SANDIA REPORT

SAND2003-4649
Unlimited Release
Printed December/2003

Large Deformation Solid-Fluid Interaction via a Level Set Approach

Authors: David R. Noble, P. Randall Schunk, Edward D. Wilkes, Thomas A. Baer, Rekha R. Rao, and Patrick K. Notz

Editor: Rekha R. Rao

Large Deformation Solid-Fluid Interaction via
a Level Set Approach

Authors: David R. Noble,
Microscale Science and Technology

P. Randall Schunk, Edward D. Wilkes, Thomas A. Baer, Rekha R. Rao, and Patrick K. Notz
Multiphase Transport Properties

Editor: Rekha R. Rao

Sandia National Laboratories
P. O. Box 5800
Albuquerque, New Mexico 87185-0834

Abstract

Solidification and blood flow seemingly have little in common, but each involves a fluid in contact with a deformable solid. In these systems, the solid-fluid interface moves as the solid advects and deforms, often traversing the entire domain of interest. Currently, these problems cannot be simulated without innumerable expensive remeshing steps, mesh manipulations or decoupling the solid and fluid motion. Despite the wealth of progress recently made in mechanics modeling, this glaring inadequacy persists.

We propose a new technique that tracks the interface implicitly and circumvents the need for remeshing and remapping the solution onto the new mesh. The solid-fluid boundary is tracked with a level set algorithm that changes the equation type dynamically depending on the phases present. This novel approach to coupled mechanics problems promises to give accurate stresses, displacements and velocities in both phases, simultaneously.

Multiple techniques are developed for addressing this multiphase problem. First, Eulerian solid mechanics is explored, seeking to describe the deformation of bodies defined by an implicit interface (not defined by a mesh contour). Second, for problems involving moderate solid deformation but large deformation of the fluid phase, an overlapping mesh approach is developed in which a Lagrangian solid moves through an Eulerian fluid.

A challenge for any of these techniques is the accurate prescription of interfacial physics. Since the interfaces no longer conform to mesh surfaces, methods are required for imposing sharp interfacial conditions along curves cutting through elements. To address this challenge, finite element methods are developed that involve enriching elements that span an interface. Degrees of freedom are added to these interfacial elements to accommodate the discontinuities present. This is a necessary step for solid-fluid interfaces, but has much broader applications. Therefore these techniques are developed in a general way and then are applied to problems ranging from solid-fluid interactions, to phase change, and to vapor-liquid interfaces with radically differing transport coefficients. As a result of this work, Sandia can now address physical process that were intractable using previous approaches.

Acknowledgment

Preface

Introduction

The level set method [Sethian, 1999] provides a technique for describing the motion of interfaces. The method has been successfully employed to simulate a number of moving interface problems. In his book, Sethian [1999] describes the method and its application to problems including multiphase flow, combustion, and semiconductor processing. In addition, the method has been used to simulate interfacial motion in solidification [Chessa et al., 2002] and solid-fluid interactions [Arienti, 2003]. At Sandia, level set methods have been primarily employed to simulate multiphase flow problems with applications to encapsulation. The original effort for this project was aimed at extending level set-based methods for solid-fluid interactions. As described in this report, however, this work has provided methods that have significantly extended Sandia's capability to address complex interfacial physics along moving boundaries with applications far beyond the original scope of this project.

Motivation

Solid-fluid interaction problems are numerous in manufacturing processes, mechanical device performance, and biological systems. The motivation for developing a level set-based method for large deformation solid-fluid interactions is to avoid the problems associated with mesh distortion. In Arbitrary Lagrangian Eulerian (ALE) methods, the material is allowed to move relative to the computational mesh, but the phase boundaries are required to coincide with mesh surfaces [Schunk et al., 2002]. Consequently, mesh distortion is reduced compared to a purely Lagrangian approach but still can be a crippling problem when there is large relative motion between the phases.
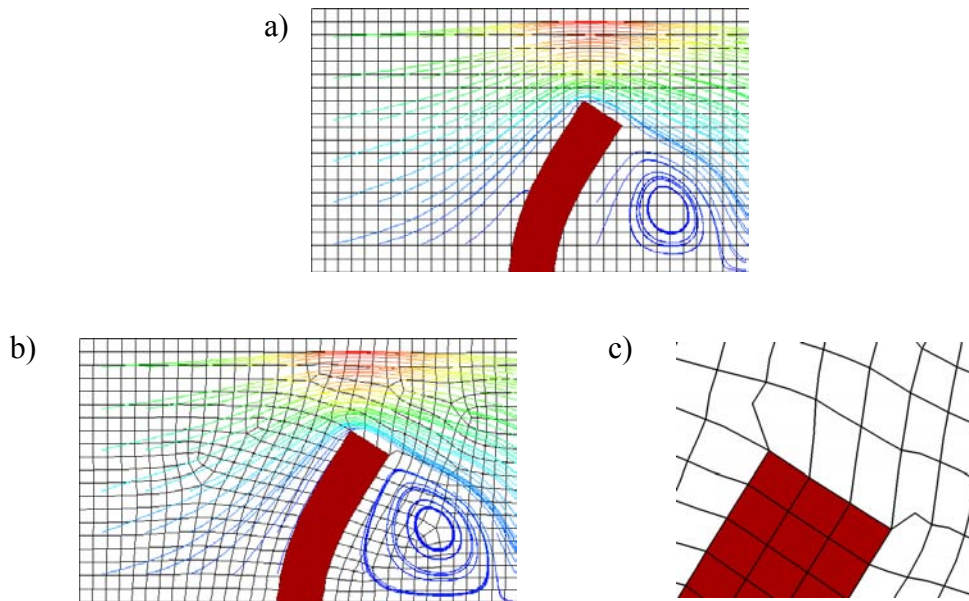
Figure P-1. Deformation of a flexible blade. a) Fully coupled overlapping mesh technique b) ALE simulation after 2 cycles of remeshing, remapping, and simulation. c) Detail of mesh around the blade tip for the ALE simulation showing the unacceptable mesh distortion.

Figure P-1 shows the steady state results for viscous flow around a deformable blade anchored to the bottom of the domain and solved using two different numerical methods. Figure P-1a gives results for the fully coupled overlapping mesh technique described in Chapters 4 and 5 and Figure P-1b gives results for the ALE solution [Schunk et al., 2002].

The mesh configuration shown in Figure P-1b follows two cycles of remeshing and remapping the solution to try to circumvent mesh distortion. Examining the mesh near the corners of the valve, Figure P-1c, reveals that the solution again needs to be remeshed and remapped in order to continue simulating the deformation. This is a costly process for the analyst and is caused by the inability of ALE techniques to address large scale relative motion of the boundaries. Even though the ALE formulation allows internal mesh motion, the motion of the boundaries away or toward one another leads to unacceptable stretching and shearing of elements.

A primary goal of this project is therefore to devise a technique in which the solid is allowed to move through an Eulerian mesh. In the methods developed in this work, the fluid mesh is purely Eulerian, even though deformable solids are moving through the fluid domain, while tight coupling is maintained at the solid-fluid interface. The Figure P-1a shows the results from this LDRD where this problem is simulated using the overlapping grid method, which requires no expensive

remeshing or remapping steps. This example is discussed further in Chapters 4 and 5.

Approach

The solid mechanics formulation has evolved as the project proceeded. As discussed below, a purely Eulerian solid mechanics approach was initially pursued. While this has certain attributes and applications that make the method interesting, the approach has both theoretical and implementation difficulties. Further work in this area is warranted. Another approach that received some attention was diffuse interface viscoelastic models that might be able to span the realms from fluid to solid mechanics. Similarly, upon further investigation, these methods did not appear to provide the most promise for practical simulations.

As an alternative, a good deal of effort was spent pursuing overlapping mesh simulations. In this method, a Lagrangian solid moves through an Eulerian fluid domain with fully coupling interactions along the intersection between the phases. For even moderate solid deformations, this method appears to provide a high accuracy approach for addressing solid-fluid interactions without remeshing. The fundamental difference between the ALE and overlapping mesh approaches is that the phase boundaries are no longer required to coincide with mesh lines of the solid phase. The solid phase is accurately described by its Lagrangian description and the fluid by its Eulerian one.

The overlapping mesh method still required some significant development of new capabilities, however. Specifically, a method is required for specifying the interfacial conditions in the fluid elements that span the solid surface. Second, a method of solving the dynamic system of equations is required. The result, however, is a robust method for simulating large deformation problems with little or no remeshing required.

In the course of this work, one unique aspect of level set methods became very apparent. Unlike the interfacial descriptions provided by volume of fluid or diffuse interface methods, the description given by a level set approach is sharp. The precise location of the interface is defined as the zero contour of the signed distance function. This distance function is defined as the shortest distance to the interface. In addition a sign convention must be employed where the distance is, for example, negative inside one of the phases and positive outside. When a level set method is employed in a finite difference or finite element code, the signed distance function describes the interfacial location with subgrid accuracy. Rather than just knowing whether a given node lies inside or outside the phase, the location of the interface between the nodes is given by the location where the distance function changes sign. This capability to precisely describe the interfacial location is unique to level set methods. Although a volume of fluid method can provide the percentage of phases present in the vicinity of a node, it cannot specifically describe the path of the interface as it passes through elements and between nodes.

While level set methods provide a method of describing sharp interfaces, it is not always obvious how to employ this information in the physics code responsible for simulating the transport processes. For example, in a solidification problem, the code used to describe the heat transfer must be enhanced to incorporate the interfacial physics along the phase boundary that is cutting

through the computational mesh. The level set method uses the interfacial velocity provided by the physics code to evolve the interface location. In diffuse interface methods, distributed source terms are employed to model the interfacial physics in an artificially wide band surrounding the interface. Because this is much simpler than enforcing the sharp interfacial physics, diffuse interface methods have been used extensively, coupled to either a level set or volume of fluid method for the interfacial description. This diffuse implementation of the interfacial physics removes a primary advantage of the level set method, however. Instead one would like to use the precise interface location provided by the level set method to enforce the interfacial physics along a sharp interface cutting between nodes of the computational mesh.

Development of sharp interface methods therefore became a primary focus of this research. While this required a significant effort not originally envisioned in this project, it provided capabilities far exceeding the original scope of this project.

Accomplishments

The accomplishments for this project can be divided into three areas. Theoretical developments have been made in the area of Eulerian solid mechanics in the context of finite element methods and a rudimentary implementation has been performed. Indicative of the change in focus for this project, much work has been done in the second area, which examines overlapping mesh techniques for solid-fluid interactions. This involved the inception, development, and implementation of a technique for tightly coupling a Lagrangian solid and Eulerian fluid represented by different meshes. Using this capability, simulations are performed that were not feasible using existing techniques. The final area of significant accomplishments is the development of a suite of mechanics for imposing sharp interfacial conditions. Previously, boundary conditions could only be applied along mesh surfaces. With this new capability boundary conditions relating to capillarity, heat transfer, kinematics, and other applied fluxes and forces can be applied along embedded interfaces in finite element simulations. As an extreme case, flow about an inviscid bubble is now possible. In this case, the viscous fluid surrounding the bubble is solved, subject to the capillary boundary condition along the boundary surface that describes the jump in pressure due to surface tension. This is one example involving completely different physics where the nature of the physical equations is dynamically changed depending on the phases present in the element. This sharp embedded physics capability promises to be instrumental in helping Sandia meet its goals to address problems like foam decomposition, laser welding, and aluminum relocation.

An additional accomplishment of this project was the development of tutorial memos on the usage of level sets [Baer, 2003] and overlapping grid methods [Schunk and Wilkes, 2003] in GOMA [Schunk et al., 2002]. These invaluable documents help disseminate the novel capabilities developed in this project as well giving training to users so that they may take advantage of the technologies developed herein.

Report Organization

This report consists of several parts that examine the development and results in these three areas of research. Chapter 1 describes the formulation for Eulerian solid mechanics developed for GO-

MA. The chapter includes discussion of the relative merits of an Eulerian solid mechanics approach and an overlapping mesh approach. The 2nd chapter examines the implementation of a 1-D prototype of an extended finite element (XFEM) method. XFEM is one of the techniques that was implemented to address the problem of imposing sharp interfacial physics along embedded interfaces. In Chapter 3 the relationship between XFEM and ghost fluid methods, which were developed in a finite difference framework for embedded discontinuities, is addressed. Chapter 4 then examines the combination of overlapping mesh techniques and XFEM for applications to multidimensional solid-fluid interactions. The relative importance of XFEM for enriching the interfacial elements is examined. The details of the overlapping mesh technique are deferred until Chapter 5. Also included are more validation problems that show that large deformation solid-fluid interactions can be accurately simulated without the difficulties caused by mesh distortion. Each chapter includes conclusions regarding the techniques developed in that chapter.

Table of Contents

1. Formulation for Eulerian Solid Mechanics in GOMA

P. Randall Schunk

1.1   Introduction

This chapter discusses several approaches/formulations we proposed--and in some cases tested--towards a capability that allows for coupled fluid-structure interaction problems to be treated in an entirely Eulerian framework. Formulations that were tested were done so with the multiphysics finite element code GOMA [Schunk et al., 2002]. Our overall goal is to reduce our reliance on moving and deforming meshes as a part of solving free and moving boundary problems. Presently we are able to solve problems in fluid mechanics with free and moving boundaries by deploying the purely Eulerian method of level set interface tracking. Despite several limitations stemming from interfacial physics resolution, this class of techniques has allowed for previously intractable free surface problems to be solved without moving meshes (we move meshes to accommodate free boundary motion with the so-called arbitrary Lagrangian/Eulerian, or ALE, mesh motion scheme). The ALE method was perfected by our research group some time ago [Sackinger et al. 1996; Cairncross et al., 2000; Baer et al., 2000]. To date, fluid/solid interaction problems can be effectively handled with ALE schemes, with the solid being treated as computational Lagrangian, but no successful formulation which allows for purely Eulerian solid mechanics coupled with purely Eulerian fluid mechanics has been advanced, to our knowledge.

Figure 1.1 diagrams the implication of the choice of reference frame on mesh and mesh motion requirements. ALE schemes require the solid and fluid phases to be meshed in a dependent, connected way, as indicated in the upper-left mesh which represents a solid ball falling through a fluid. Although this mesh is rather simple, the fluid mesh will clearly undergo major distortion if the ball moves relative to the fluid. In Chapter 5 we take a step towards our ultimate goal by allowing for independent meshes (the so-called overlapping grid method) between the two phases. Interfacial coupling can be handled in several ways, and is one of the challenges addressed throughout this effort.

We begin this chapter with a brief description of the overlapping grid approach, focusing on the portion that carries over to our Eulerian/Eulerian formulation. The details of this algorithm are discussed in Chapters 4 and 5. The rest of this chapter addresses several coupled mechanics-formulation issues that must be overcome in order to realize a completely Eulerian formulation. We conclude this chapter by proposing two approaches that are ready to be tested.

