Stable Embedded Grid Techniques in Computational Mechanics

by

Jessica D. Sanders

Department of Civil and Environmental Engineering
Duke University

Date: __________________________

Approved:

______________________________
Tod Laursen, Supervisor

______________________________
John Dolbow

______________________________
Jeff Scruggs

______________________________
Anita Layton

______________________________
Brian Mann

______________________________
Mike Puso

Dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Civil and Environmental Engineering in the Graduate School of Duke University

2010
Abstract

Stable Embedded Grid Techniques in Computational Mechanics

by

Jessica D. Sanders

Department of Civil and Environmental Engineering
Duke University

Date: __________________________

Approved:

_____________________________
Tod Laursen, Supervisor

_____________________________
John Dolbow

_____________________________
Jeff Scruggs

_____________________________
Anita Layton

_____________________________
Brian Mann

_____________________________
Mike Puso

An abstract of a dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Civil and Environmental Engineering in the Graduate School of Duke University 2010
Abstract

Engineering mechanics problems involving interfaces, whether physical or introduced by numerical methodologies, are commonplace. Just a few examples include fracture and fault mechanics, classical contact-impact, phase boundary propagation, and fluid/structure interaction. This dissertation focuses on issues of numerical stability and accuracy that must be addressed when such interfaces are included in a realistic simulation of a physical system.

We begin by presenting a novel numerical method of fluid/structure interaction that may be applied to the problem of movable devices and ocean waves. The work is done with finite differences, large motion Lagrangian mechanics, and an eye towards creating a model in which complex rigid body dynamics can be incorporated.

We then review the many advantages of embedded mesh techniques for interface representation, and investigate a completely finite element based strategy for embedding domains. The work is seen as a precursor to robust multi-physics simulation capabilities. Special attention must be given to these techniques in terms of stable and convergent representation of surface fluxes. Mesh locking and over-constraint are particularly addressed. We show that traditional methods for enforcing continuity at embedded interfaces cannot adequately guarantee flux stability, and show a less traditional method, known as Nitsche’s method, to be a stable alternative. We address the open problem of applying Nitsche’s method to non-linear analysis by drawing parallels between the embedded mesh and discontinuous Galerkin (DG)
methods, and propose a DG style approach to Nitsche’s method. We conclude with stable interfacial fluxes for a continuity constraint for a case of embedded finite element meshes in large deformation elasticity. The general conclusion is drawn that stability must be addressed in the choice of interface treatment in computational mechanics.
Contents

Abstract iv
List of Tables x
List of Figures xi
List of Abbreviations xvi
Acknowledgements xvii

1 Introduction 1
  1.1 Interfaces in Computational Mechanics . . . . . . . . . . . . . . . . . 1
  1.2 Specific Techniques Developed . . . . . . . . . . . . . . . . . . . . . . 4
  1.3 Previous Examples in Computationally Embedded Domains . . . . . 6

2 Fluid/Rigid Body Interaction with Finite Differences 9
  2.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9
  2.1.1 Wave energy harvesting . . . . . . . . . . . . . . . . . . . . . . 9
  2.1.2 Literature review . . . . . . . . . . . . . . . . . . . . . . . . . . 11
  2.1.3 An algorithm with fully Lagrangian equations of motion . . . 14
  2.2 Governing Equations . . . . . . . . . . . . . . . . . . . . . . . . . . . 15
  2.2.1 Two phase incompressible flow in an Eulerian framework . . . . 15
  2.2.2 Rigid body equations of motion . . . . . . . . . . . . . . . . . . . 17
  2.2.3 No slip boundary between solid and fluid . . . . . . . . . . . . . 20
  2.3 Fluid/Solid Interaction Model . . . . . . . . . . . . . . . . . . . . . . 20
## 2.4 Numerical Method

- **2.4.1 Time integration scheme**  
- **2.4.2 Spatial discretization**

## 2.5 Numerical Results

- **2.5.1 Flow around a cylinder**
- **2.5.2 Sedimentation of cylindrical disk**
- **2.5.3 Rigid body in a two-phase fluid: buoyancy**
- **2.5.4 Roll of a buoyant body**
- **2.5.5 Wave tank and buoy simulations**
- **2.5.6 Internal rotations prototypes**

## 2.6 Conclusion

### 3 Embedded Meshes for Linear Elasticity

- **3.1 Introduction**
  - **3.1.1 Specific challenges in stable enforcement of embedded interfacial constraints**
  - **3.1.2 Literature review - Nitsche’s method**

- **3.2 Problem Formulation**

- **3.3 Mortar Tying of a Surface to an Overlapped Volume**

- **3.4 Penalty Approximations to Mortar Lagrange Multipliers**

- **3.5 Nitsche’s Method**

- **3.6 Discrete Equation Structure**
  - **3.6.1 Unconstrained case**
  - **3.6.2 Integration of cut cells**
  - **3.6.3 Standard mortars**
  - **3.6.4 Penalty method**
  - **3.6.5 Nitsche’s method**
3.7 Numerical Examples ............................................. 76
  3.7.1 Patch tests ............................................. 77
  3.7.2 Pure bending ........................................... 78
  3.7.3 Example in curved interfaces ......................... 92
  3.7.4 Multiple inclusions .................................. 93
3.8 Conclusions .................................................... 98

4 A Discontinuous Galerkin Approach to Nitsche’s Method 99
  4.1 Introduction ............................................... 99
  4.2 Background and Literature Review ....................... 100
  4.3 Introduction to the Discontinuous Galerkin Discrete Derivative .. 101
  4.4 The discontinuous Galerkin derivative in the linear elastic case .... 108
    4.4.1 Calculation of a discrete lifting function - linear case ...... 113
    4.4.2 Pseudocode for a cut element with a DG operator .......... 116
  4.5 Numerical Results ......................................... 118
    4.5.1 Bending results ..................................... 118
    4.5.2 Multiple inclusion problems .......................... 124

5 Embedded Meshes in Non-Linear Elasticity and Other Extensions 129
  5.1 Introduction ............................................... 129
  5.2 Literature Review ......................................... 130
  5.3 Problem Formulation ...................................... 131
    5.3.1 Hyperelastic boundary value problem .................... 134
    5.3.2 Statement of virtual work for the uncoupled problem .... 135
    5.3.3 Discrete form ........................................ 136
  5.4 DG Derivative in the Non-Linear Case ...................... 139
    5.4.1 Discrete form ........................................ 140
5.4.2 Pseudocode for non-linear DG element .......................... 143
5.5 Results ................................................................. 145
  5.5.1 Bending, and convergence ..................................... 146
  5.5.2 Golf ball .......................................................... 150
  5.5.3 Reinforced conveyor belt problem ............................ 153
5.6 A Return to Multi-Physics and fluid/structure Interaction .... 156

6 Conclusions .............................................................. 162

Bibliography ............................................................... 165

Biography ................................................................. 172
List of Tables

2.1 Drag coefficient for the current study versus literature benchmarks . . 35

2.2 Experimental and numerical comparisons of decaying roll frequencies
for 6 cases. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 41
List of Figures

1.1 Computational contact mechanics; reproduced with permission from Duke Computational Mechanics Laboratory. See Puso and Laursen (2004) ................................................................. 2

1.2 Phase evolution in hydrogels; reproduced with permission from Duke Computational Mechanics Laboratory. See Dolbow et al. (2005) . . . 2

1.3 Fluid/structure interaction with an ALE (left) versus an embedded approach (right). Reproduced with permission from simulations at Lawrence Livermore National Laboratory ................................. 3

1.4 Solenoid used to propel a magnetic projectile. Image used with permission from Lawrence Livermore National Laboratory . . . . . . . 4

2.1 Schematic of point absorber style wave energy harvester . . . . . . 11

2.2 Two dimensional domain including two-phase boundary (φ = 0) and rigid body, B ................................................................. 16

2.3 MAC computational grid with face centered velocities and cell centered pressures ......................................................... 31

2.4 Velocity contours for Stokes flow around a confined cylinder .... 33

2.5 Pressure contours for Stokes flow around a confined cylinder .... 33

2.6 Surface pressures and friction for a confined cylinder in Stokes flow . 34

2.7 Lift coefficient versus time for flow around a fixed circular cylinder for Re = 200 ................................................................. 34

2.8 Wake formation and vortex flow over a cylinder for a) Re = 5 b) Re = 50 c) Re = 100 and d) Re = 200, Time = 3s, u-velocity is shown . 35

2.9 Velocity and pressure (minus hydrostatic pressure) contours for sedimentation of a cylindrical disk at t = 0.4s ......................... 37
2.10 Position and velocity of disk as a function of time versus benchmark solutions ........................................... 37
2.11 Velocity profiles for a buoyant cylinder in successively finer grids ........... 38
2.12 Error in terminal velocity for a buoyant cylinder; the error decreases at an order slightly greater than 1. A fine grid with a spatially converged solution is used as the reference solution. ............................... 38
2.13 Decay of heave for a buoyant cylinder for a) the current study and b) Donescu and Virgin (2001) ................................................. 40
2.14 Physical dimensions of barge shape for roll calculations .......................... 41
2.15 Example time histories of free decay of ship roll motion for previous (a) and current studies with second-order (b) versus first-order upwinding (c) ................................................................. 42
2.16 Setup for wave tank simulation problem ........................................... 42
2.17 Snapshots of wave tank simulations at t = 0.0s, 1.4 s, 2.4 s, 5.2 s and 8.4 s .................................................. 43
2.18 Zoomed in snapshots of wave tank simulations at t = 1.4 s, 2.4 s, 5.2 s and 8.4 s .................................................. 44
2.19 Heave motion of center of mass of a free floating buoy in wave tank ........... 44
2.20 Snapshots of wave tank simulations with rotations at t = 0.05s, 2.6s, 5.1s .................................................. 46
2.21 Rotation of rigid body and internal rotational spring/mass system in wave tank .................................................. 47
3.1 Example of embedded finite element grids .......................................... 50
3.2 Two body problem domain .......................................................... 55
3.3 Schematic of partially integrated cut element ....................................... 66
3.4 Decomposition of $\Gamma_s$ into mortar segments (a), Local coordinates for mortar integrals (b) .................................................. 70
3.5 Degrees of freedom which contribute to Nitsche interfacial stiffness as calculated over one mortar segment .................................................. 73
3.6 Straight and curved interfaces ........................................................ 77
5.10 Conforming solution for reinforced conveyor belt problem . . . . . . . 154
5.11 Mortar solution for reinforced conveyor belt problem . . . . . . . . . 155
5.12 DG derivative solution for reinforced conveyor belt problem . . . . . 155
5.13 Elastic wave propagation through an embedded mesh. Image used with permission from Lawrence Livermore National Laboratory . . . 158
5.14 Blasting loading of an embedded versus ALE mesh. Image used with permission from Lawrence Livermore National Laboratory . . . . . . 159
## List of Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALE</td>
<td>Arbitrary Lagrangian-Eulerian</td>
</tr>
<tr>
<td>CFD</td>
<td>Computational Fluid Dynamics</td>
</tr>
<tr>
<td>DG</td>
<td>Discontinuous Galerkin</td>
</tr>
<tr>
<td>FEA</td>
<td>Finite Element Analysis</td>
</tr>
<tr>
<td>PK2</td>
<td>Second Piola-Kirchoff (stress)</td>
</tr>
<tr>
<td>XFEM</td>
<td>Extended finite element method</td>
</tr>
</tbody>
</table>
Acknowledgements

I would like to express deep gratitude to my advisor, Dr. Tod Laursen, (Professor and President of Khalifa University of Science, Technology, and Research, Abu Dhabi) for his mentorship. A graduate student’s experience is greatly shaped by his or her mentor, and I am truly lucky to have had Professor Laursen usher me into engineering adulthood. In addition, I would like to acknowledge the strong co-mentorship role Professor John Dolbow offered to me.

This work was made possible by the financial support of the Department of Defense through a National Defense Science and Engineering Fellowship, Air Force Office of Scientific Research Grant FA9550-06-1-0108, The German Academic Exchange Service (DAAD) and Lawrence Livermore National Laboratory. The support is gratefully acknowledged. I am also grateful to Professor Barbara Wohlmuth, now a professor at the Technische Universität München, Munich, Germany, for inviting me to spend a semester improving my math at the Universität Stuttgart, in Stuttgart, Germany.

I have had the advantage of working with many excellent collaborators. I would like to thank Professor Peter Mucha from the University of North Carolina math department for his many meetings with me, and fruitful ideas. The two most productive periods in my graduate career were summers spent at Lawrence Livermore National Laboratory under the mentorship of Dr. Mike Puso. I am very grateful for his style of mentorship, for the fascinating challenges he provided, and his ability to
push me forward. I would also like to thank Dr. Robert Ferencz for facilitating my time at the lab. The other members of my committee, Professors Brian Mann, Anita Layton, and Jeff Scruggs, were all willing to sit down with me and talk about my work when necessary, and I appreciate their willingness to play a role in the process.

Professor Ilinca Stanciulescu, now at Rice University, has been everything I could hope for in an academic sibling and friend. I appreciate the time she has spent reading chapters, evaluating my presentations, and providing overall support for my academic ambitions. I owe an additional debt of gratitude to many of my friends and colleagues at Duke for shared expertise and friendship, in particular Justin Pogacnik, Monica Carlson, Dr. Rebecca Kunder, Dr. Maria Tanner, Dr. Bonnie Lai and Dr. Martin Heutefeuille.

I would not ever have made it to Duke without the encouragement of my very first mentors in engineering from Sandia National Laboratories, who have since provided consistent support. I would like to express a heartfelt thanks to Kenneth Gwinn, Jeff Gruda and Dr. Martin Heinstein for sending me down this path.

Finally, I am lost without the unyielding love of my family, which I have always been able to count on. Thank you.
1

Introduction

1.1 Interfaces in Computational Mechanics

The subject of this thesis is the stable numerical treatment of unconventional interfaces in computational mechanics problems. Specifically, we propose new techniques for the simulation of mechanics problems in which it makes sense for one physical domain to be embedded in another.

Engineering mechanics problems involving interfaces, whether physical or introduced by numerical methodologies, are commonplace. Just a few examples include fracture and fault mechanics, classical contact-impact (see Figure 1.1), phase boundary propagation (Figure 1.2), and fluid/structure interaction (Figure 1.3). Generally, interface treatment in computational mechanics can be divided into two classes. In the first the boundary is explicitly gridded (such as contact mechanics). We will call this interface tracking. In a second class of methods, the interface is not restrained to the computational grid, and alternative techniques are used to calculate its position and properties. Common examples for this scenario are the extended finite element method (Moës et al., 1999), used originally for crack propagation problems,
and finite difference based volume of fluid methods (Hirt and Nichols, 1981) used to calculate the free surface position of a flow. Though we draw lessons in this document about numerical stability from classical contact/impact problems, we are exclusively concerned with simulations in which the interface of concern is not aligned with the primary computational grid.
Figure 1.3: Fluid/structure interaction with an ALE (left) versus an embedded approach (right). Reproduced with permission from simulations at Lawrence Livermore National Laboratory.

The primary motivation for the techniques developed herein is their applicability to robust simulation of multi-physics problems. Three motivating examples are given as follows:

1. A buoy is moved by ocean waves. We desire the ability to predict the overall behavior (in terms of displacements, velocities, and accelerations) of a buoy in any wave environment. There is a growing need for these kind of predictive techniques due to rising interest and investment in using buoy-like structures to harvest energy from ocean waves and convert it to electricity for human use. This is a case in which equations of fluid dynamics need to be coupled with equations of rigid body motion, though they are traditionally treated with very different simulation techniques.

2. A shell structure is subjected to a blast loading. Here we might consider airplane fuselages as an example. It is common to use an Arbitrary Lagrangian-Eulerian (ALE) technique for this problem but ALE fails when the shell starts to experience large deformations/failure.
3. A rail gun uses magnetic fields to propel a projectile (see Figure 1.4). Here, the magnetic field must be computed in the air and the metal projectile which moves through the air. The large motion of the projectile makes it intractable to create a conforming interface between the grid for the magnetic field and the projectile.

It is a secondary goal of this work to provide insight into interface stability for wherever it may be appropriately applied in the broader field of computational mechanics.

1.2 Specific Techniques Developed

Considerable attention is given to the first motivating example of buoys in ocean waves. Despite the healthy state of the art when it comes to treating either structural/mechanical/dynamical systems or thermal/fluid systems, the current techniques for simulating coupled systems tend to be less generalized and less mature. This is especially the case because fluid solvers tend to be based on Eulerian methods (where meshes are fixed and do not track particles) and structural solvers are usu-
ally Lagrangian (in which the grid moves with the material). We are considering a naturally two-way coupled systems. In the ocean harvesting application, waves push on the massive buoy-like structures, which in turn makes a non-trivial impact on the wave behavior - their motions are mutually influential. We will present a stand-alone method for the simulation of rigid body motion in two-phase incompressible flow, based on completely incorporating both the Eulerian and Lagrangian frameworks for their respective strengths. The algorithm is based upon a structured finite difference fluid solver, and large motion Lagrangian mechanics for the rigid bodies. The key idea in this method is that the computational domain of the bodies is embedded in the background grid.

The remainder of the thesis will concentrate on methods for embedding one finite element mesh in another. It addresses the fidelity of finite element solutions near embedded interfaces. Much of the work presented here has been done under the supervision of Lawrence Livermore National Laboratory, where embedded techniques are being investigated for finite element analysis treatment of fluid/structure interaction simulations, as well as other coupled physics problems. In the following studies, an embedded mesh technique is investigated in a solid mechanics framework, i.e. one solid structure embedded in another. Many of the lessons from this context should be applicable to other model equations, particularly in issues concerning algorithmic surface constraints.

It is shown that standard algorithms for tying finite element domains together, namely the set of techniques known as mortar methods, can sometimes break down in the context of embedded grids in such a way that the underlying mechanics of the problem is disrupted. For a good review of mortar methods see Laursen et al. (2010). We will describe an alternate method to the mortar method, known as Nitsche’s method (Nitsche, 1971), which has been in development for elliptical PDE’s and linear finite elements over the past decade, and apply it to the problem of
embedded domains for elastic problems. An extension of the technique for non-linear solid mechanics problems based on discontinuous Galerkin (DG) methods will then be demonstrated. The emphasis will be on the consistency, robustness, stability, convergence and accuracy of these methods.

1.3 Previous Examples in Computationally Embedded Domains

Many techniques exist for the numerical coupling of equations representing disparate physical behavior. Frequently a technique is developed to be tailored to a specific problem. If we take fluid/structure interaction as an example, a simple idea would be to have two separate domains, a fixed and gridded boundary between the two, and a staggered scheme in which a fluid solver calculates motion of the fluid, and then estimates the resultant load on a solid surface. This might be reasonable, for example, for a simulation of wave sloshing against a dam, but would not be sufficient for a problem such as wave-energy harvesting. A more sophisticated approach might be based on the Arbitrary Lagrangian-Eulerian (ALE) approach (see Nomura and Hughes, 1992), in which a background fluid mesh would advect itself in response to a deforming boundary. ALE techniques were the state of the art for many years in finite element fluid/structure interaction, but still tend to break down in cases of large motion of solid structures. Recently there has been a great deal of research into embedded techniques, that is to say, where one domain overlaps another domain, and the equations are solved either monolithically or staggered, but in such a way that the discretizations of each domain do not have to alter to accommodate the other domain (e.g. Zhang et al., 2004; Zhao et al., 2008; Gerstenberger and Wall, 2008). The assumption of this document is that an embedded domain technique is a more natural and flexible framework for two-way coupled systems than either a fixed grid or ALE framework.

A vast amount of literature exists on such approaches, ranging from fictitious
domain methods (FD), which use Lagrange multipliers to enforce continuity between a body or obstacle and a background grid (e.g. Glowinski et al., 1994; Baaijens, 2001); to a variety of overset grid methods (see Brezzi et al., 2001; Houzeaux and Codina, 2003; Zhang et al., 2004) which, to a large extent, follow ideas going back at least to Noh (1964); to the Arlequin method (see Ben Dhia and Rateau, 2005); to the rapidly growing literature on how interface constraints and constitutive laws can be imposed within the context of extended finite element methods and other related approaches (see Sanders et al., 2009; Bechét et al., 2009; Gerstenberger and Wall, 2008; Dolbow and Harari, 2009).

Our approach is most closely related to an embedded mesh technique from Baaijens (2001). The proposed improvements for embedded mesh techniques concentrate on the numerical description of continuity constraints, and their stable application.

Though all of the numerical techniques presented by this document will be concerned with problems of embedded domains, they are disparate in the sense that the first method deals with fluid equations, finite differences and Lagrangian rigid body mechanics, whereas the second set of methods deals with finite elements in deformable solid mechanics. Due to the disparate nature of the topics, a more thorough introduction to the specific methods will be given at the beginning of the individual chapters, including a literature review. The document will be organized as follows: In Chapter 2 we will introduce the motivating problem of wave energy harvesting and present the algorithm for rigid bodies in incompressible two phase flow. Chapter 3 will discuss embedded finite element methods and present the problem in overlapping domains in linear elasticity. In the unique embedded domains context, the weakness of the most traditional methods of domain tying will be demonstrated. An alternate method for domain tying, Nitsche’s method, will be studied for its numerical properties. In Chapter 4 we will show that Nitsche’s method can be recast in a form that draws parallels to the discontinuous Galerkin method for finite element solid
mechanics, which gives the method greater flexibility and suggests a methodology for extension to other systems. In Chapter 5 we will use the techniques developed in Chapter 4 to extend the methodologies to problems in embedded domains for non-linear elasticity. We give a preview into how the work in embedded elasticity hints at solutions into the broader set of work being done in fluid/structure interaction in finite elements. Chapter 6 will conclude and summarize the work.
2.1 Introduction

Computational treatment of submerged bodies, particularly in the presence of free surfaces and/or breaking waves, poses several modeling challenges. The objective of this Chapter is to present a robust numerical algorithm for simulation of rigid structures interacting with incompressible two-phase viscous fluid flow in the presence of free surfaces. The numerical method is formulated in such a way that extra attenuating forces and internal dynamics can be easily introduced into the rigid body equations of motion. A motivating example where these systems might be of interest is off shore wave energy harvesting systems in which a floating structure converts mechanical oscillations to electrical energy.

2.1.1 Wave energy harvesting

Surface ocean waves have a high energy density compared to other renewable power sources, but they also pose greater physical challenges for the process of harnessing the available energy. For buoy style wave energy converters, questions of down
stream effects, optimal fleet configuration, extreme event survivability, and buoy dynamics provide ongoing challenges for a young community. The relative energy density compared to fossil fuels is still small, so efficiency is very important. For all of these reasons, accurate computational models of structures submerged in an ocean environment have the potential to ease the design and prototyping process for these systems.

Figure 2.1 gives a conceptual scheme for a point absorber style wave energy converter. In this style of design (one among dozens), a buoy system is anchored to the ocean floor, and a free outer float oscillates around the central spar. The relative motion of the two bodies, generally designed to be resonant at the dominant wave period, runs a generator. The electricity would generally be transmitted via lines on the ocean floor in the same style as off-shore wind energy systems. A good overview of the general state of the art and challenges associated with these kinds of systems is available in Scruggs and Jacob (2009).

At the most basic level, a numerical model for a point absorber would need to be able to resolve a flow field in the near vicinity of a moving structure. Because the devices are buoyant, the model would need to be able to handle a free surface. Depending on the desired level of physical fidelity, there are many more levels of complexity that could be possible. If wind is considered important, a two-phase model is necessary. A global model of array response might also be desirable. A great deal of the important design choices pertain to the device control systems, and coupling physical mechanical models to control system models is also an important avenue of research.

We will address some of the challenges of a numerical model in which we desire to resolve two-phase flow and a rigid body under large motion.
2.1.2 Literature review

Many of the challenges of fluid/structure interaction have been addressed in the numerical methods community (see Peskin, 1977; Legay et al., 2006; Zhang et al., 2004; Zhao et al., 2008). In general terms, it is more common for the computational fluid dynamics (CFD) community to apply finite difference (FD) techniques in an Eulerian framework to the spatial discretization of the Navier Stokes equations. There has also been a recent effort within the finite difference community to accurately resolve free surfaces and two-phase flows (e.g., Sussman et al., 1994). For solid mechanics, on the other hand, the numerical community usually employs Finite Element Analysis (FEA) with Lagrangian coordinate systems.

This work has been developed with an eye towards employing the full benefits of both the Eulerian and Lagrangian coordinate systems for the analysis of fluids and
solids, respectively. It is built upon existing techniques for the robust representation of free surfaces in two phase flow with finite differences (see, for example, Sussman et al. (1994)), and introduces the idea that both Eulerian and Lagrangian frameworks may be used to formulate and solve equations of motion, thus inheriting the advantages of both schemes. It is our intent that the computation strategy presented herein be extendable in a robust manner to the analysis of deformable bodies under Lagrangian coordinates, embedded in an Eulerian fluid framework.

Many schemes for fluid/structure interaction involve formulating and solving all of the equations of motion on a structured cartesian finite difference grid. In these methods, overlaying Lagrangian representations of the solid track its location, but are not used explicitly to update its position. Techniques have been developed for transferring forces or representing constraints from an overlapping solid reference frame to the fluid reference frame. Immersed boundary methods, pioneered in the 1970’s by Charles Peskin to model blood flow in the heart (Peskin, 1977), represent the action of the solid on the fluid as singular forces applied at fluid nodes in the region of an interface. In Peskin’s methods and extensions such as the Immersed Finite Element Method (Lee et al., 2008), an immersed solid is advected with a background fluid velocity. A stress is thus generated in the solid and calculated with some constitutive relationship, after which the forces are appropriately transferred back to the fluid grid for subsequent time steps. Though the methods were developed for generally deformable bodies, very stiff constitutive relationships, or other modifications, are used to approximate rigid body motion (Lee et al., 2008; Zhang and Gay, 2008).

