LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

LLNL-JRNL-533451

An embedded mesh method in a multiple material ALE

M. A. Puso, J. Sanders, R. Settgast, B. Liu

February 28, 2012

Computer Methods in Applied Mechanics and Engineering

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

An embedded mesh method in a multiple material ALE

Michael A. Puso[1,*], Jessica Sanders[1], Randy Settgast, Ben Liu

Abstract

A new approach for treating the mechanical interactions of overlapping finite element meshes is presented. Referred to as embedded mesh methods here, these overlapping mesh methods typically include a *foreground* solid mesh and a *background* Euler fluid grid or solid mesh. A number of different approaches have been used in previous work to characterize the interactions of the background and foreground meshes at the interface. Lagrange multipliers are well suited to enforce the continuity constraints but care must be taken such that the resulting formulation is stable. Several Lagrange multiplier techniques are examined in this work and applied to coupling solid meshes and fluid-structure interaction type problems. In addition, details regarding implementation in a two-step, multi-material, Arbitrary Lagrangian Eulerian (ALE) code are presented. Example problems demonstrate convergence and applicability to a range of problems. In particular, the fluid-structure interaction examples focus on blast applications.

Keywords: Coupled Euler-Lagrange, Overlapping mesh methods, Lagrange multiplier methods

1. Introduction

Methods for treating embedded (overlapping) meshes avoid body fitted meshes (Figure 1) and offer a significant simplification in model development. These methods can be used to couple a solid foreground mesh to a solid background mesh in Lagrange-Lagrange coupling. (e.g., carbon particles in rubber, aggregate in a cement matrix, etc). Typically of more interest is the Euler-Lagrange coupling of a solid foreground mesh to an Eulerian fluid background mesh (e.g. explosive blast on armor, sloshing problems etc.). Classical ALE methods [1, 2] also provide fluid structure coupling for these applications but sometimes fail to provide the appropriate mesh relaxation and terminate as demonstrated in examples herein. The utility of embedded mesh methods has led to many different approaches in the literature starting with the CEL (Coupled Euler-Lagrange) code of Noh [1] from Lawrence Livermore Laboratory in 1964. Like CEL, many of the approaches have been applied to finite difference (FD) and finite volume (FV) background grids because of the widespread use of these methods for fluids. Recently a number of embedded mesh approaches e.g. [3–11] have been applied to finite elements (FE) and coupled Euler-Lagrange is now available in some commercial FE codes [12, 13]. Nonetheless, many of these approaches have had drawbacks that have limited there popularity and proliferation. In particular, issues of consistency, stability and/or robustness are still open issues in many embedded mesh methods.

In this work, aspects of consistency, stability and/or robustness are addressed. First, it is well known [11, 14, 15] that constraints or multiplier spaces defined on the *foreground* mesh can be susceptible to mesh locking. Nonetheless, many finite element applications use this approach mainly out of simplicity [7, 8, 12, 13] Here, two different approaches for defining multipliers on the *background* mesh are applied to the embedded mesh and are found to work quite well in general application. Second, details regarding implementation of overlapping meshes into a two-step ALE method are presented. In particular, this includes: how the

*Corresponding author. Address: 7000 East Ave, Livermore, CA , 94550

Email addresses: puso1@llnl.gov (Michael A. Puso), sanders39@llnl.gov (Jessica Sanders), settgast@llnl.gov (Randy Settgast), liu15@llnl.gov (Ben Liu)

[1]Methods Development Group, Lawrence Livermore National Laboratory, Livermore, CA 94550

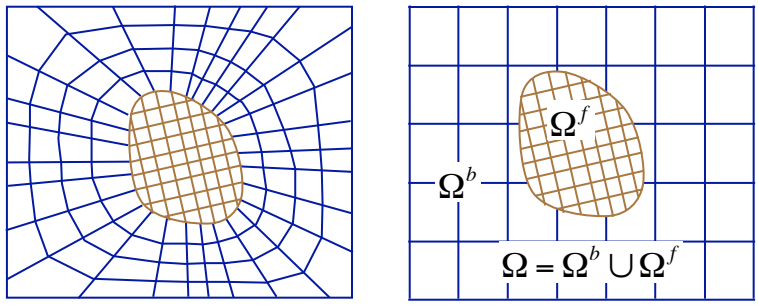

Figure 1: Body fitted mesh and embedded mesh

constraint is applied in the equations of motion, how explicit time integration and advection are performed, and how partially filled cells are treated. Overall, the techniques applied in this work appear novel in the area of embedded meshes and worked well in the example problems.

An outline of the paper is the following. Section 2 defines the boundary value problem and associated quantities used throughout the presentation. In Section 3, a number of alternative approaches from the literature are presented; the discussion pointing to relevant issues addressed in this work. In Section 4, the proposed Lagrange multiplier coupling is presented within the context of a two step (operator split) ALE approach. A box summarizing the algorithmic flow is given in Section 5. Section 6 presents example problems demonstrating convergence and versatility of the proposed method. Most embedded mesh implementations are CEL type and employ stationary Eulerian background meshes; here, moving background meshes are also considered, i.e, coupled ALE-Lagrange. Section 7 provides a summary and discussion.

2. Mathematical Problem Statement

Referring to Figure 1, foreground and background mesh quantities will be denoted by superscript f and b respectively and $\Gamma = \Omega^f \cap \Omega^b$ will represent the foreground boundary surface. Thus, positions $\boldsymbol{x}^b$ and $\boldsymbol{x}^f$ represent spatial coordinates on the background and foreground meshes respectively. For the sake of presentation, the foreground mesh will be entirely enveloped within the background mesh, no Dirichelet boundary condition will be applied on Γ and only one foreground mesh is considered. Here, the background mesh can be stationary (Eulerian) or moving with a convective velocity $\boldsymbol{c}$ with respect to the spatial coordinates (ALE) As defined in [16], $\boldsymbol{c} = 0$ for Lagrangian mesh motion and $\boldsymbol{c}$ equals the material velocity for a stationary Eulerian mesh. ALE codes typically choose $\boldsymbol{c}$ to keep the motion *nearly* Lagrangian using minimal mesh relaxation to avoid tangling. The kinematic description of the background mesh in this presentation will be Eulerian such that velocity is expressed in current (spatial) coordinates i.e., $\boldsymbol{v}^b(\boldsymbol{x}^b, t)$ and mesh motion ($\boldsymbol{c} \neq 0$) effects are accounted for by the material time derivative calculation in the usual way [16]. The foreground mesh will be treated as Lagrangian with velocity expressed in terms of material coordinates $\boldsymbol{X}^f$, i.e. $\boldsymbol{v}^f(\boldsymbol{X}^f, t)$. The continuum form of constraints between foreground and background mesh assuming no slip and are written

$$\boldsymbol{v}^b(\boldsymbol{x}, t) - \boldsymbol{v}^f({\boldsymbol{\varphi}^f}^{-1}(\boldsymbol{x}), t) = 0 \quad \boldsymbol{x} \in \Gamma \tag{1}$$

$$\boldsymbol{t}^b(\boldsymbol{x}, t) - \boldsymbol{t}^f({\boldsymbol{\varphi}^f}^{-1}(\boldsymbol{x}), t) = 0 \quad \boldsymbol{x} \in \Gamma \tag{2}$$

where $\boldsymbol{\varphi}^f$ is the deformation mapping $\boldsymbol{x}^f = \boldsymbol{\varphi}^f(\boldsymbol{X}^f, t)$, and $\boldsymbol{t}$ is the surface traction. No slip is assumed in this work, but its extension to slip conditions is straightforward. The weak form of conservation of

momentum for the coupled system is then written

$$\int_{\Omega^b} \rho^b \cdot \boldsymbol{w}^b \frac{D\boldsymbol{v}^b}{Dt}\, d\Omega + \int_{\Omega^b} \nabla \boldsymbol{w}^b : \boldsymbol{\sigma}^b\, d\Omega - \int_{\Omega^b} \boldsymbol{w}^b \cdot \boldsymbol{b}^b\, d\Omega - \int_{\Gamma^b_N} \boldsymbol{w} \cdot \boldsymbol{t}^b_N\, d\Gamma$$
$$+ \int_{\Omega^f} \rho^f \boldsymbol{w}^f \cdot \boldsymbol{v}^f_{,t}\, d\Omega + \int_{\Omega^f} \nabla \boldsymbol{w} : \boldsymbol{\sigma}^f\, d\Omega - \int_{\Omega^f} \boldsymbol{w}^f \cdot \boldsymbol{b}^f\, d\Omega - \int_{\Gamma^f_N} \boldsymbol{w} \cdot \boldsymbol{t}^f_N\, d\Gamma$$
$$= \int_{\Gamma} (\boldsymbol{t}^f \cdot \boldsymbol{w}^f + \boldsymbol{t}^b \cdot \boldsymbol{w}^b)\, d\Gamma \quad (3)$$

where ρ is density, $\boldsymbol{v}$ is velocity, D/Dt is material time derivative, $(\cdot)_{,t}$ is a partial derivative, $\boldsymbol{w}$ is a test function, $\boldsymbol{\sigma}$ is stress, $\boldsymbol{b}$ is a body force and $\boldsymbol{t}_N$ are Neumann boundary conditions on Γ_N. The material time derivative captures mesh motion $\boldsymbol{c}$ whilst the material and partial time derivatives are equivalent for the Lagrangian description of the foreground velocities $\boldsymbol{v}^f$. The right hand side terms represent the coupling terms that are fundamental to the different finite element embedded mesh approaches and will be discussed in what follows. The Eulerian and ALE descriptions for the background mesh will require additional equations for the conservation of mass and energy

$$\frac{D}{Dt}\int_{\Omega} \rho^b\, d\Omega = 0 \qquad \frac{D}{Dt}\int \epsilon\, d\Omega = \int_{\Omega} \nabla \boldsymbol{v}^b : \boldsymbol{\sigma}^b\, d\Omega \quad (4)$$

where ϵ is the energy per unit volume and adiabatic conditions are assumed.

Solution of the fully coupled system presented in Section 4 involves the weak form satisfaction of surface constraints (1,2), momentum (3) and finite volume satisfaction of conservation equations (4).

3. Alternative Approaches

A short discussion of embedded grid implementation using finite volume/differences and finite elements along with embedded Dirichelet constraint methods will provide the context for the development of the proposed approach.