1.2   Overlapping Grid Approach

As a part of the process of building a capability to model relative motion of solids through a fixed Eulerian fluid mesh, we have been advancing an overlapping grid scheme. The central idea is to

Phases "meshed" but not connected

Overlapping grids
i.e. Chapter 5

ALE and traditional GOMA

Both phases "meshed" but connected for straight forward boundary condition application

Total domain "mesh" with other means of phase representation
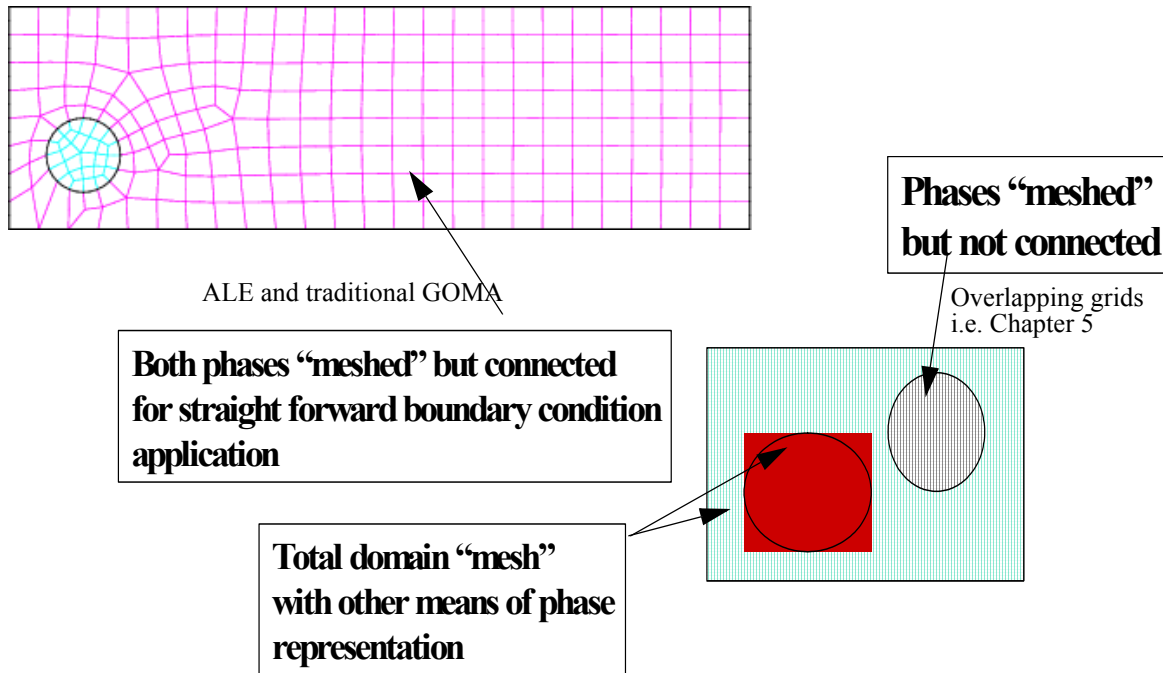
Figure 1.1 Vision of solid frame-of-reference generality. Hypothetical problem of a solid ball moving in a fluid.

treat the solid with a material-conforming Lagrangian frame of reference and the fluid in a fixed Eulerian frame of reference. Motivation for this development was primarily to deploy independent grids for each phase, thereby allowing for Lagrangian-displacement degrees-of-freedom in the solid phase and Eulerian-velocity degrees-of-freedom for the fluid phase. These variables are natural to the formulation. Details of the formulation and algorithm are presented in chapter 5 of this report.

Conserving mass and momentum present a challenge in the overlapping grid approach, as it does for a pure Eulerian/Eulerian approach. Conservation demands the accurate enforcement of the kinematic boundary condition and the surface stress conditions between the fluid and the solid. We accomplished this by 'masking' the flow in the fluid domain that underlies the solid from the rest of the flow with Lagrange multiplier constraints on the fluid-solid stress, i.e., the additional Lagrange multiplier unknowns corresponded to the stresses required to satisfy the following kinematic boundary condition:

$$\boldsymbol{v}_m = \boldsymbol{v}_f \qquad (1\text{-}1)$$

14

Here v_m is the solid velocity vector and v_f is the fluid velocity vector. The two basic classes of methods of enforcement of this condition we implemented in GOMA are:

1) Discontinuous Lagrange Multiplier equation and the "mortar-element" method, similar to the approach taken by Baaijens [2001]. This approach is basically in the class of "fictitious" domain methods.

2) Enriched finite element basis function space designed to capture discontinuities.

Chapter 5 covers the results of these application methods. A similar approach was deployed for the Eulerian mechanics formulation described next, at least with respect to the kinematic constraint.

1.3   Eulerian Mechanics Formulation

In this section we present the Eulerian Solid Mechanics formulation, which is a key building block to the overall scheme we seek. We have built up the capability of satisfying difficult stress and kinematic type boundary conditions on level set surfaces, as a part of the overlapping grid algorithm together with other developments in discontinuity capturing (cf. chapter 2 et seq.), but we also need a reliable and accurate formulation which allows for Eulerian solid mechanics and fluid mechanics solutions in regions delineated by a level set surface. Here we present the Eulerian solid mechanics formulation with a crude approach to satisfying the stress boundary condition. The combined mechanics will be solved in the same "ghosting" manner we pursued with the overlapping grid approach.

Consider the following dynamic system of equations for a transient solid mechanics problem:

$$(1-2)$$

$$\rho \frac{\partial v_m}{\partial t} + \nabla \cdot [\rho v_m (v_m - v_s)] + \nabla \cdot \sigma + f = 0$$

with the following vanishing stress condition on the boundary:

$$n \cdot \sigma = 0 \qquad\qquad (1-3)$$

Here σ is the total stress tensor of the solid. Of course, the solid tractions would not vanish at the boundary if fluid forces are present. However, our first test involves the motion of a solid in a

vacuum, driven by body forces only. Equation (1-2) is just the Cauchy momentum equation written in the frame of reference of the mesh, which is assumed to have a velocity v_s. Our goal is to take v_s=0 to achieve a completely fixed frame of reference. This equation is straightforward to solve, coupled to a continuity equation or other equation of state, provided that the boundary is well defined and the independent variable is the solid material velocity field itself. By "well-defined" boundary we mean a boundary that coincides with a mesh boundary. However, the challenge we face is to solve this equation for a material boundary moving through a fixed mesh so that outside that boundary we can solve coupled mechanics equations from different material types. Moreover, we prefer to use the material displacements from a base reference state (so-called Lagrangian variables) as dependent variables so as to avoid an incremental formulation that advances a displacement field, and hence all strain tensors, based on a derived velocity (the displacement fields and associated deformation gradient tensors are needed to calculate the stress tensor $\underset{\sim}{\sigma}$).

A description of the material boundary can be written in terms of Lagrangian invariance:

$$\frac{\partial}{\partial t}\varphi(\underset{\sim}{X}) + \underset{\sim m}{v} \cdot \nabla\varphi(\underset{\sim}{X}) = 0 \tag{1-4}$$

This equation is used to advance a level set field $\varphi(\boldsymbol{X})$ which is a signed distance function to the boundary. Note that it depends on the reference state of the solid material $\underline{\boldsymbol{X}}$, or material-point marker field.

At this point we will discuss two different approaches to calculating all required strain and stress tensors in a way which is compatible with equation (1- 4), each distinguished by the choice/ definition of the displacement field.

1.3.1 Fixed Reference State Case

In the first case we designate the independent variable of the dynamics problem as the material displacement field $\boldsymbol{d}_m$, which is defined as

$$\boldsymbol{d}_m = \boldsymbol{x}_m - \boldsymbol{X}(0) \tag{1-5}$$

Here $\boldsymbol{x}_m$ represents the deformed coordinates of the material at time t. Note that we hold the reference state $\boldsymbol{X}$ fixed in time with this definition. Specifically, we take the reference state of the solid material to be the base state defined by the level set function at the beginning of time. With this definition, the deformed coordinates of the material at time t, i.e. $\boldsymbol{x}_m$ are just the current mesh coordinates painted by the portion of the level set field that defines the solid, at least for solid-body translation. In our second formulation below we allow $\boldsymbol{X}$ to be a function of t and advance it by definition together with the displacement, viz., under solid body translation in that case, $\boldsymbol{d}_m = 0$.

16

The problem with the second approach is that the derived velocity field for the dynamics is more difficult to compute (see below). In fact, battling through this formulation it became clear why many codes use the material velocity as the independent variable and then increment displacement and relevant kinematic tensors explicitly.

In any case, the beauty of the definition (1-5) for material displacement is that the deformed coordinates of the material x_m are closely related to the current mesh reference coordinates of the material delineated by the level set field--actually they are exactly the deformed coordinates for solid-body translation. Hence, if we solve the Cauchy momentum equation (1-2) for d_m and the level set equation (1-4) for $\varphi(X)$, and hence x_m, we can nearly recover the reference state at any time. Note that a more detailed marker field within the solid is required to completely recover it.

The big challenge in both cases is to compute v_m in terms of stationary time derivatives of the only dependent vector field we have, d_m, viz. $\dfrac{\partial d_m}{\partial t}$ (recall that we have eliminated the fluid phase for the time being). This time derivative is trivial to compute on a fixed grid. This velocity field appears on the left hand side of the momentum equation and in the level set fill equation. As a side note, we use Newmark-Beta time integration schemes to evaluate the time-derivative term on the Cauchy momentum equation as well, but that is not covered here.

We start with the definition

$$-\frac{d}{dt}(x_{rs})\bigg|_X \equiv v_m \equiv \frac{dx_m}{dt}\bigg|_{X(0)} = \frac{dd_m}{dt}\bigg|_{X(0)} + \frac{d}{dt}X(0)\bigg|_{X(0)}$$

Note that x_{rs} is the grid-reference state which is the same as the material reference state at $t=0$, and so we are defining the local material velocity as the negative of the rate at which the grid points go by from an observer riding on a parcel of solid material. Equivalently, we are defining the material velocity as the rate at which the deformed material coordinates change as observed from the originating reference state. Remember that we must end up with expressions that involve gradients with respect to x_{rs} and other quantities based on the local value of d_m and its local time derivatives, because these are the only things we can compute easily. By definition the last term in the above equation is zero, and so we drop it. Based on the total differential of d_m and the chain rule:

$$\frac{dd_m}{dt}\bigg|_{X(0)} = \frac{dd_m}{dt}\bigg|_{x_{rs}} + \frac{dx_{rs}}{dt}\bigg|_{X(0)} \cdot \nabla_{rs}d_m.$$

By definition

$$\left.\frac{d\boldsymbol{x}_{rs}}{dt}\right|_{X(0)} \equiv 0$$

and so combining all of the equations above:

$$\boldsymbol{v}_m = \dot{\boldsymbol{d}}_m . \qquad (1\text{-}6)$$

Here the over-dot refers to a local time derivative.   Notice that this equation says that the Eulerian velocity field is the local-time-rate of change of the displacement field. Interestingly, there would be a correction that is related to the deformation gradient tensor had we stuck with *X(t)* being the reference state (that is discussed below in the alternate formulation).

This is the Eulerian kinematics formulation that is currently in GOMA. For our ball-drop example we solve the Cauchy momentum equation (1-2) for a rigid particle in a body-force field, for which the analytical solution will be a simple quadratic dependence of the displacement on time. We have advected a solid particle with this approach successfully, but at the time of this report we were awaiting an extension field capability in GOMA so that we can construct a smooth displacement field and hence smooth solid-velocity fields for advecting the level set with Equation (1-4). Without this capability the solid would develop unphysical stresses when passing element boundaries.

1.3.2   Time-Dependent Reference State

This formulation has the advantage that the calculated displacement field transitions more smoothly from inside the solid to outside the solid, as it is based only on mechanical deformation and not solid-body translation.   In this case we insist that

$$\boldsymbol{d}_m = \boldsymbol{x}_m - X(t) \qquad (1\text{-}7)$$

Two of these three fields are independent. We have a spatial field of reference as well, i.e. the mesh field $\boldsymbol{x}_{rs}$, but unlike the first formulation, we have no way of relating that field to these three quantities, unless one of the following is true: (a) at $t=0$ $\boldsymbol{x}_{rs}=X$, (b) under solid body translation, $\boldsymbol{x}_{rs}=X$, and (c) perhaps under linear elasticity or linearized small strain theory we could relate $\boldsymbol{x}_m$ to $\boldsymbol{x}_{rs}$ based on the current deformed level set marker field. In first formulation (cf. Section 1.3.1) we took advantage of exception (a), with the downside being that this choice creates a potentially large discontinuity at phase boundaries in the displacement field. Here we propose to add an additional equation to advance the reference state field $X(t)$.   Consider that in addition to the Cauchy momentum equation and level set field equation above we make the following changes:

- Define $\boldsymbol{d}_m = \boldsymbol{x}_m - \boldsymbol{X}(t)$ and do not use $\boldsymbol{X(0)}$. Note that the displacement field remains zero for solid body translation and rotation
- Define the independent variable for the Cauchy momentum equation to be the deformed coordinates $\boldsymbol{x}_m$. This creates some problems with boundary conditions, etc. as it deviates from solving for a displacement field, but makes for easier calculation of the inertial terms.
- Solve/advance the stress-free-state material marker field with the following equation:

$$\left.\frac{dX}{dt}\right|_X = \left.\frac{dX}{dt}\right|_{x_{rs}} + \left.\frac{d\boldsymbol{x}_{rs}}{dt}\right|_X \cdot \nabla_{rs}X = 0 \quad . \tag{1-8}$$

We will hereafter refer to this as the BIG X equation.

Notice that this expression is zero by definition, as it is the time-rate-of-change of the material marker field X as observed from a frame of reference in which that field is fixed.

- In the above expression, we note that

$$\left.\frac{d\boldsymbol{x}_{rs}}{dt}\right|_X = -v_m = -\left(\left.\frac{dX}{dt}\right|_{x_{rs}} + \left.\frac{d\boldsymbol{d}_m}{dt}\right|_{x_{rs}}\right) \tag{1-9}$$

Use this computed velocity field to advance the X field. For solid-body translation the BIG X equation above is degenerate, as $\nabla_{rs}X = \underset{\sim}{\boldsymbol{I}}$. In that case, we can simply take $x_{rs}=X$ and we do not need the BIG X equation.

We can still use the displacement field as the independent variable but then would need to make sure the proper acceleration term is used in the Cauchy momentum equation. We have implemented a BIG X equation in GOMA, but at the time this report was written it still awaits the proper velocity field. In this formulation we would simply use v_m as defined in equation (1-6). This formulation may result in much smaller discontinuities in the displacement field, but will be more expensive to run. Before completing the implementation of this formulation, we decided to complete the overlapping grid algorithm as it may allow us to solve most of our problems of immediate interest. We will return to this formulation once the aforementioned extension-field capability is implemented.

The short term prospects of running this formulation on the "ball-drop in a vacuum" test setup rest on the following outstanding issues:
- Must create a smooth d_m field using some projection scheme into neighboring elements around the solid. Currently we have to use linear elements. The volume strain tensors become too distorted and a negative Jacobian results.
- Must create a smooth velocity field using a similar scheme.
- Examine the TALE formulation [Schunk, 2000] for accuracy when mechanical deformation

is present. To create the proper strain-tensor building blocks for the stress we need *X* and *d*. Also explore solving a pseudo-problem in the fluid region for the real-solid stress field to reduce the size of the discontinuity that results from trivializing the displacement field to zero.
- Right now we satisfy the no-stress boundary condition by taking advantage of the finite element weak-form of the solid-stress equation, but do this only over element facets in the elements that contain the discontinuity. We need to contrive a scheme for satisfying these conditions over the facet representation of the zero level set.

