, John Buatti, StephenR Aylward, JamesV Miller, Steve Pieper, and Ron Kikinis. 3d slicer as an image computing platform for the quantitative imaging network. *Magnetic Resonance Imaging*, 30(9):1323–41, 11 2012.

[19] Sonia Fernández-Méndez and Antonio Huerta. Imposing essential boundary conditions in mesh-free methods. *Computer methods in applied mechanics and engineering*, 193(12):1257–1275, 2004.

[20] R. A. Gingold and J. J. Monaghan. Smoothed particle hydrodynamics: theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society*, 181(3):375–389, 1977.

[21] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, 3rd edition, 1996.

[22] Michael Griebel and Marc Alexander Schweitzer. A particle-partition of unity method for the solution of elliptic, parabolic, and hyperbolic pdes. *SIAM Journal on Scientific Computing*, 22(3):853 – 890, 2000.

[23] Olya Grove. *Heterogeneous Modeling of Medical Image Data Using B-Spline Functions*. PhD thesis, University of South Florida, 2011.

[24] W. Han and X. Meng. Error analysis of the reproducing kernel particle method. *Computer Methods in Applied Mechanics and Engineering*, 190:6157–6181, 2001.

[25] Antonio Huerta and Sonia Fernández-Méndez. Enrichment and coupling of the finite element and meshless methods. *International Journal for Numerical Methods in Engineering*, 48(11):1615–1636, 2000.

[26] T.J.R. Hughes. *The Finite Element Method*. Dover, Mineola, New York, 2000.

[27] T.J.R Hughes, J.A. Cottrell, and Y. Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194:4135–4195, 2005.

[28] Igor Kaljević and Sunil Saigal. An improved element free galerkin formulation. *International Journal for numerical methods in engineering*, 40(16):2953–2974, 1997.

[29] Jan Kaminsky, Thomas Rodt, Alireza Gharabaghi, Jan Forster, Gerd Brand, and Madjid Samii. A universal algorithm for an improved finite element mesh generation: Mesh quality assessment in comparison to former automated mesh-generators and an analytic model. *Medical Engineering & Physics*, 27(5):383 – 394, 2005.

[30] A. N. Kolmogorov and S. V. Fomin. *Introductory real analysis*. Courier Corporation, 2012.

[31] David Levin. The approximation power of moving least-squares. *Mathematics of Computation of the American Mathematical Society*, 67(224):1517–1531, 1998.

[32] Shaofan Li and Wing Kam Liu. *Meshfree particle methods*. Springer, 2004.

[33] Shaofan Li, Dong Qian, Wing kam Liu, and Ted Belytschko. A meshfree contact-detection algorithm. *Computer Methods in Applied Mechanics and Engineering*, 190:3271–3292, 2001.

[34] Wing Kam Liu, Sukky Jun, and Yi Fei Zhang. Reproducing kernel particle methods. *International Journal for Numerical Methods in Fluids*, 20(8-9):1081–1106, 1995.

[35] W.K. Liu, S. Jun, and Y.F. Zhang. Reproducing kernel particle methods. *International Journal for Numerical Methods in Fluids*, 20:1081–1106, 1995.

[36] W.K. Liu, S. Li, and T. Belytschko. Moving least square reproducing kernel method. part i: Methodology and convergence. *Computer Methods in Applied Mechanics and Engineering*, 143:422–453, 1997.

[37] Daniel Organ, Mark Fleming, T Terry, and Ted Belytschko. Continuous meshless approximations for nonconvex bodies by diffraction and transparency. *Computational mechanics*, 18(3):225–235, 1996.

[38] Antoine Rosset, Luca Spadola, and Osman Ratib. Osirix: An open-source software for navigating in multidimensional dicom images. *Journal of Digital Imaging*, 17(3):205–216, 2004.

[39] H. L. Royden. *Real Analysis*. McMillan, New York, 2 edition, 1968.

[40] Y. Saad. *Iterative Methods for Sparse Linear Systems: Second Edition*. Society for Industrial and Applied Mathematics (SIAM, 3600 Market Street, Floor 6, Philadelphia, PA 19104), 2003.

[41] Steven S Skiena. *The Algorithm Design Manual*. Springer-Verlag London, 2 edition, 2008.

[42] M. R. Stytz, G. Frieder, and O. Frieder. Three-dimensional medical imaging: algorithms and computer systems. *ACM Comput. Surv.*, 23(4):421–499, December 1991.

[43] Gregory J. Wagner and Wing Kam Liu. Hierarchical enrichment for bridging scales and mesh-free boundary conditions. *International Journal for Numerical Methods in Engineering*, 50(3):507–524, 2001.

[44] T. Zhu and S. N. Atluri. A modified collocation method and a penalty formulation for enforcing the essential boundary conditions in the element free galerkin method. *Computational Mechanics*, 21(3):211–222, 1998.

[45] Kelly H Zou, William M Wells, Ron Kikinis, and Simon K Warfield. Three validation metrics for automated probabilistic image segmentation of brain tumours. *Statistics in medicine*, 23(8):1259–1282, 2004.

Appendices

Appendix A: Useful Information

A.1 List of Symbols

Ω	Intended domain
$\hat{\Omega}_\rho$	Approximate domain
ρ	Smoothing length of meshfree particle distribution
$\mathcal{G}(x)$	Meshfree geometry function
$\partial\hat{\Omega}_\rho$	Interior of approximate domain
$\overset{\triangle}{\Omega}$	Symmetric difference of two domains.
$\mu\left(\overset{\triangle}{\Omega}\right)$	Difference volume measure
$\Phi_\rho(x)$	Kernel function
Ψ	Meshfree basis function
$\mathbf{M}$	Moment matrix
$\mathcal{F}$	Function space
$\hat{\Omega}_{\mathcal{G}}^{\mathcal{F}}$	MCIG approximate domain
β	Contact penalty
θ_n	Penetration gap