In a different approach, fictitious domain methods, or domain-embedding methods, have been applied to fluid/structure interaction in various ways starting in the early 90’s (such as Glowinski et al., 1999, 2001; Sharma and Patankar, 2005). Also referred to as Distributed Lagrange Multiplier approaches, fictitious domain methods
simplify a geometrically irregular domain for a PDE by extending the computational
domain to a larger, more geometrically regular, region and applying boundary con-
ditions on the original boundary. Glowinski applied the method to rigid particles in
a fluid flow by “filling” the rigid particle domains with fluid, solving the equations
of motion on the entire domain, and enforcing rigidity of the fluid in the particle
domain through a Lagrange multiplier field. The multiplier can be seen as a body
force required to enforce the rigidity constraint on the particles. By formulating the
equations of motions of the fluid simultaneously with the equations of the particles
and solving them together, boundary conditions between the two cancel and are
never explicitly calculated. Glowinski’s methods are based on finite elements. The
Rigid Fluid Method (Carlson et al., 2004) is a finite difference approach based on
the same principles, implemented in the presence of a model free surface (simulating
single-phase fluid motion).

In all of the preceding methods the equations of motion for both the solid and
fluid are calculated on an Eulerian grid. Though Lagrangian coordinates may be
employed to track the motion of a solid, and post-calculate stresses, they are not
used to calculate the motion or deformation itself, as would normally be done for
rigid body or solid mechanics problems. The paradigm of these methods is that the
solid is advected with the fluid motion.

A more recent body of work has begun studying rigid body/fluid interaction
using Lagrangian equations of motion for the rigid body. Additionally, an immersed
boundary method is proposed in Kim and Choi (2006), in which the equations of
motion for the body are formulated in a non-inertial frame of reference. The method
would be difficult to extend to multiple moving bodies in a given flow. The authors
of Borazjani et al. (2008) have also proposed a method with Lagrangian equations of
motion for rigid bodies in a fully three-dimensional flow, and have included a study
of the effects of strong versus loose coupling of the fluid and rigid body equations.
In this work we propose a fundamentally different method of formulating the forcing terms of the rigid body equations.

2.1.3 An algorithm with fully Lagrangian equations of motion

The following method will use the fully Lagrangian equations of motion to solve for the position and velocity of the rigid body, and in a reversal of the usual ideas, the local fluid will be constrained to move with the calculated solid motion. The approach has several advantages:

- For rigid particles, it is trivial to include the effects of the dynamics of mechanisms, including (for example) attached springs, dampers, and masses.

- In addition to fully coupled fluid/solid motion, the method can be easily applied to the one-way coupled problem in which the prescribed motion of a body forces the motion of the surrounding fluid. An example of this type will be presented in this paper, where the wave train in a wave tank is driven by an oscillating wedge.

- In the case of deformable solids, for which the method is intended to be extensible, a Lagrangian equation of motion would facilitate the full use of finite elements with all of the advantages therein, especially in cases where the strength of the material is of primary importance.

- Fully Eulerian schemes must interpolate values of a fluid velocity onto a solid domain. In the case of finite differences, it is not necessarily clear what a proper interpolation scheme would be, whereas the interpolation from a Lagrangian rigid body or a deformable finite element mesh onto fluid nodes is trivial, as in those cases the value of the velocity field is known a priori.
For the scope of this work, with wave energy harvesters as the motivating application, the most important advantage may be the lack of need to interpolate any fluid values onto the solid domain in order to advect it. For continued work, the ease of representation of complicated internal dynamics, including multiple interconnected spring mass systems, as well as power conversion systems that act to damp the mechanical motion, will also be of primary importance. The current scheme is benchmarked in two dimensions, but presented in a general way such that the algorithm is extendable to three dimensional flows. The solid/fluid interaction method has been built upon stable and spatially second-order techniques for incompressible two-phase flow (from Sussman, 2003; Morgan, 2005).

2.2 Governing Equations

2.2.1 Two phase incompressible flow in an Eulerian framework

We consider a 2D fluid domain, $\Omega$, with boundary $\partial \Omega$ as depicted in Figure 2.2. The differential equations of conservation of momentum in a continuum for a fluid are

$$\rho_f \left( \frac{Du}{Dt} \right) = \nabla \cdot \sigma_F + \rho_f g$$

(2.1)

where $\rho_f$ is the density of the fluid, $u(x)$ is the velocity of the fluid at any position, $x$, and $\sigma_F$ is the fluid stress. For an incompressible Newtonian viscous fluid, the stress is taken to be a combination of hydrostatic and shear components,

$$\sigma_F = -pI + \mu_f (\nabla u + \nabla^T u),$$

(2.2)

where $p$ is the hydrostatic pressure in the fluid and $\mu_f$ is the viscosity. Substitution of the fluid stress into the momentum equation for an Eulerian framework, and including a condition on conservation of mass, gives the familiar Navier-Stokes equations for
Figure 2.2: Two dimensional domain including two-phase boundary ($\phi = 0$) and rigid body, $B$

fluid velocity and pressure, which are

$$\rho_f \left( \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) = -\nabla p + \mu_f (\Delta \mathbf{u}) + \rho_f \mathbf{g}, \quad (2.3)$$

and

$$\nabla \cdot \mathbf{u} = 0. \quad (2.4)$$

In the level set method (Sussman et al., 1994) for tracking two-phase flow, the conservation of momentum for the fluid is given as

$$\rho_f \left( \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) = -\nabla p + \mu_f (\Delta \mathbf{u}) + \rho_f \mathbf{g} + \tau \kappa \delta(\mathbf{x}) \mathbf{n}, \quad (2.5)$$

where the last term represents a local surface tension force concentrated at the location of the phase interface. In this notation $\tau$ is the surface tension, $\kappa$ is the curvature of the interface, $\mathbf{n}$ is the surface normal at the interface, and $\delta(\mathbf{x})$ is the Dirac delta function at the location of the interface. The location between the phases is tracked as the zero contour of a scalar valued function $\phi(\mathbf{x})$ which is advected with the fluid flow:

$$\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi = 0. \quad (2.6)$$
At any position, $\mathbf{x}$, where $\phi(\mathbf{x}) > 0$ the fluid is considered to be in a liquid phase, and the gas phase exists where $\phi(\mathbf{x}) < 0$. Fluid properties such as density or viscosity in either phase can then be expressed as a function of location in the domain. For example, in the case of density where $\rho_l$ is the density of the liquid and $\rho_g$ is the density of the gas,

$$\rho_f = \rho_l H(\phi) + \rho_g (1 - H(\phi))$$

(2.7)

where the function $H(\phi(\mathbf{x}))$ is

$$H(\phi(\mathbf{x})) = \begin{cases} 0 & \phi < 0 \\ 1 & \phi > 0 \end{cases}.$$

A similar expression holds for viscosity. Finally, the system of equations representing the two-phase fluid flow is completed through appropriate boundary and initial conditions on velocity and pressure:

$$u(t, \mathbf{x})|_{t=0} = u(0, \mathbf{x})$$

$$p(t, \mathbf{x})|_{t=0} = p(0, \mathbf{x})$$

$$u(t, \partial \Omega) = u(t, \partial \Omega)$$

$$\text{and,}$$

$$p(t, \partial \Omega) = \bar{p}(t, \partial \Omega).$$

2.2.2 Rigid body equations of motion

The fluid equations in the previous section were constructed in an Eulerian framework in anticipation of using a standard finite difference technique for their numerical approximation. We construct rigid body equations of motion in a Lagrangian, or particle tracking, framework and will later discuss the consequences of needing to merge the ideas.

We consider a solid body with coordinates $\mathbf{X}$ in a reference frame with basis vectors $\{\mathbf{E}_1, \mathbf{E}_2, \mathbf{E}_3\}$ at $t = 0$. For convenience we set the origin to be the center of
mass of the body at $t = 0$. Rigid motion of the body can be expressed through the mapping

$$\varphi(t) = d(t) + \Lambda(t)X,$$

(2.9)

where $d$ is a time dependent vector (reflecting a displacement), and $\Lambda$ is an orthogonal transformation (related to the rotation of the body). While the translational part of the mapping, $d$, belongs to a cartesian space, $d(t) \in \mathbb{R}^3$, the rotational degrees of freedom are described by the proper orthogonal matrices belonging to the special orthogonal group, $\Lambda \in \text{SO}(3)$. Careful parameterization of this space is needed for subsequent algorithmic developments.

The velocity field corresponding to rigid body motion is

$$\dot{\varphi}(t) = v(t) + \dot{\Lambda}(t)X,$$

(2.10)

where $v = \dot{d}$, and $(\cdot)' = \frac{d}{dt}$. Since $\Lambda$ represents an orthogonal transformation we can say, $\dot{\Lambda}(t) = \Lambda(t)\hat{\omega}(t)$, where $\hat{\omega} = \Lambda^T(t)\dot{\Lambda}(t)$ is a skew symmetric matrix, the components of which represent an angular velocity relative to the reference frame. Here, we introduce an axial vector, $\omega$, associated with $\hat{\omega}$ in the following way:

$$[\hat{\omega}] = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}, \quad \{\omega\} = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}. $$

We refer to the vector $\omega$ as the convected angular velocity of the body.

The classical equations of motion a rigid body, in this case for an immersed rigid solid, $B$, with boundary $\Gamma^*$ completely immersed in a fluid domain $\Omega$ are:

$$\frac{dp}{dt} = m\frac{dv}{dt} = \int_{\Gamma^*} t\,d\Gamma + mg,$$

(2.11)

and

$$\frac{d\pi}{dt} = \int_{\Gamma^*} (r \times t)\,d\Gamma. $$

(2.12)
Here, \( \mathbf{p} = m \mathbf{v} \) is the total linear momentum, \( m \) is the total mass of the body, \( \mathbf{v} \) is the body’s translational velocity as measured at the center of mass, \( \mathbf{r} \) is the position vector from the center of mass to a point on the solid boundary \( \Gamma^* \), and \( \mathbf{t} \) are any surface tractions. The quantity \( \boldsymbol{\pi} \) is the total spatial angular momentum relative to the center of mass, defined via \( \boldsymbol{\pi}(t) = \mathbf{\Lambda} \mathbf{J} \mathbf{\omega} \), where \( \mathbf{J} \) is the body’s rotational inertial dyadic, somewhat analogous to mass for a rotational system. The right side of (2.11) represents the total applied force at the center of mass (a combination of surface forces and gravity), whereas the right side of (2.12) represents the total applied torque upon a current (rotated) configuration of the body. We would rather do our computations in the reference frame, however, which gives:

\[
\frac{d\boldsymbol{\Pi}}{dt} = \mathbf{\Lambda}^T(t) \left( \int_{\Gamma^*} (\mathbf{r} \times \mathbf{t}) \, d\Gamma \right),
\]

(2.13)

where \( \boldsymbol{\Pi}(t) = \mathbf{J} \mathbf{\omega} \) is the convected angular momentum relative to the center of mass.

The full systems of equations corresponding to rigid body motion for both translation (\( \mathbf{d}, \mathbf{v}, \) and \( \mathbf{a} \)) and rotation (\( \mathbf{\Lambda}, \mathbf{\omega}, \) and \( \mathbf{\alpha} \)) are given:

\[
\begin{align*}
\dot{\mathbf{d}} &= \mathbf{v} \\
\dot{\mathbf{v}} &= \mathbf{a} \\
ma &= \int_{\Gamma^*} \mathbf{t} \, d\Gamma + mg
\end{align*}
\]

(2.14a, b, c)
\[
\begin{align*}
\dot{\Lambda} &= \Lambda \dot{\omega} \quad \text{(2.15a)} \\
\dot{\omega} &= \alpha \quad \text{(2.15b)} \\
J \alpha + \omega \times J \omega &= \Lambda^T(t) \left( \int_{\Gamma^s} (r \times t) \, d\Gamma \right) \quad \text{(2.15c)}
\end{align*}
\]

Initial conditions on the particle position \((d_{t=0} = d_0, \Lambda_{t=0} = \Lambda_0,\) and velocity \((v_{t=0} = v_0, \omega_{t=0} = \omega_0)\) complete the physical system.

Our challenge, which is avoided in existing methods where these equations are folded completely into the fluid framework, is to determine the functional form of the forcing, \(t\), on the particles from the physical coupling with the surrounding fluid.

2.2.3 No slip boundary between solid and fluid

The fluid and solid equations are coupled through the no-slip condition

\[
\mathbf{u} = \mathbf{v} + (\mathbf{r} \times \omega) \quad \text{on } \Gamma^s.
\]

(2.16)

In addition there is a balance of stresses

\[
\sigma_f \mathbf{n} = t \quad \text{on } \Gamma^s,
\]

(2.17)

which states that the traction on the particle surface can be determined by balance of forces with the internal fluid stresses.

2.3 Fluid/Solid Interaction Model

As with many numerical schemes (Glowinski et al., 1999, 2001; Sharma and Patankar, 2005; Carlson et al., 2004; Lee et al., 2008) we approach the problem by first filling the solid domain with artificial fluid, which will be used to facilitate the interaction...
between the two. The kinematic boundary condition on the interface between the solid and fluid is extended to the entire domain of the artificial fluid:

\[ \mathbf{u} = \mathbf{v} + (\mathbf{r} \times \mathbf{\omega}) \text{ on } B. \]  

(2.18)

Additionally, the dynamic boundary condition is expanded to indicate that the values of fluid stress both interior and exterior to the particle domain must be equal to the surface traction and each other at the particle interface:

\[ \sigma^i_F \mathbf{n} = \sigma^o_F \mathbf{n} = \mathbf{t} \text{ on } \Gamma^* . \]  

(2.19)

The superscript \( i \) indicates a value that is supported only interior to the particle domain \( B \), in the artificial fluid. Similarly, a superscript \( o \) will indicate a fluid property in the domain exterior to the particle (i.e., \( \Omega \setminus B \)). The balance of stresses (2.19) allows us to determine the forcing terms for the rigid body equations of motion. We substitute the fluid stress for the solid traction over the interface \( \Gamma^* \) in eqns (2.11) and (2.12) to yield

\[ m \frac{d\mathbf{v}}{dt} = \int_{\Gamma^*} (\sigma_F \mathbf{n}) \, d\Gamma^* + mg \]  

(2.20)

and

\[ \frac{d\Pi}{dt} = \Lambda^T(t) \left( \int_{\Gamma^*} (\mathbf{r} \times (\sigma_F \mathbf{n})) \, d\Gamma^* \right) , \]  

(2.21)

It should be noted that here we have the choice of either \( \sigma^i_F \) or \( \sigma^o_F \); they are equivalent on the boundary by definition. Equation (2.20) is converted to a volume integral through simple application of the divergence theorem:

\[ m \frac{d\mathbf{v}}{dt} = \int_B (\nabla \cdot \sigma_F) \, dB + mg \]  

(2.22)
which is equivalent, through (2.1), to

\[ m \frac{dv}{dt} = \int_B \left\{ \rho_f \left( \frac{Du}{Dt} \right) - \rho_f g \right\} dB + mg \]  \hspace{1cm} (2.23)

Additionally, it can be shown, given the symmetry of \( \sigma_F \), that (2.21) is equivalent to

\[ \frac{d\Pi}{dt} = \Lambda^T(t) \int_B \mathbf{r} \times \left\{ \rho_f \left( \frac{Du}{Dt} \right) - \rho_f g \right\} dB. \] \hspace{1cm} (2.24)

The complete equations of motion for the coupled system are given by (2.5) for the fluid in the domain \( \Omega \setminus B \), (2.18) for the fluid in the domain \( B \), and (2.23) and (2.24) for the motion of the rigid bodies.

2.4 Numerical Method

2.4.1 Time integration scheme

Given a set of initial conditions for the fluid domain, \( u_n \), and \( p_n \), we wish to advance the solution of all the variables from \( t = n \) to \( t = n + 1 \). Time integration is based on a classic Chorin pressure projection scheme Chorin (1968). First, an intermediate velocity is calculated over the entire domain \( \Omega \).

\[ u^* = u^n + \Delta t \left[ \frac{\mu_f(\Delta u)}{\rho_f} + \frac{\tau_k \delta(d) n}{\rho_f} + g \right], \] \hspace{1cm} (2.25)

followed by a pressure projection step for \( p^{n+1} \), again, in the entire domain,

\[ \frac{\nabla^2 p^{n+1}}{\rho} = \frac{\nabla \cdot u^*}{\Delta t} \] \hspace{1cm} (2.26)

and finally a correction to the velocity field,

\[ \hat{u} = u^* + \frac{\Delta t \nabla p^{n+1}}{\rho_f}. \] \hspace{1cm} (2.27)
The solution is now divergence free, but does not represent a rigid velocity field in the domain, B. Everywhere else in the domain \( u^{n+1} = \hat{u} \). The last step would be to advance the solution in that domain from \( \hat{u} \) to \( u^{n+1} \). In fact, this is a trivial step, because we know that the velocity field in B will be given by the particle velocity \( v^{n+1} \).

This is a point of fundamental departure from previous rigid body/fluid techniques, which advect the rigid body with the fluid, and compute forces on the background fluid mesh. In our case, we are advecting the fluid with the solid motion, and calculating forces which are applied to the solid equations of motion. To compute that particle velocity, however, we do need to be able to calculate the quantity

\[
\int_B \left[ \rho_f \frac{Dn^{n+1}}{Dt} - \rho_f g \right] dB.
\]

It will be demonstrated that this can be computed with knowledge of solution in the rest of the fluid field. As a point of notation, \( \hat{\sigma} \) should be understood to be the fluid stress (2.2) at the intermediate time step \( \hat{t} \).

If the fluid stress for the artificial and real fluid is balanced on the boundary between them:

\[
\sigma_\Gamma^i n = \sigma_\Gamma^o n = t \text{ on } \Gamma^*,
\]

it follows that the integrals are also equal over the interface.

\[
\int_\Gamma \sigma_\Gamma^i n d\Gamma = \int_\Gamma \sigma_\Gamma^o n d\Gamma \text{ on } \Gamma^*.
\]

The last step of the fluid solve, going from \( \hat{u} \) to \( u^{n+1} \) will only affect the velocity field in the artificial fluid, meaning \( \hat{u} = u^{n+1} \) outside the particle domain, thus,
\[
\int_B \left[ \rho_f \frac{D\mathbf{u}^{n+1}}{Dt} - \rho_f \mathbf{g} \right] dB = \int_{\Gamma^*} \sigma_F^{i,n+1} \mathbf{n} d\Gamma^* \\
= \int_{\Gamma^*} \sigma_F^{0,n+1} \mathbf{n} d\Gamma^* \\
= \int_{\Gamma^*} \hat{\sigma}_F \mathbf{n} d\Gamma^* \\
= \int_B \left[ \rho \frac{D\hat{\mathbf{u}}}{Dt} - \rho \mathbf{g} \right] dB. \tag{2.31}
\]

In other words, the results of the integral on the left side of equation (2.30), which is needed to complete the rigid body equations of motion, can be known from an evaluation of the equivalent integral at the intermediate time \( \hat{t} \). The term \( \frac{D\hat{\mathbf{u}}}{Dt} = \frac{\hat{\Delta} \hat{\mathbf{u}}}{\hat{\Delta}t} + \hat{\mathbf{u}} \cdot \nabla \hat{\mathbf{u}} \) is divided into inertial and convective components. The inertial term is discretized with a forward Euler strategy consistent with the rest of the method, \( \frac{\Delta \hat{\mathbf{u}}}{\Delta t} = \frac{\mathbf{u} - \mathbf{u}^n}{\Delta t} \). Similar expressions may be used, as desired, to couple fluid forces to solids which are deformable or have internal dynamics.

The time advancement algorithm can now be described. Having chosen to use an explicit fractional step algorithm for the fluid algorithms, we also advance the rigid body equations of motion with an explicit scheme. The following is a standard central difference approach used to advance the rigid body quantities a single time step for generic loading:
(1) Given initial conditions, $d^n, v^n$

(2) Evaluate $f^n(d^n) = \text{applied loads}$

(3) Compute acceleration, $a^n = \frac{f^n}{m}$

(8) First partial update of velocity $v^{n+1/2} = v^n + \frac{\Delta t a^n}{2}$

(5) Update on displacement $d^{n+1} = d^n + (\Delta t)v^n + \frac{(\Delta t)^2 a^n}{2}$

(6) Evaluate $f^{n+1}(d^{n+1}) = \text{applied loads}$

(7) Compute acceleration, $a^{n+1} = \frac{f^{n+1}}{m}$

(8) Second partial update of velocity $v^{n+1} = v^{n+1/2} + \frac{\Delta a^{n+1}}{2}$

An analogous algorithm is given for the rotational system in Simo and Wong (1991). It requires defining a construction of a finite rotation from an incremental rotation via the exponential map:

$$\epsilon \in \mathbb{R} \rightarrow \Lambda = \exp[\epsilon \Lambda] := \sum_{n=0}^{\infty} \frac{1}{n!}[\epsilon \Lambda]^n \in \text{SO}(3), \quad (2.33)$$

which has a closed form expression for $\text{SO}(3)$, given an incremental angle, $\hat{\Theta}$:

$$\Lambda = \exp[\hat{\Theta}] = 1 + \frac{\sin\|\hat{\Theta}\|}{\|\hat{\Theta}\|} \hat{\Theta} + \frac{\sin^2\left(\frac{1}{2}\|\hat{\Theta}\|\right)}{2\left(\frac{1}{2}\|\hat{\Theta}\|\right)^2} \hat{\Theta} \quad (2.34)$$

The algorithm follows:
(1) Given initial conditions, $\Lambda^n, \omega^n$

(2) Evaluate $T^n(\Lambda^n) =$ applied torques

(3) Compute acceleration, $\alpha^n = J^{-1} \left[ f^n - \omega^n \times J\omega^n \right]$

(4) Update on rotation $\Theta = \Delta t \omega^n + \frac{(\Delta t)^2}{2} \alpha^n$

(5) Update on rotation $\Lambda^{n+1} = \Lambda^{n+1} \exp[\dot{\Theta}]$

(6) Evaluate $T^{n+1}(\Lambda^{n+1}) =$ applied torques

(7) Second partial update of velocity

$$\omega^{n+1} = J^{-1} \exp(-\dot{\Theta}) \left[ J\omega^n + \frac{\Delta t}{2} (T^n + T^{n+1}) \right]$$

(8) Compute acceleration, $\alpha^{n+1} = J^{-1} \left[ T^{n+1} - \omega^{n+1} \times J\omega^{n+1} \right]$

The consistent form of central differences involves an evaluation of the forcing terms at current displacement, and a velocity that is a half time step back $f^n = f^n(d^n, v^{n-1/2})$. Thus, to meld the two systems of equations (fluid and rigid body), we will use a fractional step scheme so that we may match velocity boundary conditions at the half step. The scheme is staggered, with the fluid velocity solved at time steps $n - \frac{1}{2}$ and $n + \frac{1}{2}$ and the particle variables solved at steps $n - \frac{1}{2}, n, n + \frac{1}{2}$ and $n + 1$. The level set equation is advanced with a fourth order Runge Kutta scheme.

Given initial conditions $u_{n-1/2}, d_n, v_{n-1/2}, \Lambda^n$, and $\omega^n$, the algorithm is summarized as follows:
1. Calculate the initial acceleration of the rigid body:

\[ a^n = \frac{f^n(d^n, u^{n-1/2})}{m}, \]

and

\[ \alpha^n = J^{-1} [T^n (\Lambda^n, u^{n-1/2}) - \omega^n \times J\omega^n]. \]

where for the first time step only, \( u^{n-1/2} = u^0 \).

2. Update the particle velocity, \( \mathbf{v} \) and \( \omega \),

\[ \mathbf{v}^{n+1/2} = \mathbf{v}^{n-1/2} + \Delta t a^n \]

\[ \Theta = \Delta t \omega^n + \frac{(\Delta t)^2}{2} \alpha^n \]

\[ \omega^{n+1/2} = J^{-1} \exp(-\Theta) [\omega^n + \Delta t T^n] \]

3. Update the particle position,

\[ \mathbf{d}^{n+1} = \mathbf{d}^n + \Delta t v^{n+1/2} \]

\[ \Lambda^{n+1} = \Lambda^{n+1} \exp[\Theta] \] \hspace{1cm} (2.36)

4. Set intermediate velocities equal in B,

\[ u^{n+1/2} = v^{n+1/2} + (r \times \omega^{n+1/2}) \text{ in } B. \]

The value of \( u^{n+1/2} \) in the domain \( B \) becomes the boundary condition for the fluid velocity at \( n - 1/2 \).