Note that the test problem for the ball-drop is in the directory */home/prschun/fem/gomadir/m_matl/ goma_dual_mesh/tale_eulerian.tst.*

1.4   Conclusions

During the course of our research and development of this Eulerian/Eulerian fluid/solid interaction modeling capability we realized that the most significant development hurdle in all Eulerian front tracking algorithms is the need to "sharpen" quantities on the moving boundaries, the major topic of Chapters 2, 3, and 4. Our efforts described in this chapter did not result in different goals for this LDRD project, but they convinced us that there were more important research challenges that must be met before an Eulerian/Eulerian fluid-solid capability could be realized. As a result, the algorithms discussed in this chapter were implemented but never perfected to a production capability.

2. One-Dimensional Prototyping of Extended Finite Element Algorithm

Thomas A. Baer

2.1   Introduction

The extended finite element method described in a raft of recent papers [Chessa et al., 2002; Ji et al., 2002; Wagner et al., 2001; Belytschko et al., 2001] shows considerable promise in being able to add directed modifications to the behavior of the finite element interpolation functions in the vicinity of an "off-mesh" discontinuity. In our particular case we are primarily interested in the temperature gradient jumps that might occur near a moving melt front and the pressure jumps associated with a fluid phase boundary possessing surface tension. The algorithm, however, does present some difficulties in implementation so it makes sense to approach it from a standalone one-dimensional prototype in order to evaluate its usefulness in regard to both of these problems.

The extended finite element method is a p-enrichment-like adaptivity method in that it adds additional degrees of freedom to an existing mesh rather than using refinement. However, it does not do this by simply increasing the polynomial order of the interpolating functions but by actually adding functions that have the appropriate discontinuous behavior to the interpolating functions. These functions are weighted via the partition-of-unity concept [Melenk and Babuska, 1996] to ensure that they affect only elements in the vicinity of the discontinuity curve. One issue that is introduced by doing this is the accuracy of the numerical integration of the extended basis functions. This will be discussed in due course.

2.2   Thermal Experiments

Consider a domain defined by $r \in [0,1]$ with an interfacial discontinuity at $r^* = 0.5$. The latter might be a boundary between phases of differing material properties or the point of application of a heat or momentum source. We divide this into a set of odd-numbered, equally sized elements so that the discontinuity does not coincide with an element boundary. In the results shown here only five elements were used to discretize the domain.

In standard finite elements, the shape functions associated with the nodes are used to interpolate variable fields. Take temperature as an example:

$$T(r) \;=\; \sum_n T_i N_i \qquad\qquad (2\text{-}1)$$

where T is the temperature field, T_i is the nodal temperature, and N_i are the shape functions. If we wanted to use the extend finite element approach to introducing a gradient discontinuity at the discontinuity point, for example, we first define an extending function that possesses such a discontinuity:

$$g(r) = \left\{ \begin{array}{l} r - r*, r \leq r* \\ 0, r > r* \end{array} \right\} \tag{2-2}$$

Additional degrees of freedom, a_i, are introduced to include this behavior in the solution interpolation. These degrees of freedom participate in the interpolation of the solution as follows:

$$T(r) = \sum_{N_p} T_i N_i + a_i \Psi_i \tag{2-3}$$

where $\Psi_i = N_i g(r)$, introducing the partition-of-unity concept. The unknowns, a_i, are non-zero only for the elements through which the interface discontinuity passes. Thus, the effect of the extending function is confined to only these elements and the elements they share a node with.

One of the simplest problems to apply the method to is the steady heat conduction equation with a source term:

$$\nabla \bullet (k(r) \nabla T) + f \delta(r - r*) = 0 \tag{2-4}$$

where $k(r) = k^+ H(r - r*) + k^- (1 - H(r - r*))$, H is the Heaviside function, and δ is the Dirac delta function. We impose the Dirichlet conditions: $T(0) = 0$; $T(1) = 1$.

If node k is shared by an element with the interface, the discretized equations for T_k and a_k would be, respectively:

$$(\nabla N_k, k \nabla N_{k-1}) T_{k-1} + (\nabla N_k, k \nabla N_k) T_k + (\nabla N_k, k \nabla N_{k+1}) T_{k+1} \tag{2-5}$$
$$+ (\nabla N_k, k \nabla \Psi_k) a_k + (\nabla N_k, k \nabla \Psi_{k+1}) a_{k+1} = -f N_k(r*)$$

$$(\nabla \Psi_k, k \nabla N_{k-1}) T_{k-1} + (\nabla \Psi_k, k \nabla N_k) T_k + (\nabla \Psi_k, k \nabla N_{k+1}) T_{k+1} \tag{2-6}$$
$$+ (\nabla \Psi_k, k \nabla \Psi_k) a_k + (\nabla \Psi_k, k \nabla \Psi_{k+1}) a_{k+1} = -f \Psi_k(r*)$$

Of note in evaluating these equations is the need for special treatment of the integrals associated with the inner products. Both the discontinuous nature of the thermal conductivity and of the extended shape functions must be taken into account. In one dimension, this is straightforward. For higher dimensions, this presents a significant difficulty. Moreover, integration of the source term

over the domain is also simple in one dimension but more difficult in two dimensions, and more difficult still in three dimensions.

The nodes in the elements adjacent to the "interface" element, exhibit sensitivity to the extending degrees for freedom. For example, the equation for T_{k-1} would be,

$$(\nabla N_{k-1}, k\nabla N_{k-2})T_{k-2} + (\nabla N_{k-1}, k\nabla N_{k-1})T_{k-1} + (\nabla N_{k-1}, k\nabla N_{k})T_{k} \qquad (2\text{-}7)$$
$$+ (\nabla N_{k-1}, k\nabla \Psi_{k})a_{k} = 0$$

Figure 2.1 shows a comparison of the exact solution (solid lines) and the extended finite element numerical solution (symbols) for several values of k^{+}/k^{-} ranging from 1 to 100. There is no point source ($f = 0$). Interpolation of the temperature is using linear "chapeau" functions. In general, the extended FEM captures well the discontinuity in the solution at the right point ($r^* = 0.5$). However, the solution in the left side of the domain oscillates somewhat around the exact solution. This we might attribute to the fact that the extending shape function introduces quadratic interpolation of the temperature field in the element to the left of the interface element if the extending unknown is nonzero. Consequently, the temperature derivative in this element is a linear function of temperature instead of a constant. The exact solution, however, is a constant temperature gradient. Attempting to match this requirement, in a weighted residual sense, causes a temperature gradient

less than the exact value on the left side of the element rising to a temperature gradient larger than the exact value on the other side of the same element, as indicated on the figure.
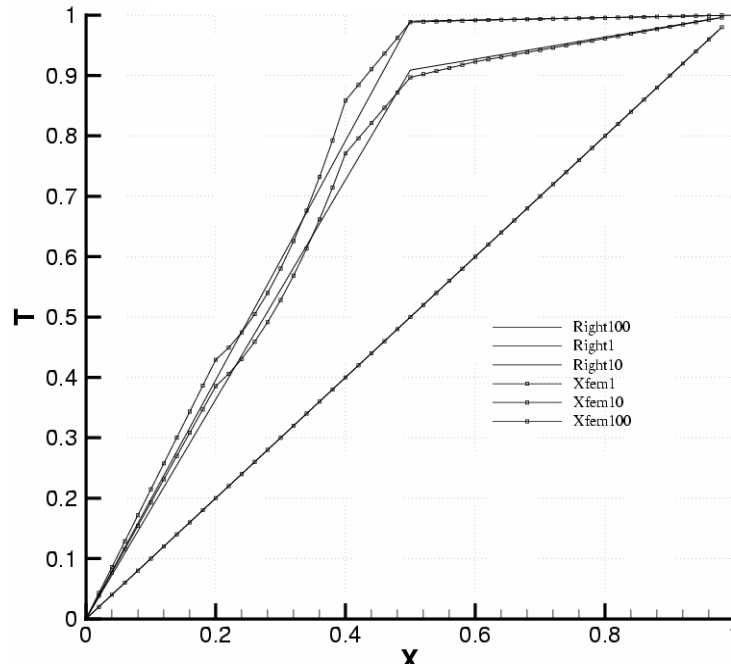
Figure 2.1: XFEM temperature field plotted with exact solution for a step change in thermal conductivity.

The response of the extending functions to differing heat sources is shown in Figure 2.2. For these computations the thermal conductivity was a constant unity and the heat source was varied from negative 10 through positive 10. The results can be seen on the figure; solid lines are the exact solution, curves with symbols are the numerical results. Interestingly enough the nodal temperature values coincide with the exact solution. Only in the interior of the elements does the temperature field differ from the exact solution. We attribute this to mismatch between derivative interpolation as discussed above. The temperature at the interface is not predicted well. Actually the temperature profile in the interface element is nearer to what one gets for a distributed source of the same strength as the point source. This suggests that the implicit smearing of the point source by the trial function weighting might have something to do with the lack of agreement with the interface temperature.
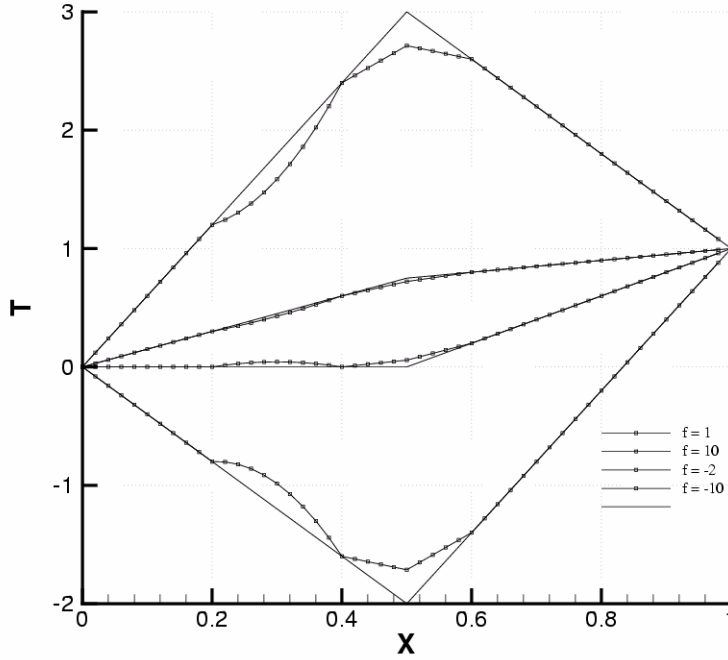
Figure 2.2: XFEM temperature field plotted with exact solution for different heat source strengths.

2.3   Pressure/Momentum Experiments

The real question that concerns us, however, is the applicability of the XFEM method to fluid problems with surface tension effects.   In particular, its applicability to a static gas bubble. To do this, a simple one-dimensional FEM prototype was constructed for solution of the static bubble problem in spherical coordinates. The governing equations for this problem are simply the weak forms of the creeping flow momentum and continuity equations:

$$0 = -\nabla P + \nabla \bullet (\mu(r)(\nabla(u + \nabla u^t))) + \frac{2\sigma}{r*}\delta(r - r*) \qquad (2\text{-}8)$$

$$\nabla \bullet u = 0$$

where u is the velocity, P is the pressure, μ is the viscosity, and σ is the surface tension. Here $r* = 0.5$ and the domain of interest was $r \in [0,1]$. The viscosity varies as a step change from 1.0 for $r >$

r^* to 10^{-3} for $r < r^*$. At the boundaries, we assign $u(0) = 0.0$ and $P(1) = 1$. The base interpolating functions for the velocity field were quadratic functions and for the pressure field, linear functions. This constitutes the standard mixed formulation for velocity/pressure solutions for solving the fully-coupled problem. The correct solution to this problem is no flow with a step change in pressure at r* of magnitude $2\sigma/r^*$.

We employed a step function as the extending function on the pressure field:

$$g(r) = \left\{ \begin{array}{l} 1, r \leq r^* \\ 0, r > r^* \end{array} \right\} \tag{2-9}$$

This function was chosen because it mimicked the form of the actual pressure change: a step from high to low pressure as r increases across the interface. The size of the step change did not seem to make any difference to the results in the extending function, since it is a linear combination.

Figure 2.3 shows the pressure solution with and without inclusion of the extending functions on the pressure field. The difference is striking. The standard Galerkin pressure response shows the oscillations that occur frequently around step changes interpolated by FEM fields. The extended pressure response is exactly the analytic solution. The velocity field for this latter case is also zero to within rounding error. The velocity field for the unextended case shows larger deviations from zero consistent with the pressure gradients associated with the oscillations. Also interesting to note is that the extended pressure field shows none of the deviation in the "off-interface" elements that was evident in the temperature experiments. Perhaps this is because the extended shape function is the same interpolation order as the standard pressure shape function and a constant multiplied by a linear function is a linear function.
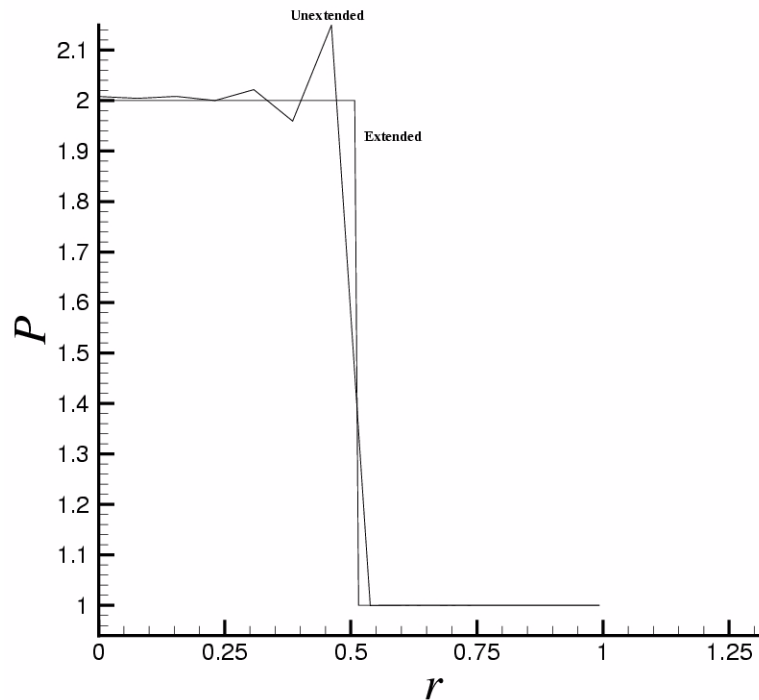
Figure 2.3: XFEM pressure field plotted with standard Galerkin FEM solution for a step change in pressure.

2.4　　Conclusions

The XFEM technique shows promise in being able to represent solution features like step changes in value or gradient which can only be approximated roughly by standard smooth FEM shape functions. This was most dramatically demonstrated by the pressure/momentum experiments which capture precisely the jump in pressure.

The technique presents some side effects as demonstrated by the temperature experiments. It introduces higher order interpolation in the elements near to the interface which may be inconsistent with the surrounding interpolating fields. A solution to this was suggested by the pressure/velocity experiments. Perhaps what is required is to choose the extending shape functions to have interpolating order no greater than the standard, unextended shape functions. For example, the temperature problem could be solved again using quadratic interpolation for the regular

temperature degrees of freedom. The extended shape functions are then formed from the product of the extending function and linear shape functions. This would be an interesting experiment.