3.1. Finite volume and finite difference applications

Many of the first embedded mesh methods for Euler-Lagrange type coupling were applied to FV and FD background grids. FE implementations are fundamentally different (e.g., how forces are transmitted at the interface); nonetheless, there are some shared issues with FV and FD worth considering. In general, most of these methods use explicit time integration and one of the issues the CEL method [1] and other methods that "cut" cells encounter is the small stable time steps due to partially filled cells. The ghost fluid method [17, 18] avoids these small time steps by simply not cutting the cell, i.e. all cells are full or vacant. Results demonstrate that induced error is often small and is order h. Variations of this method have been used successfully in the research/production codes VTF [19] and DYSMAS [20]. The time step is advanced from t_n to t_{n+1} by applying fluid pressure from t_n to the solid boundary and using solid velocities at t_n as a velocity boundary condition to the fluid background grid in a *concurrent time coupling* scheme[2]. These velocity boundary conditions are applied by using the solid foreground velocities to initialize velocities in ghost cells placed on the non-fluid side of the fluid-structure boundary via a so-call mirrored (symmetry) boundary condition approach. This ghost cell scheme facilitates the use of high resolution fluid solvers without modification. Another notable method originally applied for blood flow in the heart is the Immersed Boundary Method [21]. Here, a Dirac delta like force is applied to all solid nodes (typically membranes or shells) to constrain them to move with the background fluid. Overset grid methods [22] apply a conforming

[2]DYSMAS also provides a staggered time stepping approach they claim is more stable

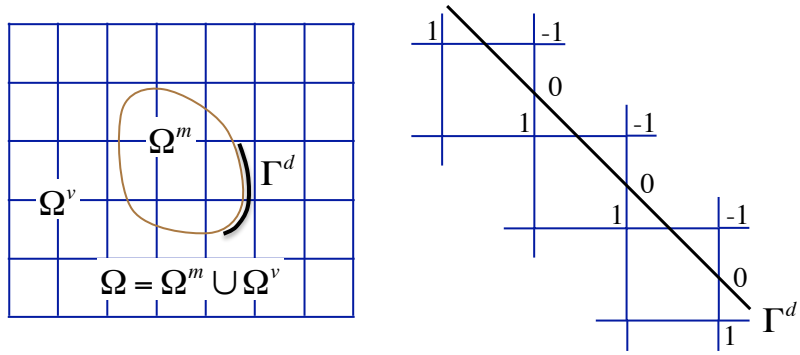

Figure 2: (a) Embedded Dirichelet domain with constraint on Γ^d (b) Example of boundary locking when the *homogenous* boundary condition is enforced point-wise on Γ^d: the bilinear quadrilaterals only admit a $1, 0, -1$ displacement mode along the boundary.

fluid mesh to the Lagrange solid and then use finite differencing to couple the fluid overset grid to the fluid background grid. The Immersed Boundary Method and many overset methods avoid the cut cells.

Finally, an embedded method using a novel material insertion technique was developed for the Sandia tool Zapotec [23, 24] coupling the production codes CTH (fluid) and PRONTO (structural). Here fluid pressures from the previous time step are applied to the solid body surface and a Lagrange step determines the velocity of all solid nodes. Next, *all* solid material and history are remapped/inserted into the background cells [3]. The velocities of the solid material are also mapped to the background grid. An Eulerian step that includes both fluid and solid material in the same discretizaion is then taken to generate a new solid pressure used to begin the same process for the next time step.

3.2. Embedded Dirichelet constraints

A significant amount of literature has focused on embedded Dirichelet constraints for finite elements [14, 15, 25, 26]. The embedded Dirichelet method (Figure 2(a)) models an arbitrary domain of interest Ω^m on a grid Ω with Dirichelet boundary conditions along Γ^d. What is pointed out nicely in [14, 15] is that Lagrange multipliers can be used to apply the constraint along Γ^d but point wise enforcement of the constraint will cause boundary locking such that only displacement modes of the form shown in Figure 2(b) are admitted. This form of locking results in the loss of the polynomial approximation properties of the affected background elements from order h^2 to order h. It turns out that this type of locking can also occur for the embedded mesh constraint when the foreground mesh is stiff compared to the background mesh and "too many" constraints are applied at the interface. In addition to the loss of accuracy, acute stress oscillations result on the foreground mesh as seen in the numerical example in Section 6.1 and Figure 11(b).

3.3. Finite element applications

Some of the first embedded mesh methods for finite elements were the fictitious domain methods [3] and immersed finite elements [5]. These methods allow the fluid background grid to cover the entire domain Ω in Figure 1 so that there are no cut cells. The fluid under the solid mesh is constrained to move with the solid; e.g. the immersed FE method uses collocation to enforce that fluid nodes under the solid mesh move with the solid. This causes an excess virtual work which is reconciled by subtracting fluid virtual work discretized on the solid mesh Ω_f. Again these methods avoid cut fluid cells and are good at handling a large number of small stiff particles immersed in a fluid. On the other hand consistency errors from collocation occur such as leaks when the solid mesh is coarse compared to the fluid grid. Furthermore, these methods

[3] Cells adjacent to the fluid structure boundary would be "mixed zones" of fluid and solid

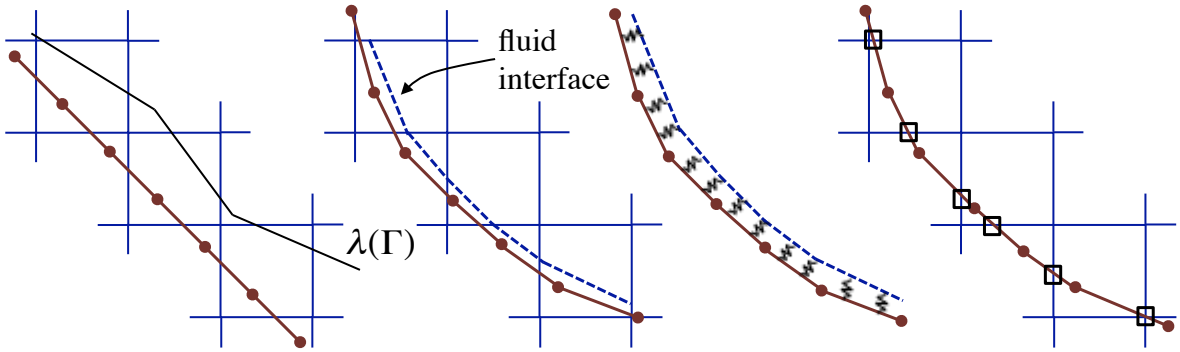

Figure 3: (a) Mortar type constraints use pressure interpolation on foreground surface (b) Surface on background fluid approximated by volume of fluid (VOF). Dotted line represents approximate surface and defines the cut cells. (c) Springs used to close gap between foreground and VOF reconstructed surface. (d) Distributed Lagrange multipliers interpolated between open squares i.e. one piecewise linear function per background cell.

originally weren't meant to be general and treat multiple fluid materials (e.g. air and water) together in the background.

There are a number of methods that define the constraints based on the foreground surface mesh (Figure 3). For example [27] uses piecewise linear Lagrange multipliers $\boldsymbol{\lambda}$ (Figure 3(a)) to approximate the traction on the surface such that the foreground and background velocities are weakly enforced i.e.,

$$\int_\Gamma \boldsymbol{\lambda} \cdot (\boldsymbol{v}^b - \boldsymbol{v}^f)\, d\Gamma = 0 \tag{5}$$

where $\boldsymbol{v}^b$ is the background and $\boldsymbol{v}^f$ is the foreground velocities. A geometric simplification of the foreground mesh surface on the background mesh is often used to cut cells *and* compute the constraint. For example, refering to 3(b) the methods in [7, 12, 13] use a volume of fluid method (VOF) with piecewise linear interface reconstruction (PLIC) [28] to define the "cut cells". Here, both void and fluid are advected and PLIC interface reconstruction is used to *approximate* the solid surface. In these applications [7, 12, 13], penalties are used to force the solid and reconstructed surfaces together at collocation points as in Figure 3(c). In the event that the foreground mesh is very stiff and fine compared to the background mesh, these aforementioned multiplier and penalty methods can suffer from the boundary locking as in Figure 11.

Legay et al. [8] uses a two dimensional distributed Lagrange multiplier method defined on the surface. Here, piecewise linear Lagrange multipliers are placed one per background element as shown in Figure 3(d). The authors did not address the issue of boundary locking but it is expected that this approach would not lock. Implementation of this distributed Lagrange multiplier approach in three dimensions would not be straightforward.

The Nitsche approach [25] was originally applied to embedded mesh boundary constraints in [4] for the Poisson's equation and fluid structure Euler-Lagrange coupling in [8]. The weak form coupling terms of (3) for the Nitsche approach are written

$$\int_\Gamma (\boldsymbol{t}^f \cdot \boldsymbol{v}^f + \boldsymbol{t}^b \cdot \boldsymbol{v}^b)\, d\Gamma = \int_\Gamma [[\boldsymbol{w}]] \otimes \boldsymbol{n}^f : \langle \boldsymbol{\sigma}(\boldsymbol{u}) \rangle \, d\Gamma + \int_\Gamma [[\boldsymbol{u}]] \otimes \boldsymbol{n}^f : \langle \boldsymbol{\sigma}(\boldsymbol{w}) \rangle \, d\Gamma - \beta \int_\Gamma [[\boldsymbol{w}]]\, [[\boldsymbol{u}]] \, d\Gamma \tag{6}$$

where $[[\boldsymbol{u}]] = (\boldsymbol{u}^f - \boldsymbol{u}^b)$, $\langle \boldsymbol{\sigma} \rangle = \alpha \boldsymbol{\sigma}^f + (1-\alpha)\boldsymbol{\sigma}^b$ and β is a stabilization term. Here, β must be chosen to be sufficiently big so that (3) is coercive. Whereas, $\alpha = 0.5$ is typically used for this approach as in [8], it was found in [29] that the boundary locking would occur even at low β and suggests that $\alpha = 0$ be used. The first Nitsche term is for consistency (i.e. patch test) and the second term containing $\langle \boldsymbol{\sigma}(\boldsymbol{w}) \rangle$ provides symmetry. This term becomes awkward for nonlinear materials since it requires the material stiffness. Wang and Belytschko [8] evaluate the necessity of the second term whereas a weak form similar to that in (6) is

developed in [11] using a discontinuous Galerkin (DG) strain which provides a symmetric formulation similar to (6) without the appearance of the material stiffness. Finally, an approach formulated as a mixed method but closely related to Nitsche and DG methods is presented in Gerstenberger and Wall [10] and shown to perform well in fluid-structure analyses.

Simulation of blast effects on structures is one of the prime applications for the CEL method. Since these analyses include transient dynamics and compressible flow, explicit time integration is usually the method of choice. The penalty methods (e.g. [12, 13]) are very efficient in an explicit scheme but must balance the strength of the penalty with stable time step. Lagrange multiplier methods are more challenging in an explicit scheme and often require a linear solution approach for the multipliers [30]. The Nitsche and related methods (e.g. [4, 8, 10, 11]) can potentially apply a small penalty or none at all. The choice of stable time steps in an explicit scheme for these types of methods is not always clear, particularly for the unsymmetric forms. As shown in [31], cut elements with small volumes require a high β in (6) for coercivity. A high β will often control the stable time step. Legay et al. [8] include the the Nitsche terms from (6) in the mass matrix of their explicit time integration schemes. The resulting system matrix is non-diagonal and requires a linear equation solver to advance the solution.