5. Solve for an intermediate fluid velocity ,

\[ \mathbf{u}^* = u^{n-1/2} + \Delta t \left[ \frac{\mu(\Delta u^{n-1/2})}{\rho_f} + \mathbf{g} \right]. \]
6. Complete a pressure projection step,

\[
\frac{\nabla^2 p}{\rho_f} = \frac{\nabla \cdot \mathbf{u}^s}{\Delta t}.
\]

7. Correct the final fluid velocity (everywhere but \(B\)),

\[
\mathbf{u}^{n+1/2} = \mathbf{u}^s + \frac{\Delta t \nabla p}{\rho_f}.
\]

8. Solve the level set equation, update the free surface position, update the position of the rigid body.

9. Increment \(n + 1 \Rightarrow n\), and \(n + \frac{1}{2} \Rightarrow n - \frac{1}{2}\) and return to Step 1.

For any explicit scheme there is a time step restriction to maintain stability. In this case, the time step is taken to be the lesser of the convective (\(\Delta t^{\text{conv}}\)) and viscous (\(\Delta t^{\text{visc}}\)) stable time steps, where

\[
\Delta t^{\text{conv}} = \frac{C}{\left( \frac{u_{\text{max}}}{\Delta x} + \frac{v_{\text{max}}}{\Delta y} \right)}
\]

(2.37)

\[
\Delta t^{\text{visc}} = \frac{1}{2 \left( \left( \frac{\mu_f}{\rho_f} \right)_{\text{max}} \left( \frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right) \right)}.
\]

(2.38)

For all of the numerical examples, the stable time step is recalculated at every step.

The simple fluid time stepping scheme that the above algorithm is based upon does not represent the most sophisticated option available for these kinds of solvers. For simulations with high Peclet numbers, the coefficient on the time step, \(C\), might have to be less than 0.1 to achieve stable results. Better options might be implicit solvers for the advective terms, or at least a multi-step Runge-Kutta explicit solver, for example. The preceding scheme was chosen because it was already built into a
fluid solver with an option for a free surface. Fluid coupling to rigid bodies was added on a foundation of an original fluid solver was written in Fortran90 by Nathaniel Ray Morgen at the Georgia Institute of Technology. Details can be found in Morgan (2005).

In addition, the preceding time integration scheme represents a loose coupling between the equations of motion for the rigid bodies and fluid. That is to say, the forcing terms for the rigid body equations are dependent only on the location of the body at the preceding time step. A loose coupling strategy is computationally efficient, but is known to cause stability problems in certain interaction regimes, most importantly those in which the solid has very low mass. Additionally, the second-order accuracy of the central differences may be compromised by the staggered approach, rendering the method first order in time. A strong coupling integration scheme for these kinds of equations is proposed in Borazjani et al. (2008) and shown to alleviate many of the stability issues. In a strongly coupled scheme the motion of the body is determined at least in part by the updated position, and thus requires iterative solving.

2.4.2 Spatial discretization

The Eulerian fluid domain is discretized with a staggered finite difference grid, as in Jiang and Peng (2000), with face centered velocities and cell centered pressures. Figure 2.3 gives a schematic of the grid. Viscous terms in the Navier Stokes equations are approximated with second-order central differences. A second-order upwinding stencil is applied to the convective terms in which, for example:

\[
\frac{\partial \hat{u}}{\partial x} = u^+ u^- + u^- u^+, \tag{2.39}
\]

where at a nodal point \((i, j)\),
\[ u^\pm = 0.5(u_{i,j} \pm |u_{i,j}|), \quad (2.40) \]
\[ u_x^- = \frac{3u_{i,j} - 4u_{i-1,j} + u_{i-2,j}}{2\Delta x}, \quad (2.41) \]

and
\[ u_x^+ = \frac{-u_{i+2,j} + 4u_{i+1,j} - 3u_{i,j}}{2\Delta x}, \quad (2.42) \]

and similar expressions hold for the terms \( u_B \partial_B u \), \( u_B \partial_B v \) and \( v_B \partial_B v \). In some cases we will compare the effects of using first order upwinding, in which
\[ u_x^- = \frac{u_{i,j} - u_{i-1,j}}{\Delta x}, \quad (2.43) \]

and
\[ u_x^+ = \frac{u_{i+1,j} - u_{i,j}}{\Delta x}. \quad (2.44) \]

The discrete values of \( \phi \) are stored at cell centers and its spatial gradient is calculated with either ENO or WENO schemes (Harlow and Welch, 1965), depending on the proximity of a nodal point to the domain boundary.

The discrete form of the step function \( H(\phi) \) is given by
\[
H^d(\phi) = \begin{cases} 
0 & \phi < -\epsilon \\
\frac{1}{2} \left( 1 + \frac{\phi}{\epsilon} + \frac{1}{\pi} \sin \left( \frac{\pi \phi}{\epsilon} \right) \right), & |\phi| \leq \epsilon \\
1 & \phi > \epsilon
\end{cases}
\]
(2.45)

where
\[ \epsilon = 1.5 \cdot \Delta x. \]

The particle boundary is defined for convenience through a second level set function, \( \phi_B \), and step function \( H^d(\phi_B) \), where the step function is 1 in the solid domain and 0 in the fluid. The particle level set and heaviside functions are not smoothed and
used for geometric tracking purposes only and are advected with the particle velocity rather than the level set equations.

Evaluation of the rigid body forces and torques requires integrals of fluid quantities over the body, $B$. The integral of any quantity, $f$, over the particle domain, $B$ embedded in a computation domain $\Omega$ may then be calculated through evaluation of

$$\int_B f dB = \int_\Omega f(x) H^d(\phi_B(x)) d\Omega.$$  
$$= \sum_{\text{cell}} f(x_{i,j}) H^d_{i,j}(\phi_B(x_{i,j})) \Delta x \Delta y \quad (2.46)$$

2.5 Numerical Results

2.5.1 Flow around a cylinder

A standard benchmark for flow solvers and rigid bodies is the case of a rigid cylinder in a uniform flow field for a range of Reynolds numbers. For very low Reynolds ($Re < 1$) the solution approaches Stokes flow (in a finite domain) and little separation
of flow occurs downstream from the cylinder. For this case, near field analytical solutions exist for pressure distribution over the surface of the cylinder (Wagner et al., 2001). To compare against this solution, we construct a square channel $[2\text{cm} \times 2\text{cm}]$ with a cylinder of diameter $0.2\text{cm}$ and apply an inlet velocity $u_\infty$ at the left boundary. The right boundary has an outflow conditions and symmetry is applied at the walls, that is $u_y = 0$ and $\frac{du}{dn} = 0$. The Reynolds number in this case is $0.1$. Velocity and pressure contours for a uniform grid $100\times100$ are given in Figures 2.4 and 2.5. The confined numerical solution for velocity and pressure approaches the unconfined analytical solution in the near field to the cylinder. To demonstrate this, the pressure and friction (calculated as $\mu_f(\nabla u + \nabla^T u)$) are reconstructed on the cylinder’s surface via a finite difference interpolation operator given in Taira and Colonius (2007). The solution for pressure, for example, at any Lagrangian point $(\xi_k, \eta_k)$ on the boundary can be constructed from the values at surrounding nodes via

$$p_k = \Delta x \Delta y \sum_i p_i r(x_i - \xi_k)r(y_i - \eta_k) \tag{2.47}$$

where

$$r(d) = \begin{cases} \frac{1}{2\epsilon} \left(1 + \cos \left(\frac{\pi r}{\epsilon}\right)\right), & r < -\epsilon \\ 0, & |r| \leq \epsilon \\ 0, & r > \epsilon \end{cases} \tag{2.48}$$

is the interpolation operator with $\epsilon = 1.5\Delta$. The indices $k$ and $i$ indicate nodal points and not vector indices. Figure 2.6 shows the comparison of the analytical Stokes flow surface pressure and friction distribution versus numerical results.

To study the behavior of the system (wake formation and vortex shedding) at higher Reynolds numbers, the numerical domain is extended to a long channel, $[2 \text{cm} \times 10 \text{cm}]$. The boundary conditions remain the same. As the Reynolds number
Figure 2.4: Velocity contours for Stokes flow around a confined cylinder

Figure 2.5: Pressure contours for Stokes flow around a confined cylinder
Figure 2.6: Surface pressures and friction for a confined cylinder in Stokes flow.

Figure 2.7: Lift coefficient versus time for flow around a fixed circular cylinder for $Re = 200$ increases, separation of the flow occurs and a stable laminar vortex appears in the wake of the cylinder (Figure 2.8). Von Karman vortex shedding is known to occur at Reynolds numbers for this case starting around $Re = 45$.

Lift and drag coefficients in this case are an important measure of the accuracy of an algorithm. The lift and drag coefficients, $C_L$ and $C_D$ respectively are calculated as

$$C_{(L \text{ or } D)} = \frac{F_{(L \text{ or } D)}}{0.5\rho_f U_\infty^2 d}$$

where $d$ is the diameter of the cylinder. $F_L$ and $F_D$ are the total lift and drag forces,
Figure 2.8: Wake formation and vortex flow over a cylinder for a) Re = 5 b) Re = 50 c) Re = 100 and d) Re = 200, Time = 3s, u-velocity is shown

Table 2.1: Drag coefficient for the current study versus literature benchmarks

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>4.32</td>
<td>4.42</td>
<td></td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>1.60</td>
<td>1.74</td>
<td>1.38</td>
<td>1.81</td>
</tr>
<tr>
<td>100</td>
<td>1.60</td>
<td>1.46</td>
<td>1.26</td>
<td>1.76</td>
</tr>
<tr>
<td>200</td>
<td>1.58</td>
<td>1.29</td>
<td>1.18</td>
<td>1.76</td>
</tr>
</tbody>
</table>

which can be calculated via the total traction acting on the cylinder at a given time:

\[
F_L = \int_0^{2\pi} |\boldsymbol{\sigma} \cdot \boldsymbol{n}| \sin(\theta) \, d\theta \quad \text{and} \quad F_D = \int_0^{2\pi} |\boldsymbol{\sigma} \cdot \boldsymbol{n}| \cos(\theta) \, d\theta \quad (2.50)
\]

where \(\theta\) is measured from the stagnation point. For Reynolds numbers higher than Re = 40, \(C_L\) and \(C_D\) are time dependent. Time averaged values for drag coefficient are given in Table 2.1 and compared to standard literature numerical and experimental results. A plot of lift coefficient versus time for Re = 200 is given in Figure 2.7.

2.5.2 Sedimentation of cylindrical disk

A second numerical problem that includes rigid body dynamics and has been extensively studied is that of the sedimentation of a circular disk. Construction of the
problem is done in such a way that it matches numerical studies from the Distributed Lagrange Multiplier approach (Glowinski et al., 2001) and IFEM (Lee et al., 2008) approaches. A cylinder of 0.25 cm radius starts at rest in the center of a 2D channel with width 2cm and length 6cm. The fluid properties are designated to approximate water, with a density of $\rho_f = 1.0(g/cm^3)$ and viscosity $\mu_f = 0.01(Ns/m^2)$. The solid has a density $\rho_s = 1.2(g/cm^3)$ and will accelerate downwards in the fluid until viscous drag forces balance gravity and a terminal velocity is reached. Velocity and pressure contours for the fluid domain with a uniform grid ($\Delta x = \Delta y = 0.04$) are given in Figure 2.9. Figure 2.10 shows reasonable agreement with previous studies for the time history of position and velocity of the rigid particle.

A more challenging problem in terms of stability, especially for loosely coupled time integration schemes, is that in which a solid that is less dense than a surrounding fluid rises due to buoyancy effects. Thus the problem was repeated but with the cylinder starting at rest at $y = 2.0$ cm and a density of $\rho_s = 0.8(g/cm^3)$. Figure 2.11 shows three different velocity profiles of the cylinder as it rises in the fluid, corresponding to successively finer grids. Figure 2.12 shows the final upwards “terminal” velocity to be consistently converging to a stable solution (given by an appropriately refined reference grid).

2.5.3 Rigid body in a two-phase fluid: buoyancy

One example of the motion of a rigid body in a two phase fluid is the decaying oscillation of a buoyant body on a free surface. There have been analytical and numerical studies to provide benchmarks (Zandergen et al., 1993; Donescu and Virgin, 2001) and in this case we study a circular cylinder in a tank of water and air, with density half that of water such that the buoyant equilibrium point is approximately half submerged (the density of air being 1000 times smaller does comparatively little to change the final equilibrium position). To match the simulations of Donescu and
**Figure 2.9:** Velocity and pressure (minus hydrostatic pressure) contours for sedimentation of a cylindrical disk at $t = 0.4s$

**Figure 2.10:** Position and velocity of disk as a function of time versus benchmark solutions
Figure 2.11: Velocity profiles for a buoyant cylinder in successively finer grids

Figure 2.12: Error in terminal velocity for a buoyant cylinder; the error decreases at an order slightly greater than 1. A fine grid with a spatially converged solution is used as the reference solution.
Virgin (2001), which treated non-linear free-surface/rigid body interaction with an implicit boundary element technique, a cylinder with radius 2m is positioned in a tank 50 meters in length with rigid walls and a no-slip condition at the ends (a condition that allows reflected waves). The depth of the tank is 4 meters. The initial position of the cylinder’s center of mass is 0.5 meters above the free surface. In both the current and previous numerical studies (Figure 2.13), the heave motion of the cylinder decays over a 10 second simulation period and the center of mass settles to the predicted position (indicated by the dotted line) based on buoyancy. For the numerical results of Donescu and Virgin (2001) the damping ratio is approximately 0.19, and in our numerical results it is slightly less (0.15). Damping ratio is calculated as

\[
Z = \frac{d}{\sqrt{(2\pi)^2 + d^2}}, \quad (2.51)
\]

where \(d\) is the logarithmic decrement of two successive periods of oscillation, \(T_1\) and \(T_2\):

\[
d = \ln \left( \frac{T_1}{T_2} \right). \quad (2.52)
\]

A parametric study of the problem shows that the two greatest factors that affect damping ratio are the value of gravity and water depth, with a deeper basin decreasing the damping ratio.

2.5.4 Roll of a buoyant body

The in-plane rotation (or roll) of a buoyant body can represent a large part of the total motion, and has the potential to have a profound impact on the energetic uptake of wave energy harvesters. Decades of research from the ship capsize community (Liaw
Figure 2.13: Decay of heave for a buoyant cylinder for a) the current study and b) Donescu and Virgin (2001)

and Bishop, 1995) has shown that heave and roll are strongly coupled motions under wave excitations for non-axisymmetric bodies.

Experimental and two dimensional numerical results from a study of free decay of roll motion for a barge (Chen et al., 2001a,b) in a wave basin provide a suitable benchmark for the roll of an oblong object, and are repeated with the current method.

The shape of the barge, with dimensions are shown in Figure 2.14. The water depth is set to 12 ft and two different values for draft, or the distance from the still water line to the barge depth, are used (4 and 6 feet). In addition, three different values of center of gravity (CG) and radius of gyration (RG) are used, giving a total of six test cases, A1 - B3. Radius of gyration is related to the rotational inertia of the barge through the relationship $I = m(RG)^2$. Natural roll periods are determined for the six cases by giving the body an initial angle of 10 degrees from horizontal. Results for the natural roll frequencies, compared to the numerical calculations of Chen et al. (2001b) and original experimental data from Chen et al. (2001a), are given in Table 2.2. Additionally, the time history of the roll angle of the barge is shown for one example case (A1) in comparison to the previous numerical study (Figure 2.15) with both second and first order upwinding. It can be seen that the current study is once
Table 2.2: Experimental and numerical comparisons of decaying roll frequencies for 6 cases.

<table>
<thead>
<tr>
<th>Case</th>
<th>A1</th>
<th>A2</th>
<th>A3</th>
<th>B1</th>
<th>B2</th>
<th>B3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Draft (ft)</td>
<td>4.0</td>
<td>4.0</td>
<td>4.0</td>
<td>6.0</td>
<td>6.0</td>
<td>6.0</td>
</tr>
<tr>
<td>CG (ft)</td>
<td>9.13</td>
<td>10.23</td>
<td>11.18</td>
<td>9.24</td>
<td>10.33</td>
<td>11.30</td>
</tr>
<tr>
<td>RG (ft)</td>
<td>10.31</td>
<td>10.41</td>
<td>10.09</td>
<td>10.11</td>
<td>10.23</td>
<td>10.11</td>
</tr>
<tr>
<td>Exp roll period (s)</td>
<td>5.4</td>
<td>5.9</td>
<td>6.5</td>
<td>7.3</td>
<td>8.9</td>
<td>12.3</td>
</tr>
<tr>
<td>Num roll period (s)</td>
<td>5.73</td>
<td>6.18</td>
<td>6.76</td>
<td>8.01</td>
<td>10.36</td>
<td>15.69</td>
</tr>
<tr>
<td>Current study (s)</td>
<td>5.36</td>
<td>5.80</td>
<td>6.26</td>
<td>7.12</td>
<td>8.4</td>
<td>11.86</td>
</tr>
</tbody>
</table>

again slightly less dispersive than previous numerical studies. The slight blip in the consistent decay of the roll angle at $t = 20s$ comes from influence of reflected waves in the current study.

2.5.5 Wave tank and buoy simulations

The goal of the presented algorithm will be to compare numerical results with a scale wave tank with buoy prototypes to replicate energy harvesting systems. The simulated wave tank is 5 meters long with a water depth of 0.5 meters. A wedge-
Figure 2.15: Example time histories of free decay of ship roll motion for previous (a) and current studies with second-order (b) versus first-order upwinding (c).

Figure 2.16: Setup for wave tank simulation problem

A shaped wave generator with variable frequency oscillates at one end. The motion of a buoy with a height of 20 cm, width of 10 cm, weight of 15 g, and center of gravity 5 cm below the geometric center, is studied under steady state waves conditions generated in the tank with the wave generator moving at 1 Hz. The numerical domain is shown in Figure 2.16. Single time snapshots of the simulation are shown in Figure 2.17, with close ups of the buoy shown at the same times in 2.18 and a time history of the heave of the cylinder is given in Figure 2.19.
Figure 2.17: Snapshots of wave tank simulations at $t = 0.0\ \text{s}, 1.4\ \text{s}, 2.4\ \text{s}, 5.2\ \text{s}$ and $8.4\ \text{s}$
Figure 2.18: Zoomed in snapshots of wave tank simulations at t = 1.4 s, 2.4 s, 5.2 s and 8.4 s

Figure 2.19: Heave motion of center of mass of a free floating buoy in wave tank
2.5.6 *Internal rotations prototypes*

This section presents a simple example of recent work done to build upon the preceding algorithms into systems of rigid bodies with internal dynamic components. The coding and simulation works were the effort of Curtis Lee at Duke University. A one-degree of freedom rotational spring and mass was included in the core of the buoy in the wave tank via an addition term in (2.12) and one additional equation representing the motion of the device. This gives, as a set of equations representing rotation of the device:

$$\frac{d\pi^1}{dt} = \int_{\Gamma^s} (\mathbf{r} \times \mathbf{t}) \, d\Gamma^s + \mathbf{T}_i,$$

and

$$\frac{d\pi^2}{dt} = -\mathbf{T}_i,$$

where $\mathbf{T}_i$ is an internal force between the two rotating masses, and is potentially a function of both the relative rotational positions and velocities.

Figure 2.20 shows snapshots of the simulation, in which a wave generator operates for a period of three seconds and then the basin is allowed to settle. A fluid velocity map is printed on top of the level set contours. Figure 2.21 gives the rotation angle of the outer and inner masses over simulation time. The spring is sufficiently stiff to keep them relatively in sync. Clearly the example is qualitative as opposed to quantitative, but should give a preview into what kind of capabilities may be possible for modeling internal mechanisms.
Figure 2.20: Snapshots of wave tank simulations with rotations at t = 0.05s, 2.6s, 5.1s
Figure 2.21: Rotation of rigid body and internal rotational spring/mass system in wave tank
2.6 Conclusion

The first part of this work has presented an algorithm for the simulation of the two-dimensional motion of rigid bodies and their interactions with two-phase incompressible flow and large amplitude free surface motion. The method uses a finite difference approach to the two-phase fluid problem that takes advantage of the current level of sophistication of the level set method. However, the standard paradigm for tracking solid motion in a fluid, that is to say, solving for solid motion on a fluid grid, has been abandoned in favor of a Lagrangian approach to rigid body motion, in order to facilitate the anticipated addition of internal dynamics to the solid equations of motion. The current method shows good agreement with standard benchmarks. However, there are some concerns with the approach, which may either indicate necessary future work, or a switch to a different approach for a general method of fluid/structure interaction. The kinds of comparisons that can be made in two dimensions, either to analytical benchmarks or experimental data, are very limited. It is also of concern that some of the two-phase simulations display a loss of mass. This is generally associated with the level set method for two-phase flow, and certain corrections do exist in the literature (see, for example, Sussman and Puckett, 2000). For the remainder of this thesis, we will turn our attention to finite element approaches to problems of multiple overlapping domains.
3

Embedded Meshes for Linear Elasticity

3.1 Introduction

In the remainder of this work, we consider a methodology of overlapping finite element grids. The basic idea behind the approach is to formulate finite element equations of motion over two separate domains, and then couple them by geometrically embedding one in the other, and appropriately matching their motion at the interface without any re-meshing or algorithmic (non-physical) mesh advection. A conceptual schematic of the idea is shown in 3.1.

The broader goal is to develop and implement an approach that can accommodate a host of multi-physics simulations. In principle, the two domains could support completely different physical equations. The background grid might represent a fluid, while the embedded body could be a structure. In some cases, it may make sense to treat the domains with different frameworks (Eulerian versus Lagrangian, similar to what was presented in the preceding chapter). For any class of problem, the method has the potential to simplify model development and avoid re-meshing algorithms. An implementation of a basic embedded mesh strategy is in existence, at
various stages for various multi-physics strategies, in the codes of Lawrence Livermore National Laboratory.

In this work we will focus our attention on accurate and stable methods to enforce continuity constraints across embedded grids, such as that shown in Figure 3.1. There are a large range of possible techniques and model equations, but we choose solid mechanics as a suitable platform, as it can demonstrate the important geometrical and numerical mechanisms.

This chapter is organized as follows. The remainder of the introduction will give an overview of the primary problem we see with traditional methods of interface constraint, that is, that in some situations we run into “mesh locking”. We propose to change the usual strategy and employ a method of weak constraint known as Nitsche’s method. Section 3.2 introduces the problem formulation and governing equations for the embedded mesh formulation for a linear elastic problem. Section 3.3 describes a naive implementation of the mortar approach, within the context of an overlapping elasticity problem. Section 3.5 describes the application of Nitsche’s method to a linear form of the problem, including a derivation of the weighted form. Section 3.4 describes a commonly used penalty approximation to mortar methods. Section 3.6 overviews the global equation structure produced by the procedure, while
Section 3.7 gives some numerical demonstrations of the technique. The presentation of methods and results mostly refers to the two dimensional case, but is not, in principle, restricted. A discussion of the challenges of three dimensions will be given in Section 3.8 along with a more general conclusion.

3.1.1 Specific challenges in stable enforcement of embedded interfacial constraints

In our applications of overlapping finite element grids, we seek to demonstrate and address the issue of mesh locking which may occur when the number of constraints imposed on the surface or volume exceeds the number of degrees of freedom available to render these constraints independent of the physical deformations the elements should allow. This generally occurs when a domain with a very fine mesh is embedded in a coarser background grid. Oscillations of the Lagrange multipliers (when they are used to impose the constraints) may also accompany this situation. This problem manifests itself in both loss of overall accuracy and suboptimal convergence rates in affected problems. On the other hand, a relative scarcity of constraints can lead to underrepresentation of the compatibility conditions and inaccurate representations of surface fluxes. An additional challenge in solving these problems is that we generally wish that low-order approximations (typically linear velocities or displacements, spatially constant fluxes) be exactly representable across tied overlap regions (at the very least for flat interfaces, and ideally more broadly). Though this is a problem that may occur in any embedded mesh situation, will we use elastic solid mechanics problems as a template for studying the behavior of various techniques for matching disparate approximations at an embedded interface.

Many, if not most, of these challenges have been met within the contexts of the methods mentioned in the introduction of this dissertation. For this work, however, we ask the question whether such requirements can be met under rather specific circumstances. Specifically, our goals and assumptions are:
1. We are most interested in avoidance of mesh locking as manifested, for example, by lack of optimality in bulk $L_2$ and $H_1$ norms of error.

2. The method should be applicable in both two and three spatial dimensions (as is usual in computational mechanics, the latter case is much more difficult in general).

3. Conservation of linear momentum (for all geometries) is an important feature we wish to maintain, as is passage of patch tests (at least for flat interfaces, but preferably curved as well).

4. We wish to employ a framework in which we enforce surface-to-volume constraints, as opposed to the more commonly applied surface-to-surface or volume-to-volume constraints treated in contact.

It should be mentioned that items 2 and 3 are particularly demanding, especially in light of recent efforts in the literature. For example, Bechét et al. (2009) proposes a particularly ingenious way of sorting through nodes and constraints in the context of an enriched interface method, identifying so-called “vital nodes” that ensure a stable multiplier representation through use of the trace of the element shape functions as a multiplier basis. The resulting method is convincingly demonstrated to satisfy inf-sup stability conditions, though application of the method is involved and possibly inconvenient. It has not, to the author’s knowledge, been applied to the problem of overlapping domains.