Underlying all of these experiments is the fact that the energy or force introduced at the interface can by integrated exactly and consistently. As noted above this is next to trivial for one dimensions. For two and three dimensions, this requirement becomes less certain. How this would affect the performance of this method is uncertain, but it probably would degrade its effectiveness. We come back to the same issue: the real key to sharper interfaces is the accuracy of the integration.

3. A Hybrid Ghost Fluid – Extended Finite Element Method

David R. Noble

3.1   Background

Recent publications describe methods for embedding interfacial jumps within finite difference [Fedkiw et al., 1999] and finite element methods [Belytschko, 2001]. These methods seek to decouple the interfacial motion from the mesh. Using these methods [Belytschko, 2001], the moving interfaces found in multiphase problems have been simulated on a fixed mesh. This requires methods for embedding interfacial discontinuities within a finite difference stencil or a finite element.

At first glance it appears that the methods used by finite difference and finite element practitioners are quite different. A class of finite difference methods has been termed "ghost fluid" methods [Fedkiw et al., 1999]. Normally, a finite difference stencil for a node close to the interface would incorporate nodal values from both sides of the interface. This causes unphysical solutions, however, when a discontinuity cuts through this stencil. The discontinuity violates the finite difference assumption that the derivatives are continuous within the stencil. Ghost fluid methods resolve this by extrapolating nodal values that are consistent with each side of the interface. Instead of sampling values from both sides of the interface, values that would normally come from the other side of the interface are replaced with these consistent extrapolated values. The discontinuity is effectively removed from the finite difference operators, and the discontinuity is captured.

Concurrent with these developments, new algorithms have been proposed for embedding interfacial discontinuities in finite element methods. A class of these methods has been termed extended finite element methods (XFEM) [Belytschko, 2001]. In this approach the elements near the interface are augmented with additional degrees of freedom that can accommodate the interfacial jumps. Depending on the application, the constraint equations for these additional unknowns can be derived from the standard Galerkin method, and may involve additional penalized conditions, or may incorporate additional Lagrange multipliers. The last approach provides the additional degrees of freedom so that the normal element equations are satisfied along with the interfacial conditions.

The purpose of this memo is to describe the interrelation between ghost fluid and extended finite element methods. The implementation of these methods in a finite element code is explored. One advantage of typical ghost fluid methods is that the nodal values are modified rather than the structure of the finite difference stencil. A similar method is described for finite element methods where the element assembly is unchanged but the nodal values are manipulated. These methods are explored for the case of a single quadratic finite element applied to energy equation with an embedded jump in the temperature gradient.

3.2   Extended Finite Element Implementation for Ghost Fluids

To accommodate the interfacial discontinuities in extended finite element methods, additional degrees of freedom are added to element near the interface. The resulting expression for the temperature in an element can be written as,

$$T(x) = \sum_i N_i(x)T_i + \sum_j N_j(x)g_j(x)a_j \tag{3-1}$$

Here, the a_j are the enriching degrees of freedom. The basis function for these degrees of freedom can be considered to have two components. The first, $N_j(x)$ describes the typical variation within an element (piecewise constant, linear, or quadratic). The second portion $g_j(x)$ is a discontinuous extending function that enables the resulting temperature field to contain discontinuities in value, gradient, or both. A number of extending functions have been described for enriching the finite elements near the interface. In this memo, the extending function is given by,

$$g_j(x) = H\left[-\phi(x)\phi(x_j)\right] \tag{3-2}$$

Where $H[\phi]$ is the Heaviside function that is zero for $\phi < 0$ and unity for $\phi > 0$. To facilitate the comparison with ghost fluid methods, the enriching degrees of freedom are written as,

$$a_j = \hat{T}_j - T_j. \tag{3-3}$$

When substituted into the expression for the temperature, these give,

$$T(x) = \sum_i N_i(x)T_i + \sum_j N_j(x)H\left[-\phi(x)\phi(x_j)\right]\left(\hat{T}_j - T_j\right). \tag{3-4}$$

A final simplification can be made by assuming that the interpolating functions for the enriching degrees of freedom is the same as that for the regular degrees of freedom:

$$T(x) = \sum_i N_i(x)T_i + \sum_i N_i(x)H\left[-\phi(x)\phi(x_i)\right]\left(\hat{T}_i - T_i\right). \tag{3-5}$$

It is helpful to introduce notation to distinguish between the temperature fields that appear on the two sides of the interface. The "positive version" of the temperature field comes from evaluating equation (3-5) with $\phi > 0$. Likewise, the "negative version" of the temperature field is found using $\phi < 0$. These give,

$$T^+(x) = \sum_i N_i(x)T_i + \sum_i N_i(x)H\left[-\phi(x_i)\right]\left(\hat{T}_i - T_i\right) \tag{3-6}$$

$$T^-(x) = \sum_i N_i(x) T_i + \sum_i N_i(x) H\left[\phi(x_i)\right]\left(\hat{T}_i - T_i\right).$$ (3-7)

By considering the role of the Heaviside function these can be written as,

$$T^+(x) = \sum_{i \in \phi(x_i) > 0} N_i(x) T_i + \sum_{i \in \phi(x_i) \leq 0} N_i(x) \hat{T}_i$$ (3-8)

$$T^-(x) = \sum_{i \in \phi(x_i) \leq 0} N_i(x) T_i + \sum_{i \in \phi(x_i) > 0} N_i(x) \hat{T}_i$$ (3-9)

where the special case of $\phi(x_i) = 0$ has been avoided by including it with negative ϕ values.

Examination of these expressions reveals that the nodal $\hat{T}_i$ values are serving as ghost values. For the "positive version" of the field, which is used where $\phi(x) > 0$, the temperature is obtained from the nodal temperatures for nodes on the positive side of the interface along with "ghost" temperatures from the negative side of the interface. Another aspect of this particular extending function is that the "ghost" temperatures only appear in elements that contain discontinuities.

3.3   Problem Description

The method described in the previous section is applied to the energy equation with a single quadratic finite element with three nodes. A quadratic element is used because it reveals a number of issues not apparent in the simpler linear element. The basis functions for this element are given by,

$$N_1(x) = \frac{1}{2} x(x-1) \quad N_2(x) = 1 - x^2 \quad N_3(x) = \frac{1}{2} x(x+1)$$ (3-10)

To simplify the analysis, the physical and elemental coordinates are made to be coincident by choosing -1<x<1. Also, the location of the discontinuity is fixed at x=1/2. These simplifications are not necessary but greatly clarify the presentation. To finish the problem description, boundary conditions are needed as well as interfacial conditions. For this example problem, the temperature is set to zero at x=-1. At x=1, the temperature is set to unity. For the interfacial conditions, we require that the temperature is continuous across the interface and the gradient in the negative portion of the domain is 10 times that of the positive portion. This type of constraint is common in phase change problems or in multimaterial problems with differing conductivities.

When introducing the 3 nodal temperatures and 3 ghost values, this problem has 6 unknowns. The equations for these degrees of freedom are developed in two different ways below. First, a standard Galerkin finite element approach is taken. This reveals the effect of the extending function and interfacial conditions on the resulting system of discrete equations. Then a ghost fluid perspective is applied. These various techniques are useful in understanding the development of con-

sistent and efficient formulations of extended finite element methods. The exact solution for this problem is given by,

$$T_1 = 0 \qquad T_2 = \frac{20}{31} \qquad T_3 = 1$$

$$\hat{T}_1 = \frac{27}{31} \qquad \hat{T}_2 = \frac{29}{31} \qquad \hat{T}_3 = \frac{40}{31} \qquad\qquad (3\text{-}11)$$

This solution can be plotted as,
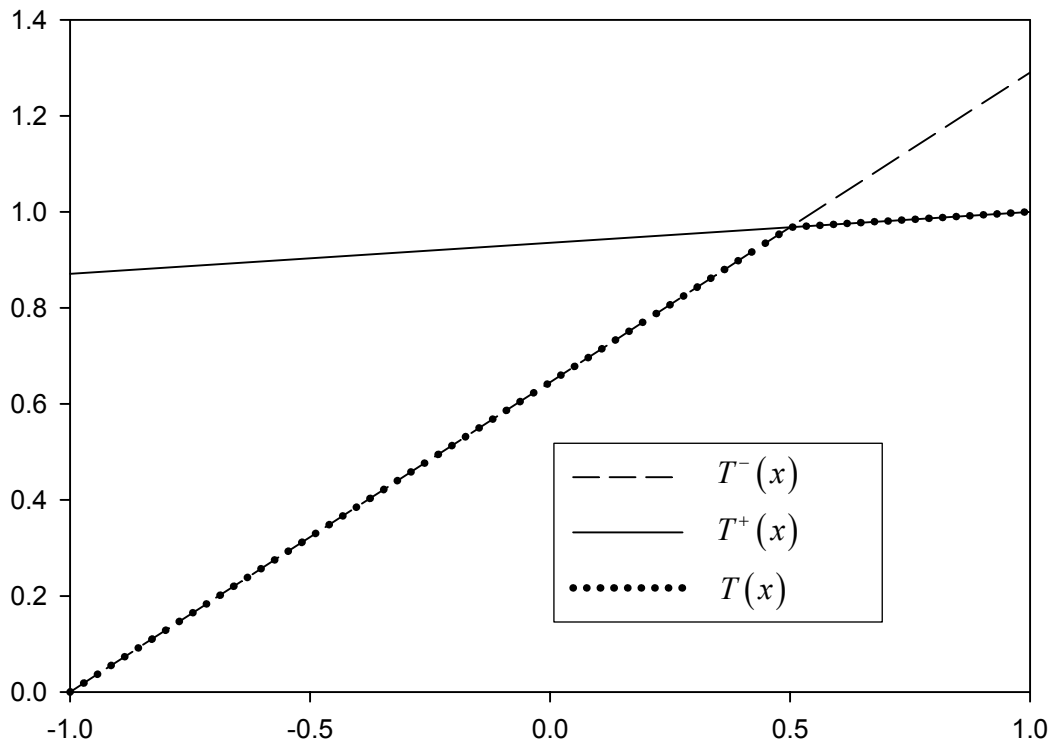
Figure 3.1: Exact solution for one-dimensional conduction problem with embedded interface

3.4   Galerkin Approach

The discrete system of equations may be formed by weighting the residual with the discontinuous basis functions:

$$\int_{\Omega} \left\{ N_i(x) H\left[\phi(x)\phi(x_i) \right] (\nabla \cdot k \nabla T) \right\} d\Omega = 0 .$$

(3-12)

This integral has a non-zero contribution only where the distance function has the same sign as the distance function at the node i. This can be expressed as,

$$\int_{\Omega \in \phi(x)\phi(x_i)>0} \left\{ N_i(\nabla \cdot k \nabla T) \right\} d\Omega = 0 .$$

(3-13)

Integrating this expression by parts results in,

$$\int_{\Omega \in \phi(x)\phi(x_i)>0} \left\{ \nabla N_i \cdot k \nabla T \right\} d\Omega - \int_{\Gamma \in \phi(x)=0} \left\{ N_i \mathbf{n} \cdot k \nabla T \right\} d\Gamma = 0 .$$

(3-14)

This is a very similar to the standard finite element equation for the nodal temperatures. It only differs in that the volume integral is over the portion of the element with the same sign as that at the node. In addition the boundary integral is evaluated over the surface where $\phi(x)=0$. Unlike typical finite element methods, this surface will not, in general, coincide with the element boundaries where the basis function is zero. It should also be noted that the outward normal vector is given by,

$$\mathbf{n} = \pm \frac{\nabla \phi}{|\nabla \phi|} .$$

(3-15)

For the negative domain the sign in this expression is positive, and for the positive domain the sign is negative. The equation for the "ghost" temperatures is derived similarly and becomes,

$$\int_{\Omega \in \phi(x)\phi(x_i)<0} \left\{ \nabla N_i \cdot k \nabla T \right\} d\Omega - \int_{\Gamma \in \phi(x)=0} \left\{ N_i \mathbf{n} \cdot k \nabla T \right\} d\Gamma = 0 .$$

(3-16)

For the 1-D problem at hand with $\phi(x)=x-1/2$, the six equations become,

$$\int_{-1}^{1/2} k^- \partial_x N_1 \partial_x T^- dx - \left(N_1 k^- \partial_x T^- \right)\Big|_{x=1/2} = 0$$

$$\int_{-1}^{1/2} k^- \partial_x N_2 \partial_x T^- dx - \left(N_2 k^- \partial_x T^- \right)\Big|_{x=1/2} = 0$$

$$\int_{1/2}^{1} k^+ \partial_x N_3 \partial_x T^+ dx + \left(N_3 k^+ \partial_x T^+ \right)\Big|_{x=1/2} = 0$$

$$\int_{1/2}^{1} k^+ \partial_x N_1 \partial_x T^+ dx + \left(N_1 k^+ \partial_x T^+ \right)\Big|_{x=1/2} = 0 \tag{3-17}$$

$$\int_{1/2}^{1} k^+ \partial_x N_2 \partial_x T^+ dx + \left(N_2 k^+ \partial_x T^+ \right)\Big|_{x=1/2} = 0$$

$$\int_{-1}^{1/2} k^- \partial_x N_3 \partial_x T^- dx - \left(N_3 k^- \partial_x T^- \right)\Big|_{x=1/2} = 0$$

The next step is to introduce the boundary conditions. At $x=-1$, the temperature is set to zero. At $x=1$, the temperature is set to unity. The obvious way to introduce these Dirichlet conditions is to replace the equations for T_1 and T_3:

$$T_1 = 0$$

$$\int_{-1}^{1/2} k^- \partial_x N_2 \partial_x T^- dx - \left(N_2 k^- \partial_x T^- \right)\Big|_{x=1/2} = 0$$

$$T_3 = 1$$

$$\int_{1/2}^{1} k^+ \partial_x N_1 \partial_x T^+ dx + \left(N_1 k^+ \partial_x T^+ \right)\Big|_{x=1/2} = 0 \tag{3-18}$$

$$\int_{1/2}^{1} k^+ \partial_x N_2 \partial_x T^+ dx + \left(N_2 k^+ \partial_x T^+ \right)\Big|_{x=1/2} = 0$$

$$\int_{-1}^{1/2} k^- \partial_x N_3 \partial_x T^- dx - \left(N_3 k^- \partial_x T^- \right)\Big|_{x=1/2} = 0$$

In this problem, the fluxes that appear in the interface integrals are part of the solution rather than applied boundary conditions. These may be found using a number of standard techniques from finite element methods. First, they could be established as additional unknowns, and additional equations could be formulated for them. This is analogous to a Lagrange multiplier implementation. If a balance is known for the gradients, like the one in the problem posed here, it is also possible to eliminate one of the gradients in terms of the other. Finally, a penalty method can be used to satisfy the interfacial condition by replacing the gradient term with a penalized term. Combining these last two techniques for the problem yields,

$$T_1 = 0$$

$$\int_{-1}^{1/2} k^- \partial_x N_2 \partial_x T^- dx - 10\beta \left(N_2 \left(T^+ - T^- \right) \right) \Big|_{x=1/2} = 0$$

$$T_3 = 1$$

$$\int_{1/2}^{1} k^+ \partial_x N_1 \partial_x T^+ dx + \beta \left(N_1 \left(T^+ - T^- \right) \right) \Big|_{x=1/2} = 0$$

(3-19)

$$\int_{1/2}^{1} k^+ \partial_x N_2 \partial_x T^+ dx + \beta \left(N_2 \left(T^+ - T^- \right) \right) \Big|_{x=1/2} = 0$$

$$\int_{-1}^{1/2} k^- \partial_x N_3 \partial_x T^- dx - 10\beta \left(N_3 \left(T^+ - T^- \right) \right) \Big|_{x=1/2} = 0$$

where β is a large penalty parameter. Solving the resulting discrete equations gives the solution,

$$T_1 = 0$$

$$T_2 = \frac{20}{31}$$

$$T_3 = 1$$

$$\hat{T_1} = \frac{27\beta + 2}{31\beta + 2}$$

(3-20)

$$\hat{T_2} = \frac{29\beta + 2}{31\beta + 2}$$

$$\hat{T_3} = \frac{40\beta}{31\beta + 2}$$

In the limit of a large penalty parameter, this recovers the exact solution.