4. Proposed ALE Embedded Grid Approach

In this section, the discrete forms for momentum and Lagrange multiplier constraints are defined. Implementation into a two step ALE approach is then presented. Finally, aspects of the Lagrange solve and embedded grid modified advection remap steps are described.

4.1. Discrete background element and void domains

The background domain Ω^b (Figure 1) is composed of elements $e \in E$ with domain Ω_e defined in the standard way. That is, Ω_e is defined over the entire isoparametric space of the element defined by nodal coordinates and may exist inside, outside or partially outside Ω^b. The subset of elements that are cut by the foreground surface is E^c and the associated cut volumes Ω_e^c are defined

$$E^c = \{e \in E : \Omega_e \cap \Gamma\} \qquad \Omega_e^c = \Omega_e \cap \Omega^b \; e \in E^c \tag{7}$$

The entire background domain will typically be meshed and $\Omega^e \cap \Omega^f$ will be referred to as "void" in the following as there will only be fluid on one side of the interface Γ in this work; extensions to doubly wetted interfaces with fluid on both sides of a shell foreground interface will be the subject of future work.

A significant computational effort involves locating cut cells and a detailed description of the algorithms used is beyond the scope of this article. It suffices to say that cells are cut by determining intersections of foreground surface segments and background cells using computational geometry. This contrasts methods such as [7] that advect void and fluid and use PLIC interface reconstruction to compute cut cell volumes. The search for solid surface-background cell intersections begins with an expensive bucket type sort at problem initialization. Then, by limiting the motion of the foreground surface nodes to move no more than one background cell per time step, efficient incremental searches (linear in cost) can be exploited to find cut elements for all following time steps. This motion limitation is enforced by appropriate time step control and is consistent with CFL stability requirements for the explicit time integration of the background fluids. Cells statuses are: cut, void or filled (e.g. all fluid for Euler-Lagrange). Because of the aforementioned motion limitation, cells status can be monitored recursively such that filled cells must be cut before they transition to void and vice-versa.

4.2. Discrete forms

Standard FE approximations are defined for velocity

$$\boldsymbol{v}^{b(h)} = \sum_{A \in S^b} N_A^b \boldsymbol{v}_A^b \quad \boldsymbol{v}^{f(h)} = \sum_{A \in S^f} N_A^f \boldsymbol{v}_A^f$$
$$S^b = \{\text{background nodes}\} \quad S^f = \{\text{foreground surface nodes}\} \tag{8}$$

and similarly the test functions $\boldsymbol{w}^{b(h)}$ and $\boldsymbol{w}^{f(h)}$ are used in (3) and provide the usual discrete forms for *diagonal* mass matrices and force vectors

$$\begin{array}{lll} M_{AB}^b = M_A^b\,\delta_{AB} & \boldsymbol{f}_A^{b(int)} = \int_{\Omega^b} \nabla N_A^b \cdot \boldsymbol{\sigma}^b \, d\Omega & \boldsymbol{f}_A^{b(ext)} = \int_{\Omega^b} N_A^b \boldsymbol{b}^b \, d\Omega + \int_{\Gamma_N^b} N_A^b \boldsymbol{t}_N^b \, d\Gamma \\ M_{AB}^f = M_A^f\,\delta_{AB} & \boldsymbol{f}_A^{f(int)} = \int_{\Omega^f} \nabla N_A^f \cdot \boldsymbol{\sigma}^f \, d\Omega & \boldsymbol{f}_A^{f(ext)} = \int_{\Omega^f} N_A^f \boldsymbol{b}^f \, d\Omega + \int_{\Gamma_N^f} N_A^f \boldsymbol{t}_N^f \, d\Gamma \end{array} \tag{9}$$

where nodal masses M_A are formed from surrounding element masses as usual (refer to Section 4.4.2 and 4.4.3 regarding element mass of cut cells). Where appropriate, subscripts for nodal velocities and other discretized quantities will be dropped, for example nodal momentum can be written one of two ways

$$\boldsymbol{p}_A^b = \sum_{B \in S^b} M_{AB}^b \boldsymbol{v}_B^b \quad \forall A \in S^b \quad \Longleftrightarrow \quad \boldsymbol{p}^b = \boldsymbol{M}^b \boldsymbol{v}^b \tag{10}$$

4.3. Lagrange multipliers

As noted in Section 3.3, a number of finite element embedded mesh implementations [7, 12, 13, 27] employ forms of Lagrange multiplier or penalty collocation that use the foreground mesh to define the number of discrete constraints and thus may encounter boundary locking in certain cases. A notable exception was the distributed Lagrange multiplier method [8] which was developed in two dimensions but would require non-trivial extension to three dimensions. In contrast, two different Lagrange multiplier methods that are defined on the background mesh and simple to implement in three dimensions, are proposed here.

4.3.1. Piecewise constant multipliers

The first approach comes from [26] where embedded Dirichelet constraints were applied to the Poisson problem in two dimensions. Using the prescribed stabilization, the approach was shown to be LBB stable and uses a piecewise constant multiplier space defined on cut elements (7) as follows

$$\mathcal{W}_0^h = \{\boldsymbol{\lambda}^h : \boldsymbol{\lambda}^h = \sum_{e \in E^c} P_0^e(\boldsymbol{x}) \boldsymbol{\lambda}_e ; P_0^e(\boldsymbol{x}) = \left\{ \begin{array}{ll} 0 & \boldsymbol{x} \in \Omega_e \\ 1 & \boldsymbol{x} \notin \Omega_e \end{array} \right. \} \tag{11}$$

Whereas [26] used scalar element pressure, the multipliers are element traction vectors $\boldsymbol{\lambda}_e \in \mathbb{R}^3$ here. To stabilize the "checkerboard" mode that can result from (11), the method penalizes the difference in adjacent background element pressures $\boldsymbol{\lambda}_e$. To make this precise, let F be the set of all shared element faces chosen from the set of background elements E i.e.,

$$F = \{(e_{f1}, e_{f2}) : meas(\partial\Omega_{e_{f1}} \cap \partial\Omega_{e_{f2}}) > 0 \; \forall \, e_{f1}, e_{f2} \in E ; e_{f1} \neq e_{f2}\}$$
$$F^c = \{f \in F : meas(\partial\Omega_{e_{f1}} \cap \partial\Omega_{e_{f2}} \cap \Gamma) > 0\} \tag{12}$$

where f is a shared element face denoted by the element pairs (e_{f1}, e_{f2}) and F^c are those f cut by interface Γ. Given an arbitrary traction $\boldsymbol{\mu}^h \in \mathcal{W}_0^h$ and definition (12), the method uses the following discrete stabilized constraint

$$\int_\Gamma \boldsymbol{\mu}^h \cdot (\boldsymbol{v}^{b(h)} - \boldsymbol{v}^{f(h)}) \, d\Omega - \sum_{f \in F^c} \gamma_f h A_f (\boldsymbol{\mu}_{e_{f1}} - \boldsymbol{\mu}_{e_{f2}}) \cdot (\boldsymbol{\lambda}_{e_{f1}} - \boldsymbol{\lambda}_{e_{f2}}) / \Delta t = 0 \tag{13}$$

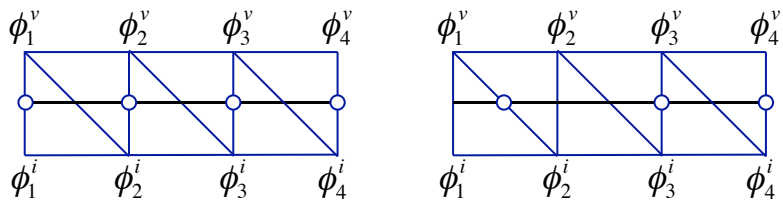

Figure 4: Two different edge multiplier candidates for vital vertex method. Background mesh is blue and foreground mesh is black. Circles are "vital vertices" and represent edges chosen to define multipliers

where the first term in (13) defines the velocity constraint, the second term stabilizes traction fields $\boldsymbol{\lambda}^h$, γ_f is the stabilization term, h is background element size and A_f is the area of face f. Whereas [26] used a displacement constraint, the velocity constraint is used in (13) for the fluid structure interaction problem leading to the time Δt in the second term of (13).

4.3.2. Nodally defined multipliers

The multiplier space in Section 4.3.1 requires a stabilization term and choice of this term is discussed in Section 6. Definition of a nodal or edge defined multiplier space can avoid the checkerboard mode and be a good candidate for a stable space. One such method is the method of vital vertices and was used primarily in [15] to enforce embedded Dirichelet constraints as it was proven LBB stable. Applied to two dimensional triangular discretizations, this method introduces multipliers on select cut edges necessary to produce consistent pressure fields (i.e. reproduce constants). The arrangement of the multipliers is not unique and an example of this method[4] is shown in Figure 4. Circles are the "vital vertices" and denote cut edges possessing multipliers. Background shape functions ϕ_a^v and and ϕ_a^i are identified with the underlying Lagrange polynomials N_A used in (8) but renumbered in an edgewise fashion for presentation. Here, ϕ_a^v lie on the void side and ϕ_a^i lie on the interior (e.g. fluid) side of cut edges. The key ingredient in choosing the special vital vertex edges is that none of the vital vertex edges are allowed to share a common void or interior node (easily verified from Figure 4). Figure 4(a) illustrates the most obvious arrangement whilst (b) represents another viable option. The discrete multiplier for cases (a) and (b) are written

$$\boldsymbol{\lambda}^h = (\phi_1^v + \phi_1^i)\boldsymbol{\lambda}_1 + (\phi_2^v + \phi_2^i)\boldsymbol{\lambda}_2 + (\phi_3^v + \phi_3^i)\boldsymbol{\lambda}_3 + (\phi_4^v + \phi_4^i)\boldsymbol{\lambda}_4 \qquad \text{case (a)} \tag{14}$$

$$\boldsymbol{\lambda}^h = (\phi_1^v + \phi_2^v + \phi_2^i + \phi_2^v/2)\boldsymbol{\lambda}_1 + (\phi_2^v/2 + \phi_3^v + \phi_3^i)\boldsymbol{\lambda}_2 + (\phi_4^v + \phi_4^i)\boldsymbol{\lambda}_4 \qquad \text{case (b)} \tag{15}$$

A set of rules was proposed in [15] to generate the vital vertices but the authors suggest that the best approach is still an open issue. The method is elegant and well founded in the theory but does introduce some "bookkeeping" issues that can complicate a parallel implementation. In particular, applying the rules to determine vital vertices for elements at the boundary of parallel domains is not intractable but relatively messy.