Here we begin with a rather standard mortar method to join the surface of one elastic domain to the image of this surface in an overlapped second elastic domain. Though the method is broadly applicable in many cases, this method displays mesh locking in the case where the overlapping mesh is both finer and stiffer than the underlying grid, as will be demonstrated in our numerical examples (Section 3.7).
A penalty approximation of mortar constraints is also studied, and shown to exhibit sub-optimal convergence rates due to a loss of consistency. We finally study Nitsche’s method as an alternative approach to applying the continuity constraints. Nitsche’s method has already been studied for overlapping domains for the Laplace equation by Hansbo et al. (2003), though the issue of mesh locking is not investigated, probably due to the fact that in that study the “stiffness” of the domains is always equivalent. In this case, we uncover the surprising results that for the same pathological situation in which standard application of mortar constraints produces mesh locking, the typical application of Nitsche’s method also produces mesh locking and sub-par convergence behavior. Here we show, however, that a weighted Nitsche’s method, biased to material stiffnesses, alleviates locking and recovers convergence of the relevant norms.

3.1.2 Literature review - Nitsche’s method

Nitsche’s method was originally proposed in Nitsche (1971) as a method for weakly imposing essential boundary condition for the finite element approximation of elliptical PDEs. A relationship to stabilized finite element methods was later established in Stenberg (1995), who showed that Nitsche’s method could be derived for the stabilized Lagrange multiplier formulation of Barbosa and Hughes (1991). The method has witnessed a resurgence in recent years. It was used in Griebel and Schweitzer (2002) and Fernández-Méndez and Huerta (2004) for boundary condition enforcement in mesh-free methods, in Fritz et al. (2004) for domain decomposition purposes, and in Wriggers and Zavarise (2008) for contact. In Hansbo et al. (2003) and Hansbo and Hansbo (2002) the method is used for embedded interfaces, and for an overlapping finite element method based for scalar elliptical equations, similar in principle to the technique to be discussed in this chapter. The method was not, however, established for elasticity, nor was the concept of mesh locking addressed.
The concept of stabilization of ungridded interfaces was addressed in Sanders et al. (2009) in the context of the extended finite element method.

The use of Nitsche’s method requires choice of an algorithmic stabilization parameter. Though it does not enforce a physical constraint, the choice of the parameter is important to the results in terms of stability of the discrete system. Both Sanders et al. (2009) and Dolbow and Harari (2009) give techniques for its proper estimation. The latter gives a few particularly efficient closed form analytical expressions for the stabilization term in certain cases. As the choice of this term is perhaps the most confounding aspect of the class of methods stemming from Nitsche’s original work, the existence of such expressions makes the process much more palpable.

3.2 Problem Formulation

We consider, as a model problem, the perfect adhesion of two linear elastic bodies in a static framework. As depicted in Figure 3.2, we assume that one body, \( B^1 \), overlays the second body, \( B^2 \), over some overlap region, and we denote the intersection of \( \partial B^1 \) (the boundary of \( B^1 \)) with \( B^2 \) as \( \Gamma_\ast \). In what follows, the coordinates in bodies \( B^i \), \( i = 1, 2 \), will be denoted as \( \mathbf{x}^1 \) and \( \mathbf{x}^2 \), respectively. The entire computational domain is considered as the union of \( B^1 \), with the uncovered region of \( B^2 \), \( \Omega = B^1 \cup (B^2 \backslash B^1) \).

Our formulation will be general to cases of both complete (as shown) and partial overlap of the bodies. Excluding \( \Gamma_\ast \), the rest of the computational boundary can be divided into Dirichlet and Neumann parts on each body, \( \Gamma_{d}^{(m)} \) and \( \Gamma_{h}^{(m)} \) respectively. A super script, \( (m) \), is used to denote a quantity that is valid over region \( B^m \), with \( m = 1, 2 \).

The system of model equations with the primary unknown displacement, \( \mathbf{u}^{(m)} \),
for static equilibrium, completed with boundary and initial conditions, is

\[ \nabla \sigma^{(m)} + f^{(m)} = 0 \quad \text{in } \Omega \quad (3.1a) \]

\[ u^{(m)} = \bar{u}^{(m)} \quad \text{on } \Gamma_d^{(m)} \quad (3.1b) \]

\[ \sigma^{(m)} n^{(m)} = h^{(m)} \quad \text{on } \Gamma_h^{(m)} \quad (3.1c) \]

\[ u^{(1)} = u^{(2)} \quad \text{on } \Gamma_s \quad (3.1d) \]

\[ \sigma^{(1)} n^{(1)} = -\sigma^{(2)} n^{(2)} \quad \text{on } \Gamma_s \quad (3.1e) \]

where the strain is taken as the symmetric part of the displacement gradient:

\[ \varepsilon(u)^{(m)} = \frac{1}{2}(\nabla u^{(m)} + \nabla^T u^{(m)}), \]

and the Cauchy stress is a linear function of the strains:

\[ \sigma^{(m)} = C^{(m)} : \varepsilon^{(m)}(u). \]

Here, \( C^{(m)} \) is the fourth order tensor of material properties for body \( m \) according to Hooke’s Law.
3.3 Mortar Tying of a Surface to an Overlapped Volume

We start by defining the spaces, $S$ and $V$, defined over each subdomain, $B^1$ and $B^2$, that will contain the solution and trial functions respectively:

$$S^{(m)} = \{ u(x) | u(x) \in H^1(B^m), u(x) = u(x) \text{ on } \Gamma_d \}$$

$$V^{(m)} = \{ w(x) | w(x) \in H^1(B^m), w(x) = 0 \text{ on } \Gamma_d \}.$$

It is important to note that while functions are continuous within each subdomain, the restriction of the continuity to individual bodies means the entire solution may be discontinuous at the boundary $\Gamma_d$. The mortar approach, used for example by Baaijens (2001), would (weakly) constrain the surface of $B^1$ to move in concert with the underlying mesh of $B^2$ via a mixed variational form. We choose to enforce the constraints (3.1d) and (3.1e) via a traction field, belonging to space $L$, represented by $\lambda$ acting over the interface $\Gamma_d$,

$$L = \{ \lambda(x) | \lambda(x) \in L^2(\Gamma_d) \}.$$

The mixed variational statement for this method is:

Find $(u^{(1)}, u^{(2)}, \lambda) \in S^{(1)} \times S^{(2)} \times L$ such that

$$\sum_{m=1}^{2} \int_{B^m} \varepsilon(w^{(m)}) \cdot \sigma^{(m)} \, dB + \int_{\Gamma_d} \lambda \cdot \llbracket w \rrbracket \, d\Gamma =$$

$$\sum_{m=1}^{2} \int_{B^m} w^{(m)} \cdot f^{(m)} \, dB + \sum_{m=1}^{2} \int_{\Gamma_h^{(m)}} w^{(m)} \cdot h^{(m)} \, d\Gamma, \quad (3.2)$$

and

$$\int_{\Gamma_d} \mu \cdot \llbracket u \rrbracket \, d\Gamma = 0 \quad (3.3)$$

for all $(w, \mu) \in V^{(1)} \times V^{(2)} \times L$. 

56
A gap function is defined on $\Gamma_*$ as $[[u]] = u^1 - u^2$. Equation (3.2) is a statement of conservation of momentum for the entire system. The “constraint” virtual work,

$$G^C = \int_{\Gamma_*} \lambda \cdot (w^1 - w^2) \, d\Gamma_*,$$  \hspace{1cm} (3.4)

represents the additional internal work the system performs in order to maintain a zero gap function. Equation (3.3) is a constraint equation enforcing that the gap must vanish in a weak, or integral, sense.

A major disadvantage of mixed methods (called mixed because they solve simultaneously for two different fields, in this case displacement and traction) is that when discretized, their performance is dependent on the finite element subspaces satisfying the so-called LBB (or inf-sup) conditions. For embedded interface methods, this rules out many convenient choices for the subspaces. The various problems with getting stable and convergent results will be demonstrated in Section 3.7.

### 3.4 Penalty Approximations to Mortar Lagrange Multipliers

One way of avoiding a mixed variational formulation would be to apply a penalty approximation of the mortar constraints. Instead of explicitly solving for the Lagrange multiplier field, we approximate it with

$$\lambda \approx \frac{\alpha}{2} [[u]],$$ \hspace{1cm} (3.5)

where the scalar constant $\alpha$ can be interpreted as a gap stiffness, and will be referred to as the penalty parameter. The associated variational form is:
Find \((u^{(1)}, u^{(2)}) \in S^{(1)} \times S^{(2)}\) such that

\[
\sum_{m=1}^{2} \int_{B^m} \varepsilon(w)^{(m)} : \sigma^{(m)} dB + \alpha \int_{\Gamma_s} \|[(u)] \cdot [w]\| d\Gamma =
\]

\[
\sum_{m=1}^{2} \int_{B^m} w^{(m)} \cdot f^{(m)} dB + \sum_{m=1}^{2} \int_{\Gamma_h^{(m)}} w^{(m)} \cdot h^{(m)} d\Gamma
\]

for all \(w \in V^{(1)} \times V^{(2)}\). (3.6)

The advantage of the penalty method is that there are no additional global constraint equations and only one unknown field (displacement). However, the penalty method is variationally inconsistent, and discrete results are overly sensitive to \(\alpha\). A classic result by Babuška (1973) indicates that to achieve spatial convergence, the penalty parameter must grow at a rate that outpaces the other terms in the stiffness matrix. Additionally, the constraint is only perfectly represented in the limit \(\alpha \to \infty\). For the discretized method, this eventually leads to poor conditioning of the linear algebraic system of equations.

### 3.5 Nitsche’s Method

Unfortunately, given the way we choose our constraints in the discrete form of the standard mortar method, we find ourselves subject to the widely identified problem of mesh locking (see, for example, thorough descriptions and examples in Bechét et al. (2009) or Lew and Buscaglia (2008)). The logical choice in this case is to define the basis for the discrete Lagrange multipliers mesh on the surface \(\Gamma_s\). Within the context of our formulation, the problem occurs on the master side (body 2), where the interface is effectively embedded with the volume. Particularly for the case where the discretization of the slave side is comparatively fine relative to the master side volume discretization, use of the standard mortar approximations will give rise to increasingly severe over-constraint as the mesh is refined. A possible way
of circumventing these problems is to apply a weak constraint enforcement based on Nitsche’s method. The standard application of Nitsche’s method for the two-body problem is:

Find \( (u^{(1)}, u^{(2)}) \in S^{(1)} \times S^{(2)} \) such that

\[
\frac{2}{\mu} \int_{B^m} \varepsilon(w^{(m)}) : \sigma^{(m)} \ dB - \int_{\Gamma_s} ([w] \otimes n^{(1)}) : \langle \sigma \rangle \ d\Gamma
\]

\[
- \int_{\Gamma_s} ([u] \otimes n^{(1)}) : \langle \sigma(w) \rangle \ d\Gamma + \beta \int_{\Gamma_s} [[u]] : [[w]] \ d\Gamma =
\]

\[
\sum_{m=1}^2 \int_{B^m} w^{(m)} \cdot f^{(m)} \ dB + \sum_{m=1}^2 \int_{\Gamma_h^{(m)}} w^{(m)} \cdot h^{(m)} \ d\Gamma
\]

for all \( w \in V^{(1)} \times V^{(2)}. \) (3.7)

Here the brackets \( \langle \rangle \) indicate an average quantity, \( \langle a \rangle = \frac{1}{2} (a^{(1)} + a^{(2)}) \), and for linear elasticity the operator \( \sigma(w) \) is defined:

\[
\sigma(w) = C : \varepsilon(w).
\] (3.8)

With the exception of the second and third terms on the left side, the equilibrium condition is the same as that for the penalty method. Nitsche’s method assures continuity of the solution vector at \( \Gamma_s \) by directly correcting a consistency error that arises due to mismatching weighting functions at the interface. That correction comes from the extra terms not present in a penalty formulation.

Nitsche’s method can be easily derived from a stored energy functional approach, as follows: The total potential energy of the two body system can be computed as the difference between energy due to internal and external forces:

\[
\Pi(u) = \frac{1}{2} \sum_{m=1}^2 \int_{B^m} \varepsilon^{(m)} : \sigma^{(m)} \ dB - \sum_{m=1}^2 \int_{\Gamma_h^{(m)}} u^{(m)} \cdot h^{(m)} \ d\Gamma.
\] (3.9)
In the case of a mixed mortar method there is an additional potential due to the enforced continuity across $\Gamma_*$, which is the integral of the unknown surface traction, $\lambda$, dotted with the displacement gap:

$$\Pi(u)^{Log} = \int_{\Gamma_*} \lambda \cdot [[u]] \, d\Gamma.$$  \hfill (3.10)

To obtain Nitsche’s method, we approximate the surface traction as an average of the stresses on either side of the interface, dotted into the normal. A small penalty traction is added, which would not add energy if the displacement gap were exactly zero everywhere on the surface,

$$\lambda \approx -\langle \sigma \rangle \cdot n^{(1)} + \frac{\beta}{2} [[u]].$$ \hfill (3.11)

The interfacial potential energy associated with this approach is:

$$\Pi(u)^{Nit} = \int_{\Gamma_*} \left( -\langle \sigma \rangle \cdot n^{(1)} + \frac{\beta}{2} [[u]] \right) \cdot [[u]] \, d\Gamma.$$ \hfill (3.12)

Stationarity of the potential energy imposes that the first variation of the total potential energy, $\Pi(u)^{Tot} = \Pi(u) + \Pi(u)^{Nit}$, be zero. This gives the following variational formation:

Find $((u^{(1)}, u^{(2)}) \in \mathbf{S}^{(1)} \times \mathbf{S}^{(2)}$

$$\delta \Pi(u) + \delta \Pi(u)^{Nit} = 0$$

for all $w \in V^{(1)} \times V^{(2)}$. \hfill (3.13)

which gives Nitsche’s method exactly as it is written in (3.7). In Stenberg (1995), it is shown that any convex combination of flux operators will yield a consistent method in this context. We take $\langle \sigma \rangle$ to be $[\gamma \sigma^{(1)} + (1 - \gamma) \sigma^{(2)}]$, where $0 \leq \gamma \leq 1$. A generalized form of Nitsche’s method for elasticity becomes:
Find \(( u^{(1)}, u^{(2)} ) \in S^{(1)} \times S^{(2)} \)

\[
\sum_{m=1}^{2} \int_{B^{(m)}} \varepsilon(w)^{(m)} : \sigma^{(m)} dB - \int_{\Gamma_s} \left( \left[ [w] \right] \otimes n^{(1)} \right) : \left[ \gamma \sigma^{(1)} + (1 - \gamma) \sigma^{(2)} \right] d\Gamma \\
- \sum_{m=1}^{2} \int_{B^{(m)}} w^{(m)} : f^{(m)} dB + \sum_{m=1}^{2} \int_{\Gamma^{(m)}_h} w^{(m)} : h^{(m)} d\Gamma = \sum_{m=1}^{2} \int_{B^{(m)}} w^{(m)} : f^{(m)} dB + \sum_{m=1}^{2} \int_{\Gamma^{(m)}_h} w^{(m)} : h^{(m)} d\Gamma \\

\]

for all \( w \in V^{(1)} \times V^{(2)} \).

A choice of \( \gamma = \frac{1}{2} \) returns the original form given by (3.7).

**Consistency of Nitsche’s method**

The variational problem is consistent in the sense that, for \( u \) satisfying (3.1), the statement (3.14) is also true.

For \( u \) satisfying (3.1), the second and third terms of (3.14) are identically zero, leaving the statement

\[
\sum_{m=1}^{2} \int_{B^{(m)}} \varepsilon(w)^{(m)} : \sigma^{(m)} dB - \int_{\Gamma_s} \left( \left[ [w] \right] \otimes n^{(1)} \right) : \left[ \gamma \sigma^{(1)} + (1 - \gamma) \sigma^{(2)} \right] d\Gamma \\
- \sum_{m=1}^{2} \int_{B^{(m)}} w^{(m)} : f^{(m)} dB + \sum_{m=1}^{2} \int_{\Gamma^{(m)}_h} w^{(m)} : h^{(m)} d\Gamma = \sum_{m=1}^{2} \int_{B^{(m)}} w^{(m)} : f^{(m)} dB + \sum_{m=1}^{2} \int_{\Gamma^{(m)}_h} w^{(m)} : h^{(m)} d\Gamma \\

\]

for all \( w \in V^{(1)} \times V^{(2)} \). (3.15)

We apply Green’s first identity to (3.15), and split the boundary integral into \( \Gamma_h \), \( \Gamma_d \), and \( \Gamma_s \) for each body. Since \( w \) belongs to \( V^{(m)} \), the integrals over \( \Gamma_d \) vanish.
Grouping terms according to their domain of integration gives

$$\sum_{m}^{2} \int_{B^{m}} \mathbf{u}^{(m)} \cdot (\nabla \cdot \sigma^{(m)} - f^{(m)}) \, dB$$

$$+ \sum_{m}^{2} \int_{\Gamma^{(m)}_{h}} \mathbf{w}^{(m)} \cdot (\sigma^{(m)} n^{(m)} - h^{(m)}) \, d\Gamma$$

$$- \int_{\Gamma_{*}} ([w] \otimes n^{(1)}) : [\gamma \sigma^{(1)} + (1 - \gamma) \sigma^{(2)}] \, d\Gamma$$

$$+ \sum_{m}^{2} \int_{\Gamma_{*}} \mathbf{w}^{(m)} \sigma^{(m)} n^{(m)} \, d\Gamma = 0$$

for all $\mathbf{w} \in \mathbf{V}^{(1)} \times \mathbf{V}^{(2)}$. (3.16)

The first two integrals vanish due to (3.1a) and (3.1c), leaving it only necessary to show that the last two add to zero. To do so we use the fact that $\mathbf{n}^{(1)} = -\mathbf{n}^{(2)}$ and $\sigma^{(1)} n^{(1)} = -\sigma^{(2)} n^{(2)}$ to manipulate the integral

$$\int_{\Gamma_{*}} ([w] \otimes n^{(1)}) : [\gamma \sigma^{(1)} + (1 - \gamma) \sigma^{(2)}] \, d\Gamma =$$

$$\int_{\Gamma_{*}} [w] \cdot (\gamma \sigma^{(1)} n^{(1)} + \sigma^{(2)} n^{(1)} - \gamma \sigma^{(2)} n^{(1)}) \, d\Gamma =$$

$$\int_{\Gamma_{*}} [w] \cdot (\sigma^{(1)} n^{(1)}) \, d\Gamma = \sum_{m}^{2} \int_{\Gamma_{*}} \mathbf{w}^{(m)} \sigma^{(m)} n^{(m)} \, d\Gamma. \quad (3.17)$$

Thus, the last two terms of (3.15) cancel each other and the consistency of the variational form is established.

It is the second term of equation (3.14) that assures consistency of the variational form, and likewise the absence of this term that renders a standard penalty method variationally inconsistent. The third term provides symmetry and adjoint consistency, allowing the interpretation of the bilinear form as the minimization of a
potential energy functional. Unfortunately, subtraction of the surface integrals introduces a stability problem stemming from the lack coercivity of the bilinear functional. A penalty term is added to restore coercivity, given an appropriate choice of the stability parameter, $\beta$. Proofs of consistency and coercivity and are shown for Nitsche’s method for elliptical problems in Hansbo et al. (2003).

Only certain choices of the parameters $\gamma$ will give solutions free of the locking behavior that plagues the constrained problem in the case of the mortar method. In the case of a fine and stiff overlapping mesh, neither a perfect average ($\gamma = 1/2$) nor a choice of only the stress operator from the overlapping side ($\gamma = 1$, as implemented for the Laplace equation in Hansbo et al. (2003)) gives locking free results. Stress operators formed on the underlying side ($\gamma = 0$), or carefully weighted combinations, will result in locking free and fully convergent approximations, as shown in subsequent numerical studies (Section 3.7).

3.6 Discrete Equation Structure

3.6.1 Unconstrained case

This section considers the discrete form of the part of the equations treating the material stiffness, i.e., the variational equation without the interior constraint terms. The domains, $B^1$ and $B^2$ are divided independently into elements (see Figure (3.1)). In the following description, elements in the two bodies are denoted as $B^1_e$ and $B^2_e$, respectively, and boundary facets in $\Gamma_s$ are denoted as $\Gamma_e$. We choose to use standard linear shape functions, for both the displacement and trial functions, so for each body, $(m)$,

$$ u^{(m)} \approx u^{(m)h} = \sum_A N_A^{(m)} d_A^{(m)} $$

(3.18)

$$ v^{(m)} \approx v^{(m)h} = \sum_B N_B^{(m)} c_B^{(m)} $$

(3.19)
where $N_A$ and $N_B$ are standard linear finite element shape functions, $A$ and $B$ index over the nodes in the body, and $\{ \mathbf{d}_A^{(e)} \}$ represents the set of unknown displacement coefficients.

We use the standard vector notation where, in 2 dimensions,

$$\varepsilon(\mathbf{u}) = \begin{bmatrix} u_{1,1} \\ u_{2,2} \\ u_{1,2} + u_{2,1} \end{bmatrix},$$

$$\sigma(\mathbf{u}) = \begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{12} \end{bmatrix},$$

and

$$\sigma = D : \varepsilon(\mathbf{u}),$$

where $D$ is the matrix of elastic constants:

$$D = \begin{bmatrix} \lambda + 2\mu & \lambda & 0 \\ \lambda & \lambda + 2\mu & 0 \\ \text{symmetric} & \mu \end{bmatrix}.\quad (3.23)$$

The discretization of strain is done in an element with the typical $\mathbf{B}$ matrix formulation:

$$\varepsilon(\mathbf{u}) = \frac{1}{2}(\nabla \mathbf{u} + \nabla \mathbf{u}^T) = \mathbf{B}^e \mathbf{d}_e.$$

where

$$\mathbf{B}^e = [\mathbf{B}^e_1; \mathbf{B}^e_2; \ldots; \mathbf{B}^e_{n_{en}}].\quad (3.24)$$

Here, $\mathbf{d}_e$ is a vector of local displacement degrees of freedom, $n_{en}$ is the number of nodes in an element, (four for a two dimensional linear quadrilateral), and,

$$\mathbf{B}^e_A = \begin{bmatrix} N_{A,1} & 0 \\ 0 & N_{A,2} \\ N_{A,2} & N_{A,1} \end{bmatrix}.\quad (3.25)$$
Plugging the discretization into the unconstrained equations gives, for an element in body \( (m) \),
\[
\left( k^{(m)*} \right) = \int_{B^m_e} (B^e)T DB^e dB^m_e. \tag{3.26}
\]
Standard finite element assembly gives a discrete stiffness for each body,
\[
K^{(m)} = \frac{1}{n_{el}} \sum_{e=1}^{n_{el}} \left( k^{(m)*} \right).
\]
where \( n_{el} \) is the number of elements in the discretization of \( B^m \). The system of equations for the two body system, uncoupled and not subject to surface constraints is,
\[
K^{(1)} d^{(1)} + K^{(2)} d^{(2)} = f^{(ext)1} + f^{(ext)2},
\]
or,
\[
\begin{bmatrix}
K^1 & 0 \\
0 & K^2
\end{bmatrix}
\begin{bmatrix}
d^1 \\
d^2
\end{bmatrix}
= \begin{bmatrix}
f^{(ext)1} \\
f^{(ext)2}
\end{bmatrix}. \tag{3.27}
\]

### 3.6.2 Integration of cut cells

While the discretization and assembly of the overlapping grid is entirely standard, some special accommodations must be made for the background grid, where some elements may be indiscriminately cut by the surface of the overlapping body. Elements of the underlying grid that are completely covered by the inclusion are considered void and do not contribute to the discrete stiffness equations. In a sharp interface capture technique similar to that used in the extended finite element method (see, for example, Moës et al., 1999), elements that are cut by \( \Gamma_e \) contribute a partial stiffness, due to the uncovered area. All of the degrees of freedom of cut elements remain active, but integration only occurs over the material, or uncovered region of cut elements. It is not possible to use the standard optimal methods of Gauss quadrature (generally two points in each direction for linear quadrilaterals and hexes) over the
polygonal, possibly non-convex region that represents the active area. The region is divided into integration triangles (Figure 3.3), which are evaluated with a sufficiently large Gauss quadrature rule based on the background element interpolations (general linear or bilinear shape functions). In this case, six points for a triangle is used.

The standard expression for a local stiffness matrix for a discrete element integrated with $g$ Gauss points is

$$k^e = \sum_{g=1}^{n_g} \left( B^T(\xi) DB(\xi) j \right) W_g,$$  \hspace{1cm} (3.28)

where $\xi$ are the natural coordinates of a reference element, $j$ is the jacobian of the coordinate transformation from $x$ to $\xi$, $n_g$ is the number of Gauss points, and $W_g$ are appropriate Gauss quadrature weights. If we introduce a new coordinate system, $r$, based on the natural coordinates of an integration triangle, the expression for a cut cell divided into $n_t$ integration regions is

$$k_{cut}^e = \sum_{i=1}^{n_t} \left( \sum_{l=1}^{n_g} \left( B^T(r) DB(r) j \right) W_g \right),$$  \hspace{1cm} (3.29)
where \( j \) is now the jacobian of the coordinate transformation from \( x \) to \( r \). Assembly of the modified element stiffness matrices follows standard finite element procedures. There will be some nodes for this technique that lie in the void region of \( B^2 \), but still contribute support to the displacement approximation.

### 3.6.3 Standard mortars

In this work, we have used either two-dimensional bi-linear quadrilaterals, or linear triangles for the discretization of the bodies. To complete the discretization for the mixed variational statement, we must choose interpolant shape functions for the discrete multipliers \( \lambda_h \), (as well as their variations, \( \mu \)). \( \lambda \) can be physically interpreted as the traction field over the interfaces enforcing a tied constraint. We have defined this space on the embedded interface, \( \Gamma_s \), which fixes the definitions of master and slave sides in the mortar formulation. According to standard terminology, the side on which the multiplier field is defined is the non-mortar, or slave side. The background elements are then the mortar, or master side.