3.5   Ghost Fluid Approach

As mentioned previously, one of the attractive features of the finite difference implementation of the ghost fluid method is that the finite difference stencil is unmodified. This raises the question of whether a finite element equivalent is feasible. One way to address this is to consider the effect of performing the integrals that result from the Galerkin approach over the entire element rather than just the portion where the discontinuous basis function is non-zero. One consequence of this alteration is that the interface integrals from the integration by parts are moved to the element boundaries. For internal degrees of freedom (i.e. node 2 of our quadratic element) the interface integrals are completely eliminated. The resulting set of equations is,

$$T_1 = 0$$

$$\int_{-1}^{1} k^- \partial_x N_2 \partial_x T^- dx = 0$$

$$T_3 = 1$$

$$\int_{-1}^{1} k^+ \partial_x N_1 \partial_x T^+ dx + \left(N_1 k^+ \partial_x T^+ \right)\Big|_{x=-1} = 0$$

(3-21)

$$\int_{-1}^{1} k^+ \partial_x N_2 \partial_x T^+ dx = 0$$

$$\int_{-1}^{1} k^{-} \partial_x N_3 \partial_x T^{-} dx - \left(N_3 k^{-} \partial_x T^{-} \right)\Big|_{x=1} = 0$$

The advantage to this technique is that the integrals over the portions of the element are replaced with integrals over the entire element. This avoids the need for adaptive quadrature or subdividing the element into subelements. On the other hand, the remaining surface integrals are not specified in terms of known quantities at the interface or even the interfacial constraints. Using a penalty formulation for these terms can get around this, however. Alternatively, Lagrange multipliers could be used to associate the unknown element fluxes with interfacial constraints. It is not possible to use the flux balance to equate these integrals as we did in the Galerkin approach since they are evaluated at locations other than the interface. Instead two separate penalty expressions are developed using the two interfacial conditions to produce,

$$T_1 = 0$$

$$\int_{-1}^{1} k^{-} \partial_x N_2 \partial_x T^{-} dx = 0$$

$$T_3 = 1$$

$$\int_{-1}^{1} k^{+} \partial_x N_1 \partial_x T^{+} dx + \beta \left(10 \partial_x T^{+} - \partial_x T^{-} \right)\Big|_{x=1/2} = 0 \tag{3-22}$$

$$\int_{-1}^{1} k^{+} \partial_x N_2 \partial_x T^{+} dx = 0$$

$$\int_{-1}^{1} k^{-} \partial_x N_3 \partial_x T^{-} dx - \beta \left(T^{+} - T^{-} \right)\Big|_{x=1/2} = 0$$

This has basically tied the flux at $x=-1$ to the flux matching term at the interface. The flux at $x=1$ is tied to the temperature matching constraint at the surface. The resulting solution is,

$$T_1 = 0$$

$$T_2 = \frac{20\beta^2 - 2\beta}{31\beta^2 - 23\beta + 2}$$

$$T_3 = 1$$

$$\hat{T}_1 = \frac{27\beta^2 - 23\beta + 2}{31\beta^2 - 23\beta + 2}$$

(3-23)

$$\hat{T}_2 = \frac{29\beta^2 - 23\beta + 2}{31\beta^2 - 23\beta + 2}$$

$$\hat{T}_3 = \frac{40\beta^2 - 4\beta}{31\beta^2 - 23\beta + 2}$$

Examination of this result reveals the exact solution is again recovered as the penalty parameter approaches infinity. Thus, this method is just as effective as the standard Galerkin method for obtaining the solution but did not require integrals over the subelements. While it is expected that the method can be applied to higher dimensions, care will be required in associating the boundary fluxes to the interface constraints. Also, it may be desirable, from a numerical standpoint, to eliminate the use of a penalty method for imposing the interfacial conditions. As mentioned above, this might be possible by introducing a Lagrange multiplier that is discretized along element surfaces and is constrained by the interfacial matching conditions. These options will be examined further in future.

4. Finite Element Simulations of Fluid-Structure Interactions Via Overlapping Meshes and Sharp Embedded Interfacial Conditions

4.1   Introduction

Fluid-structure interaction problems are common in fields ranging from biological systems to manufacturing. Heart valves, lungs, and other tissue motion along with suspension flows, brazing, and gravure coating are just a few of the systems in which coupled solid-fluid motion is a controlling factor [De Hart et al., 2003].

The simplest of these applications involves relatively small deformations of the geometry of the fluid and solid regions but still require solving the coupled fluid and solid transport equations. For rigid solids, the solid momentum equations are greatly simplified, but the task of detailed simulation remains formidable. For deformable solids, the interfacial traction and no-slip boundary conditions must be satisfied on the solid-fluid interface. In the small deformation regime, moving mesh methods have been successful [Cairncross et al., 2000]. With larger solid motion, however, the fluid domain may be significantly deformed, resulting in excessive mesh distortion. This can occur even when the solid motion is purely due to rigid body motion such as in particle flows. The situation is further complicated when the bodies are deformable. For these applications it is highly desirable to separate the phase boundaries from the computational mesh.

Recently, a number of researchers have developed methods for moving interface problems that allow the interfaces to move through the mesh. Glowinski *et al* [2001] examined suspension flows using distributed Lagrange multipliers to impose rigid motion over the domain occupied by the particles. Baaijens [2001] used overlapping meshes for the solid and fluid and coupled the motion along the interface using a mortar element method. The solid was treated with a Lagrangian description while the fluid was Eulerian. A finite-difference-based method was also developed for an Eulerian fluid and Lagrangian solid [Fedkiw, 2002].

One issue that was not specifically addressed in the finite element methods described above [Glowinski et al., 2001; Baaijens, 2001] is how to account for the discontinuities that occur at the solid-fluid interface. For moving mesh methods, this is readily handled since the mesh moves with the phases, and the discontinuities therefore coincide with mesh boundaries. When the phases move through the mesh, however, the discontinuities associated with the interface also move. Fedkiw [Fedkiw, 2002] handled this in a finite difference framework using the ghost fluid method [Fedkiw et al., 1999]. Ghost values were inserted in the finite difference stencils that spanned the interface to avoid differencing across the discontinuity. In contrast, the finite element work by Baaijens [2001] did not specifically account for the discontinuities.

A class of methods termed extended finite element methods (XFEM) have been developed to address discontinuities within an element [Belytschko, et al., 2001]. The methods have been

applied to a number of applications including solidification [Chessa et al., 2002] and fluid dynamics [Chessa and Belytschko, 2003]. These methods enrich the elements that span the interface in order to accommodate the discontinuities within these elements.

In this chapter, the ideas developed by Baaijens [2001] are combined with the XFEM approach [Belytschko, et al., 2001; Chessa et al., 2002; Chessa and Belytschko, 2003] to address fluid-structure interactions. The solid motion is Lagrangian with its mesh overlapping that of an Eulerian fluid. The effects of introducing the XFEM method in the fluid elements containing the solid-fluid interface are examined.

4.2   Approach

4.2.1   Coupled Fluid-Structure Interactions via Lagrange Multipliers

At solid-fluid interfaces, two interfacial conditions must be satisfied. The normal stresses must match, and the no-slip condition must be satisfied. Baaijens [2001] proposed that these conditions be satisfied using a Lagrange multiplier for the no-slip condition with the resulting value of the Lagrange multiplier corresponding to the interfacial traction force. The same approach is taken here, with the Lagrange multiplier implemented as a piecewise constant on the exterior faces of the solid elements. (See Chapter 5 for details of the algorithm.)

In this algorithm, the no-slip (or kinematic) condition is imposed on both phases at the interface as augmenting conditions on the main problem. This facilitates the necessary coupling of fluid and solid equations as the subset of fluid-phase elements overlapped by the solid changes in time. All nodal unknown values and interpolation functions from both phases are available for the assembly of the kinematic residuals and their sensitivities to unknowns in either phase. The new unknowns corresponding to these constraints are the Lagrange multipliers, which in this formulation represent the interfacial traction forces required to maintain no-slip. The constraints are numerically coupled to the main linear system through a bordering algorithm [Chan and Resasco, 1986], which is essentially a block elimination of the augmented equations. This coupled approach to solving the fluid mechanics and solid mechanics equations is implemented in GOMA, a multiphysics, multi-dimensional finite-element computer code developed at Sandia, described further in [Cairncross et al., 2000; Schunk et al., 2002].

4.2.2   Extended Finite Element Method for Fluid-Structure Interactions

While the velocity is continuous at a solid-fluid interface, the gradient of velocity is not. In the fluid, the force balance and no-slip conditions give rise to viscous stresses with resulting gradients in velocity. In contrast, the velocity field in the solid is a combination of rigid body motion and deformation. Typically, the velocity gradient in the solid is much smaller. This discontinuity cannot be addressed using a standard C^0 finite element discretization that requires a continuous gradient of velocity within an element. In XFEM [Belytschko, et al., 2001; Chessa et al., 2002; Chessa

and Belytschko, 2003], additional degrees of freedom are introduced in interfacial elements to capture the discontinuity.

In this study, the velocity and pressure fields are enriched. The velocity is given by,

$$v(x) = \sum_i N_i(x)v_i + \sum_i N_i(x)g_i(x)a_i^v.$$ (4-1)

Here, the a_i^v are the enriching degrees of freedom. The basis function for these degrees of freedom can be considered to have two components. The first, $N_i(x)$, describes the typical continuous variation within an element. The second portion, $g_i(x)$, is a discontinuous extending function that enables the resulting field to contain discontinuities in value, gradient, or both. A number of extending functions have been described for enriching the finite elements near the interface [Belytschko, et al., 2001; Chessa et al., 2002; Chessa and Belytschko, 2003]. Here, a different extending function is employed,

$$g_i(x) = H\left[-S\left[\phi(x_i)\right]\phi(x)\right],$$ (4-2)

where $H[\phi]$ is the Heaviside function that is zero for $\phi < 0$ and unity for $\phi > 0$, and $S[\phi]$ is the sign function that is -1 for $\phi < 0$ and $+1$ for $\phi > 0$. An additional modification compared to previous work is to write the enriching degrees of freedom as deviations from the nodal velocity:

$$a_i^v = \hat{v}_i - v_i.$$ (4-3)

When substituted into the expression for the velocity (equation (4-1)), these give,

$$v(x) = \sum_i N_i(x)v_i + \sum_i N_i(x)H\left[-S\left[\phi(x_i)\right]\phi(x)\right](\hat{v}_i - v_i).$$ (4-4)

It is helpful to introduce notation to distinguish between the velocity fields that appear on the two sides of the interface. The "positive version" of the velocity field comes from evaluating equation (4-4) with $\phi > 0$. Likewise, the "negative version" of the velocity field is found using $\phi < 0$. These give,

$$v^+(x) = \sum_i N_i(x)v_i + \sum_i N_i(x)H\left[-\phi(x_i)\right](\hat{v}_i - v_i)$$ (4-5)

$$v^-(x) = \sum_i N_i(x)v_i + \sum_i N_i(x)H\left[\phi(x_i)\right](\hat{v}_i - v_i).$$ (4-6)

By considering the role of the Heaviside function these can be written as,

$$v^+(x) = \sum_{i \in \phi(x_i) > 0} N_i(x)v_i + \sum_{i \in \phi(x_i) \leq 0} N_i(x)\hat{v}_i$$ (4-7)

41

$$v^-(x) = \sum_{i \in \phi(x_i) \leq 0} N_i(x)v_i + \sum_{i \in \phi(x_i) > 0} N_i(x)\hat{v}_i .$$

(4-8)

Examination of these expressions reveals that the nodal $\hat{v}_i$ values are serving as "ghost" values. For the "positive version" of the field, which is used where $\phi(x) > 0$, the velocity is obtained from the nodal velocities for nodes on the positive side of the interface along with "ghost" velocities from the negative side of the interface. Another aspect of this particular extending function is that the "ghost" velocities only appear in elements that contain discontinuities. Unlike other enriching functions that have been proposed previously, there are no partially enriched elements. The support for the enriching degrees of freedom is limited to the elements that contain discontinuities. It is apparent that the degrees of freedom can be separated into two sets, one for each side of the interface. The basis functions associated with each side of the interface are zero on the other side of the interface. This type of discontinuity in the basis functions is beneficial for formulating weakly integrated interfacial conditions. Just as for external boundaries, a boundary integral results from the integration-by-parts of the stress divergence term. The interfacial traction can then be imposed by including a term of the form,

$$\int_\Gamma N_i \mathbf{n} \cdot \mathbf{S} d\Gamma ,$$

(4-9)

where Γ is the surface defined by $\phi(x) = 0$ and $\mathbf{S}$ is the applied interfacial stress. Unlike typical finite element methods, this surface will not, in general, coincide with the element boundaries. In this study, the Lagrange multiplier, which enforces the no-slip constraint, supplies the interfacial traction, $\mathbf{n} \cdot \mathbf{S}$.

4.3   Results

Two problems are simulated to test the accuracy of the numerical methods. In the first test flow past a stationary cylinder is simulated. In this case, the solid is held rigid, and the interfacial stresses are determined such that the zero velocity condition is satisfied. The second test examines a fluid-structure interaction where a flexible blade is deformed by viscous flow. For both problems, the Lagrange multiplier approach is employed. The results obtained using both standard finite elements and extended finite elements are evaluated by comparison with ALE predictions.

4.3.1   Flow about a circular cylinder

Figures 4.1 shows the domain and streamlines for flow about a circular cylinder for the overlapping grid and ALE approach. The flow pressure is specified at the inlet and zero normal stress conditions are applied on the remaining boundaries. Here, the boundary conforming mesh results are expected to be accurate since standard Dirichlet conditions are applied along the cylinder surface.

42

a)

b)
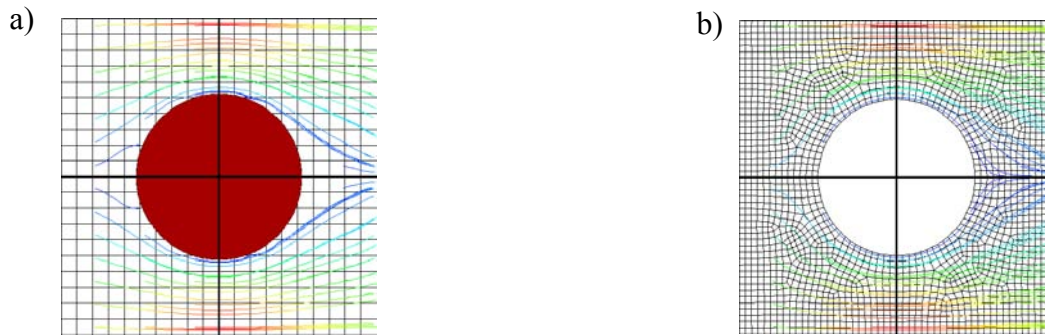
Figure 4.1: Flow about a stationary cylinder. a) Result of non-conforming mesh simulation with Lagrange multiplier to enforce the no-slip condition b) Result from simulating the flow using a boundary conforming ALE mesh. The heavy horizontal and vertical lines indicate the directions along which the streamwise velocity is plotted in the following figures.