In contrast, a method that uses just the void nodes of cut elements to define the multipliers is examined here. Definitions (7) and (8) define the set of void nodes $S^{v,e}$ and interior nodes $S^{i,e}$ on element e as

$$S^{v,e} = \{A \in S^b : \boldsymbol{x}_A^b \cap \Omega_e \neq 0, e \in E^c, \boldsymbol{x}_A \notin \Omega^b\} \quad S^{i,e} = \{A \in S^b : \boldsymbol{x}_A^b \cap \Omega_e \neq 0, e \in E^c, \boldsymbol{x}_A \in \Omega^b\}. \tag{16}$$

Now for each $e \in E^c$, a partition of unity of the void nodes on each element e leads to the definition of the multiplier shape functions and resulting space

$$\phi_A^e(\boldsymbol{\xi}) = \frac{N_A^b(\boldsymbol{\xi})}{\sum_{B \in S^{v,e}} N_B^b(\boldsymbol{\xi})} \quad \forall\, A \in S^{v,e} \qquad \mathcal{W}^h = \{\boldsymbol{\lambda}^h : \boldsymbol{\lambda}^h = \sum_{e \in E^c} \sum_{A \in S^{v,e}} \phi_A^e \boldsymbol{\lambda}_A\}. \tag{17}$$

[4]Taken from Section 4 in [15]. Here, the notation of void ϕ_a^v and interior ϕ_a^i nodes takes the place of ϕ_a^1 and ϕ_a^2 from [15]

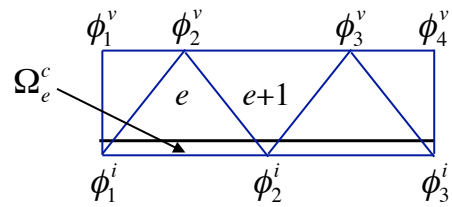

Figure 5: Example of small cut element volume Ω_e^c.

This interpolation space reproduces constants by construction. When the interface Γ is located such that the volume of the cut element is vanishingly small, $\Omega_e^c \to 0, e \in E^c$ (Figure 5), $(17)_1$ cannot be used. Here, the formulation reverts to the stabilized piecewise constant i.e. ϕ_A^e is replaced with $\phi^e = P_0^e$ from (11) for an element $e \in E^c$ when $\Omega_e^c < 0.001\Omega_e$. This rarely occurs and will be considered a degenerative case[5]. Let $\boldsymbol{\mu}^h \in \mathcal{W}^h$, the Lagrange multiplier constraint over $\Gamma_e = \Gamma \cap \Omega_e, e \in E^c$ is written

$$\int_{\Gamma_e} \boldsymbol{\mu}^h (\boldsymbol{v}^{b(h)} - \boldsymbol{v}^{f(h)})\, d\Gamma = \sum_{A \in S^{v,e}} \boldsymbol{\mu}_A \int_{\Gamma_e} (\sum_{B \in S^{v,e}} \phi_A^e N_B^b \boldsymbol{v}_B^b + \sum_{C \in S^{i,e}} \phi_A^e N_C^b \boldsymbol{v}_C^b - \sum_{D \in S^f} \phi_A^e N_D^f \boldsymbol{v}_D^b)\, d\Gamma \tag{18}$$

There is typically one constraint per void node (except for degenerative cases) and the void nodes $B \in S^{v,e}$ are effectively *slaved* to the interior $C \in S^{i,e}$ and foreground $D \in S^f$ nodes. The slaved void node portion of (18) (first term in parenthesis on right side) can be orthogonalized by following the dual mortar approach in [32]. To this end, ϕ_A^e in (18) is replaced with $\bar{\phi}_A^e$ which is defined by the linear combination of shape functions from (17)

$$\bar{\phi}_A^e = \sum_{C \in S^{v,e}} \phi_C^e \psi_{AC}^e \tag{19}$$

with coefficients ψ_{AC}^e. Replacing ϕ with $\bar{\phi}$ in (18) and using the partition of unity shape functions from (17), the coefficients are chosen to orthogonalize the "slave" term i.e.

$$\int_{\Gamma_e} \bar{\phi}_A^e N_B^b\, d\Gamma = \int_{\Gamma_e} \frac{N_C^b N_B^b}{\sum_{D \in S^{v,e}} N_D^b} \psi_{AC}^e = \mathcal{M}_{CB}^e \psi_{AC}^e = \delta_{AB} \int_{\Gamma_e} N_B^b\, d\Gamma \quad \forall A \in S^{v,e} \tag{20}$$

where no sum on B is implied. The matrix $\mathcal{M}_{CB}^e$ is positive definite and invertible when $\Omega_e^c > 0$. The rank of $\mathcal{M}_{CB}^e$ is determined by the number of void nodes i.e. $|S^{v,e}|$. For the degenerative case defined $\Omega_e^c < 0.001\Omega_e$, the constant interpolation ($\phi^e = P_0^e$) is used on e and no orthogonalization is performed. The main advantage of the (dual form) orthogonalization is that it expedites convergence of the iterative equation solver introduced in the next section by reducing the coupling.

4.4. Two step ALE

Many finite element ALE production codes employ a two step ALE approach (Figure 6(a)) [2, 12, 13, 33, 34] based on an operator split at each time step $t_n \to t_{n+1}$:

1. *Lagrange Step*: Treat deformation as Lagrangian over step while observing Lagrange multiplier embedded grid constraints.
2. *Advection Remap Step*: Relax background mesh and remap mass, energy, momentum and all history variables to new mesh configuration. Eulerian relaxation returns mesh to original configuration, i.e. no mesh motion. ALE relaxation moves mesh based on some criteria.

[5]For Euler-Lagrange coupling, this may often occur at initialization but rarely after significant flow.

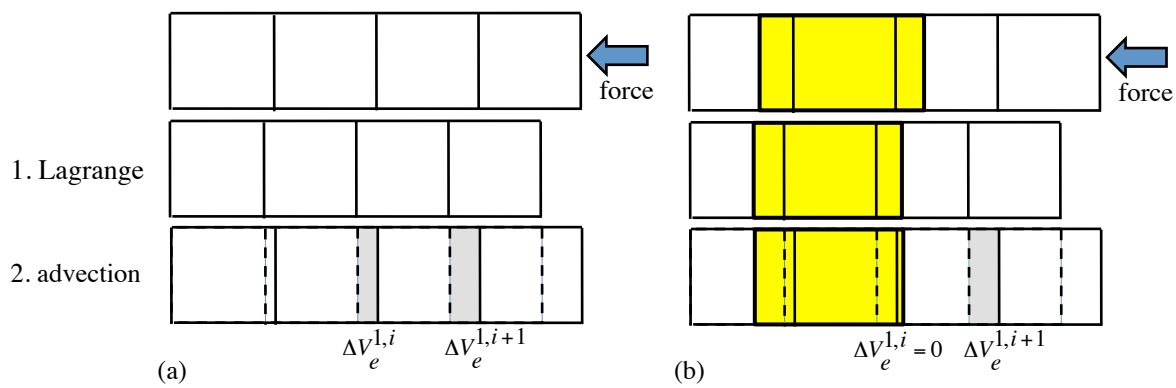

Figure 6: Two step ALE process: 1. Lagrange deformation 2. Advection remap where material is fluxed between old mesh (dotted line) and relaxed mesh (solid line) (a) No embedded mesh with single fluid material. (b) Embedded mesh with foreground in yellow, void "under" foreground solid and single fluid on background mesh. Lagrange step deforms background and foreground mesh. Advection remap relaxes background mesh and produces zero flux at embedded mesh interface

The following details this process and the modifications required for implementing the Lagrange multiplier constraints and special advection treatment at the interface of the foreground and background meshes for Euler-Lagrange coupling and ALE-Lagrange coupling (i.e. non-trivial mesh motion on background mesh).

4.4.1. Lagrange step: equations of motions

To advance from time t_n to t_{n+1}, the equations of motion are solved considering Lagrangian motion of foreground *and* background meshes over the step. Here background advection terms are neglected and Ω_n^b is considered the reference configuration. To facilitate the presentation, nodal velocities and forces are partitioned as follows:

$$\begin{aligned} \boldsymbol{v}^{b_c},\, \boldsymbol{v}^{f_c},\, \boldsymbol{f}^{b_c},\, \boldsymbol{f}^{f_c} &\equiv \text{nodal velocities,forces } \textit{with} \text{ embedded mesh constraints} \\ \boldsymbol{v}^{b_u},\, \boldsymbol{v}^{f_u},\, \boldsymbol{f}^{b_u},\, \boldsymbol{f}^{f_u} &\equiv \text{nodal velocities,forces } \textit{without} \text{ embedded mesh constraints} \end{aligned} \tag{21}$$

Using the notation (21), the discrete equations of motion (22) and constraint forms (13) or (18) can then be written for Euler-Lagrange coupling

$$\begin{aligned} \boldsymbol{M}_n^{b_u}(\hat{\boldsymbol{v}}_{n+1}^{b_u} - \boldsymbol{v}_n^{b_u})/\Delta t + \boldsymbol{f}_n^{b_u(int)} - \boldsymbol{f}_n^{b_u(ext)} &= 0 \\ \boldsymbol{M}_n^{f_u}(\boldsymbol{v}_{n+1}^{f_u} - \boldsymbol{v}_n^{f_u})/\Delta t + \boldsymbol{f}_n^{f_u(int)} - \boldsymbol{f}_n^{f_u(ext)} &= 0 \\ \boldsymbol{M}_n^{b_c}(\hat{\boldsymbol{v}}_{n+1}^{b_c} - \boldsymbol{v}_n^{b_c})/\Delta t + \boldsymbol{f}_n^{b_c(int)} - \boldsymbol{f}_n^{b_c(ext)} + \boldsymbol{G}^{b_c{}^T}\boldsymbol{\lambda}_{n+1} &= 0 \\ \boldsymbol{M}^{f_c}(\boldsymbol{v}_{n+1}^{f_c} - \boldsymbol{v}_n^{f_c})/\Delta t + \boldsymbol{f}_n^{f_c(int)} - \boldsymbol{f}_n^{f_c(ext)} - \boldsymbol{G}^{f_c{}^T}\boldsymbol{\lambda}_{n+1} &= 0 \\ \boldsymbol{G}_n^{b_c}\hat{\boldsymbol{v}}_{n+1}^{b_c} - \boldsymbol{G}_n^{f_c}\boldsymbol{v}_{n+1}^{f_c} - \boldsymbol{C}_n^{stab}\boldsymbol{\lambda}_{n+1} &= 0 \end{aligned} \tag{22}$$

where the matrices $\boldsymbol{G}^b$, $\boldsymbol{G}^f$ and $\boldsymbol{C}^{stab}$ incorporate the constraints and stabilization from (13) and (18) . The velocity $\hat{\boldsymbol{v}}_{n+1}^b$ is an intermediate velocity that is modified to get $\boldsymbol{v}_{n+1}^b$ after the advection remap step (Section 4.4.4) to account for local momentum conservation. The background mass and the constraint matrices in (22) are based on the t_n configuration for say Euler-Lagrange but are constant for Lagrange-Lagrange coupling.