Our choice for an interpolant on the slave surface, \( \Gamma_s \), is the same field used to discretize the displacement over the surface, that is, piecewise linear shape functions supported by the existing nodes:

\[
\lambda = \sum_{A} n^s N_A^1(\xi) \lambda_A, \tag{3.30}
\]

where \( \xi \) is the piecewise surface parameterization and \( n^s \) represents the number of surface nodes (i.e., the nodes of body 1 on \( \Gamma_s \)). The shape functions \( \tilde{N}_A \) are functions used to interpolate the multipliers over \( \Gamma_s \); in the baseline mortar method, these are taken to coincide with the surface trace of the standard finite element shape functions \( N_A \). This choice is simple and convenient, but by no means unique. It is also possible, for example, to take the trace of the master side shape functions, or to judiciously
select nodes on the slave surface as supports for the multiplier field, as opposed to using them all.

We will reconsider our expressions for discrete trial functions, and express their restriction to the boundary $\Gamma_*$, as

$$\mathbf{v}^{(1)} \approx \mathbf{v}^{(1)h} = \sum_B N_B^{(1)}(\xi^{(1)}) \mathbf{c}_B^{(1)}$$

(3.31)

$$\mathbf{v}^{(2)} \approx \mathbf{v}^{(2)h} = \sum_C N_C^{(2)}(\xi^{(2)}) \mathbf{c}_C^{(2)}$$

(3.32)

As a notational point, $\xi^{(1)}$ are natural parent coordinates of the one dimensional shape functions formed by the union of slave elements with $\Gamma_*$, and $\xi^{(2)}$ are natural parent coordinates of the two dimensional shape functions from underlying master elements (Figure 3.4). One may then substitute equations (3.30), (3.31), and (3.32) into the mortar constraint virtual work (3.4) to obtain the discrete expression:

$$G^c = \sum_A \sum_B \sum_C \lambda_A \cdot \left[ n_{AB}^{(1)} \mathbf{c}_B^{(1)} - n_{AC}^{(2)} \mathbf{c}_C^{(2)} \right]$$

(3.33)

where the expressions for $n_{AB}^{(1)}$ and $n_{AC}^{(2)}$ are

$$n_{AB}^{(1)} = \int_{\Gamma_*} N_A^{1}(\xi^{(1)}) N_B^{1}(\xi^{(1)}) \, d\Gamma$$

(3.34)

$$n_{AC}^{(1)} = \int_{\Gamma_*} N_A^{1}(\xi^{(1)}) N_C^{2}(\xi^{(2)}) \, d\Gamma.$$  

(3.35)

Remark: Equation (3.33) contains a coupling between a surface quantity - parameterized by the one dimensional shape function $N_A(\xi^{(1)})$, and a volume quantity - parameterized by the two dimensional shape function $N_C(\xi^{(2)})$. This marks a departure from mortar methods as they are generally used for contact mechanics, domain decomposition, or other surface-to-surface constraints.
Evaluation of mortar integrals

The integrations in (3.34) and (3.34) are performed over subdivisions of $\Gamma_s$. We call these subdivisions mortar sections, indexed via $s$. They are determined by sectioning $\Gamma_s$ into segments to which only one master and slave element contribute shape functions (Figure 3.4). Integration is done via numerical quadrature. A parameterization, $r$, is introduced, which runs from -1 to 1 over each mortar. If we define the slave surface parent coordinates of the two ends of a mortar segment with $\xi_{a}^{(1)}$ and $\xi_{b}^{(1)}$, a mapping from the mortar coordinates, $r$, to the parent slave surface coordinates, $\xi$, is given by

$$\xi^{(1)} = \frac{1}{2}(1-r)\xi_a^{(1)} + \frac{1}{2}(1-r)\xi_b^{(1)}$$

(3.36)

Similarly, $\xi_{a}^{(2)}$ and $\xi_{b}^{(2)}$ are given as the master element reference coordinates for the mortar endpoints. The mapping from the mortar coordinates, to element coordinates, $\xi$, is given by

$$\xi^{(2)} = \frac{1}{2}(1-r)\xi_a^{(2)} + \frac{1}{2}(1-r)\xi_b^{(2)}$$

(3.37)

We can now express the mortar integrals over a single segment as

$$\left( n_{AB}^{(1)} \right)_{\Gamma_s} = \int_{-1}^{1} N_A^{(1)} \left( \xi^{(1)}(r) \right) N_B^{(1)} \left( \xi^{(1)}(r) \right) \left\| \frac{\partial \mathbf{x}^{(1)h}}{\partial \xi^{(1)}} \right\| \frac{\partial \xi^{(1)}}{\partial r} \, dr$$

(3.38)

$$\left( n_{AC}^{(2)} \right)_{\Gamma_s} = \int_{-1}^{1} N_A^{(1)} \left( \xi^{(1)}(r) \right) N_C^{(2)} \left( \xi^{(2)}(r) \right) \left\| \frac{\partial \mathbf{x}^{(1)h}}{\partial \xi^{(1)}} \right\| \frac{\partial \xi^{(1)}}{\partial r} \, dr$$

(3.39)

which can be calculated with standard Gauss quadrature (3 points in our case).
Global equation structure

The sets $S^1$ and $S^2$ contain the solid 1 interface nodes and the solid 2 background nodes adjacent to the interface boundary, i.e.,

$$S^1 = \{ A | x_A^1 \in \Gamma \}, \quad S^2 = \{ A | x_A^2 \in \Omega_\epsilon \}, \quad \text{and} \quad \bar{e} = \{ e | B_\epsilon^2 \cap \Gamma \}. \quad (3.40)$$

In addition, let the vector $\bar{d}^1$ be the collection of nodal displacements $\bar{d}_A^1, A \in S^1$ and the vector $\bar{d}^2$ of nodal displacements $\bar{d}_A^2, A \in S^2$. Using these definitions, and assuming that $\bar{N}_A$ are the normal element shape functions $N_A$, substitution of (3.30) in (3.3) yields the discrete constraint matrix equations corresponding to the standard mortar method

$$G^1 \bar{d}^1 - G^2 \bar{d}^2 = 0, \quad (3.41)$$

where matrices $G^1$ and $G^2$ are defined as

$$G^1 = [G^1_{AB}] = \sum_{s=1}^{n_s} \left( n_{AB}^{(1)} \right)_{\Gamma_s}, \quad G^2 = [G^1_{AC}] = \sum_{s=1}^{n_s} \left( n_{AC}^{(2)} \right)_{\Gamma_s} \quad \forall A, B \in S^1, \ C \in S^2 \quad (3.42)$$

---

1 Actually, we abuse notation slightly here, as the elements of the $G^i$ are given in terms of $G^i_{AB} \mathbf{I}$, where $\mathbf{I}$ is the $2 \times 2$ identity matrix in two dimensions, and $3 \times 3$ in three dimensions. We will continue this slight notational abuse throughout the thesis.
To illustrate the imposition of the interface constraints in terms of global equation structure, we partition the nodes in terms of solid interface nodes $S^1$ (3.40), body 2 solid nodes adjacent to interface $S^2$ (3.40) and all other nodes $A \setminus (S^1 \cup S^2)$. This partitioning yields the following discrete matrix equations

$$
\begin{bmatrix}
K^* & K^{*,1} & K^{*,2} & 0 \\
[K^{*,1}]^T & K^1 & 0 & [G^1]^T \\
[K^{*,2}]^T & 0 & K^2 & -[G^2]^T \\
0 & G^1 & -G^2 & 0 \\
\end{bmatrix}
\begin{bmatrix}
\frac{d}{d^1} \\
\frac{d}{d^2} \\
\lambda \\
0 \\
\end{bmatrix}
= 
\begin{bmatrix}
\tilde{f}^{(ext)} \\
\tilde{f}^{(ext)}^1 \\
\tilde{f}^{(ext)}^2 \\
0 \\
\end{bmatrix},
$$

(3.43)

where $K^*$ is the stiffness associated with nodes $A \setminus (S^1 \cup S^2)$ and $K^{*,i}$ is the stiffness coupling nodes in $S^i$ to the forces associated with non-interface nodes, for $i = 1, 2$. The partitioning is such that body 1 interface nodes occupy the second row in (3.43), body 2 interface nodes occupy the third row in (3.43) and all other nodes occupy the first row. The fourth row incorporates the constraint equations. The quantities $\tilde{f}^{(ext)}^1$ and $\tilde{f}^{(ext)}^2$ in (3.43) represent the externally imposed forces involving interface nodes. The computation of quantities associated with $S^2$ must reflect the partially filled elements (see Figure 3.3) due to the overlapping mesh of body 1.

### 3.6.4 Penalty method

The discrete form of the penalty method will involve evaluating the expression for virtual work associated with the penalty method,

$$
G^P = \beta \int_{\Gamma_*} [\mathbf{u}] \cdot [[\mathbf{w}]] \, d\Gamma.
$$

(3.44)

To expand the expression, we need to be explicit about the restriction of the displacements to $\Gamma_*:

$$
\mathbf{u}^{(1)} \approx \mathbf{u}^{(1)h} = \sum_D N_D^{(1)} (\xi^{(1)}) \tilde{d}_D^{(1)}
$$

(3.45)

$$
\mathbf{u}^{(2)} \approx \mathbf{u}^{(2)h} = \sum_E N_E^{(2)} (\xi^{(2)}) \tilde{d}_E^{(2)}.
$$

(3.46)
Plugging (3.45), (3.46), (3.32) and (3.31) into (3.44) gives

\[ G^P = \beta \left\{ \left( \int_{\Gamma_*} N_B^1 N_D^1 d \Gamma_* \right) d_D^{(1)} - \left( \int_{\Gamma_*} N_C^2 N_D^1 d \Gamma_* \right) \bar{d}_D^{(1)} \right. \]

\[ \left. - \left( \int_{\Gamma_*} N_B^1 N_E^2 d \Gamma_* \right) d_E^{(2)} + \left( \int_{\Gamma_*} N_C^2 N_E^2 d \Gamma_* \right) \bar{d}_E^{(2)} \right\}, \quad (3.47) \]

or

\[ G^P = \beta \left\{ K_p^{(1,1)} d_D^{(1)} - K_p^{(1,2)} d_D^{(1)} - K_p^{(2,1)} d_E^{(2)} + K_p^{(2,2)} d_E^{(2)} \right\}. \quad (3.48) \]

The global system of equations becomes

\[
\begin{bmatrix}
K^* & K^*_{,1} & K^*_{,2} \\
[K^*_{,1}]^T & \tilde{K}^1 + K_p^{(1,1)} & K_p^{(1,2)} \\
[K^*_{,2}]^T & K_p^{(2,1)} & \tilde{K}^2 + K_p^{(2,2)}
\end{bmatrix}
\begin{bmatrix}
d \\
\bar{d}_D^{(1)} \\
\bar{d}_E^{(2)}
\end{bmatrix}
= \begin{bmatrix}
f^{(\text{ext})} \\
f^{(\text{ext})}_{1} \\
f^{(\text{ext})}_{2}
\end{bmatrix}. \quad (3.49)
\]

As in the case of the mortar method, surface integrals are evaluated over mortars, and the global quantities are assembled as, for example,

\[ K^{(1,1)} = [K_{BD}] = \sum_{s=1}^{n_s} \left( \int_{\Gamma_*} N_B^1 N_D^1 d \Gamma_* \right)_{\Gamma_s}, \quad (3.50) \]

and similarly for the other terms.

### 3.6.5 Nitsche’s method

To form the global stiffness for Nitsche’s method we must evaluate the extra virtual work terms in equation (3.14) in a discrete setting:

\[ G^N = K^N d + (K^N)^T d \]

\[ = - \int_{\Gamma_*} \left[ [w] \otimes n^{(1)} : [\gamma \sigma^{(1)} + (1 - \gamma) \sigma^{(2)}] \right] d \Gamma \]

\[ - \int_{\Gamma_*} \left[ [u] \otimes n^{(1)} : [\gamma \sigma(w)^{(1)} + (1 - \gamma) \sigma(w)^{(2)}] \right] d \Gamma. \quad (3.51) \]
We will describe the local calculations necessary to evaluate the first integral and take advantage of the fact that second is the transpose of the first, due to the matching spaces for displacements and their variations. As in the case of the standard mortars and the penalty method, it will make sense to evaluate the discrete integral over individual mortar sections. In this case, however, there are extra degrees of freedom pulled into the calculation over a single mortar, due to presence of the stress operators on both sides of the interface. For convenience, we introduce a special vector of local nodal displacements associated with a single mortar segment, \( s \), corresponding to all of the degrees of freedom involved both in the discrete gap function, and the slave and master side stress operators for elements associated with a single mortar (1 each, master and slave). Figure 3.5 shows an example cut element and resulting modified local displacement vector.

The clearest way to present the local integrals that make up mortar-wise contri-
butions to the Nitsche related stiffness is through indicial notation. The local discrete
stiffness that results from plugging our discrete approximations for both $u^{(m)}$ and
$\sigma^{(m)}$ for a single mortar, $s$, into the first integral of 3.51 is

$$k_{pq}^{N,s} = \left[ \begin{array}{c} k_{pq}^s \end{array} \right] = \int_{\Gamma_s} ( N_A^{(1)} - N_A^{(2)} ) \left[ \gamma C_{ijkl}^{(1)} N_{B,l}^{(1)} + (1 - \gamma) C_{ijkl}^{(2)} N_{B,l}^{(2)} \right] n_j^{(1)} \ d\Gamma. \quad (3.52)$$

The indexing is done with

$$p = 2(A - 1) + i \text{ and } q = 2(B - 1) + k,$$

where $A$ indexes over the nodes which support the gap function on $\Gamma_s$, and $B$ indexes
over all of the local degrees of freedom in the vector $d^{N,s}$. $N_A^{(1)}$ is restricted to those
shape functions associated with the slave side, and $N_A^{(2)}$ is similarly restricted to
master side nodes. An example of the resulting structure for the local stiffness
matrix over the mortar in Figure (3.5) is shown below:

$$k_{pq}^{N,s} = \left[ \begin{array}{cccccccc}
 d_1^{(1)} & d_1^{(1)} & \ldots & d_4^{(1)} & d_1^{(1)} & d_1^{(1)} & \ldots & d_4^{(1)} \\
 d_1^{(2)} & d_1^{(2)} & & d_4^{(2)} & d_1^{(2)} & d_1^{(2)} & & d_4^{(2)} \\
 & & & \ddots & & & \ddots & \\
 d_4^{(2)} & & & & d_4^{(2)} & & & \\
 \cdot & \cdot & & & \cdot & \cdot & & & \\
 \cdot & \cdot & & & \cdot & \cdot & & & \\
 d_2^{(2)} & & & & d_2^{(2)} & & & \\
 d_2^{(2)} & & & & d_2^{(2)} & & & \\
 \end{array} \right]
$$

(3.53)
In an algorithmic sense, it may be more convenient and efficient to formulate the local stiffness associated with a mortar integral of the “Nitsche” terms with a vector scheme, similar to the \( B^TDB \) formulation traditionally used for local standard finite element stiffness procedures.

Each quadrant of (3.53) is filled with contributions from slightly different operators, making it easier to evaluate the local Nitsche stiffness contributions of each quadrant specifically. We will take, as an example, the expression \( \gamma N_A^{(1)} C_{ijkl} N_B^{(i)} n_j^{(1)} \), and build an algorithmically more efficient vector expression for it, based on the definitions (3.20), (3.21), and (3.23). We maintain the standard expression for the gradient matrix \( B \), given by (3.24) and (3.25). A second matrix, \( Z^e \), is introduced as, for example,

\[
Z^{e,1} = [Z^{e,1}_1, Z^{e,1}_2, \ldots, Z^{e,1}_{n_en}],
\]

where

\[
Z^{e,1}_A = \begin{bmatrix}
N_A^{(1)} n^{(1)} & 0 \\
0 & N_A^{(2)} n^{(2)} \\
N_A^{(1)} n^{(2)} & N_A^{(1)} n^{(1)}
\end{bmatrix},
\]

and \( n_{en} \) is the number of gap nodes on \( B^1 \). This allows us to express

\[
k^{(N,s,quad \ 1)} = \gamma N_A^{(1)} C_{ijkl} N_B^{(i)} n_j^{(1)} = \gamma \int_{\Gamma_s} (Z^{e,1})^T D^1 B^{e,1} d\Gamma_s, \quad (3.54)
\]

and similarly for the other three quadrants.

**Global equation structure**

The coupling of additional internal degrees of freedom in the local operator imply that the global equation structure for Nitsche’s method unfortunately does not fit conveniently in the partitioning scheme introduced for the mortar and penalty for-
mulations. We can say that
\[ \mathbf{K} + \mathbf{K}^N + (\mathbf{K}^N)^T + \mathbf{K}^S = \mathbf{F}, \]  
(3.55)
where
\[ \mathbf{K}^n = \frac{1}{{n_s}} \sum_{s=1}^{n_s} \mathbf{k}^{N,s} , \]
and the “stabilization” term, \( \mathbf{K}^S \), is evaluated in exactly the same manner as the traditional penalty term.

The choice of the parameter \( \beta \) is important to the stability of the problem. We refer to \( \beta \) as a stabilization parameter in this case rather than a penalty parameter because it no longer acts primarily to enforce the constraint. Indeed, Nitsche’s method with any value of \( \gamma \) is able to pass a patch test exactly for any value of \( \beta \), including zero. However, in the discrete sense, the subtraction of the extra interface terms leads to a global set of equations that is no longer guaranteed to be positive definite. This is related to the loss of coercivity in the functional form of the method. The penalty style operator restores coercivity (in the continuous problem), or positive definiteness (in the discrete problem) to the system of equations. A stability parameter that is too small produces unstable results, whereas a stability parameter that is too large essentially forces the problem to behave as a penalty method, which produces the same locking problems as typical Lagrange multipliers. Efficient methods to estimate optimal parameters are presented in Sanders et al. (2009) and Dolbow and Harari (2009), but the problem of choosing an effective stability parameter stands out as one of the primary drawbacks of Nitsche’s method as it is traditionally formulated.

3.7 Numerical Examples

Though ultimately intended for three dimensional use, algorithms based on the preceding methods were prototyped into a two dimensional MATLAB code with linear
triangle and quadrilateral finite elements. Finite element meshes were created in CUBIT, by Sandia National Laboratories (cubit.sandia.gov).

3.7.1 Patch tests

Nitsche’s method, whether classical or weighted, is distinguished immediately over the standard mortar method and the penalty method through patch test performance. Figure 3.6 shows two meshes with internal interfaces. Patch tests were solved by applying a linear displacement field (of 100% strain in both directions, and in shear) on all exterior nodes. For a method to pass the patch test, the internal nodes on both sides of the interface should be able to exactly represent the linear displacement field in the body, and the stresses should be uniform in either body. Mortar methods are well known to able to exactly pass patch tests for the straight interface, but small errors ($\approx 0.1\%$) were observed in the case of a curved interface. Penalty meshes cannot exactly pass a patch test except in the limit as $\beta \to \infty$, but error decreased proportionally to the value of $\beta$. Nitsche’s method passed patch tests to machine precision for any choice of $\beta$ for both the straight and curved interfaces.
3.7.2 Pure bending

A simple example can demonstrate the strengths and weaknesses of all of the proposed methods. As shown in Figure 3.7(a), we construct a rectangular \([1 \times 1.5]\) background domain (domain 1), with the origin at the center of the left boundary, and embed a smaller \([1 \times 0.625]\) rectangular body (domain 2), in left portion, so that the boundaries coincide on three surfaces, and continuity only needs to be enforced on a vertical surface at \(x = 0.625\). The crosshatch mesh in Figure 3.7(a) is constructed to represent a “worst case scenario” for the locking phenomenon, with the constrained surface cutting diagonally across the background mesh. In the case of the mesh shown, there is always an ample “material” region in cut elements. An alternate location for the interface (at \(x = 0.745\)) places the interface in a different relative position with the background elements, and is used to identify effects of elements with very small active regions.

A linear distribution of normal forces on the left and right faces should result in bending stresses dependent only on \(y\) (Figure 3.7(b)), regardless of the material parameters of the problem. No shear should be present. In this case, the material properties are either uniform throughout the entire domain \((E_1 = E_2 = 50, \nu_1 = \nu_2 = 0.3)\), or comparatively stiff in the overlapping section \((E_1 = 50000, E_2 = 50)\). The equivalent moment exerted by the external loads is \(M = 1.5\). Units are deliberately left out here, given that they can be consistently chosen in any system. We seek to realize the regular behavior of the solution without mesh dependency effects at the interface. Resultant bending stresses using standard mortar constraints are shown in Figures 3.8 and 3.9 for several cases. Next to the contoured stress plots for each is a plot of the surface traction for that particular case.

The various cases presented for standard mortars are, respectively,

1. A fine mesh overlaying a coarser mesh, with the same material parameters for
both bodies.

2. Overlapping and underlying meshes that are comparable in terms of mesh density, with an embedded stiffness 1000 times greater than the underlying stiffness.

3. A fine mesh overlaying a coarser mesh with an embedded stiffness 1000 times greater than the underlying stiffness.

4. Case (3) subject to mesh refinement on both bodies (maintaining the same relative sizes).

Though the mortar formulation is able to perform reasonably well in cases (1) and (2), mesh locking is apparent in cases (3) and (4). Figure 3.10 studies the behavior of the penalty method for the cases (3) and (4). For low penalty numbers, the stress field appears qualitatively improved over the mortar method. As the penalty number increases for the same mesh, locking reappears. For the penalty method, we can post-calculate the traction on the interface as the discrete approximation of
\begin{align}
\lambda &\approx \frac{\alpha}{2}[[u^h]]. 
\end{align}

(3.56)

The quality of the solution is also recovered with the use of the weighted Nitsche’s method with \( \left( \gamma = \frac{E_2}{E_1+E_2} \right) \), as shown in Figure 3.11, or for a single sided Nitsche’s method \( (\gamma = 0) \), as shown in Figure 3.12. In this case, the surface tractions are calculated as

\begin{align}
\lambda &\approx -\langle \sigma_h \rangle \cdot n^{(1)} + \frac{\beta}{2}[[u^h]]. 
\end{align}

(3.57)

It is also clear that the correct choice of a stabilization parameter is very important for the accuracy of the problem. A stabilization parameter which is too low will not guarantee positive definiteness of the global system of equations, which is important for a stable result. A parameter which is too high, however, will drive the penalty style term in Nitsche’s method to dominate the constraint enforcement. At high penalty parameters, the penalty method starts to behave similar to the mortar method, so high values of both \( \alpha \) and \( \beta \) can produce the same locking effect seen in the standard mortar case. Plots of the value of the stabilization parameter versus the error in terms of both \( L^2 \) and energy norms are given in Figures 3.13 for the weighted Nitsche’s method for case (4), and demonstrate the idea that there exists a best value for the parameter. The worst error is seen at values just below the positive definiteness threshold, and the greatest accuracy tends to occur with \( \beta \) only slightly greater. For the finer mesh for this problem (case (4)), a value of \( \beta = 2000 \) gives good accuracy for Nitsche’s method. Finally, stress contours are given for the various methods with an alternate interface location in Figures 3.14 and 3.15.

For this particular study, a value of \( \gamma = 0 \) gave very similar results to the weighted method, although this is not always the case. It is worth noting that the standard application of Nitsche’s method using a perfect average \( (\gamma = 0.5) \) is problematic. In
this case, the penalty number had to be very high to overcome the positive definite
ness threshold. The non-physical oscillations in stress in Figure 3.16(a), come from
a non-positive definite discrete system. When the number is high enough, however,
to overcome the restriction, the locking behavior is already present (Figure 3.16(b)).

It appears, from a visual study of the stress contours that the penalty method
and a weighted Nitsche’s method tend to perform similarly, meaning there may be
no advantage to going to the more complicated method. The advantage can be seen,
however, by investigating the convergence properties. Convergence plots for L2 dis-
placement and energy norms are given in Figure 3.18 for the problematic cases (3
and 4) for both the mortar, penalty, and Nitsche’s formulation. For generality, con-
vergence plots are based on data from unstructured background grids. An example
grid is shown in Figure 3.17. Since the ordinary metrics for convergence can be de-
ceiving in the consideration of a two body problem in which the stiffness of one body
is much greater than the stiffness of the other, we use a scaled form of the standard
error norms for the bulk displacement norm

\[ \|e(x)\|_{L_2} = \left( E_2 \int_{B^1} e \cdot e \, dB + E_1 \int_{B^2} e \cdot e \, dB \right)^{1/2}, \quad (3.58) \]

where \( e = u^h - u \), and an energy norm,

\[ \|e(x)\|_e = \left( \int_{\Omega} (\sigma^h - \sigma) : (\sigma^h - \sigma) \, d\Omega \right)^{1/2}. \quad (3.59) \]

The penalty method appears to function very well in terms of convergence in the en-
ergy norms (see Figure 3.18), which is not surprising given the excellent stress fields.
The method does not, however, converge optimally in the L2 norm of displacement.
In both cases it is necessary to scale the parameter with the mesh size, \( h \), at a rate
of \( \alpha \propto \frac{1}{h} \). The fundamental difference between the penalty method and Nitsche’s
method in terms of how well they converge in the displacement norm comes from how well they satisfy the displacement constraint at the interface.