In Figure 4.2, the streamwise velocity is plotted along the streamwise direction. Due to the large size of the cylinder compared to the computational domain, the magnitude of the streamwise velocity component is small along this line. By comparing the results using the boundary conforming mesh, it is apparent that even without extended finite elements, the Lagrange multiplier

approach produces reasonably accurate velocity fields away from the interface. Near the interface, however, it is apparent that the extended finite element method performs much better.

Figure 4.2: Streamwise velocity component along streamwise direction using extended and standard finite element methods for flow about circular cylinder.

This improved accuracy can be understood by considering the effect of the Lagrange multiplier constraint on the velocity field. Unless the field is enriched, the no-slip condition requires that the regular piecewise quadratic velocity be zero along the embedded surface of the cylinder. The finite element interpolant is unable to accommodate this discontinuity in the velocity gradient. This results in unphysical oscillations near the surface. In addition, the velocity inside the cylinder is non-zero. The extended finite element method, on the other hand, is able to accommodate this discontinuous gradient. The no-slip boundary condition is handled very well, without introducing unphysical oscillations in the solution. The velocity within the body is identically zero.

In Figure 4.3, the streamwise velocity is plotted along the transverse direction. The streamwise velocity component is much larger along this line. Consequently, the errors associated with the boundary condition are relatively small and the effect of the XFEM is not as pronounced.
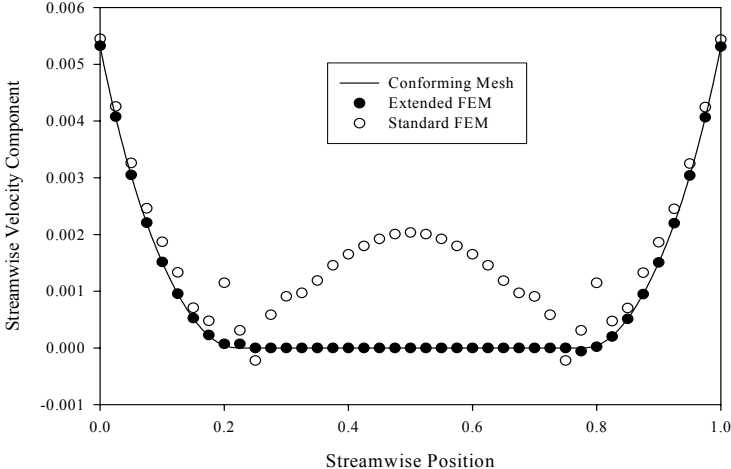
44
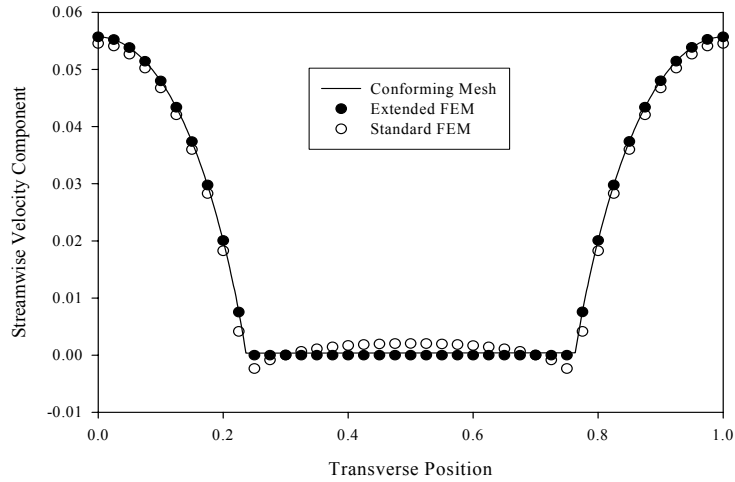
Figure 4.3: Streamwise velocity component along direction normal to flow using extended and standard finite element methods for flow about circular cylinder.

4.3.2 Flow over a deformable blade

To assess the accuracy of the combined Lagrange multiplier/XFEM approach for fluid-structure interactions, the deformation of a flexible blade is examined. Figure 4.3 shows the blade as it deforms due to the flow, which is driven by a specified inlet pressure. The remaining boundaries have zero normal stress conditions. The deformed blade is shown at three times leading up to its steady state deformation. The final deformation agrees very closely with the steady solution for the boundary conforming mesh, which is shown with its mesh.

45

Figure 4.4: Deformation of flexible blade by viscous flow from left to right. The steady solution for boundary conforming simulation is shown in black with its mesh. The extended finite element method predictions for the blade location are shown at three times with gray outlines. The solution at the final time corresponds very closely with the boundary conforming solution.

Figure 4.5 shows the transient deformation of the blade using the standard and extended finite element methods. The displacement of the forward edge of the blade is plotted as it approaches the steady deformation. The steady solution obtained using the boundary conforming solution is also shown. It is apparent that the method performs well, even without the XFEM. There is good agreement between the standard element and XFEM solutions during the transient, and both solutions agree closely with the boundary conforming solution at steady state. In [Baaijens, 2001] it was noted that moving the solid with the Lagrange multiplier on the solid surface requires accelerating the fluid under the solid. For this problem, the density ratio between solid and fluid for the blade deformation is 1000, so this effect is not apparent here. It is expected that this may be important when the solid density is small or comparable to that of the liquid. However, the extended finite element approach does not accelerate the fluid under the solid, circumventing this issue.
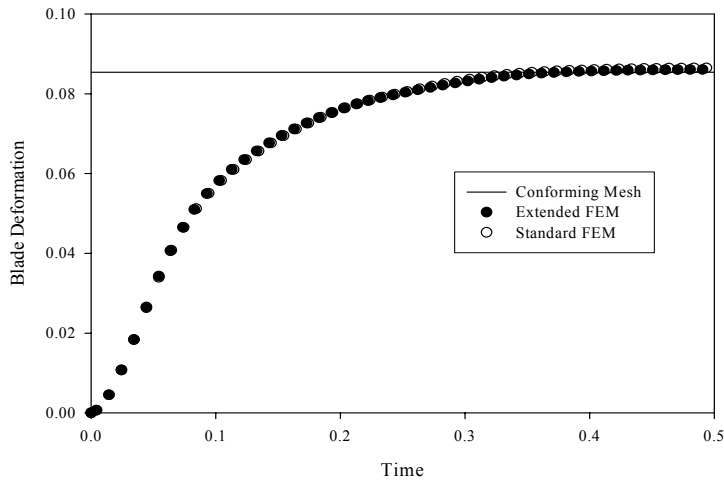
Figure 4.5: Deformation of a blade under viscous flow using extended and standard finite element methods.

Figure 4.6 shows the streamwise velocity component plotted along the streamwise direction at y=0 through the center of the blade. Similar to the result for the stationary cylinder, the simulation without using extended finite elements shows some wiggles in the solution and non-zero flow through the blade. There appears to be a small but significant error in the velocity in the wake region. It is interesting that this type of error did not significantly impact the prediction of the deformation. The fact that the errors occur in the wake region suggests that capturing the discontinuous velocity gradient may be more important for higher Reynolds number flows. Applications that involve solids in near contact, like suspension flows, may also be particularly sensitive to any errors near boundaries or in regions of recirculating flow. The extended finite element method may provide a more accurate approach for these flows.
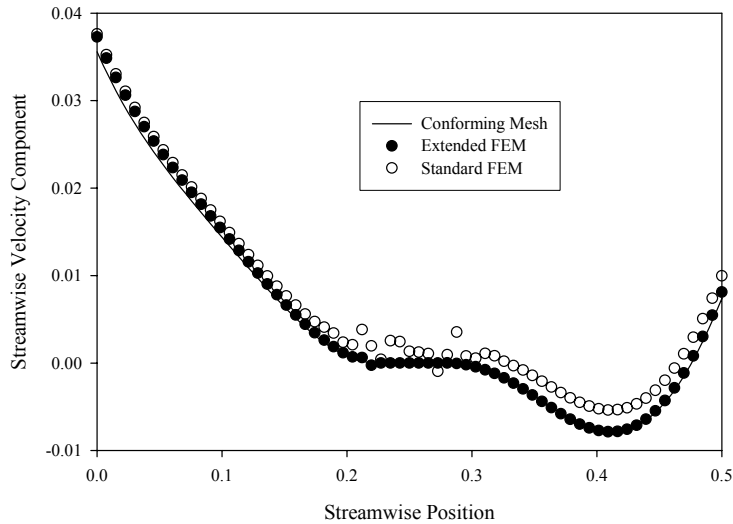
Figure 4.6: Streamwise velocity component along streamwise direction using extended and standard finite element methods for flow about deformable blade.

4.4   Conclusions

A combined Lagrange multiplier, XFEM method is developed for addressing fluid-structure interactions. The Eulerian fluid and Lagrangian solid are coupled using a Lagrange multiplier that imposes the no-slip condition, yielding the interfacial traction. A new extending function is proposed which limits the support of the extending degrees of freedom to elements that contain discontinuities. When the resulting residual equations are integrated by parts, a natural boundary condition appears allowing the interfacial traction to be specified along the interface cutting through the elements.

The method is applied to flow about rigid and deformable bodies. The flows are also simulated using the Lagrange multiplier approach without enriching the finite elements. The results are compared with solutions from boundary conforming simulations. While the extended finite element method is found to significantly improve the accuracy of the solution in the vicinity of the solid-liquid interface, both methods are shown to provide accurate predictions of the solid deformation and the fluid velocity away from the interface. Further study is required for solids with density below that of the fluid or for multiple interacting solids like that found in suspension flows.

5. An Overlapping Grid Algorithm for Finite Element Solution of Solid-Fluid Interaction Problems

Edward D. Wilkes, David R. Noble, P. Randall Schunk, Rekha R. Rao, and Thomas A. Baer

5.1   Introduction

There are several types of physical processes that involve motion of both solids and fluids; some examples are fluidized-bed chemical reactors, coating flows, and particle settling in fluid containers. Such processes may consist of a solid body which is set into motion by a surrounding fluid (e.g. fluidization) or a fluid which is forced to flow in response to the motion of a solid body (e.g. a pump impeller). Moreover, the solid object may be either rigid or deformable. In any of these cases, the physics of the two phases are strongly coupled, and the extent of relative motion between the phases is typically large.

It is highly desirable to be able to model such solid-fluid processes with powerful and robust numerical algorithms such as the finite element method. However, the nature of these problems present some challenges. One such challenge is to maintain acceptable element quality during mesh motion when the solid-fluid interface displacement becomes large relative to the size of the computational domain. This is illustrated in Figure 5.1 by a problem that involves a solid ball falling downward through a column of fluid; here, the meshes for both phases are prescribed to undergo arbitrary Lagrangian-Eulerian (ALE) motion such that both domains conform to the motion of the interface [Cairncross et al., 2000]. This widely-used method simplifies the task of tracking the transient interface location and the specification of interaction boundary conditions (e.g. continuity of velocity and stress) and works well for small relative motion between phases, but will ultimately fail as shown when the degree of fluid domain deformation reaches a certain point. Continuation of the transient solution beyond this point would typically require frequent (and often undesirable) remeshing and solution remapping steps.
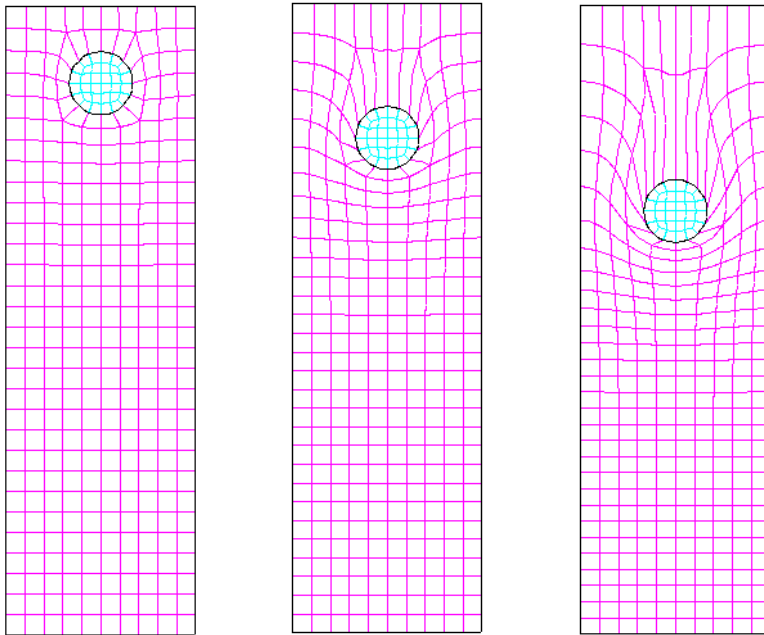
Figure 5.1: Evolution of conforming ALE meshes for a falling ball.

The approach taken in this work is to construct two independent but overlapping meshes, thereby requiring that only the solid mesh conform to the moving boundary. Doing so allows the solid-phase variables (e.g. solid displacement), which are typically Lagrangian, to be solved on a mesh with Lagrangian mesh motion, while the fluid variables (e.g. velocity, pressure) may be solved on an Eulerian mesh. The fluid mesh can then be fixed; in lieu of solving equations for mesh motion, a level set interface is used to track the location of the solid boundary from within the fluid. This interface provides a locus within a fluid element along which liquid-side contributions to the interaction conditions are to be imposed, while the corresponding solid-side contributions are simply integrated along the element sides which comprise the moving solid boundary. This approach is similar to the fictitious domain/mortar element method of Baaijens [Baaijens, 2001].

As suggested above, the key to this method is the ability to impose continuity-type constraints which require contributions from the primitive variables in both phases. An additional difficulty arises when sparse matrix formats e.g. MSR, VBR [Schunk et al., 2002] are used to store the non-zero entries of the Jacobian matrix. These formats are based on pre-determining the neighbors of each element to decide which equations (matrix rows) depend on which variables (matrix columns), then allocating a single linear array to hold only these nonzero entries and a set of auxiliary arrays to match positions in this array to row and column indices in the full matrix. In this

method, however, the dependencies of equations on one mesh to variables on the other mesh (referred to here as cross-mesh sensitivities) result in a dependency mapping which is not known a priori and changes as the relative motion progresses, and therefore cannot be incorporated into such a sparse matrix allocation in any practical way. This is a particular concern for the full-Newton finite element method, whose convergence relies heavily on having a complete set of these sensitivities.

In the overlapping grid algorithm we propose here, the interfacial constraints are imposed as augmenting conditions, which are assembled separately from the remainder of the equations but coupled to them by means of a bordering algorithm. This effectively adds an equation to the overall system, which then requires a corresponding unknown to maintain closure. Hence, the Lagrange multiplier is treated as an additional primitive variable on one of the two phases, as will be explained in section 5.2. This Lagrange multiplier treatment is consistent with the equation system in Baaijens' [Baaijens, 2001] paper, although no mention is made of how his interaction constraints were handled or how (if at all) cross-mesh sensitivities were included.