The unconstrained velocities $\boldsymbol{v}_{n+1}^{b_u}$ and $\boldsymbol{v}_{n+1}^{f_u}$ are easily computed as a consequence of the diagonal mass matrix. On the other hand, despite explicit time integration, the Lagrange multipliers preclude the same simple solution for the constrained $\boldsymbol{v}_{n+1}^{b_c}$ and $\boldsymbol{v}_{n+1}^{f_c}$ velocities requiring a solution of the linear system of these equations. A solution approach similar to [30] is followed here such that $n+1$ velocities are solved for in terms of multipliers $\boldsymbol{\lambda}_{n+1}$ in the third and fourth equation and substituted into the last constraint equation

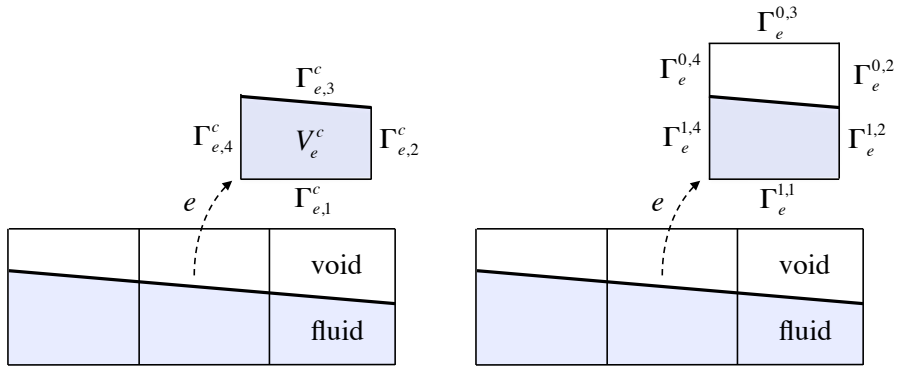

Figure 7: (a) Cut elements at interface with surfaces $\Gamma^c_{e,f}$ used for computing $\boldsymbol{\beta}^c_{e,A}$ for case of simple planar foreground surface. (b) Material area on element faces used for advection. Here $\Gamma^{k,i}_e$ is area of material k on face i

leading to a reduced set of equations

$$\boldsymbol{r} = (\boldsymbol{G}^b_n \boldsymbol{M}^{b\,-1}_n \boldsymbol{G}^{b\,T}_n + \boldsymbol{G}^f_n \boldsymbol{M}^{b\,-1}_n \boldsymbol{G}^{f\,T}_n + \boldsymbol{C}^{stab}_n/\Delta t)\boldsymbol{\lambda}_{n+1} + \boldsymbol{G}^b_n \boldsymbol{M}^{b\,-1}_n \bar{\boldsymbol{f}}^b_n - \boldsymbol{G}^f_n \boldsymbol{M}^{f\,-1}_n \bar{\boldsymbol{f}}^f_n = 0 \tag{23}$$

where subscript c is dropped from b_c and f_c, $\bar{\boldsymbol{f}}_n = \boldsymbol{f}^{(int)}_n - \boldsymbol{f}^{(ext)}_n - \boldsymbol{M}_n \boldsymbol{v}_n/\Delta t$ and $\boldsymbol{r}$ becomes the residual in a matrix free, conjugate gradient solution scheme. A diagonal preconditioner $\boldsymbol{D}$ is used to facilitate the solution and is defined:

$$D_A = \sum_B G^b_{AB} G^b_{AB}/M^b_B + \sum_C G^f_{AC} G^f_{AC}/M^f_C \tag{24}$$

where A refers to the multiplier index i.e. $\boldsymbol{\lambda}_A$, M^b_B/M^f_C are background/foreground nodal masses and the diagonal term C^{stab}_{AA} from (13) is neglected.

4.4.2. Lagrange step: strain, force, inertia and stable time step calculations

Background elements cut by the foreground surface require special calculations for constraint, internal and inertial forces. Cell cuts are computed by finding the intersection of foreground surface with individual background cells. The integrals (13,18) are computed numerically by using Gauss quadrature on each individual segment and boundary cell intersection and assembled into the constraint matrices $\boldsymbol{G}^b$ and $\boldsymbol{G}^f$. The cut element strain displacement matrix $\boldsymbol{\beta}^c_{e,A}$ is used to compute the internal forces $\boldsymbol{f}^{(int)}$

$$\boldsymbol{\beta}^c_{e,A} = \frac{1}{V^c_e}\int_{\Omega^c_e} \nabla N^b_A \, d\Omega = \frac{1}{V^c_e}\sum_{f\in S^c_e}\int_{\Gamma^c_{e,f}} N_A \boldsymbol{n}^f_e \, d\Gamma \qquad \boldsymbol{f}^{(int)}_A = \int_{\Omega^c_e} \nabla N^b_A \cdot \boldsymbol{\sigma}^b \, d\Omega \approx \boldsymbol{\beta}^c_{e,A} \cdot \boldsymbol{\sigma}^b_e V^c_e \tag{25}$$

where $\Gamma^c_{e,f}$ are the exposed faces of cut elements as depicted in Figure 7(a), V^c_e is the volume of the cut element and $\boldsymbol{n}^f_e$ are the face normals. For cut hexahedrals, the strain and stress are assumed constant and numerical integration over the surface in (25) accommodates *exact* patch test satisfaction. For cut tetrahedrals, the strain and stress *are* constant and matrix $\boldsymbol{\beta}^c_{e,A} = \nabla N^b_A$ on Ω_e is constant and identical to the uncut tetrahedral.

Element mass matrices are computed using cut element volumes V^c_e and density ρ^b_e [6] with all mass lumped so element nodal mass M^e_B is computed from element mass M_e

$$M_e = \rho^b_e V^c_e, \quad M^e_B = 1/n_e \, M_e \quad \text{where} \quad n_e = 8 \text{ (hexahedrals)} \quad n_e = 4 \text{ (tetrahedrals)} \tag{26}$$

[6]For elements with multiple materials (e.g. air and water), the mass of each individual species based on its volume fraction and density is summed to compute total mass (see Section 4.4.3).

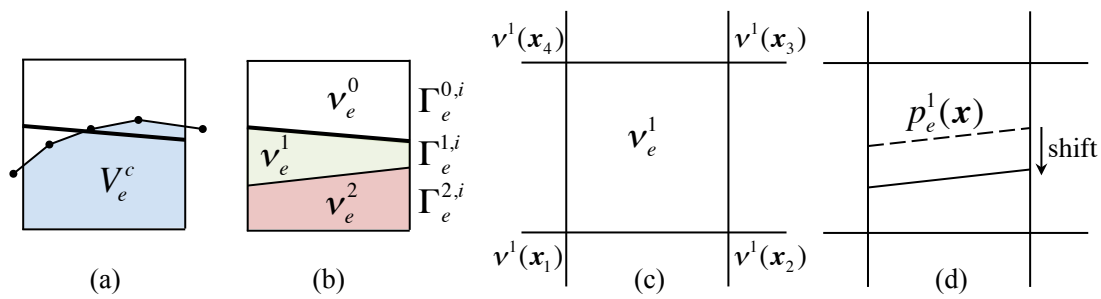

Figure 8: (a) Curved surfaces approximated by plane for calculation of areas in (25) and (29). An average normal and volume V_e^c of curved surface is used to define plane. Shaded volume V_e^c under curve can be computed using tetrahedrals and/or a signed distance function when algorithm detects corners (b) Multiple fluids species are characterized by interface reconstruction using volume fractions ν_e^k and Youngs' method to compute plane between $k = 1$ (green) and $k = 2$ (pink) fluids. Planes are then used to compute advection areas $\Gamma_e^{1,i}$ and $\Gamma_e^{2,i}$. (c) Material 1 nodal average volume fractions. (d) Plane computed from (31) is then shifted to match void and material 1 volume.

Explicit time integration is conditionally stable and the critical time is computed from maximum element frequency $\Delta t_c = \max_{e\in E}(2/\omega_{max}^e)$. The maximum frequency for a compressible fluid with bulk modulus K on a cut element volume is bounded by the Raleigh quotient

$$\omega_{max}^e \leq \frac{K\int_{\Omega_e^c}(\nabla\cdot\boldsymbol{v}^b)^2\,d\Omega}{\sum_B^{n_e} M_B^e \boldsymbol{d}_B^e\cdot\boldsymbol{d}_B^e} = \frac{K\sum_A^{n_e}(\boldsymbol{\beta}_{e,A}^c\cdot\boldsymbol{d}_A^e)^2 V_e^c}{(1/n_e)\rho_e^b V_e^c\sum_B^{n_e}\boldsymbol{d}_B^e\cdot\boldsymbol{d}_B^e} \leq c_e^2 n_e \sum_A^{n_e}||\boldsymbol{\beta}_{e,A}^c||^2 \tag{27}$$

where c_e is the sound speed. At this point, it is very important to recognize that the frequency (27) and thus *critical time step is not drastically affected by the amount of fluid in the cell*, that is when $\Omega_e^c \ll \Omega_e$. In fact, the amount of fluid in the cell has no effect for cut tetrahedral and triangular cells since $||\boldsymbol{\beta}_A^e||$ is identical for cut and full cells. So, for a right triangular cell with sides h, the frequency is $\omega_{max}^e = 2\sqrt{3}\,c_e^2/h$ regardless of the location of the cut. The amount of fluid does have a small effect on cut quad and hex cells and this effect could be bounded by applying a mean value theorem to the first volume integral in (25). For the linearized Lagrange-Lagrange coupling, stability is guaranteed by observing the critical time step given by (27) as long as the equations of motion (22) are solved to sufficient accuracy. Of course penalty methods or staggered methods could be used to avoid the linear solution required in (22), but as mentioned in Section 3.3, provably stable staggered schemes are not as easily established for partially filled cells.