(a) Case (1) bending stresses  
(b) Case (1) surface tractions

(c) Case (2) bending stresses  
(d) Case (2) surface tractions

**Figure 3.8**: Standard mortar constraints
Figure 3.9: Standard mortar constraints
(a) Case (3) bending stresses, $\alpha = 667$

(b) Case (3) surface tractions, $\alpha = 667$

(c) Case (4) bending stresses, $\alpha = 2,000$

(d) Case (4) surface tractions, $\alpha = 2,000$

(e) Case (4) bending stresses, $\alpha = 50,000$

(f) Case (4) surface tractions, $\alpha = 50,000$

Figure 3.10: Penalty method
Figure 3.11: Weighted Nitsche's method
(a) Case (3) bending stresses, $\beta = 583$

(b) Case (3) surface tractions, $\beta = 583$

(c) Case (4) bending stresses, $\beta = 1,750$

(d) Case (4) surface tractions, $\beta = 1,750$

(e) Case (4) bending stresses, $\beta = 50,000$

(f) Case (4) surface tractions, $\beta = 50,000$

Figure 3.12: Single sided Nitsche’s method
Figure 3.13: Error norms versus stabilization parameter for the weighted Nitsche’s method, case (4)
Figure 3.14: Mortar (a) and penalty (b) method with an alternate interface location \((x = 0.745)\)

(a) Case (4) bending stresses, \(\beta = 1,750\)  
(b) Case (4) bending stresses, \(\beta = 2,000\)

Figure 3.15: Weighted Nitsche’s methods with an alternate interface location \((x = 0.745)\)
(a) Case (4) bending stresses, $\beta = 10,000$

(b) Case (4) bending stresses, $\beta = 1,000,000$

Figure 3.16: Classic ($\gamma = 0.5$) Nitsche’s method
Figure 3.17: Example unstructured bending mesh
Figure 3.18: Convergence rates of the various methods; for linear elements, optimal slope in the L2 norm is 2 (a). Optimal slope in the energy norm is 1 (b).
3.7.3 Example in curved interfaces

Numerical interface methods are notoriously easier for straight surfaces over curved ones. A good example is the failure of mortar methods to pass patch tests on curved interfaces. For this reason two more examples were constructed to test the methods for a composite bodies with curved interfaces. The first is shown in Figure 3.19(a). It is, roughly speaking, a stiff thick walled cylinder \((B^1_{1} \text{ with } E_1 = 50000, \nu = 0.3)\), and a soft, nearly incompressible material inside \((B^1_{1} \text{ with } E_1 = 50, \nu = 0.45)\). Locking is apparent for mortar methods only for cases where there is a variable stress field over the length of the interface. Thus, the boundary conditions involve a changing pressure field on the exterior of the cylinder.

Due to the lack of an analytical solution for this case, comparisons are made to a refined finite element mesh in which the interface between the two materials is explicitly gridded. Figure 3.19 shows radial stress contours for the standard solution, a mortar interface method in which locking is apparent at the embedded interface, and a weighted Nitsche method solution. Stress contours for a penalty method and a single sided Nitsche method are qualitatively indistinguishable from the weighted Nitsche’s method.

In this first example, the locking effect at the mortar interface tends to decrease sufficiently that optimality is not lost in the displacement error norms, as it was in the flat interface example. It is clear, however, from a qualitative evaluation of the results, that the accuracy of the weighted Nitsche’s method is better in terms of stresses along the interface.
3.7.4 Multiple inclusions

In the final example, three inclusions, one straight and two curved, are embedded in a background grid. Two of the inclusions are stiffer than the surrounding material, and one is significantly softer. Material properties and boundary conditions are shown in Figure 3.20. The domain is stretched in the vertical direction via a non-uniform exterior load. A reference solution for the given boundary conditions is shown via a refined conforming mesh (Figure 3.21). The solution in terms of normal yy stresses is given in Figure 3.22 for standard mortars, Figure 3.23 for the penalty method, and Figure 3.24 for a weighted Nitsche’s method. A single sided method is not studied in this case because in one case the background grid functions as the “stiff” side.
whereas in the other two it is the “soft” side. Though it is obviously possible for a different “side” to be chosen for different inclusions, it is easier simply to allow the weighting parameter, $\gamma$ to automatically address the issue of which side of the interface is stiffer, and bias the method to the softer material. Convergence plots are shown for the multiple inclusion problem in Figure 3.25.

As in the first example, mortar methods do not attain optimal convergence in the L2 norm, while penalty methods do not achieve an optimal rate in the energy norm. The weighted Nitsche’s method is the only method that exhibits fully optimal convergence rates for both norms.
Figure 3.21: Conforming solution for \( yy \) stresses in a multiple inclusion problem

Figure 3.22: Standard mortar solution for \( yy \) stresses in a multiple inclusion problem
Figure 3.23: Penalty solution for yy stresses in a multiple inclusion problem

Figure 3.24: Weighted Nitsche solution for yy stresses in a multiple inclusion problem
Figure 3.25: Convergence rates for standard mortars, a penalty method, and a weighted Nitsche’s method
3.8 Conclusions

It can be inferred that there are major advantages to using a Nitsche’s method approach to embedded finite element constraints. With the correct biasing, Nitsche’s method never, to our knowledge, displays the same problems with mesh locking that standard mortar methods may. Additionally, there appear to be few extra challenges to extending Nitsche’s method to three dimensions. For the case of embedded meshes, regardless of the constraint enforcement, the computational geometry of re-triangulating cut cells is much more complicated in three dimensions. This is commonly simplified by cutting the background cells using a single linear fit to the overlapping interface instead of the precise geometry (this has the potential to cause problems at embedded corners). The discrete expressions for the stress operators are not more complicated than the standard expressions used to calculate element stiffness. The primary challenge appears to be the integration over interface facets. These kinds of challenges have been addressed in the mortar methods literature, in for example, Puso (2003).

There are a couple of obvious drawbacks, however, to Nitsche’s method. The method is clearly quite sensitive to the choice of stabilization parameter. Additionally, attempts to extend Nitsche’s method as it was presented in this chapter to non-linear materials have neither a precedent nor an obvious way forward, due to the specific form of the symmetry term. The remainder of this document proposes a modification to the traditional way of thinking about Nitsche’s method, so that it may be consistently extended to non-linear materials.
4 A Discontinuous Galerkin Approach to Nitsche’s Method

4.1 Introduction

Nitsche’s method, as posed in Chapter 3, poses some outstanding challenges for extension to large deformation, and other non-linear mechanics problems. In particular, it is not necessarily clear how we can obtain a natural extension of the operator $\sigma(w)$, to a non-linear material law. Additionally, current techniques for estimating the stability parameter, $\alpha$, rely heavily on linear analysis. Correct estimation of $\alpha$ is crucial to the success of Nitsche’s method. We address these problems by returning to the linear case, seeing that we can reformulate Nitsche’s method in a way that makes both an estimate for $\alpha$ and an explicit expression for $\sigma(w)$ unnecessary. The key is that we can draw comparisons between Nitsche’s method and the interior penalty method frequently used discontinuous Galerkin (DG) applications. DG methods pose the variational form of the problem over each individual element separately. Whereas in standard finite elements, continuity between the elements is built into the discrete space, it must be enforced in the variational form of DG meth-
ods. However, the techniques used to create this enforcement will give us insight into other problems of weak enforcement.

The chapter will be organized as follows. In Section 4.2 we review the literature on discontinuous Galerkin methods and the various methods of interior penalties. Section 4.3 introduces and derives the concept of a DG derivative (the key to our algorithm) in the case of Laplace’s equation for a traditional DG approach. In Section 4.4 we show the DG technique as applied to the linear elastic embedded mesh problem. Results are reviewed in Section 4.5. The linear reformulation of the problem provides a framework for a straightforward extension to (non-linear) large deformation problems given in Chapter 5.

4.2 Background and Literature Review

Discontinuous Galerkin (DG) methods have been used for the numerical solution of PDEs since their introduction in Reed and Hill (1973). The class of methods has since been studied under many variations. During the early development, DG methods developed independently under two umbrella terms: The discontinuous Galerkin method (e.g. Brezzi et al., 2000; Bassi and Rebay, 1997) and interior penalty methods (e.g. Baker, 1977; Arnold, 1982).

Elastic one-field problems with kinematic constraints, such as incompressibility, pose a well-known locking problem for conforming approximations. Locking in this context (which is similar in spirit to interface locking) motivated studies of DG methods for these elliptical classes of problems, along side other techniques such as reduced integration and assumed strain methods. DG methods avoid locking for incompressible elasticity through their relaxation of the assumptions of continuity at element boundaries (followed by subsequent weak enforcement). The techniques in this chapter are largely motivated by the work of Lew et al. (2004) and Brezzi et al. (2000), which extend the ideas of the original DG methods as applied to linear
elasticity by Bassi and Rebay (1997). The work of Bassi and Rebay is based on a particular choice of flux approximations at element boundaries. Many alternative choices exist, yielding slightly different primal formulations (see, for example, Bassi et al., 1997). An excellent review of DG methods, Arnold et al. (2002), compares the many choices of fluxes, and the resulting primal variational formulations. A close parallel is drawn to the kinds of variational statements that appear in Nitsche’s method and its related techniques. Additional acknowledgments of the close relationship between DG methods and Nitsche’s method can be seen in works like Gracie et al. (2008), which successfully blends the extended finite element method with DG methods for embedded discontinuities.

We will borrow the key idea from the DG literature that a discontinuous solution field can produce a special kind of strain operator, called the DG derivative. For a traditional discontinuous Galerkin method, this operator is used to describe the strain (or gradient) field in an approximation space in which continuity is not assumed between elements. In our case, the solution is potentially discontinuous at the boundary between the overlapping and underlying bodies. Using the DG derivative instead of a standard continuous strain operator for elasticity will yield a variational formulation with special treatment of the approximation space in the near field of the interface. We will be able to show that the variational formulation resulting from using this operator is essentially equivalent to Nitsche’s method prior to discretization. But by using the altered strain operator we will be able to perform domain integrals instead of boundary integrals. We start by demonstrating the original and traditional application of the DG derivative.

4.3 Introduction to the Discontinuous Galerkin Discrete Derivative

A great part of the following presentation is based upon notation from the review paper of discontinuous Galerkin methods presented by Arnold et al. (2002). Consider
the Laplace equation for a scalar field, $u$, as a model for an elliptic BVP in a domain $\Omega$ with boundary $\partial \Omega$.

$$-\triangle u = f \quad \text{in } \Omega$$
$$u = 0 \quad \text{on } \partial \Omega$$

(4.1)

The problem can be re-written as a first order system, which is the usual starting point for DG methods:

$$\sigma = \nabla u \quad \text{in } \Omega$$
$$-\nabla \cdot \sigma = f \quad \text{in } \Omega$$
$$u = 0 \quad \text{on } \partial \Omega$$

(4.2)

If we choose trial functions $\tau$ and $v$, we can form variational statements for each equation over any domain, $E$, that is a subset of $\Omega$, and has a boundary $\partial E$. The development of these statements follows the usual rules of integration by parts and application of the divergence theorem:

$$\int_E \sigma \cdot \tau \, dE = - \int_E u \nabla \cdot \tau \, dE + \int_{\partial E} u \tau \cdot n \, d\partial E$$

(4.3)

$$\int_E \sigma \cdot \nabla u \, dE = \int_E f \, v \, d\Omega + \int_{\partial E} v \sigma \cdot n \, d\partial E$$

(4.4)

A simple, two element example of a triangulation of the domain is shown in Figure 4.1. In this framework, we define the elements individually as $E^+$ and $E^-$. 

**Figure 4.1**: Two element domain for discontinuous Galerkin method
and the boundary between them as $\Gamma_*$, while the exterior boundary is $\Gamma_0$. The entire triangulation is the union of the two elements:

$$T_h = E^+ \cup E^-.$$ 

Other important geometrical definitions concern the decomposition of the boundary into internal and external components:

$$\partial E^+ \cup \partial E^- = \Gamma,$$ (4.5)  

$$\partial E^+ \cap \partial E^- = \Gamma_*,$$ (4.6)  

and  

$$\Gamma_0 = \Gamma \setminus \Gamma_*.$$  

The finite element spaces associated with the triangulation will be

$$S_h = \{ u_h | u_h \in H^1(E) \ \forall E \in T_h \}$$  

$$V_h = \{ \sigma_h | \sigma_h \in H^1(E) \ \forall E \in T_h \}$$  

The $H^1$ continuity of the finite element spaces is, in this case, restricted to the interior of each element, and not required across element boundaries. A discrete variational form for any element, $E$, would then be:

Find $u_h \in S_h$ and $\sigma_h \in V_h$ such that for all $v \in S_h$ and $\tau \in V_h$

$$\int_E \sigma_h \cdot \tau \, dE = - \int_E u_h \nabla_h \cdot \tau \, dE + \int_{\partial E} \hat{u} (\tau \cdot n) \, d\partial E$$  

$$\int_E \sigma_h \cdot \nabla_h v \, dE = \int_E f \, v \, d\Omega + \int_{\partial E} v (\hat{\sigma} \cdot n) \, d\partial E$$  

for every $E \in T_h$. 

103
The numerical fluxes, $\hat{\sigma}$ and $\hat{u}$, are approximations of the quantities $\sigma$ and $u$ on the boundary $\partial E$. There is more than one reasonable choice for these quantities, and we will delay making that choice. If we sum over elements of the triangulation, $T_h$, we get

$$\int_{T_h} \sigma_h \cdot \tau \, dT = -\int_{T_h} u_h \nabla_h \cdot \tau \, dT + \int_{\partial E^+} \hat{u}(\tau \cdot n^+) \, d\partial E + \int_{\partial E^-} \hat{u}(\tau \cdot n^-) \, d\partial E$$

$$\int_{T_h} \sigma_h \cdot \nabla h v \, dT = \int_{T_h} f v \, dT + \int_{\Gamma} [\hat{v}](\tau \cdot n) \, d\Gamma$$

where in this case, $[a] = a^+ - a^-$. 

Remark: Jump terms similar to those in equations (4.9) and (4.10) do not appear in the variational forms for traditional finite element formulations because continuity of the solution spaces across element boundaries is built into the discrete space, and thus boundary jumps in those spaces are inherently zero.

It is now necessary to choose a form for the approximate flux operator, $\hat{\sigma}$. A simple choice would be the numerical solution on one or the other side of the boundary, and in this case we choose $\hat{\sigma} = \sigma_h^\pm$. Since we are most interested in the expression
on the surface $\Gamma_*$ we will assume that both the choice of the numerical flux $\hat{u}$ and the weighting space $v$ are 0 on the exterior boundary, $\Gamma_0$.

We do not actually want to solve an equation in $\sigma$, but rather the primal variable, $u$, so the next step is to find an expression for $\sigma_h$ in terms of $u_h$. A second application of integration by parts to equation (4.10) gives (recalling that the terms over $\Gamma_0$ have dropped out):

$$
\int_{T_h} \sigma_h \cdot \tau \, d\mathcal{T} = \int_{T_h} \nabla_h u_h \cdot \tau \, d\mathcal{T} + \int_{\Gamma_*} [[\hat{u}_h - u_h]] (\tau^+ \cdot n^+) \, d\Gamma.
$$

A choice of $\hat{u}_h = 0$ on $\Gamma_*$, which was originally proposed in Bassi and Rebay (1997), gives

$$
\int_{T_h} \sigma_h \cdot \tau \, d\mathcal{T} = \int_{T_h} \nabla_h u_h \cdot \tau \, d\mathcal{T} - \int_{\Gamma_*} [[u_h]] (\tau^+ \cdot n^+) \, d\Gamma. \tag{4.11}
$$

In order to solve for $\sigma_h$ in (4.11), we would like to have all of the integrals over the same region, $\Omega$, whereas at the moment the last term is an integral over the internal boundary, $\Gamma_*$. To convert this to a domain integral we introduce a lifting operator, $r$, that is defined (for any domain $\Omega$ with internal or external boundary $\Gamma$) via the following integral expression for a scalar field, $a$, and a vector field $b$:

$$
\int_{\Omega} r(a) \cdot b \, d\Omega = -\int_{\Gamma} a(b \cdot n) \, d\Gamma \tag{4.12}
$$

Inserting $u_h$ and $\tau$ into the definition for $a$ and $b$ gives

$$
\int_{T_h} r([[u_h]]) \cdot \tau \, d\mathcal{T} = -\int_{\Gamma_*} [[u_h]] (\tau \cdot n) \, d\Gamma. \tag{4.13}
$$

Now, we can substitute the last term in (4.11) with an equivalent domain integral given in terms of the lifting operator, $r$. Along with the definition of $\hat{\sigma}$, this gives us an equation that we can solve for $\sigma_h$.  

105
\[ \sigma_h = \nabla_h u_h + r([u_h]). \]

Plugging (4.14) into (4.10) gives

\[ \int_{T_h} (\nabla_h u_h - r([u_h])) \cdot \nabla_h v \, dT - \int_{\Gamma_a} [[v]] (\nabla_h u_h - r([u_h])) \cdot n \, d\Gamma = \int_{T_h} f \, v \, dT. \]

Simplifying and applying the definition of \( r \) gives a final variational expression: Find \( u_h \in S_h \) such that for all \( v \in S_h \)

\[ \int_{T_h} \nabla_h u_h \cdot \nabla_h v \, dT - \int_{\Gamma_a} [[u_h]] (\nabla_h v \cdot n) \, d\Gamma - \int_{\Gamma_a} [[v]] (\nabla_h u_h \cdot n) \, d\Gamma + \int_{\Gamma_a} r([u_h]) \cdot r([v]) \, d\Gamma = \int_{T_h} f \, v \, dT \]

for every \( E \in T_h \).

Equation (4.15), which is, barring some slight modifications, the interior penalty version of the discontinuous Galerkin method due to Bassi and Rebay, has a remarkable similarity with Nitsche’s method for the Laplace equation. In this case, however, the boundary \( \Gamma_a \) is comprised of all of the internal element boundaries. Here the extra “stabilization” term arises spontaneously, and without a leading coefficient. The evaluation of this term depends on being able to evaluate the operator, \( r \), in a discrete setting. This will be addressed in detail in Section 4.4, but it is worth noting here that contributions to \( r \) will come only from boundaries where there is a potential displacement jump.

There is more than one way of deriving equation (4.15), but the preceding method gives a nice physical interpretation of the lifting operator, \( r \). The lifting operator is the mechanism by which we go from a quantity which is defined on a boundary (a set of zero measure) to a quantity which is defined on a domain. The quantity on
the boundary, $[[u]]$, is a primal variable, and the $r$ can be understood to represent a related gradient.

A faster and more flexible way of arriving at (4.15) is by using an energy functional. An appropriate energy functional in the case of the Laplace equation is

$$\Pi(u) = \int_{\Omega} \frac{1}{2} (\nabla u)^2 \, d\Omega$$

(4.16)

The solution of system (4.1) coincides with finding the stationary point of $\Pi(u)$, which involves setting the first variation with respect to $u$ to zero:

Find $u \in H^1$ such that

$$\delta[\Pi(u)] = \int_{\Omega} \nabla u \cdot \nabla v \, d\Omega = 0$$

for all $v \in H^1$. (4.17)

When moving to a triangulation, $T_h$, and a discrete operator for $\nabla u$ we use the DG derivative $\nabla u \approx \sigma_h = \nabla_h u_h + r([u_h])$, which gives a discrete variational system:

Find $u_h \in S_h$, such that

$$\int_{T_h} (\nabla_h u_h + r([u_h])) \cdot (\nabla_h v + r([v_h])) \, dT = 0.$$ 

for all $v \in S_h$ (4.18)

The system can be expanded into its separate components,

$$\delta[\Pi(u)] \approx \int_{T_h} \nabla_h u_h \cdot \nabla_h v \, dT + \int_{T_h} \nabla_h u_h \cdot r([v_h]) \, dT$$

$$+ \int_{T_h} r([u_h]) \cdot \nabla_h v \, dT + \int_{T_h} r([u_h]) \cdot r([v]) \, dT = 0.$$
Finally, a discrete variational system is obtained:

Find $u_h \in S_h$, such that

$$
\int_{T_h} \nabla_h u_h \cdot \nabla_h v \ dT + \int_{\Gamma_*} [[v]] \left( \nabla_h u_h \cdot n \right) \ d\Gamma \\
+ \int_{\Gamma_*} [[u_h]] \left( \nabla_h v \cdot n \right) \ d\Gamma + \int_{T_h} r([u_h]) \cdot r([v]) \ dT = 0
$$

for all $v \in S_h$. (4.19)

The preceding derivation gives two viable options for forming the discrete system. The more traditional option is to evaluate each term in (4.19) separately, involving integrals over the interior boundaries. In some cases, it might even be more convenient to replace the last term with a more traditional penalty term, in which case the variational form becomes identical to that of Nitsche, and discrete evaluation of $r$ is completely avoided. The modification of the last term does not affect the proof of consistency argument. A second option is to evaluate (4.18) by calculating a discrete form of $r$, and treating it as a second part of the standard gradient operator. The second method has an advantage of simplicity of implementation, and, as we will show in Chapter 5, it is more easily extended to non-linear equations. Though the two methods are variationally identical, they differ due to the discrete approximation and evaluation of $r$. In the next section, we will take the idea of a discontinuous Galerkin derivative and show how it can be extended to the case of embedded meshes in linear elasticity.

4.4 The discontinuous Galerkin derivative in the linear elastic case

Moving from a scalar Laplacian equation to the equations of elasticity requires some careful updates in notation. In addition, we wish to adapt the idea of a discontinuous Galerkin derivative for an embedded mesh application, as opposed to a true discon-
tinuous Galerkin application. We will reuse the problem setup, including domain composition and solution spaces used in (3.1).

A DG style strain operator in a domain \( \Omega = B^{(1)} \cup B^{(2)} \) with a possible jump in a vector displacement \( u \) on a boundary \( \Gamma \) (either a part of the boundary of \( \Omega \) or an internal lower dimensional feature) is

\[
D_{DG} u = \frac{1}{2} (\nabla u + \nabla^T u) + R([[u]]) - \varepsilon(u) + R([[u]]),
\]

where we define the jump in any field as \([[a]] = a^{(1)} - a^{(2)}\) and the “lifting operator”, \( R \), through its action on a vector field \( a \) and a tensor field \( b \) in the following way:

\[
\int_{\Omega} R(a) : b \, d\Omega = - \int_{\Gamma} a \otimes n^{(1)} : b^{(2)} \, d\Gamma.
\]

It will make sense in this case for the vector field to be a displacement jump and tensor field to be strain, giving

\[
\int_{\Omega} R([[u]]) : \sigma \, d\Omega = - \int_{\Gamma} [[u]] \otimes n^{(1)} : \sigma^{(2)} \, d\Gamma.
\]

The definition is not unique in terms of which normal and which stress field is used. Similarly to our physical interpretation for the Laplace equation, we are now converting a displacement jump, defined over a boundary, to a spatial gradient defined over a corresponding domain. We are enforcing continuity by insisting that the set of zero measure over which the jump is defined be subject to the same continuum assumptions as the rest of the domain.

We can then formulate Nitsche’s method from an energy functional, \( \Pi \), in which we replace the standard strain, \( \varepsilon(u) \), with a modified strain operator, \( D_{DG} u \).

As a first step, we derive a statement of virtual work for a single linear elastic body, \( \Omega \), through a stored energy function. An appropriate function for the stored
energy of a linear elastic body undergoing small deformations is \( W = \varepsilon : \sigma = \varepsilon : C \varepsilon \). The total potential energy for the body undergoing deformation, \( u \), can be given as the difference of the potential energy of internal and external forces:

\[
\Pi(u) = \Pi^{\text{int}}[u] - \Pi^{\text{ext}}[u],
\]  

(4.23)

where

\[
\Pi^{\text{int}}[u] = \int_\Omega W \, d\Omega = \int_\Omega \varepsilon(u) : C : \varepsilon(u) \, d\Omega,
\]  

(4.24)

and

\[
\Pi^{\text{ext}}[u] = \int_\Omega u f \, d\Omega.
\]  

(4.25)

By the stationary principle, an equilibrium configuration of the body can be found by setting the first variation of (4.23) to zero, giving a standard variational form, for the linear elastic single body problem:

Find \( u \in S \) such that

\[
\int_\Omega \varepsilon(w) : C : \varepsilon(u) \, d\Omega - \int_\Omega w f \, d\Omega = 0
\]

for all \( w \in V \).  

(4.26)

Now we return to the application in which there is one body embedded in another, and a potential displacement at the embedded boundary, \( \Gamma_* \). Due to the displacement jump, it is no longer sufficient to use the standard strain operator, \( \varepsilon(u) \). The modified operator \( D_{DG} \) (equation (4.20)) is used instead. The internal potential energy (4.24) becomes

\[
\Pi[u] = \int_\Omega D_{DG} u : C : D_{DG} u \, d\Omega.
\]  

(4.27)
Taking the first variation with respect to displacement gives our variational form:

Find \( \mathbf{u} \in S \) such that

\[
\int_{\Omega} (\varepsilon(\mathbf{w}) + \mathbf{R(}[\mathbf{u}])] \cdot C : (\varepsilon(\mathbf{u}) + \mathbf{R(}[\mathbf{u}])] \cdot \mathbf{d\Omega} = \int_{\Omega} \mathbf{w} f \mathbf{d\Omega}
\]

for all \( \mathbf{w} \in V. \) (4.28)

**Remark:** In a traditional discontinuous Galerkin context, the variational statement with the “DG derivative” is only given after a discrete approximation space has been introduced, due to the discontinuities in the approximation space. In this case, we have a discontinuity in the solution space of the problem, so can introduce the variational form given above even before going to a discrete approximation.