5.2   Numerical Algorithm

As indicated in the previous section, the class of problems considered here consist of two phases which are coupled at a common interface, but otherwise would act independently. Although the focus of this work is on fluid-structure interactions, the coupled phases could also both be immiscible fluids, as in the case of an emulsion.

5.2.1   Governing Equations

The governing equations, written here in the weak Galerkin finite element residual form, include a fluid momentum balance:

$$\int_V \phi_v^i \left[\rho_f \frac{D\underline{v}}{Dt} + \nabla \bullet \tau + \rho_f \underline{F} \right] dV - \int_\Gamma \phi_k^i \underline{\gamma} d\Gamma = 0 \tag{5-1}$$

and mass balance:

$$\int_V \phi_c^i [\nabla \bullet \underline{v}] dV = 0 \tag{5-2}$$

a solid side momentum balance:

$$\int_S \phi_x^i [\rho_s \underline{\ddot{x}} + \nabla \bullet \sigma + \rho_s \underline{F}] dV + \int_\Gamma \phi_k^i \underline{\gamma} d\Gamma = 0 \tag{5-3}$$

The kinematic constraint at the solid-fluid interface is:

$$\int_\Gamma \phi_\gamma^{\ i} [\dot{\underline{x}} - \underline{v}] d\Gamma = 0 \qquad\qquad (5\text{-}4)$$

and the level set function is evaluated at each fluid mesh node by:

$$f = \Theta \lVert \underline{x}_i - \underline{x}_s \rVert \qquad\qquad (5\text{-}5)$$

In equations (5-1)-(5-5), ρ_f and ρ_s are the fluid and solid material densities, respectively, $\underline{v}$ is the fluid velocity, $\underline{F}$ represents any body forces such as gravity, τ is the fluid stress tensor, γ is the Lagrange multiplier vector unknown, $\underline{x}$ is the solid displacement vector unknown, σ is the solid stress tensor, f is the level set unknown, Θ is a step function which is -1 for points within the region occupied by the solid and +1 outside this region, $\underline{x}_i$ and $\underline{x}_s$ are the position vectors of a fluid node and of the closest point to it on the solid boundary, respectively, V is the fluid volume domain, S is the solid volume domain, Γ is the solid boundary (interface) surface domain, and the ϕ^i are finite element weight functions assigned to the equation residuals as indicated by their sub-scripts.

Boundary conditions at locations other than on the solid boundary (interface) will depend on the specific problem being solved; these may include no-slip and no-penetration conditions at walls, fully-developed inflow profiles, axisymmetry conditions, etc. Interfacial conditions are discussed in the following subsection.

5.2.2   Lagrange Multiplier Constraints

In this formulation, the solid is treated as Lagrangian and the fluid as Eulerian. As previously indicated, it is desirable to first assemble the independent equation systems for each individual phase, then augment both with the necessary interfacial constraints. Following the formulation of Baaijens [Baaijens, 2001], equation (5-4) is applied as a Lagrange multiplier constraint on the fluid and solid velocities, such that they are required to be equal on the interface Γ. The new unknown which is introduced along with this constraint is then the Lagrange multiplier vector γ.

The physical problem also requires a balance of stress across the interface. This is implemented here by augmenting the momentum equation in each phase with a surface term which represents the change in total stress due to interaction with the other phase; these are the surface integral terms in equations (5-1) and (5-3) in variational weak form [Morse and Feshbach, 1953]. In sum-mary, the kinematic constraint, equation (5-4) is used to solve for the interfacial stress correction value which results in continuity of phase velocity. Thus, the physical meaning of γ is clear along the solid boundary. Although γ is defined throughout one of the two domains, it is not significant away from the interface. This Lagrange multiplier treatment allows for a complete, fully coupled problem statement; details of the solution method are discussed in the following section.

5.3 Computational Method

The algorithm described in section 5.2 was implemented in GOMA, a full-Newton finite element program developed at Sandia National Laboratories which contains the necessary equations, boundary conditions, transient nonlinear solver, and bordering algorithm necessary for this class of problems.

5.3.1 Equation Assembly

This algorithm involves assembly of the primary equation terms separately from the augmenting condition terms. Equations (5-1), (5-2), (5-3), and (5-5) are assembled on their respective volume domains in residual form:

$$\underline{R} = [\underline{R}_m, \underline{R}_c, \underline{R}_x, \underline{R}_f] = \underline{0} \tag{5-6}$$

Equation (5-6) lacks a Lagrange multiplier residual because its terms are handled in the augmenting conditions. Any boundary conditions other than those at the solid-fluid interface are then applied to these residuals. As each residual term is assembled, the unknowns upon which is depends are noted, and corresponding sensitivity terms for unknowns in the same phase are assembled to construct the Jacobian matrix:

$$[J] = \frac{d\underline{R}}{d\underline{x}} \tag{5-7}$$

However, sensitivities of the surface integral terms of Equations (5-1) and (5-3) are also deferred to the augmenting conditions.

These are the steps which are normally used for simple Newton's method applications, and could be used to solve a problem with two uncoupled, non-interacting phases by iteratively solving:

$$[J](-\Delta\underline{x}) = \underline{R} \tag{5-8}$$

5.3.2 Augmenting Condition Assembly

Terms which may involve unknowns from both phases are more convenient to include in augmenting conditions. To do this, Equation (5-4) is assembled as an augmenting constraint which is discretized on each element on or overlapping the solid-fluid interface. The solid-side contributions are evaluated along the element side(s) which coincide with the interface. In the fluid elements, the level set function f is used to determine if the interface passes through the element: if f changes sign anywhere on the element, then it contains part of the interface. In this case, the nodal values of f are used to construct the locus of the interface within the element, and the fluid-side term contributions are evaluated along this locus.

The overlapping grid problem consists of N unknowns (all variables including the Lagrange multipliers discretized over both domains) and M augmenting constraints. As indicated above, the main residual is a vector of length N and the Jacobian is an NxN matrix. The augmenting condition residuals are loaded into vector $\underline{G}$ of length M. The sensitivities of these residuals to all variables on both meshes are computed and loaded into submatrix $\underline{C}$ of dimension MxN. Finally, the sensitivities of the surface integral terms in Equations (5-1) and (5-3) to the Lagrange multiplier unknowns are computed and loaded into submatrix $\underline{\underline{B}}$ of dimension NxM. This yields the following augmented system:

$$\begin{bmatrix} J & B \\ C & D \end{bmatrix} \begin{bmatrix} -\Delta x \\ -\Delta y \end{bmatrix} = \begin{bmatrix} R \\ G \end{bmatrix} \tag{5-9}$$

where Δy is the vector of updates to the Lagrange multiplier unknowns. Submatrix D is not relevant to this algorithm, so it is loaded with zeros.

5.3.3   Augmented system solution

The augmented system, equation (5-9), is then solved for Δx and Δy at each Newton iteration by means of a block-elimination bordering algorithm similar to that of Chan and Resasco [Chan and Resasco, 1986], which is incorporated into the GOMA nonlinear solver. One Newton iteration then consists of the initial NxN solve of $\underline{J}$ (equation (5-8)), M resolves of $\underline{J}$ using right-hand sides taken from the columns of $\underline{B}$, one solve of the MxM Schur complement $\underline{S}$, and updates to all unknowns. This is done within the framework of a predictor-corrector strategy: an initial solution guess is predicted, usually from the previous time step, then Newton iterations are perform to correct the current solution until the residual norm and/or solution update norm fall below specified tolerances.

Because the algorithm performs M resolves per Newton iteration, the primary concern in the selection of a linear solver is the ability to perform fast resolves of a given matrix after the initial solve. Experience indicates that direct sparse-matrix solvers such as UMFPACK [Davis and Duff, 1997] have the best resolve performance, in which case time taken for the resolves and other steps of the bordering algorithm is typically on the same order as the initial solve for $M \leq 50$. Iterative linear solver algorithms such as GMRES take considerably longer to perform the resolves. This algorithm appears to exhibit superlinear $(\alpha \sim 1.5)$ convergence of the nonlinear solver, similar to that reported by Baaijens for his algorithm [Baaijens, 2001].

5.4   Results and Discussion

The results presented here illustrate the use of the overlapping grid algorithm for various solid-fluid interaction configurations. In each example, a fixed Eulerian mesh (relative to the fluid) is used for the fluid domain, while a dynamic Lagrangian mesh, equation (5-3), is used for the solid

domain. In each case, a compressible neo-Hookean solid moves through an incompressible Newtonian fluid.

5.4.1 Spherical Ball Falling Through Fluid Due to Gravity

A very simple example of a solid-fluid interaction is the settling of a sphere under gravity. Here, the fluid is contained in a vertical cylinder with a radius of $4R_s$, where R_s is the sphere radius. The cylinder has a closed bottom. The solid material density is 10 times that of the fluid.

Figure 5.2 shows fluid streamlines for the fluid moving around the sphere, which illustrate the recirculation regions which form on the sides of the sphere as the fluid flows with a net upward motion; this recirculation region travels downward with the sphere. Figure 5.2 also shows contours of the shear component of the solid stress (σ_{rz}), which prove that the algorithm allows the stress due to fluid motion to be transmitted to the solid, which undergoes a small degree of deformation in the process.
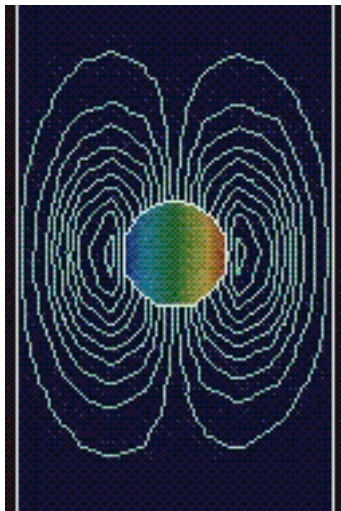

Figure 5.2: Streamlines and solid shear stress for ball dropping in a cylinder of radius $4R_s$.

5.4.2 Spherical Ball Validation

To validate the overlapping grid algorithm, we solved the falling ball problem with and without the overlapping grid algorithm, for one set of parameters. Specifically, the properties and parameters correspond to fluid water, the cylinder radius is 5 cm and the ball radius is 0.05 cm such that the radius ratio is 100; all other conditions are as above. The Reynolds number for this case is

about 5.0e-5, or low enough to approximate creeping flow, such that an estimate of the sphere's terminal velocity can be obtained for both algorithms before element distortion becomes significant.

The comparison case employs moving ALE meshes for both meshes, as shown in Figure 5.1. Therefore, a mesh equation similar to equation (5-3) is also needed for the fluid. In this case, the interfacial constraints are applied directly on the boundary shared by the meshes, so the Lagrange multiplier and level set equations are not necessary.
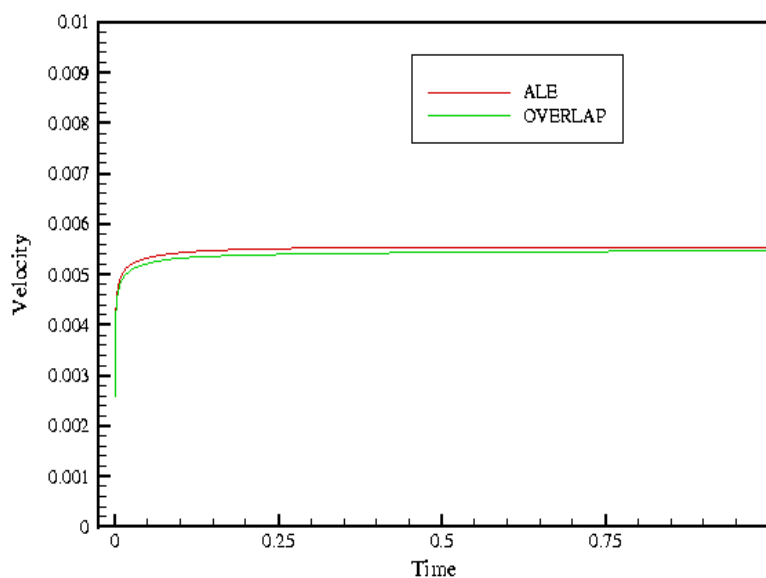

Figure 5.3: Velocity profiles for falling spherical ball using fixed and conforming fluid meshes.

Figure 5.3 shows the evolution in time of the sphere settling velocity for the two mesh configurations. Although there is a slight difference in the transient velocities approaching the terminal velocity, the velocity reached was 5.534e-3 mm/s for the ALE case and 5.475 mm/s for the overlapping grid case, or about 1% difference between the cases. This shows that we are able to solve the problem and arrive at a similar solution for each case. In the creeping-flow regime, the correlations of Bird *et al.* [Bird et al., 1960] predict a terminal velocity of 5.0e-3 mm/s for our Reynolds number, which is within about 10% of the computed value by either method. In future, mesh refinement should reduce this discrepancy. No significant difference in the transient flow evolution was observed between the two cases.

5.4.3   Solid Blade Moving Through a Uniform Fluid Channel

Attention is now turned to solid-fluid interaction problems where the deformation of the solid is significant. Here, a rectangular blade of a flexible solid moves along the bottom of a rectangular channel of fluid. The blade height is 63% of the channel height and its width is 1/3 of its height. The channel height is 12% of its width. The Young's modulus of the solid is 2.5e+5 and its Poisson's ratio is 2.5e+4. The solid density is 10 times the fluid density. Two flow configurations are considered for this case: in the first a fluid pressure gradient will be used to initiate blade motion, and in the second the blade is pulled through the fluid from the bottom, which drives fluid motion.
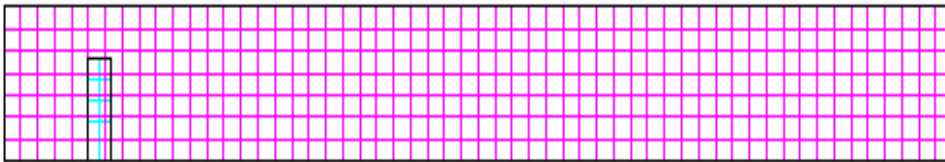

Figure 5.4: Typical overlapping grid for flexible solid blade in a rectangular channel.

The two-phase mesh shown in Figure 5.4 is used for both configurations, which will differ primarily by the imposed boundary conditions.

In the first configuration, a steady simple shear flow is imposed on the fluid by prescribing a fixed velocity at the top of the channel, which pushes the blade through the channel while its bottom is free to slide along the bottom channel surface. Here, the horizontal (x) velocity at the ends of the channel vary linearly in the vertical (y) direction. In this case, the top of the blade moves faster than the bottom because the impingement momentum is higher there, resulting in a small amount of forward bending deformation of the blade. The resulting distortion of the shear flow is illustrated in Figure 5.5. As shown, the primary effect of the presence of the blade is to reduce the vertical velocity gradient in its vicinity; once the blade passes a given horizontal location, the gradient is restored.
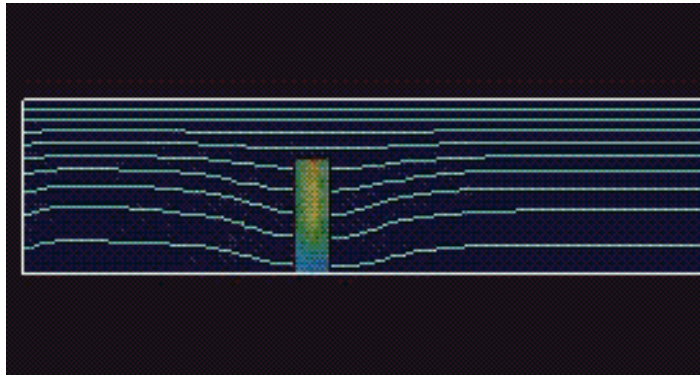
Figure 5.5: Blade pushed by shear flow. Fluid streamlines and solid shear stress shown.