4.4.3. Advection remap step: flux calculation

After the Lagrange step, the mesh is relaxed to a new configuration to avoid mesh tangling. With the Eulerian approach, the mesh is returned to the original mesh position after each step. Referring to Figure 6, this process requires mass, energy and all other cell centered variables (stress, plastic strain, etc.) to be fluxed and thus "remapped". Multiple material quantities in the background will be distinguished by the superscript $k = 1, n_{mat}$. Superscript $k = 0$ will be reserved for void although it is not a material. A simple single material case ($n_{mat} = 1$) in Figure 6(b) shows void $k = 0$ located under the yellow foreground region with fluid material $k = 1$ everywhere else. These background (typically fluid) quantities are treated as cell centered and local flux conservation is enforced using the FV method. For example, the simple first order upwind method uses the upwind cell values to compute fluxes to update the background hexahedral element

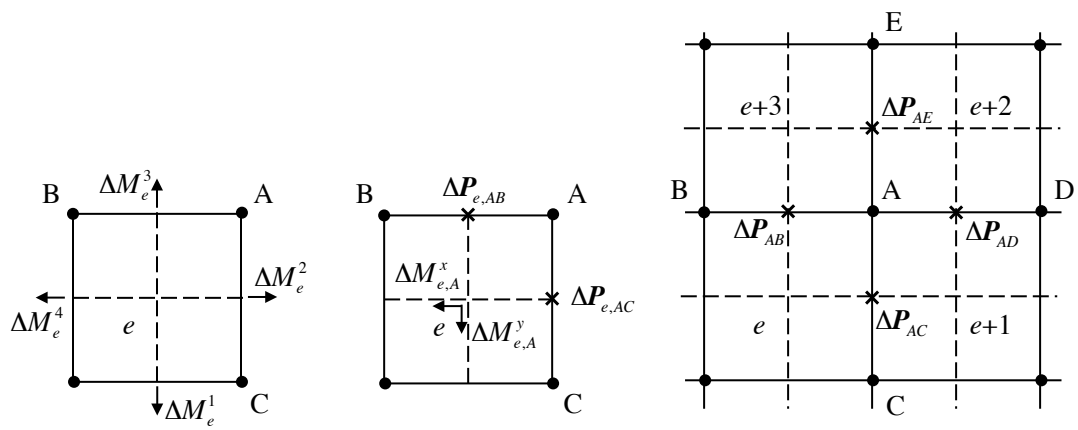

Figure 9: Dual mesh (c) used for momentum advection of node A. (a) Mass fluxes computed on faces of element e (b) Face mass fluxes are centered to element. Center mass flux is then centered on edges AB and AC to get edgewise momentum fluxes. (c) Element edge fluxes are summed $\Delta \boldsymbol{P}_{AB} = \Delta \boldsymbol{P}_{e,AB} + \Delta \boldsymbol{P}_{e+3,AB}$ to get the edge fluxes for node A

fluid mass M_e^1 and energy E_e^1

$$M_{e,n+1}^1 = M_{e,n}^1 + \sum_{i=1}^{6} \rho_n^{1,\bar{e}^i} \Delta V_{e,n+1}^{1,i} \qquad E_{e,n+1}^1 = E_{e,n}^1 + \sum_{i=1}^{6} \epsilon_n^{1,\bar{e}^i} \Delta V_{e,n+1}^{1,i} + \boldsymbol{\sigma}_e^b : \nabla \boldsymbol{v}_e^b V_e^c \tag{28}$$

$$\Delta V_{e,n+1}^{1,i} = -\Delta t \int_{\Gamma_{e,n}^{1,i}} \hat{\boldsymbol{v}}_{n+1}^b \cdot \boldsymbol{n} \, d\Gamma \quad i = 1,6 \qquad \text{volume flux} \tag{29}$$

where $\rho_n^{\bar{e}^i}$ and $\epsilon_n^{\bar{e}^i}$ are respectively the mass and energy densities at time n of the upwind element $\bar{e}^i$ of the face i. In addition, an Eulerian background mesh was assumed to compute the volume flux (29), i.e. convected velocity $\boldsymbol{c} = \hat{\boldsymbol{v}}_{n+1}^b$. Referring to Figure 7(b), $\Gamma_{e,n}^{1,i}$ in (29) is the *exposed* area of material 1 on the ith face of element e based on the t_n configuration. [7] Calculation of $\Gamma_{e,n}^{1,i}$ is key to the appropriate advection in the cut cells since no fluid flows from void cells under the foreground mesh into cut cells as demonstrated in the example of Figure 6(b) i.e. $\Delta V_e^{1,i} = 0$ since $\Gamma_e^{1,i} = 0$. Curved interfaces are approximated by planes as shown in Figure 8(a).

Multiple fluid materials ($n_{mat} > 1$) require some definition of an interface between the different fluids. So, with $n_{mat} = 2$ as in Figure 8(b), a modified Youngs' method based on the volume fractions of ν_e^1 and ν_e^2 is used to compute a plane between the fluid material 1 and 2. The volume fraction ν_e^k of material k is the ratio of the volume of material V_e^k to element volume V_e . Whereas the void volume is calculated from geometry i.e. $V_e^0 = V_e - V_e^c$, other material volumes are advected using $(30)_1$. The volume fraction is calculated in $(30)_2$ and normalized to sum to one in $(30)_3$

$$V_{e,n+1}^k = \nu_n^k V_e + \sum_{i=1}^{6} \Delta V_{e,n+1}^{k,i} \; (k = 1, n_{mat}) \qquad \hat{\nu}_e^k = V_{e,n+1}^k / V_e \qquad \nu_{e,n+1}^k = \hat{\nu}_e^k / \sum_{k=0}^{n_{mat}} \hat{\nu}_e^k \tag{30}$$

Considering the two dimensional case in Figure 8(c), material 1 volume fractions from surrounding elements are averaged to get nodal volume fractions $\nu^1(\boldsymbol{x}_j)$, $j = 1,4$. Referring to (31), the coefficients of the plane

[7] The flux (29) accounts for the mesh motion from the Lagrange step $\hat{\boldsymbol{v}}^b \Delta t$ and neglects so called corner fluxes.

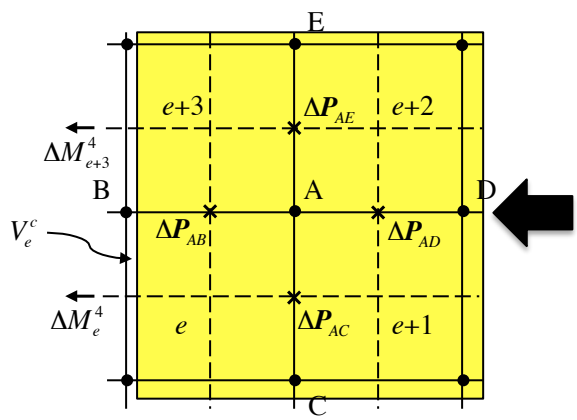

Figure 10: Background cell being "fluxed out" by yellow foreground mesh from time n to $n+1$. On the left side, elements e and $e+3$ have small fluid volumes and no vertical flow is assumed there.

p_e^1 are chosen to minimize the error function E.

$$p_e^1(\boldsymbol{x}) = ax + by + c \qquad \mathrm{E}(a,b,c) = \sum_{i=1}^{4} (p_e^1(\boldsymbol{x}_i) - (\nu^1(\boldsymbol{x}_i) - \nu_e^1))^2 \tag{31}$$

Because the value of ν_e^1 will be bounded above and below by the nodal volume fractions, the plane will intersect the element but it is important to shift the plane to capture the exact volume. Here the plane is then shifted so that the upper volume exactly contains both the void 0 and fluid material 1. The last fluid material 2 is thus located below the plane. Extension to three dimensions and three or more materials is straightforward. As with most interface reconstruction algorithms, there is some order dependence based on the material numbers that can affect the results.

Remark 1 *What distinguishes the technique here is that the void is located via geometry of the foreground surface as in Figure 8(a). The interface reconstruction (31) is only applied when more than one fluid material is used i.e. $n_{mat} > 1$. The interface reconstruction is not used to locate (approximate) the interface between the void and the fluid as in [7].*

When possible, the second order Van Leer approach [35] is used such that a combination of downwind, upwind and upwind plus one cell values are used to compute the fluxes in (28). This cannot always be achieved near the embedded mesh interface since there may be no upwind plus one cell in Ω^b, i.e. it is under the foreground mesh.

4.4.4. Advection Remap Step: momentum advection

Mesh relaxation also advects momentum locally. Because momentum is node centered, algorithms that advect it on a staggered (i.e. dual) mesh as in Figure 9 are often preferred [36]. The algorithm used here is based on the undocumented algorithm used in the ALE3D code [34] and conserves linear momentum. The algorithm is illustrated in Figure 9, which shows node A at the center of four elements e to $e+3$. The dual mesh yields a face for every edge emanating from node A in the dual cell and the algorithm computes $\Delta\boldsymbol{P}_{edge}$ along each of these edges. Figure 9(a) shows a mass flux ΔM_e^i defined on each face, i, of element e

(32). The flux is then recentered (directionally) in the element via (33), as in Figure 9(b).

$$\Delta M_e^i = \sum_k \rho^{k,\bar{e}} \Delta V_e^{k,i} \quad \text{total mass flux of all materials } k \text{ on face } i \text{ of element } e \tag{32}$$

$$\Delta M_{e,A}^x = 1/2(\Delta M_e^4 - \Delta M_e^2) \quad \Delta M_{e,A}^y = 1/2(\Delta M_e^1 - \Delta M_e^3) \quad \text{centered mass flux} \tag{33}$$

The element center based mass flux is then centered on edges and multiplied by the upwind nodal velocity to get the increment of momentum for the associated dual mesh face. For example, for edge AB, the momentum increment $\Delta \boldsymbol{P}_{e,AB}$ in Figure 9(b) is computed

$$\Delta \boldsymbol{P}_{e,AB} = 1/2\, \Delta M_{e,A}^x \hat{\boldsymbol{v}}_{AB}^b \quad \text{where} \quad \hat{\boldsymbol{v}}_{AB}^b = \begin{cases} \hat{\boldsymbol{v}}_A^b, & \Delta M_{e,A}^x \geq 0 \\ \hat{\boldsymbol{v}}_B^b, & \Delta M_{e,A}^x < 0 \end{cases} \tag{34}$$

where the factor 1/2 results from the cell centered mass flux being shared by the top and bottom edges. The process is repeated on all elements with edges associated with node A and and the results are summed to get $\Delta \boldsymbol{P}_{AB}$ and the like as in Figure 9(c). Finally, the velocity at node A is then modified to reflect the momentum update

$$\boldsymbol{v}_{A,n+1}^b = (M_{A,n} \hat{\boldsymbol{v}}_{A,n+1}^b + \Delta \boldsymbol{P}_{AB} + \Delta \boldsymbol{P}_{AC} + \Delta \boldsymbol{P}_{AD} + \Delta \boldsymbol{P}_{AE})/M_{A,n+1} \tag{35}$$

where $M_{A,n}$ and $M_{A,n+1}$ are the nodal masses computed at at node A at configuration n and $n+1$ respectively. These nodal masses are computed from the assembled element masses M_e which are updated from n to $n+1$ using mass fluxes (32). The motion of the overlapping foreground mesh can cause a cell to be "fluxed out" as in Figure 10. Node A will be the upwind node and assuming no vertical flux in elements e and $e+3$, (35) can be written

$$\boldsymbol{v}_{A,n+1}^b = (M_{A,n} \hat{\boldsymbol{v}}_{A,n+1}^b + \Delta \boldsymbol{P}_{AB})/M_{A,n+1} \tag{36}$$

$$= (M_{A,n} + \Delta M_e^4/4 + \Delta M_{e+3}^4/4) \hat{\boldsymbol{v}}_{A,n+1}^b / (M_{A,n} + \Delta M_e^4/4 + \Delta M_{e+3}^4/4) = \hat{\boldsymbol{v}}_{A,n}^b \tag{37}$$

which is valid when $M_{A,n+1} \to 0$ but not necessarily stable. To avoid such problems a very small void mass is added to (26) $M_e = \rho_e^b (V_e^c + \epsilon(V_e - V_e^c))$. All problems in Section 6 were run with $\epsilon = 10^{-4}$ or less. By construction, the algorithm conserves linear momentum except when void mass is used, although the effects appear small. Extension to non-structured and three dimensional meshes is straightforward. The algorithm described is a first order upwind method but a velocity gradient along with monotonicity constraints is used to improve the accuracy when applied away from embedded mesh boundaries.