Equation (4.28) can be expanded and re-written using the definition of \( \mathbf{R} \):

Find \( \mathbf{u} \in S \) such that

\[
\sum_{m=1}^{2} \int_{B^{(m)}} \varepsilon(\mathbf{w})^{(m)} : \mathbf{\sigma}^{(m)} \mathbf{dB} - \int_{\Gamma^*} [\mathbf{w}] \otimes \mathbf{n}^{(1)} : \mathbf{\sigma}^{(2)} \mathbf{d\Gamma}
\]

\[
- \int_{\Gamma^*} [\mathbf{u}] \otimes \mathbf{n}^{(1)} : \mathbf{\sigma}^{(2)} \mathbf{d\Gamma} + \int_{\Omega} \mathbf{R([w])} : \mathbf{C : R([u])} \mathbf{d\Omega} = \int_{\Omega} \mathbf{w} \mathbf{f} \mathbf{d\Omega}
\]

for all \( \mathbf{w} \in V. \) (4.29)

**Remark:** The lifting operator used for this derivation only contains terms from the background, simplifying its discretization for this application, but alternate formulations are possible, including one which more precisely mimics the original average operators in Nitsche’s method, or even the weighted average. As presented, (4.28) and the definition of the lifting operator in (4.22) are most similar to a single sided application (based on the master side) of Nitsche’s method.
Nitsche’s method is traditionally implemented through integration of the continuous strain over the body, and integration of the jump conditions over the interface with direct assembly into the stiffness matrix. Equation (4.28) gives an alternative method. When the background body, $B^{(2)}$, is discretized, its elements are divided into three categories (see figure 4.2). Elements completely covered by the embedded body are void, elements completely free of the embedded body contribute the standard stiffness to the discrete system, based on the continuous strain operator, but elements which contain a portion of $\Gamma_*$ now contribute to both the continuous strain operator and the discrete lifting operator. In other words, in a discretized
system, we calculate $D_{DG}^e = \varepsilon_h(u_h) + R_h^e([u_h])$ in elements (as indicated by the superscript $e$) whose displacement fields are crossed by the interface (and are thus discontinuous), instead of the only standard strain $\varepsilon_h(u_h)$.

4.4.1 Calculation of a discrete lifting function - linear case

In a modified DG element we need to be able to form a discretization of the term $R([u])$. This is a local calculation which involves choosing a basis for $R$. In order to develop a discrete definition, we first need to establish a couple of definitions. A section of $\Gamma_\alpha$ that overlaps a specific element, $e$, will be called $\Gamma_e$. For convenience, we introduce a special vector of displacements,

$$
\mathbf{d}^e = \left\{ \begin{array}{c}
d_{1x}^{e(1)} \\
d_{2x}^{e(2)} \\
\end{array} \right\},
$$

(4.30)
corresponding to all of the displacement degrees of freedom involved in the discrete gap function over \( \Gamma^e \). Figure 4.3 shows an example.

Now, assuming that \( R \) has support functions \( N_a \) over the entire area of a DG element, we can write

\[
R([u]) = N_b[R_a]d. \tag{4.31}
\]

For clarity it is worth presenting both vector and indicial notation, so

\[
R_{ij}([u]) = N_b[R_{bij}]u_i = N_b[R_{baj}]d_{ai}, \tag{4.32}
\]

where \( a \) is indexed over all of the elements involved in the gap function as defined over \( \Gamma^e \). We solve for \( [R_{baj}] \) through a discrete form of the equation that defines \( R \). Equation (4.22), written over a single element, \( e \), is

\[
\int_{B_e} \mathbf{R}([u]) : \mathbf{\sigma} \, dB = -\int_{\Gamma_e} ([u] \otimes \mathbf{n}^{(1)}) : \mathbf{\sigma} \, d\Gamma,
\]

or,

\[
\int_{B_e} R_{ij}([u])\sigma_{ij} \, dB = -\int_{\Gamma_e} [u_i] \sigma_{ij} n_j^{(1)} \, d\Gamma
\]

If we assume that the basis functions \( N_b \) are constant over an element, then \( R_{ij} \) and consequently \( \sigma_{ij} \) are both constant and can be pulled out of the integral. Presuming that \( \sigma_{ij} \) is invertible, we get

\[
R_{ij}([u]) \int_{B_e} dB = -\int_{\Gamma_e} [u_i] n_j^{(1)} \, d\Gamma.
\]

or
\[ R_{ij}([u]) = -\frac{1}{V_e} \int_{\Gamma_e} (N_a^{(1)} - N_a^{(2)}) n_j d_{ai}^g d\Gamma, \quad (4.33) \]

where \( V_e \) is the volume (or area in two dimensions) of the element. As has become usual with cut elements, the volume integrals are only performed over the material, or uncovered area, meaning that \( V_e \) is the area of the material section of the element.

We are ready to define the element-wise discrete form of the variation of strain.

\[ D_{DG} = \varepsilon(u) + R([u]) = Bd + B^M d^g, \quad (4.34) \]

where \( B \) and \( d \) are the standard B matrix and element displacement vector (see Section 3.6) and

\[ B^M = -[B^M_1, B^M_2, -B^M_3, \ldots, -B^M_{n_g}] \quad (4.35) \]

\[ B^M_a = \frac{1}{V_e} \begin{bmatrix} \int_{\Gamma_e} N_a n_1 & 0 \\ 0 & \int_{\Gamma_e} N_a n_2 \\ \int_{\Gamma_e} N_a n_2 & \int_{\Gamma_e} N_a n_1 \end{bmatrix} \quad (4.36) \]

Here \( n_g \) is the number of nodes involved in the special element displacement vector \( d^g \). The final expression for a modified element stiffness is

\[ k^e = \int_{B_e} (B + B^M)^T D (B + B^M) dB_e, \quad (4.37) \]

which should be assembled using a local ID array taking into account all of the degrees of freedom in \( d \) and \( d^g \). An example of the resulting structure local for the local stiffness matrix over \( \Gamma_e \) in Figure 4.3 is shown below:
4.4.2 Pseudocode for a cut element with a DG operator

1. Determine number of local degrees of freedom, \( n_{eq} \), \( (n_{eq} = 12 \) in example). 

2. \( k^e = 0 \) \((n_{eq} \times n_{eq})\)

   \( R_{aj} = 0 \) \((n_{eq} \times 2)\)

3. Divide uncovered area into integration triangles

4. Loop over mortars

   \[ R_{aj} = R_{aj} + \int_{\Gamma_e} N_a^{(1)} n_j d\Gamma_e \]

   \[ R_{aj} = R_{aj} - \int_{\Gamma_e} N_a^{(2)} n_j d\Gamma_e \]

   end loop over mortars

5. Calculate “uncovered” area, \( V_e \)

6. \( R_{aj} = \frac{R_{aj}}{V_e} \)
7. Loop over integration triangles (Area = A_T)

(a) \( k_s^e = 0 \) \((n_{eq} \times n_{eq})\)

(b) Loop over Gauss points, \( r \)

i. Calculate \( N_{A_\xi}(\xi(r)) \)

ii. Assemble \( B \) and \( B^M \)

iii. \( k_s^e = k_s^e + (B + B^M)^T D (B + B^M) (A_T g_w) \)

iv. end loop over Gauss points

(c) \( k^e = k^e + k_s^e \), end loop over integration triangles
4.5 Numerical Results

We will return, for the most part, to the numerical example problems presented in Chapter 3, so that we may make a meaningful comparison of the discontinuous Galerkin style formulation to the methods previously studied. In this spirit, we start with the rectangular domain under a constant moment, and finish with a multiple inclusion problem. The version of the DG method as outlined is most similar to a one-sided Nitsche’s method, though it would be possible, with a little more development, to write the DG derivative to include contributions from both bodies. As implemented, the only contribution comes from the background grid. In such a case, the method is somewhat tailored to a situation in which the inclusion is either of comparable stiffness to the background grid, or much stiffer. The advantage of a stiffness weighted Nitsche’s method was that if the inclusion was much softer, the weighting would adjust the operator contributions to the soft side. With a one-sided method we no longer have that luxury. Thus, inclusions that are very soft (close to void) may not achieve the desired accuracy. We will concentrate on the situation for which the method was designed, that is, stiff overlapping meshes.

As before, the results for the traction over the interface are of interest. We consistently post-calculate the results in this case using the DG derivative:

\[ \lambda \approx (n^1)^T (\sigma n^1)_{\Gamma_s} = n^T [C : (\varepsilon(u) + R([u]))] n^1. \]  

(4.39)

4.5.1 Bending results

Figure 4.4 gives stress contours for the rectangular problem exactly as it was presented Section 3.7 for cases 3 and 4 when solved with a DG derivative technique. The accuracy of the results in this case is as good or better as the accuracy of any of the original methods, but with the added advantage that there is zero extra penalization - no stability parameter was estimated. All of the subsequent plots of
stress contours and convergence plots in this chapter represent cases of a zero extra
stability parameter.

The plots of surface traction are obviously piecewise constant in this case, as opposed to previous methods which showed up as either $C^0$ continuous or piecewise linear but discontinuous. The nature of these traction plots comes from the piecewise constant approximation for the lifting operator (and stress field) over cut elements. Thus, we get one value for every $\Gamma_e$.

Figure 4.5 shows the results for the modified mesh in which there are very small integration regions for certain cut cells. As before, the results appear to be fairly insensitive to the position of the interface. Plots of the convergence behavior in the relevant error norms are given in Figure 4.6. In this case, the method appears to converge as well as either style of Nitsche’s method, with the added advantage that there is no additional stability parameter. As an added assurance that the “automatic” stabilization is sufficient, we study the effects of adding an additional penalty term. Figure 4.7 shows the relevant error norms for a given mesh (case 3), and how they are affected by an increasing coefficient on an extra stability term, identical to the one used in a penalty method. Unlike similar graphs for a standard Nitsche’s method (Figures 3.13), there is no spike near an instability threshold.
(a) case (3) bending stresses

(b) case (3) surface tractions

(c) case (4) bending stresses

(d) case (4) surface tractions

Figure 4.4: Bending results from DG derivative
Figure 4.5: Bending results from DG derivative and an alternative interface location
Figure 4.6: Spatial convergence of the DG derivative versus other methods for bending
Figure 4.7: Effect of additional penalization for case (4)
4.5.2 Multiple inclusion problems

Due to the one-sided nature of the method, we modify the multiple inclusion problem described in Section 3.7 to have all stiff inclusions. The new setup for the problem, including material properties and boundary conditions is given in Figure 4.8. A conforming solution for $yy$-stresses is shown in Figure 4.9. Embedded meshes with a mortar constraint, a penalty constraint, and the DG derivative are shown in Figures 4.10, 4.11 and 4.12, respectively. Convergence properties in the error norms for the three methods, as well as a one-sided Nitsche’s method, are given in Figure 4.13. In both norms, convergence rates are better for the DG derivative method than for any of the other three. Results are shown with zero extra stabilization for the method, meaning that there is also no extra choice or scaling of a stability parameter. The behavior of the previously studied methods for this slightly altered problem are consistent with other numerical examples. The slightly greater than optimal convergence rates that occur here for the DG method are probably due to the comparison that is being made to a fine conforming mesh, rather than an analytical solution.
Figure 4.8: Setup for a multiple inclusion problem; all inclusions are stiff.
Figure 4.9: Reference solution with a conforming finite element grid for stiff inclusion example

Figure 4.10: Mortar solution for stiff inclusion example
Figure 4.11: Penalty solution for stiff inclusion example

Figure 4.12: DG derivative solution for stiff inclusion example
Figure 4.13: Spatial convergence to a fine conforming mesh for stiff inclusion problem
5.1 Introduction

This chapter proposes a method for stable enforcement of embedded grid constraints in a finite deformation framework. We continue to deal exclusively with elastic material behavior (no energy dissipation), even in large deformation. However, when we admit the possibility of large deformations during loading, we must account for the fact that the deformed (or current) shape of the body is different than the initial (reference, or material) configuration. The deformed configuration is unknown at the start of the analysis and thus is determined as a part of the solution process, giving rise to an inherently non-linear problem.

The method described herein is a logical extension of the linear discontinuous Galerkin (DG) interface method described in Chapter 4, reconstructed around a finite deformation framework. The chapter is organized as follows: Section 5.2 reviews the current state of literature for non-linear applications of both Nitsche and DG methods. Section 5.3 covers the basic problem formulation for a two-body fi-
finite deformation hyper-elastic boundary value problem and its standard solution in a finite element setting. Section 5.4 reformulates the problem starting with the assumption that a DG derivative is used instead of standard gradient operators. Results comparing the three methods are reviewed in section 5.5.

5.2 Literature Review

Literature reflecting the extension of either discontinuous Galerkin methods, or Nitsche’s methods to non-linear systems, such as large deformation elasticity is relatively scarce.

Current foundations for discontinuous Galerkin methods in non-linear finite elasticity are presented in Noels and Radovitzky (2006) and Ten Eyck and Lew (2006). The work of Noels and Radovitzky (2006) derives a DG method based on a Hu-Washizu-de Veubeke (see Hu (1955), Washizu (1955) and Fraijs de Veubeke (1951)) energy functional which leads to a three-field approximation space. The resulting variational formulation looks very much like a Nitsche method with non-linear stress and strain operators. A key exception is that the symmetry term is conspicuously absent. This circumvents the problem of finding an appropriate expression and possibly a linearization of the stress variation operator, and simplifies the implementation. It is also a great advantage for the extension to non-linear elastodynamics, which was outlined in Noels and Radovitzky (2008). Unfortunately, the absence of the symmetry term means that the variational form can no longer be seen in terms of minimization of an energy. Though not variationally inconsistent, it may result in sub-optimal spatial convergence.

The work of Ten Eyck and Lew (2006) is a natural extension of Lew et al. (2004) to non-linear hyper-elasticity. Its foundations are numerical fluxes found in the extensions of Bassi/Rebay fluxes to linear elasticity. The approach is to formulate a single field potential energy functional, with the caveat that the deformation gradient
is evaluated through the DG derivative. As demonstrated in the previous chapter for linear elasticity, the DG derivative essentially contains all the information about the choices of approximate numerical fluxes, and its use can circumvent the need to explicitly evaluate surface integrals. Stability of the formulation is discussed via the problem of coercivity of the linearized elasticity problem. Though restrictions on the elastic moduli are shown to result in the desired coercivity problem, a complete case for overall stability of the formulation remains an open question.

We will draw on the basic ideas in Ten Eyck and Lew (2006), in that we will take advantage of their justification for extension of a DG derivative for a hyperelastic material. In our case we will apply it to the discontinuity that exists in the functional form of our problem - the interface, as opposed to standard DG discrete discontinuity which occurs at the boundary of elements.

5.3 Problem Formulation

We start by considering the two domains shown in Figure 5.1. Our formulation is general to cases of both complete (as shown) and partial overlap of the bodies.

Figure 5.1: Two body problem in finite deformation
The coordinates of body \(m\) in the material configuration are denoted as \(X^{(m)}\) and coordinates in the current configuration are denoted as \(x^{(m)}\). As before, the entire computational domain is considered as the union of \(B^1\), with the uncovered region of \(B^2, \Omega = B^1 \cup (B^2 \setminus B^1)\). Excluding \(\Gamma\), the rest of the computational boundary can be divided into Dirichlet and Neumann parts on each body, \(\Gamma_d^{(m)}\) and \(\Gamma_h^{(m)}\) respectively.

The deformation of the bodies are found through the mapping

\[
x^{(m)} = \varphi^{(m)} \left( X^{(m)} \right).
\]  

(5.1)

We continue to use \(u\) to indicate a total deformation,

\[
x^{(m)} = X^{(m)} + u^{(m)}(X^{(m)}),
\]  

(5.2)

but the most important measure of deformation is the deformation gradient given by

\[
F^{(m)} = \frac{\partial x^{(m)}}{\partial X^{(m)}} = \frac{\partial \varphi^{(m)}}{\partial X^{(m)}} = 1 + \frac{\partial u^{(m)}}{\partial X^{(m)}}. 
\]  

(5.3)

It is occasionally convenient to express quantities in terms of indicial notation. It is particularly useful in this case to distinguish between quantities defined in the material configuration, \(X\), and the current configuration, \(x\). Unless otherwise noted, we take uppercase letters to index quantities in the material configuration and lower case indices to index to the current configuration. Thus, it is possible to express the deformation gradient as

\[
F_{iJ}^{(m)} = \frac{\partial x_i^{(m)}}{\partial X_j^{(m)}} = \delta_{iJ} + \frac{\partial u_i^{(m)}}{\partial X_j^{(m)}}. 
\]  

(5.4)

The problem is posed in a total Lagrangian framework, meaning that we use stress and strain measures in the material configuration. We may use the deformation gradient to update to the current configuration. It is most convenient to use, as a
measure of strain, either the right Cauchy-Green deformation tensor:

\[ C^{(m)} = \left( F^{(m)} \right)^T F^{(m)}, \]  

(5.5)

or the Green strain tensor

\[ E^{(m)} = \frac{1}{2} (C^{(m)} - I). \]  

(5.6)

Two important quantities describing the strain behavior will be the first invariant of strain:

\[ I^{(m)} = Tr \left( C^{(m)} \right), \]  

(5.7)

and the jacobian

\[ J^{(m)} = \det \left( F^{(m)} \right) \]  

(5.8)

which is a measure of total volumetric change.

For hyper-elastic material behavior, we assume the existence of some stored energy function, \( W \), which represents the work required to produce a strain, and from which we can compute stress by differentiating with respect to strain. This assumption guarantees the path independence of the material behavior. It is also restricted to have the dependance \( W = W(C) \) due to material frame indifference. Additionally, if the material is isotropic, the energy only depends on strain invariants. The model we use here is a standard Neo-Hookean model, described by

\[ W^{(m)} = \frac{1}{2\mu} \left[ I^{(m)} - 3 - 2 \ln \left( J^{(m)} \right) \right] + \frac{\lambda}{4} \left[ \left( J^{(m)} \right)^2 - 1 \right] - \frac{\lambda}{2} J^{(m)}. \]  

(5.9)

A convenient measure of stress is the first Piola-Kirchoff stress, which is obtained via direct differentiation of the stored energy function:

\[ P^{(m)} = \frac{\partial W^{(m)}}{\partial F^{(m)}}. \]  

(5.10)
Though it is convenient to pose the boundary value problem in terms of $P^{(m)}$, it is easier to evaluate the constitutive equations through the second Piola-Kirchoff stress (PK2), which is

$$S^{(m)} = 2 \frac{\partial W^{(m)}}{\partial C^{(m)}} = \mu \left[ 1 - \left( C^{(m)} \right)^{-1} \right] + \frac{\lambda}{2} \left[ \left( J^{(m)} \right)^2 - 1 \right] \left( C^{(m)} \right)^{-1}, \quad (5.11)$$

or, in indicial notation,

$$S^{(m)}_{IJ} = 2 \frac{\partial W^{(m)}}{\partial C^{(m)}} = \mu \left[ \delta_{IJ} - \left( C^{(m)}_{IJ} \right)^{-1} \right] + \frac{\lambda}{2} \left[ \left( J^{(m)} \right)^2 - 1 \right] \left( C^{(m)}_{IJ} \right)^{-1}.$$ 

The relationship between the two stress measures is

$$P^{(m)} = S^{(m)} \cdot \left( F^{(m)} \right)^T. \quad (5.12)$$

### 5.3.1 Hyperelastic boundary value problem

For simplicity, we consider the case in which the materials are not subjected to body loads, though we allow the possibility of surface loads $T^{(m)}$, and prescribed boundary displacements, $\bar{u}^{(m)}$. The equations of motion for the finite deformation case, in which the bodies $B^1$ and $B^2$ are forced to move in concert at $\Gamma_\ast$, are

$$\nabla_0 \cdot P^{(m)} = 0 \quad \text{in } \Omega \quad (5.13a)$$

$$u^{(m)} = \bar{u}^{(m)} \quad \text{on } \Gamma_d^{(m)} \quad (5.13b)$$

$$P^{(m)} n^{(m)} = T^{(m)} \quad \text{on } \Gamma_h^{(m)} \quad (5.13c)$$

$$u^{(1)} = u^{(2)} \quad \text{on } \Gamma_\ast \quad (5.13d)$$

$$P^{(1)} n^{(1)} = -P^{(2)} n^{(2)} \quad \text{on } \Gamma_\ast. \quad (5.13e)$$

Equations (5.13d) and (5.13e) represent the coupling through enforced continuity at the domains’ shared boundary. Before treating the entire system, we review a the standard method for treating the two body problem in the absence of coupling.
5.3.2 Statement of virtual work for the uncoupled problem

To define the virtual work, we start by defining the solution and weighting space, \( S^{(m)} \) and \( V^{(m)} \), consisting of solutions \( u \) and their variations \( \delta u \)

\[
S^{(m)} = \{ u^{(m)} | u^{(m)} \in H^1(B^m), u = \bar{u} \text{ on } \Gamma_d \}
\]

\[
V^{(m)} = \{ \delta u^{(m)} | \delta u^{(m)} \in H^1(B^m), \delta u = 0 \text{ on } \Gamma_d \}
\]

From the stored energy function, \( W \), we deduce a total potential energy for the system:

\[
\Pi(u) = \Pi^{int}(u) - \Pi^{ext}(u)
\]

\[
= \sum_{m=1}^{2} \int_{B^m} W^{(m)}(C^{(m)}) dB - \int_{\Gamma^m_h} u^{(m)} \cdot T^{(m)} d\Gamma.
\]  \( (5.14a) \)

We use stationarity of the potential energy to obtain a virtual work statement for the case in which continuity is not enforced at \( \Gamma_* \), that is, omitting \( (5.13d) \) and \( (5.13e) \). Taking the first variation of \( (5.14a) \), and setting it to zero gives:

Find \( (u^1, u^2) \in (S^1 \times S^1) \) such that

\[
\int_{B^0_0} (\delta F^{(m)})^T : P^{(m)} dB = \int_{\Gamma^m_h} \delta u^{(m)} \cdot T^{(m)} d\Gamma
\]

for all \((v^1, v^2) \in (V^1 \times V^2)\), \( (5.15) \)

or, equivalently:

Find \( (u^1, u^2) \in (S^1 \times S^1) \) such that

\[
\int_{B^0_0} \delta E^{(m)} : S^{(m)} dB = \int_{\Gamma^m_h} \delta u^{(m)} \cdot T^{(m)} d\Gamma
\]

for all \((\delta u^1, \delta u^2) \in (V^1 \times V^2)\). \( (5.16) \)
5.3.3 Discrete form

Spatial discretization is done over the two domains independently. A finite element in body $B^{(m)}$ will be denoted $B^{(m)}_{e}$. Finite dimensional subspaces of the solutions and their variations are denoted $S^{(m)h}$ and $V^{(m)h}$ via

$$S^{(m)h} = \{ u^{(m)h} | u^{(m)h} \in C^0(B^{(m)}), u = \varphi \text{ on } \Gamma_{d}^{(m)} \}$$

$$V^{(m)h} = \{ \delta u^{(m)h} | \delta u^{(m)h} \in C^0(B^{(m)}), \delta u = 0 \text{ on } \Gamma_{d}^{(m)} \}.$$  

Standard bilinear shape function technology is used, such that

$$u^{(m)h} = \sum_{A} N_{A}(\xi^{(m)}) d_{A}^{(m)} \quad (5.17)$$

$$\delta u^{(m)h} = \sum_{B} N_{B}(\xi^{(m)}) e_{B}^{(m)}, \quad (5.18)$$

where $\xi$ are the parent coordinates of a reference element, and $d$ is the vector of nodal displacement unknowns. The discrete deformation gradient is evaluated via

$$F^{(m)} = I + \frac{\partial N_{A}^{(m)}}{\partial X^{(m)}} d_{A}^{(m)}, \quad (5.19)$$

or

$$F_{iJ}^{(m)} = \delta_{iJ} + \frac{\partial N_{A}^{(m)}}{\partial X_{j}^{(m)}} d_{iA}^{(m)}.$$  

The discrete tensor expressions for $C^{(m)}$, $E^{(m)}$, and $S^{(m)}$ can be evaluated via $(5.19, (5.5), (5.6)$ and $(5.11)$. The variation of Green’s strain, $\delta E$, is

$$\delta E_{iJ}^{(m)} = \frac{1}{2} \left( \frac{\partial \delta u_{i}^{(m)h}}{\partial X_{i}^{(m)}} F_{iJ}^{(m)} + \frac{\partial \delta u_{j}^{(m)h}}{\partial X_{j}^{(m)}} F_{ij}^{(m)} \right). \quad (5.20)$$
We convert this to Voigt vector form:

\[
\delta \mathbf{E}^{(m)} = \begin{bmatrix}
F^{(m)}_{11} \delta u^{(m)}_{i,1} \\
F^{(m)}_{12} \delta u^{(m)}_{i,2} \\
F^{(m)}_{11} \delta u^{(m)}_{i,1} + F^{(m)}_{12} \delta u^{(m)}_{i,1}
\end{bmatrix} = \mathbf{B}_A^{(m)} \mathbf{d}_A^{(m)},
\]

where

\[
\mathbf{B}_A^{(m)} = \begin{bmatrix}
F^{(m)}_{11} N^{(m)}_{A,1} & F^{(m)}_{21} N^{(m)}_{A,1} \\
F^{(m)}_{12} N^{(m)}_{A,2} & F^{(m)}_{22} N^{(m)}_{A,2} \\
F^{(m)}_{11} N^{(m)}_{A,1} + F^{(m)}_{12} N^{(m)}_{A,2} & F^{(m)}_{21} N^{(m)}_{A,2} + F^{(m)}_{22} N^{(m)}_{A,1}
\end{bmatrix}.
\]