In the second configuration, the bottom of the blade is moved along the channel bottom at a fixed velocity. Here, the ends of the channel are closed, and zero velocity is imposed on the top and ends of the channel. As shown in Figure 5.6, this induces a net recirculation flow throughout the channel. The streamlines show that the top of the blade acts as a source of vorticity which is carried along with the blade as it moves. The top of the blade has to exert considerable force on the fluid to the right of it to push it along, so it is not able to move forward at the same velocity as the bottom of the blade, which results in a large degree of backward bending deformation. This deformation is reflected in the shear stress contour in Figure 5.6.
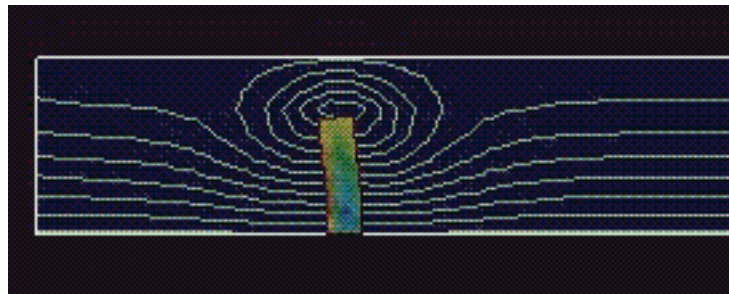

Figure 5.6: Blade dragged from the bottom. Fluid streamlines and solid shear stress shown.

5.4.4　　　　　Solid Blade Moving Through a Channel with a Gravure-Type Top Surface

For this case, the flat top of the channel is replaced with a grooved top, as illustrated in Figure 5.7. This more closely simulates a typical blade coating process, in which solid-fluid interactions are more complex and therefore more important to understand. Such configurations clearly indicate the necessity of this method. Whereas the problems with a straight channel (Figures 5.4-5.6) could also have been formulated in the frame of reference of the blade (with a fixed fluid mesh), the introduction of geometric irregularities, such as that shown in Figure 5.7, preclude the ability to set the moving solid and container stationary relative to each other in any frame of reference.
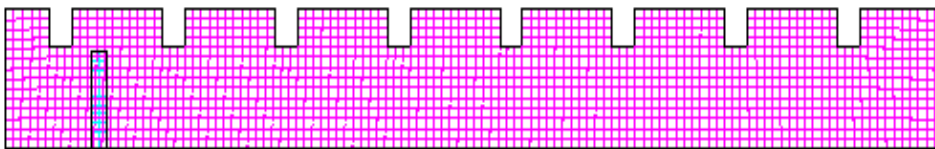

Figure 5.7: Overlapping grid for flexible solid blade in a grooved rectangular channel.
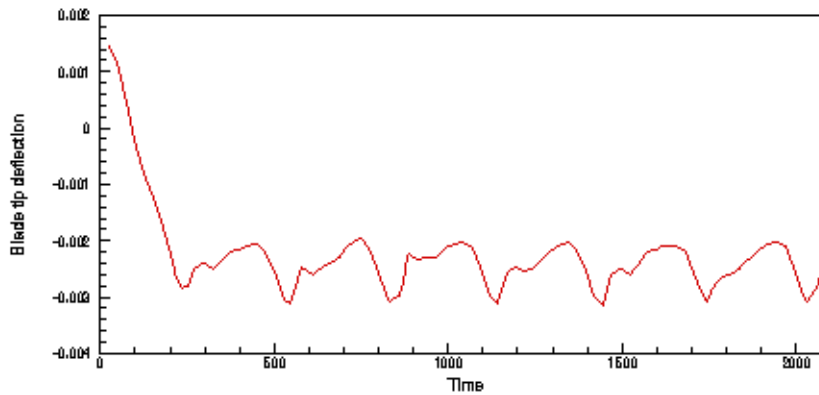

Figure 5.8: Deflection of the blade tip as it moves through the channel.

For this case, the geometry of the top of the channel now includes a series of regularly-spaced grooves, such that the gap between the grooves and the undeformed blade is reduced to 5% of the channel height. Otherwise, all conditions are the same as in the previous case. Here, the blade passes alternately along grooves and spaces between grooves as it moves through the channel. Thus, the force exerted on the "tip" (top right corner) of the blade varies in a sinusoidal pattern, as shown by the tip deflection profile in Figure 5.8.
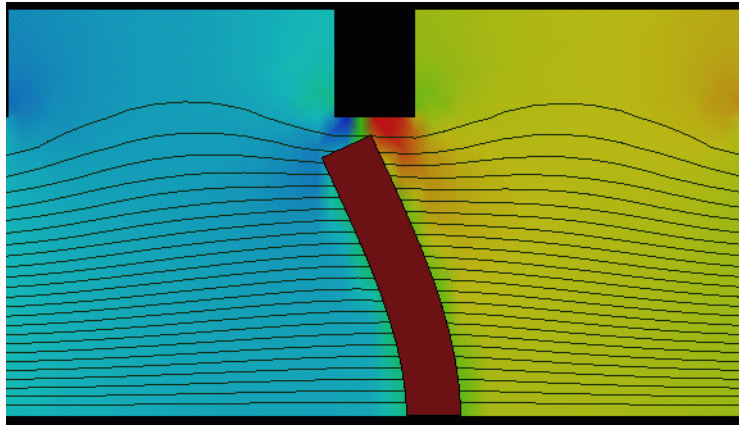
Figure 5.9: Streamlines and pressure contours as blade moves along a groove.

As shown in Figure 5.9, as the blade passes by a groove, the fluid gap height above the blade is significantly reduced. Thus, more force is required to push fluid backward in this narrow gap, as indicated by the sharp pressure gradient in this region. Also, the degree of recirculation around the tip is considerably smaller at this instant. As the blade passes between grooves (Figure 5.10), the fluid flow more closely resembles the flat-top case discussed in the previous subsection, but with some distortion along the top due to its irregular geometry.
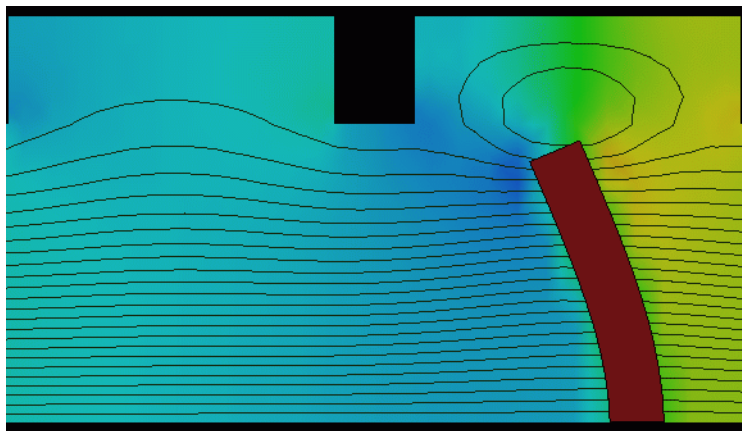

Figure 5.10: Streamlines and pressure contours as blade moves between grooves.

5.5 Conclusions

This work has demonstrated an algorithm which accommodates finite element solution of fluid-structure interaction problems by allowing each phase to have an independent mesh suited to the variables which are to be defined on it. Full coupling of the phases is achieved through a Lagrange multiplier formulation for the interfacial interaction terms. All necessary Jacobian sensitivities, including cross-mesh terms, are included through the use of augmenting conditions. The algorithm was validated against both theory and a fully-conforming mesh result using the classical falling sphere problem. Investigation of other solid-fluid configurations shows that this algorithm is capable of capturing many physical phenomena of coupled fluid and solid mechanics. The algorithm is versatile enough to be extended to a wide variety of such problems, and its numerical performance is shown to be at least equal to Baaijens' [Baaijens, 2001] algorithm.

During the course of performing these simulations, it was noted that the scheme performed best when the grid size of the solid and fluid were comparable. It is suspected that the system of equations becomes ill conditioned when the grid sizes are significantly disparate. This is especially evident when the solid element size is small compared to that of the fluid. In this case a multitude of Lagrange multipliers are associated with the kinematic condition within the same fluid element. While this should be investigated further, it does not seem unreasonable that this leads to an underconstrained system of equations.

References

Arienti, M, Hung, P., Morano, E. and Shepherd, J. S., "The level set approach to Eulerian-Lagrangian coupling," *Journal of Computational Physics,* **185**(1), 213, (2003).

Baaijens, F. P. T. "A fictitious domain/mortar element method for fluid-structure interaction," *International Journal for Numerical Methods in Fluids*, **35**(7), 743 (2001).

Baer, T. A., "New and Improved Level Set Tutorial," Internal Sandia Memorandum to Distribution, December, 2003.

Baer, T.A., Cairncross, R.A., Schunk, P.R., Sackinger, P.A., Rao, R.R., "A finite element method for free surface flows of incompressible fluids in three dimensions. Part II. dynamic wetting lines," *International Journal for Numerical Methods in Fluids,* **33(3)**, 375, (2000).

Belytschko, T., Moes, N., Usui, S., Parimi, C., "Arbitrary discontinuities in finite elements," *International Journal for Numerical Methods in Engineering,* **50(4)**, 993, (2001).

Bird, R. B., Stewart, W. E. and Lightfoot, E. N., *Transport Phenomena*, Wiley, New York, 1960, 192.

Cairncross, R.A., Schunk, P.R., Baer, T.A., Rao, R.R., Sackinger, P.A., "A finite element method for free surface flows of incompressible fluids in three dimensions. Part I. Boundary fitted mesh motion.," *International Journal for Numerical Methods in Fluids,* **33(3)**, 375, (2000).

Chan, T. F. and Resasco, D. C. "Generalized deflated block-elimination," *SIAM J. Numer. Anal.* **23**, 913 (1986).

Chessa, J., Belytschko, T., "An extended finite element method for two-phase fluids," *Journal of Applied Mechanics-Transactions of the ASME,* **70(1)**, 10, (2003).

Chessa, J.; Smolinski, P.; Belytschko, T., "The extended finite element method (XFEM) for solidification problems," *International Journal for Numerical Methods in Engineering,* **53(8)**, 1959, (2002).

Davis, T. A. and Duff, I. S. "An unsymmetric-pattern multifrontal method for sparse LU factorization," *SIAM J. Matrix Analysis and Applications* **18**(1), 140 (1997).

De Hart, J., Peters, G.W.M, Schreurs, P.J.G., Baaijens, F.P.T., "A three-dimensional computational analysis of fluid-structure interaction in the aortic valve," *Journal of Biomechanics,* **36(1)**, 103, (2003).

Fedkiw, R.P., "Coupling an Eulerian fluid calculation to a Lagrangian solid calculation with the ghost fluid method," *Journal of Computational Physics.* **175(1)**, 200 (2002).

Fedkiw, R.P., Aslam, T., Merriman, B., Osher, S. "A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method)," *Journal of Computational Physics,* **152(2)**, 457, (1999).

Glowinski, R., Pan, T.W., Hesla, T.I., Joseph, D.D., Periaux, J., "A fictitious domain approach to the direct numerical simulation of incompressible viscous flow past moving rigid bodies: Application to particulate flow," *Journal of Computational Physics,* **169(2)**, 363, (2001).

Ji, H., D. Chopp, J.E. Dolbow, "A hybrid extend finite element/level set method for modeling phase transformations," *International Journal for Numerical Methods in Engineering*, **54**, 1209-1233, (2002)

Melenk, J.M., I. Babuska, "The partition of unity finite element method: Basic theory and applications," *Comp. Meth. Appl. Mech. and Eng.*, **139,** 289, (1996)

Morse, P. M. and Feshbach, H., *Methods of Theoretical Physics,* McGraw-Hill, New York, 1953, 275.

Sackinger, P. A., Schunk, P. R. and Rao, R. R., "A Newton-Raphson Pseudo-Solid domain Mapping Technique for Free and Moving Boundary Problems: A Finite Element Implementation", *Journal of Computational Physics*, **125**, 83 (1996).

Schunk, P. R., "TALE: An Arbitrary Lagrangian-Eulerian Approach to Fluid-Structure Interaction Problems," Sandia National Laboratories Report, SAND2000-0807, 2000.

Schunk, P. R., Sackinger, P. A., Rao, R. R., Chen, K. S., Baer, T. A., Labreche, D. A., Sun, A. C., Hopkins, M. M., Subia, S. R., Moffat, H. K., Secor, R. B., Roach, R. A., Wilkes, E. D., Noble, D. R., Hopkins, P. L., and Notz, P. K. "GOMA 4.0 - A Full-Newton Finite Element Program for Free and Moving Boundary Problems with Coupled Fluid/Solid Momentum, Energy, Mass, and Chemical Species Transport: User's Guide," Sandia National Laboratories Report SAND2002-3204, 2002.

Schunk, P. R. and Wilkes, E. D., "GOMA's Overset Mesh Method: User Tutorial," Internal Sandia Memorandum to Distribution, December, 2003

Sethian, J. A., *The Level Set Method and Fast Marching Methods*, Cambridge University Press, Cambridge, United Kingdom, 1999.

Wagner, G.J., N. Moës, W.K. Liu, T. Belytschko, "The extended finite element for rigid particles in Stokes flow," *International Journal for Numerical Methods in Engineering*, **51**, 293, (2001).

Distribution

MS 0316	L. C. Musson	9233	1
MS 0323	D. L. Chavez	1011	1
MS 0427	J. R. Weatherby	2134	1
MS 0824	W. L. Hermina	9110	1
MS 0826	S. N. Kempka	9113	1
MS 0826	D. R. Noble	9113	10
MS 0826	P. A. Sackinger	9113	1
MS 0834	M. R. Prairie	9112	1
MS 0834	T. A. Baer	9114	1
MS 0834	A. M. Grillet	9114	1
MS 0834	M. M. Hopkins	9114	1
MS 0834	P. K. Notz	9114	1
MS 0834	R. R. Rao	9114	10
MS 0834	R. A. Roach	9114	1
MS 0834	P. R. Schunk	9114	5
MS 0834	E. D. Wilkes	9114	1
MS 0834	J. E. Johannes	9114	1
MS 0834	E. S. Hertel	9116	1
MS 0835	E. A. Boucheron	9141	1
MS 0835	S. W. Bova	9141	1
MS 0835	J. M. McGlaun	9140	1
MS 0835	S. R. Subia	9141	1
MS 0836	R. O. Griffith	9117	1
MS 0841	T. C. Bickel	9100	1
MS 0847	H. S. Morgan	9120	1
MS 0847	D. J. Segalman	9124	1
MS 9018	Central Technical Files, 8945-1		1
MS 0899	Technical Library, 9616		2