5. Algorithm

The following box summarizes the code flow. Background materials are given $k = 1, n_{mat}$ with $k = 0$ being void.

$t_n = 0$
Compute cut cells, void cells using expensive search algorithms
Compute V_e^c, $\boldsymbol{\beta}_{e,A}^c$, $\boldsymbol{G}^b$, $\boldsymbol{G}^f$ on cut cells
do while $t_n < t_{term}$
  $t_n = t_n + \Delta t$
  * ***Perform Lagrange Step*** *\
  Compute and assemble forces (use $\boldsymbol{\beta}_{e,A}^c$ and V_e^c on cut background elements)
  Solve iteratively equations of motion (22) for $\hat{\boldsymbol{v}}^b$, $\boldsymbol{v}^f$ and $\boldsymbol{\lambda}$
  Update geometry
  Update stress $\boldsymbol{\sigma}_{e,n}^k$ based on incremental strain and constitutive laws
  * ***Perform Advection Step*** *\
  Relax mesh
  Get advected areas $\Gamma_{e,n-1}^{k,i}, k = 1, n_{mat}$ from volume fractions $\nu_{e,n}^k$ (Sect. 4.4.3)
  Get time t_n volume flux (29) $\Delta V_{e,n}^{k,i}$ using $\Gamma_{e,n-1}^{k,i}$
  Get cut cells, void cells using cheaper incremental search algs. (Sect.4.1)
  Get V_e^c, $\boldsymbol{\beta}_{e,A}^c$, $\boldsymbol{G}^b$, $\boldsymbol{G}^f$ on cut cells
  Update volume fractions using V_e^c and (30)
  Mass and energy are updated using (28) or the modified Van Leer version
  Advect stress, history variables: divide quantity by element volume and advect like mass density
  Advect momentum to get $\boldsymbol{v}_n^b$ from $\hat{\boldsymbol{v}}_n^b$ as in (35)
end while

6. Examples and Implementation Aspects

The following examples will be used to demonstrate convergence and robustness of the proposed embedded mesh methods. The first two examples are elasticity problems using Lagrange-Lagrange coupling and are solved using implicit time integration with a direct solver. They illustrate the mesh locking that occurs when foreground multipliers [8] are used and the optimal convergence of the proposed piecewise constant multipliers and nodal multipliers from Sections 4.3.1 and 4.3.2. The remaining examples are coupled Euler-Lagrange and ALE-Lagrange using the explicit time integration two step ALE method of Section 4.4. Here comparison to conforming ALE results is used to demonstrate convergence.

The piecewise constant multipliers require a stabilization term. For elasticity $\gamma_f = \frac{\alpha}{\rho c^2}$ is used where α is a scalar parameter, and c and ρ are the background material sound speed and density. These quantities were not available to the explicit ALE coupling code. Instead a scheme that replaces $\gamma_f h A_f$ in (13) with a combination of diagonal terms (24) [9] defined on adjacent elements of face f i.e.

$$\gamma_f h A_f = \alpha \sqrt{D_{e_{f1}} D_{e_{f2}}} \Delta t^2 \quad \forall f \in F^c \tag{38}$$

where F^c are element pairs of cut faces defined in (12). Results in the following examples were computed with $\alpha = 0.1$, and convergence tolerance on the conjugate solver was $1 \times 10^{-9} ||\boldsymbol{r}_0||$. All coupled Euler-Lagrange examples use units of cm, gm, μs, Mbar, had initial (ambient) pressure conditions of $p = 1 \times 10^{-6}$ Mbar and included models for air, C4, TNT and aluminum. The ideal gas law was used for air $p = (\gamma - 1)\bar{\epsilon}/V$, $\gamma = 1.4$ where $\bar{\epsilon}$ is the energy density (energy per unit initial volume) and V is the relative volume. The following

[8] i.e. $\boldsymbol{\lambda} = \sum_A N_A^f \boldsymbol{\lambda}_A$ where N^f are foreground surface FE shape functions
[9] For constants, Lagrange multipliers can be indexed by element $e \in E^c$ so A is replaced by e in (24)

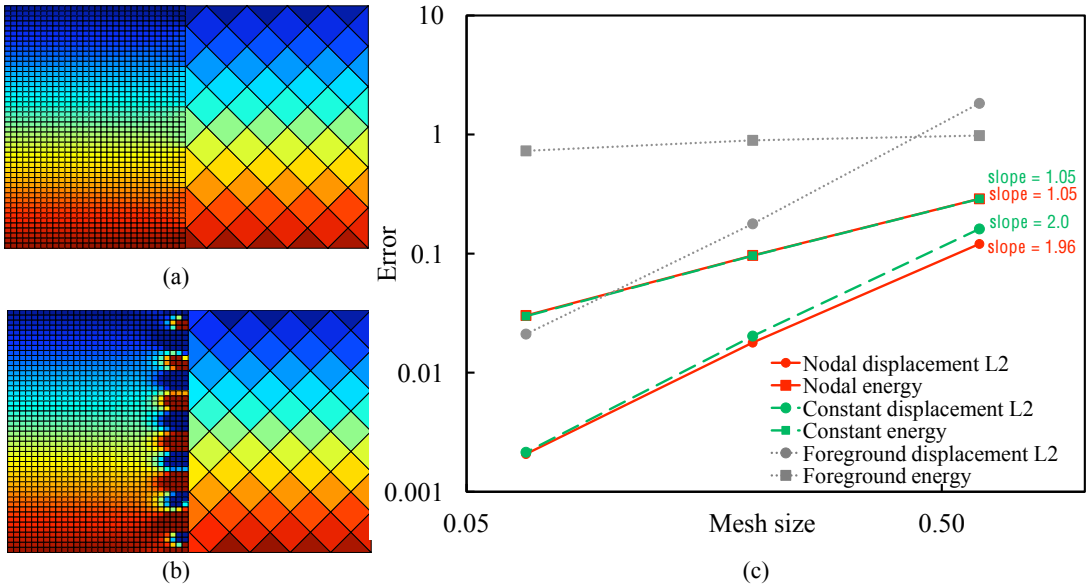

Figure 11: (a) Beam bending results using nodal multipliers (b) Beam bending results using foreground multipliers (c) L_2 displacement error and energy error plot from nodal, constant and foreground Lagrange multiplier results.

JWL equations of state [34] were used for the explosives:

$$p = A(1 - \frac{\omega}{R_1 V})\exp(-R_1 V) + B(1 - \frac{\omega}{R_2 V})\exp(-R_2 V) + \frac{\omega}{V}\bar{\epsilon} \tag{39}$$

$$\text{C4:} \quad \{A, B, R_1, R_2, \omega, \bar{\epsilon}_0, \rho_0\} = \{6.0977, 0.1295, 4.5, 1.4, 0.25, 0.090, 1.60\} \tag{40}$$

$$\text{TNT:} \quad \{A, B, R_1, R_2, \omega, \bar{\epsilon}_0, \rho_0\} = \{4.5138, 0.1058, 4.5, 1.5, 0.25, 0.078, 1.63\} \tag{41}$$

where ρ_0 is initial relative density, $\bar{\epsilon}_0$ is inital energy density and no deviatoric strength was included. In addition, no burn model was used such that entire volume of explosive was lit at $t = 0$. The aluminum had a shear modulus $G = 0.267$Mbar, Poisson's ratio 0.34 and power law plasticity type yield $\sigma_y = 0.0042(1 + 965\varepsilon)^{0.1}$ for small strain at 25°C where ε is plastic strain.

6.1. Linear elastic beam

Referring to Figure 11(a), Lagrange-Lagrange type coupling is used to couple a stiff, fine mesh ($E = 1000$, $\nu = 0.3$) with soft, coarse mesh ($E = 1$, $\nu = 0.3$). The two meshes form a short beam with $L = 6$ and $h = 3$ and meet at the center ($x = 3$). A bending moment is applied on the right end ($x = 6$) and the boundary conditions $u(0, y) = v(0, 0) = 0$ are applied on the left ($x = 0$). The coarse mesh is mostly quadrilaterals with some triangles and was chosen to highlight the locking exhibited with foreground multipliers. Bending stress σ_{yy} is shown in Figure 11 where it is seen that nodal multiplier stress in 11(a) is smooth but the foreground multiplier stress in 11(b) is highly oscillatory. Locking occurs because the background mesh is diagonal to the foreground surface producing the effect shown in Figure 2(b). The L_2 displacement and energy error in Figure 11(c) are optimal for both constant and nodal background multipliers and suboptimal for foreground type.

6.2. Linear elastic stiff inclusions

This example in Figure 12(a) incorporates stiff inclusions in a soft matrix under given loading conditions. Figure 12(b) shows pressure results on a very fine conforming mesh for comparison to embedded meshes. Figure 12(c) demonstrates poor pressure results for foreground multipliers and Figure 12(d) demonstrates

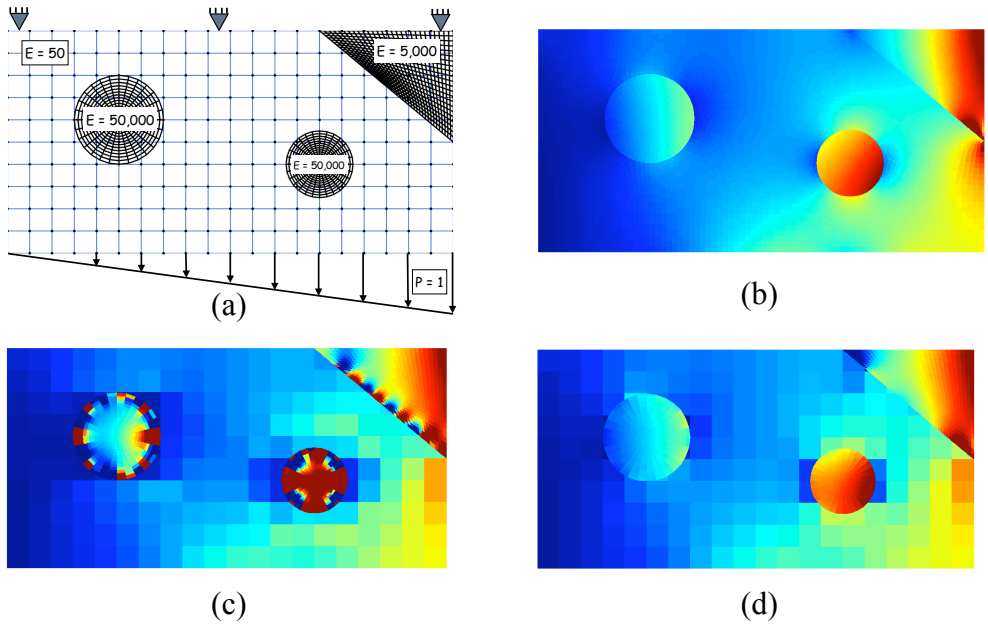

Figure 12: (a) Stiff inclusions in soft matrix (b) pressure result on a very fine conforming mesh (c) results from foreground multipliers on coarse mesh (d) results from nodal background multipliers on coarse mesh.

good pressures on a coarse mesh for nodal multipliers. Using the fine conforming mesh as reference, L_2 displacement and energy error is plotted in Figure 13. Again background type multipliers converge optimally whilst foreground show poor energy norm convergence due to stress oscillations.