With these definitions in hand, we form the left side of 5.15 and obtain internal forces as a function of displacement for an element in body \(m\):

\[
f^{\text{int},e,m}(\mathbf{d}) = \int_{B^m_e} \mathbf{B}_A^T \mathbf{S} dB.
\]

The global internal force is then assembled:

\[
\mathbf{F}^{\text{int},m} = \sum_{e=1}^{n_{el}} A_e \left( f^{\text{int},e,m} \right).
\]

A global statement of equilibrium is

\[
\mathbf{R}(\mathbf{d}) = 0,
\]

where the global force vector, \(\mathbf{R}\) includes external body and Neumann boundary forces, and internal forces due to element stresses via

\[
\mathbf{R}(\mathbf{d}) = \mathbf{F}^{\text{ext}} - \mathbf{F}^{\text{int}}.
\]

Here, \(\mathbf{F}^{\text{int}}\) includes uncoupled contributions from both bodies. We use the consistent Newton-Raphson method, and generate the following linear system to be solved
in an iteration, \( k \):

\[
\mathbf{R} + \left[ \frac{\partial \mathbf{R}}{\partial \mathbf{d}} \right]^k \Delta \mathbf{d}^k = 0, \tag{5.27}
\]

or, after rearranging,

\[
\mathbf{K}^k \Delta \mathbf{d}^k = \mathbf{R}, \quad \mathbf{K}^k := \left[ \frac{\partial \mathbf{F}^{(\text{int},k)}}{\partial \mathbf{d}} \right]^k \tag{5.28}
\]

with an update on \( \mathbf{d} \):

\[
\mathbf{d}^{i+1} = \mathbf{d}^i + \Delta \mathbf{d}. \tag{5.29}
\]

A consistent linearization of (5.24) is needed to correctly evaluate \( \mathbf{K} \). Taking a directional derivative of the internal force term, with respect to \( \mathbf{d} \) gives

\[
\frac{\partial \mathbf{F}^{\text{int}}(\mathbf{d})}{\partial \mathbf{d}} = \int_{\Omega} \mathbf{B}^T \mathbf{D} \mathbf{B} \, \mathbf{d} \, d\Omega + \int_{\Omega} \frac{\partial \mathbf{B}}{\partial \mathbf{u}} \mathbf{S} \, \mathbf{d} \, d\Omega \tag{5.30}
\]

where \( \mathbf{D} \) is a matrix of tangent moduli deduced (through Voigt notational compression) from material constitutive relations \( A_{IJKL} \), where

\[
A_{IJKL} = 2 \frac{\partial^2 W}{\partial C_{IJ} \partial C_{KL}} = \lambda J^2 C^{-1}_{IJ} C^{-1}_{KL} + \mu \left[ 1 + \frac{\lambda}{2\mu} (1 - J^2) \right] (C^{-1}_{IK} C^{-1}_{JL} + C^{-1}_{IL} C^{-1}_{JK}). \tag{5.31}
\]

If the consistent tangent, \( \mathbf{K} \), is recalculated at every iteration, \( k \), the norm of the \( \mathbf{R}, \| \mathbf{R} \| \), is expected to converge upon zero at a quadratic rate, assuming that the initial guess for displacement \( \mathbf{d} \) is in the basin of attraction of the solution.

The expression for \( \mathbf{K} \) in (5.28) assumes that \( \mathbf{F}^{(\text{ext})} \) is independent of \( \mathbf{d} \). This is not always the case, as in the case of pressure loads, but a consistent linearization
of a pressure term is a fairly standard procedure (see, for example, Zienkiewicz and Taylor, 1967).

5.4 DG Derivative in the Non-Linear Case

The discontinuous Galerkin derivative gives a natural framework for the extension of Nitsche’s method into non-linear problems. To see this we must re-derive the variational form of the non-linear problem based on different assumptions about the gradient operators. First, we revisit the choice of solution spaces and the idea of the displacement gradient. Instead of defining separate solution spaces for each body, we make one definition of $S$ and $V$ containing the set of solutions and their variations, respectively:

$$S = \{ u | u \in H^1(\Omega), u = \bar{u} \text{ on } \Gamma_d, \ u \text{ discontinuous on } \Gamma_s \}$$

$$V = \{ \delta u | \delta u \in H^1(\Omega), \delta u = 0 \text{ on } \Gamma_d, \delta u \text{ discontinuous on } \Gamma_s \}.$$  

We propose that in the presence of a discontinuous displacement field, such as that found in $\Omega$ for this problem, an appropriate gradient is:

$$D_{DG}u = F + R([u]). \tag{5.32}$$

Here $R([u])$ is a lifting function defined as in (4.20) and $F$ is the standard gradient of a continuous field. We then proceed to form an energy functional for some hyperelastic material with a stored energy function, $W = W(X, D_{DG}u)$.

$$\Pi(u) = \int_\Omega W(X, D_{DG}u) \, d\Omega - \int_{\Gamma_h} \left( \begin{array}{c} u^{(m)} \\ T^{(m)} \end{array} \right) \cdot \left( \begin{array}{c} n \end{array} \right) \, d\Gamma. \tag{5.33}$$

We may re-use, for example, the energy density function associated with a hyperelastic Neo-hookean material (Equation (5.9)), with the understanding that the quantities $C, E, J$ and $I$ are functions of $D_{DG}u$:

$$C^{DG} = (D_{DG}u)^T D_{DG}u. \tag{5.34}$$
\[ E^{DG} = \frac{1}{2} \left( C^{DG} - 1 \right), \]  
(5.35)

\[ I = Tr(C_{DG}), \]  
(5.36)

and

\[ J = \det(D_{DG}u) \]  
(5.37)

Stresses are now also a function of the DG derivative, for example the PK2 stress becomes:

\[ S^{DG} = 2 \frac{\partial W(X, D_{DG}u)}{\partial C^{DG}}. \]  
(5.38)

We seek a deformation, \( u \), that satisfies stationarity of (5.33). This gives a variational problem:

Find \( u \in S \) such that

\[ \int_{\Omega} \frac{\partial W}{\partial D_{DG}u}(X, D_{DG}u) : D_{DG}\delta u \, d\Omega = \int_{\Gamma^{(m)}_h} \delta u^{(m)} \cdot T^{(m)} \, d\Gamma \]

for all \( \delta u \in V \),

(5.39)

which is equivalent to:

Find \( u \in S \) such that

\[ \int_{\Omega} \frac{\partial W}{\partial C_{DG}}(X, D_{DG}u) : \delta E_{DG} \, d\Omega = \int_{\Gamma^{(m)}_h} \delta u^{(m)} \cdot T^{(m)} \, d\Gamma \]

for all \( \delta u \in V \).

(5.40)

### 5.4.1 Discrete form

Despite the redefinition of the solution space, the domain is discretized in the same way as in the uncoupled case, that is, the bodies are gridded independently, with the same discretization mismatch allowed to occur at \( \Gamma_u \). Finite dimensional subspaces
Figure 4.2: The background grid divided into traditional (white), void (gray) and DG elements (teal)

in this case are:

\[ S^{(m)h} = \left\{ u^{(m)h} | u^{(m)h} \in C^0(B^m), u = \varphi \text{ on } \Gamma_d^{(m)} \right\} \]

\[ V^{(m)h} = \left\{ \delta u^{(m)h} | \delta u^{(m)h} \in C^0(B^m), \delta u = 0 \text{ on } \Gamma_d^{(m)} \right\} . \]

For clarity, we reproduce figure 4.2, which shows the overlapping body superposed over the background discretized body 2. For simplicity, the overlapping elements are not shown.

As in the linear case, all elements of body 2 will contribute to the discrete stiffness of the system. Body 1 contains three categories of elements: (1) those that are void,
(2) those that are complete, and (3) cut elements which contribute partial stiffness. We call these elements the DG elements. In the third category of elements, the displacement field is supported by the master element nodes only in the material region, and the overall solution is supplanted by the slave elements where they overlap. In DG elements, which potentially support a discontinuous solution, the displacement gradient is calculated discretely as

$$D_{DG} u^h = 1 + \frac{\partial N_A}{\partial X} d_A + R([u^h]),$$

or

$$D_{DG} = \delta_i + \frac{\partial N_A}{\partial X_I} d_{AI} + R^h_i([u^h]).$$

Away from the interface, the discrete displacement gradient reduces to a standard form (5.19).

As in section 4.4.1, we choose the support of $R$ to be piecewise constant over cut elements. The discrete expression for $R$ is then (due to the development in section 4.4.1):

$$R_{iJ}([u]) = \frac{1}{V_E} \int_{B_E} (N_A^{(1)} - N_A^{(2)}) n_J d_{AI} d\Omega,$$

where $V_E$ is the volume, or area of the material region of the cut element, and $n_J$ is the normal, in material coordinates, of $B^1$. The vector $d_{AI}^g$ is a set of nodal degrees of freedom associated with the support of the gap function.

The variation of strain in DG elements becomes

$$\delta E_{ij}^{DG} = \frac{1}{2} \left( \frac{\partial \delta u_i}{\partial X_I} (F_{iJ} + R_{iJ}) + \frac{\partial \delta u_i}{\partial X_J} (F_{iJ} + R_{iJ}) \right).$$

142
In Voigt vector notation this is

$$\delta E^{DG} = \left\{ \begin{array}{c} F_{i1}\delta u_{i,1} \\ F_{i2}\delta u_{i,2} \\ F_{i1}\delta u_{i,2} + F_{i2}\delta u_{i,1} \end{array} \right\} + \left\{ \begin{array}{c} R_{i1}\delta u_{i,1} \\ R_{i2}\delta u_{i,2} \\ R_{i1}\delta u_{i,2} + R_{i2}\delta u_{i,1} \end{array} \right\}$$

$$= Bc + B^M c^g$$

where $B$ is as defined in (5.22), and

$$B^M = -[\begin{bmatrix} B^M_1 & B^M_2 & B^M_3 & \cdots & B^M_{n_d} \end{bmatrix}],$$

$$B^M_A = \left\{ \begin{array}{cc} R_{11}N_{A,1} & R_{21}N_{A,1} \\ R_{12}N_{A,2} & R_{22}N_{A,2} \\ R_{11}N_{A,2} + R_{12}N_{A,2} & R_{21}N_{A,2} + R_{21}N_{A,1} \end{array} \right\}.$$  (5.47)

Local evaluation of the internal forces in DG elements gives

$$f_{DG}^{int,e,m}(d) = \int_{B^e} (B + B^M)^T S_{DG} dB.$$  (5.48)

The consistent tangent associated with the linearization of this term for a Newton-Raphson iteration is:

$$\frac{\partial f^{int}(d)}{\partial d} = \int_{B^e} (B + B^M)^T D^{DG}(B + B^M) dB + \int_{B^e} \frac{\partial (B + B^M)}{\partial d} S^{DG} dB.$$  (5.49)

Integration of the DG elements is done with Gauss quadrature over sub-triangles. Standard finite element assembly of element internal forces and tangent stiffness applies, and the global problem is solved with Newton-Raphson iteration.

5.4.2 Pseudocode for non-linear DG element

The following is an algorithm for finding the internal force and consistent tangent associated with a hyper-elastic DG element. Figure 4.3 is repeated here for reference.
Figure 4.3: DG element with associated degrees of freedom

1. Determine number of local degrees of freedom, \( n_{eq} \).

2. \( k^e = 0 \ (n_{eq} \times n_{eq}) \)
   \[ f^{int} = 0 \ (n_{eq} \times 1) \]
   \[ R_{AJ} = 0 \ (n_{eq} \times 2) \]

3. Divide uncovered area into integration triangles

4. Loop over mortars
   \[ R_{AJ} = R_{AJ} + \int_{\Gamma_e} N_A^{(1)} n_J \ d\Gamma \]
   \[ R_{AJ} = R_{AJ} - \int_{\Gamma_e} N_A^{(2)} n_J \ d\Gamma \]
   end loop over mortars

144
5. Calculate “uncovered” area, $V_E$

6. $R_{AJ} = \frac{R_{AJ}}{V_E}$

7. Loop over integration triangles (Area = $A_T$)

   (a) $k^e_s = 0 \ (n_{eq} \times n_{eq})$

   $f^{\text{int}}_s = 0 \ (n_{eq} \times 1)$

   (b) Loop over Gauss points (local coordinates $r$, gauss weights $g_w$)

      i. Calculate $\beta = N_{J\xi}(\xi(r))$

      ii. $H_{iJ} = \frac{\partial N_A}{\partial X_J} d_Ai + R_{AJ}d_Ai$

      iii. $D_{iJ} = H_{iJ} + \delta_{iJ}$

      iv. $J = \det(D)$

      v. $C_{IJ} = D^T D$

      vi. Compute $S$ and $A_{IJKL}$ via constitutive equation

      vii. Assemble $B$ and $B^M$ via (5.22) and (5.47)

      viii. $k^e_s = k^e_s + (B + B^M)^T D^{DG} (B + B^M) (A_T g_w) + (\beta)^T S^{DG}(\beta) (A_T g_w)$

      $f^{\text{int}}_s = f^{\text{int}} + (B + B^M)^T S^{DG}(A_T) (g_w)$

      ix. end loop over Gauss points

   (c) $k^e = k^e + k^e_s$

      $f^{\text{int}} = f^{\text{int}} + f^{\text{int}}_s$

   (d) end loop over integration triangles

5.5 Results

For our non-linear examples, we measure the accuracy and optimality of our methods in comparison to standard finite element solutions.
5.5.1 Bending, and convergence

If we return to the rectangular problem, it is now reasonable to apply large pressure loads, and visualize the displacement.

The geometry and material properties for the rectangular problem in Section 3.7 are recycled here. Loading, however, is increased 100 times, and reversed in direction. Figure 5.2 shows a conformed solution. For the sake of rendering, the gridding is not shown. Figure 5.3 shows a mortar solution, both in terms of x directional stresses and interface tractions. We cannot plot an exact solution, but it is reasonable to expect that the traction should somehow resemble a linear function from one side to the other, which is clearly not the case. Results for the DG derivative formulation are shown in Figure 5.4 and appear qualitatively improved, as we have come to expect. Convergence rates to a fine conforming mesh are given in Figures 5.5 and 5.6 for mortar methods, a penalty method, and the DG derivative method. The expected results, based on the linear numerical tests, are that the mortar method does not converge properly in the energy norm, the penalty method does not converge optimally in the L2 norm of displacements, and that the DG derivative method converges optimally for both. This is nearly the case. Both the penalty method and the DG derivative method have slightly less than optimal rates in the energy norm. The DG derivative has a slightly better rate than the penalty, but the reason that the slope is not quite 1 remains a question.
Figure 5.2: Converged conforming finite element solution for non-linear bending

Figure 5.3: Mortar solutions for non-linear bending: (a) Cauchy stresses in x, (b) surface tractions
Figure 5.4: DG solutions for non-linear bending: (a) Cauchy stresses in x, (b) surface tractions

Figure 5.5: Spatial convergence for L2 norm in displacement for non-linear bending
Figure 5.6: Spatial convergence for energy norm in displacement for non-linear bending
5.5.2 Golf ball

We will give two more examples of embedded meshes for non-linear elastic materials. The first is a model which appears to be satisfactorily represented by either a mortar or a DG method, and the second will again demonstrate the advantages of the DG method for a physically reasonable system. Common examples of composite bodies in which elastic material properties are very different is sports equipment, where strength and weight are competing factors in performance. Here we choose to model a golf ball under a pressure load (see Figure 5.7). Golf balls are typically made of either one or two layers over a solid or gel core. In this model the outer shell and inner core have a stiffness difference by a factor of 30, and the inner core has $\nu = 0.45$ whereas the outer core has $\nu = 0.3$. The embedded meshes given approximately the same mesh density. It is obvious in this case that a conforming mesh is just as simple to build, though this may not be as convenient if the outer layer is comparatively very thin. However, circular inclusions may be a very desirable use of embedded grids for the reason that conforming meshes cannot properly capture free rotations of circular inclusions, due to the linear spatial interpolation of the geometry. If the continuity constraints on a circular inclusion are only normal in direction, there would be no spatial impediment to rotation. In this case, full continuity is enforced, and the angular stresses are studied for a conforming mesh (Figure 5.8(a)), a mortar method (Figure 5.8(b)), and the DG method (Figure 5.8(c)). It is difficult to discern any major qualitative differences between the three methods.
Figure 5.7: Setup for loaded golf ball problem
Figure 5.8: $\theta\theta$ stresses for loaded golf ball problem
5.5.3 Reinforced conveyor belt problem

The final example is a cross-sectional plane strain simulation of loading on a reinforced conveyor belt. These kinds of systems are used for heavy duty industrial applications such as mining. Steel cords are generally used as the reinforcement for a rubber belt. Both materials can be reasonably modeled with a Neo-Hookean constitutive model in the elastic regime, although their overall strength and compressibility are very different. A setup is shown in Figure 5.9. The rubber belt is modeled with an effective stiffness of $E = 0.1$ and Poisson's ratio of $\nu = 0.45$. These material parameters can be converted to the lamé parameters, $\lambda$ and $\mu$, which are direct inputs to the Neo-Hookean material model given by (5.9). The relationship is:

$$\lambda = \frac{E \nu}{(1 + \nu)(1 - 2\nu)}$$  \hspace{1cm} (5.50)

$$\mu = \frac{E}{2(1 + \nu)}$$  \hspace{1cm} (5.51)

The steel inclusions are modeled with a stiffness of $E = 200$ and $\nu = 0.3$. The right side of the problem is considered a symmetry plane, and we apply a pressure loading. The loading is deliberately larger than would be physically reasonable for industrial applications, so as to allow us to observe very large deformations of the materials. A conforming solution is given in Figure 5.10. Though overall displacement seems reasonable in the mortar solution (Figure 5.11), it is clear that the stress values in the steel are non-sensical – this is due to the stiffness difference between the materials. The relative mesh densities were allowed to be similar in this case, to demonstrate that only one of our two important parameters - relative mesh density and relative stiffness, can be enough to produce a locking effect. The solution due to the DG method returns a reasonable approximation of the stress field in comparison to the conforming solution (Figure 5.12).
Figure 5.9: Setup for reinforced conveyor belt problem

Figure 5.10: Conforming solution for reinforced conveyor belt problem
Figure 5.11: Mortar solution for reinforced conveyor belt problem

Figure 5.12: DG derivative solution for reinforced conveyor belt problem
5.6 A Return to Multi-Physics and fluid/structure Interaction

In the beginning of this document, we stated that the intention of the methods to be studied was to take steps towards techniques that would robustly handle simulation of multi-physics phenomenon. Though we focused on the fluid/structure interaction problem in Chapter 2, the remainder of the work focused exclusively on the solid mechanics applications of embedded finite element methods. It is relevant then, to focus a bit on how these methods could be expanded to treat fluid/structure interaction in finite elements. In fact, embedded mesh techniques have already been proposed for fluid/structure interaction using a sharp interface capture method similar to that which we have outlined for structural mechanics, (see, for example, Gerstenberger and Wall, 2008).

The problem of interface stability for these kinds of techniques is relevant. Embedded fluid/structure interaction techniques would have the same shortcomings in terms of over-constraint and mesh locking as the purely solid mechanics case. It is an important next step for this work to be applied to a more general fluid/structure interaction framework. The other key issues for an embedded approach to full fluid/structure interaction can be summed up as follows:

1. The structural algorithms should be formulated and validated in a dynamic framework. This is something that has already been successfully done for the basic embedded mesh framework, though not for the discontinuous Galerkin approach. A relevant benchmark for embedded dynamics, reproduced with permission from Lawrence Livermore National Laboratory, is demonstrated in Figure 5.13. In the simple two dimensional example, an elastic wave is sent through a uniform material. In the control case, the mesh is completely regular. The same numerical experiment is conducted on an embedded mesh in which the materials are the same. Examination of the results shows that there are
no spurious oscillations due to mesh effects at the embedded boundary. It is not clear, however, that the DG derivative will not adversely affect the stable time-step for dynamic simulations.

2. There must be a mechanism in place for the solid to advect in the background fluid grid.

Figure 5.14 demonstrates the motivating example given in the introduction of a submerged structure being subject to a blast load. The image is reproduced courtesy of Lawrence Livermore National Laboratory and is indicative of the current state of embedded technology. The anticipation is that the quality of the solution in cases like this can be improved by improving the quality of the interface interactions.
Figure 5.13: Elastic wave propagation through an embedded mesh. Image used with permission from Lawrence Livermore National Laboratory
**Figure 5.14**: Blasting loading of an embedded versus ALE mesh. Image used with permission from Lawrence Livermore National Laboratory.
FIGURE 5.14: Blasting loading of an embedded versus ALE mesh. Image used with permission from Lawrence Livermore National Laboratory
**Figure 5.14:** Blasting loading of an embedded versus ALE mesh. Image used with permission from Lawrence Livermore National Laboratory
In this dissertation, the treatment of embedded interfaces for computational mechanics problems has been extensively discussed. Some of the key achievements of this work are summed up as follows:

1. A new formulation for the simulation of rigid body interaction with two-phase flow was proposed and benchmarked. The major idea of the method was that the use of a fully Lagrangian rigid body solver would increase accuracy and flexibility over methods that incorporate the rigid body equations into the fluid solver. In this case, it was determined that the state of the art for free surface flows has emerged from the finite difference/level set community, so the algorithm was based upon those developments. The largest advantage of the method is that it provides an intuitive and simple framework for consistently incorporating internal mechanisms into the rigid bodies. This was motivated by the need for a simulation technique for the mechanisms inside ocean wave energy harvesting devices.
2. The stability of embedded finite element meshes for structural mechanics problems was analyzed. In the case of linear elasticity, traditional methods for enforcing continuity constraints at embedded boundaries, were shown to have either stability or consistency problems. In particular, mortar methods, which are standard for treatment of gridded interfaces, can result in over-constraint and locking behavior, which causes a loss of energy convergence and unstable interfacial fluxes. Penalty methods tend to behave better than mortar methods for energy considerations, but failed to sufficiently enforce continuity constraints, resulting in loss of optimality in the displacement norms. Nitsche’s method for weak constraint was shown to be capable of restoring convergence optimality and increasing overall accuracy of the solution in these cases. The major disadvantage of Nitsche’s method is clearly the need to judiciously choose an algorithmic stabilization parameter.

3. A re-formulation of Nitsche’s method was presented, also for linear elasticity, in which a close parallel to the interior penalty methods associated with the discontinuous Galerkin method was exploited. A lifting operator was introduced such that Nitsche’s methods’ boundary integrals could be evaluated as volume integrals and rolled into the definition of element strain. An additional advantage occurs from this method in that a stability term naturally comes from the modified element strains. The method is shown to perform as well or better than a standard Nitsche’s method in all of the test problems with no additional stabilization.

4. The discontinuous Galerkin version of Nitsche’s method was applied to the case of embedded meshes for non-linear elasticity. Use of this method circumvented the need to find an expression for the stress variation - previously an outstanding question for non-linear applications on Nitsche’s method. Optimal
convergence in displacement and near-optimal energy convergence was demonstrated with comparison to conformal finite element solutions. In all cases, the discontinuous Galerkin method performed as well or better than either standard mortars or penalty methods.

Several compelling avenues of research exist for the methods described here. In particular, they were developed with the intention of incorporation into a multi-physics framework. The next most logical step would be to adapt the DG derivative method on elasto-dynamics problems. There are open questions as to how the DG derivative terms would affect the stable time step for those kinds of problems. Further anticipated applications include incorporation into a hydro-code, so as to confidently solve problems such as blast loading, and ultimately into electro-magnetic computations for problems such as the magnetic projectile. It is also anticipated that many lessons can be drawn from this work pertaining to current work in Nitsche’s method and all of its applications.


Biography

Jessica Duvall Sanders was born in Albuquerque, New Mexico on August 27th, 1983. She holds a Bachelor of Science in Mechanical Engineering from the University of New Mexico, a Master of Science in Civil and Environmental Engineering from Duke University, and a Doctor of Philosophy in Civil and Environmental Engineering from Duke University.

Jessica held the National Defense Science and Engineering Graduate Fellowship from 2006 - 2009 at Duke. In addition, she was the recipient of the Deutscher Akademischer Austausch Dienst (DAAD) Fellowship for research in Germany in fall of 2009. She has authored or co-authored several peer reviewed journal articles, including “On methods for stabilizing constraints over enriched interfaces in elasticity”, (Sanders et al., 2009), “A new method for simulating rigid body motion in incompressible two-phase flow” (Sanders et al., 2010) and “Mortar contact formulations for deformable-deformable contact: past contributions and new extensions for enriched and embedded interface formulations” (Laursen et al., 2010). She will be starting a position as a post-doctoral researcher at Lawrence Livermore National Laboratory, Livermore, California in January of 2011.