6.3. Piston problem

A piston problem is used to check how well the proposed coupled Euler-Lagrange approach conserves energy and behaves over long times. An air filled $1 \times 1 \times 2$ cm rectangular cylinder contains a 0.4 cm aluminum piston (Fig 14). At $t = 0$ the piston is located at the center and has initial velocity of 5 km/s. The gas is fully contained and the piston should oscillate at ≈ 300 hz indefinitely since there is no inherent damping. A two dimensional 1×2 cm model with 0.4 cm wide piston at a $5:1$ slant was also considered.

The straight case piston velocity results from nodal Lagrange multiplier and conforming mesh ALE in Figure 14(d) show very good agreement. At the last time in the plot $t = 1 \times 10^5 \mu s\,(0.1s)$, the piston has a kinetic energy loss of 1.2% after 4×10^5 time steps [10] for both constant and nodal multipliers and 0.7% loss for the conforming mesh. At $t = 1s$, 4×10^6 time steps, the energy loss is 10% and 6% for embedded mesh and conforming mesh respectively. These energy losses are mostly due to the bulk viscosity and the known lack of energy conservation of the advection algorithms from Sections 4.4.3 and 4.4.4. The latter is more prevalent for the embedded mesh since the background mesh is Eulerian (fixed) whilst the ALE mesh is nearly Lagrangian for this problem and requires little, to no advection. A small energy loss may also be due to the void mass introduced in Section 4.4.4.

The slanted piston velocity results shown in Figure 14(e) were generated using constant Lagrange multipliers and the results are visually identical to the straight piston as expected. The energy loss after $t = 0.1\,s$ and $t = 1\,s$ is 0.27% and 1.6% respectively; less than the previous analysis due to the finer background mesh.

[10] Fixed time steps of $0.25\mu s$ were used for both embedded grid and conforming mesh to make the comparison.

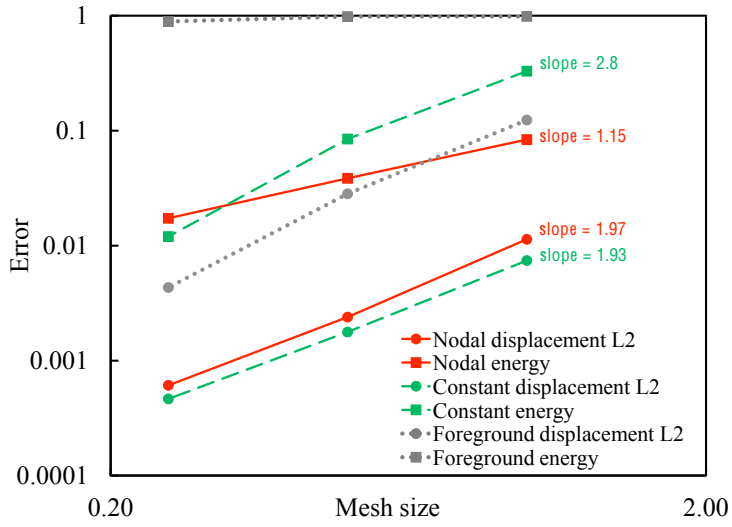

Figure 13: Plot of L_2 displacement and energy error for stiff inclusions.

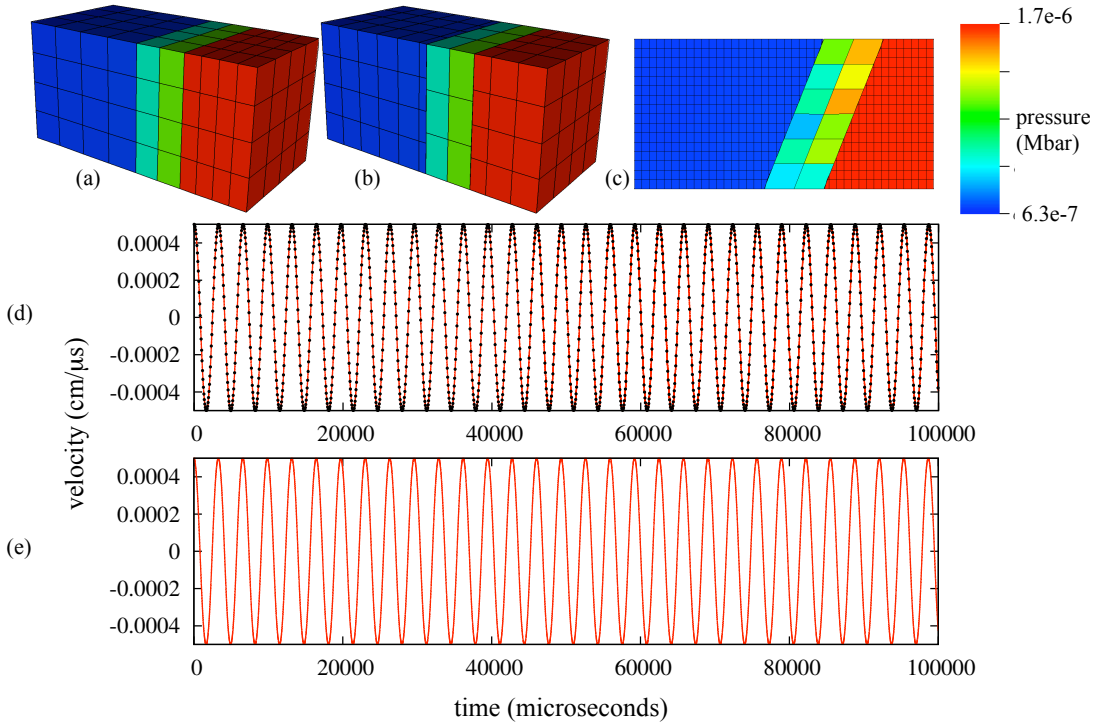

Figure 14: Piston problem: (a) Conforming mesh and (b) embedded mesh for three dimensional piston and (c) two dimension slanted piston. Pistons are shown at full compression i.e. $v \approx 0$. (d) Comparison of embedded (red line) and conforming (black dot) mesh velocities out to 0.1 s. (e) Velocity for slanted mesh.

6.4. Two dimensional explosively driven plate

A mesh convergence study is performed on a two dimensional blast problem using a conforming mesh as the reference solution and results from Euler-Lagrange coupling (Eulerian background mesh) and ALE-Lagrange coupling (moving background mesh) are considered. Referring to Figure 15, an 8 cm diameter charge of C4 is placed at a 13 cm standoff from a 5 cm thick aluminum plate which is surrounded by air. Outflow boundary conditions are applied at the left, right and top of the mesh and a symmetry condition is applied at the bottom. Figure 15 also displays mesh motion for the ALE-Lagrange coupling. Here, the mesh movement attempts to remain nearly Lagrangian and also adapt the mesh to follow the shock wave. In Figure 16, coupled Euler-Lagrange displacement and pressure are compared at different time states and velocity and pressure results are plotted in Figures 17(a,b). Three meshes were used in the convergence study and the L_2 norm of the difference in displacement at $t = 5000\,\mu$s between embedded and conforming mesh ($\int_{\Omega_f} ||\boldsymbol{d}_{emb} - \boldsymbol{d}_{conf}||\, d\Omega$) is shown in Figure 17(c). Differences between nodal and constant multipliers were neglegible in the comparisons. For example, displacement in results from constant and nodal multipliers was less than 0.1% different with coupled Euler-Lagrange. Overall, the conforming and embedded mesh results compare well and appear to converge to a common solution at a reasonable rate.

6.5. Three dimensional explosively driven plate

Referring to Figure 18, a 12 lb TNT charge is placed 5 in under compacted soil and 23 in from the bottom of a 2 in thick aluminum plate with dimensions (132×76) in plate. The soil is modeled using a porous geological material model. The sequence of deformation states are shown in Figures 18 and 19(a). The plot of velocity versus time at the center and (far) outside corner of the plate is shown in Figure 19(b). Here embedded mesh and conforming results are seen to compare well and the constant and nodal multiplier results are again virtually the same.

6.6. Blast on structural shell pipe

A mesh convergence study is performed on the two dimensional blast problem shown in Figure 20. Here, an 8 cm diameter C4 charge is placed in air at an 4 cm standoff from a 2 mm thick, 12 cm radius aluminum pipe modeled with structural shells. Void (i.e. no material) is contained in the pipe. Four different mesh sizes were considered, the coarse mesh results of the nodal multiplier embedded mesh are shown in Figure 20(top). As with the other coupled Euler-Lagrange example problems, constant and nodal multiplier results are nearly indiscernible. In Figure 20(bottom), the shell profile for the embedded mesh (black) and conforming mesh (grey) are compared. At time $t = 180\,\mu$s, some difference in the profile is recognized and the conforming mesh analysis terminates at time $t = 127\,\mu$s due to mesh tangling as seen in Figure 21(a). That is, the mesh relaxation could not reconcile the "bubble" formed in the side of the shell. The displacement of the bottom, left node resulting from the finest mesh is plotted in Figure 20(b) and the difference in this displacement for embedded and conforming mesh at time $t = 120\,\mu$s is plotted for four different mesh sizes in Figure 21(c) where it is seen to converge to a common solution.

6.7. Three dimensional shell pipe and cylinder model

The last examples demonstrate the robustness of the proposed embedded mesh approaches. The first example in Figure 22 is a three dimensional version of the pipe in Section 6.6 with pipe length $L = 40$ cm. The second example has a 5 cm spherical C4 charge in a cylinder with dimensions $L = 40, R_i = 10$ and thickness $t = 3$ cm. The ends of the cylinder ($z = \pm 20$) have a 1 cm thick slot that extends 6 cm in the x and y directions off the axis of symmetry. The amount of deformation occurring in these problems would be difficult for a conforming ALE approach. In addition, both figures show no discernible penetration of the gasses into the void regions and the void calculated on the background grid effectively matches the interior of the solid foreground mesh. A conforming mesh of the slotted pipe is difficult to build but is relatively simple for the embedded mesh method.

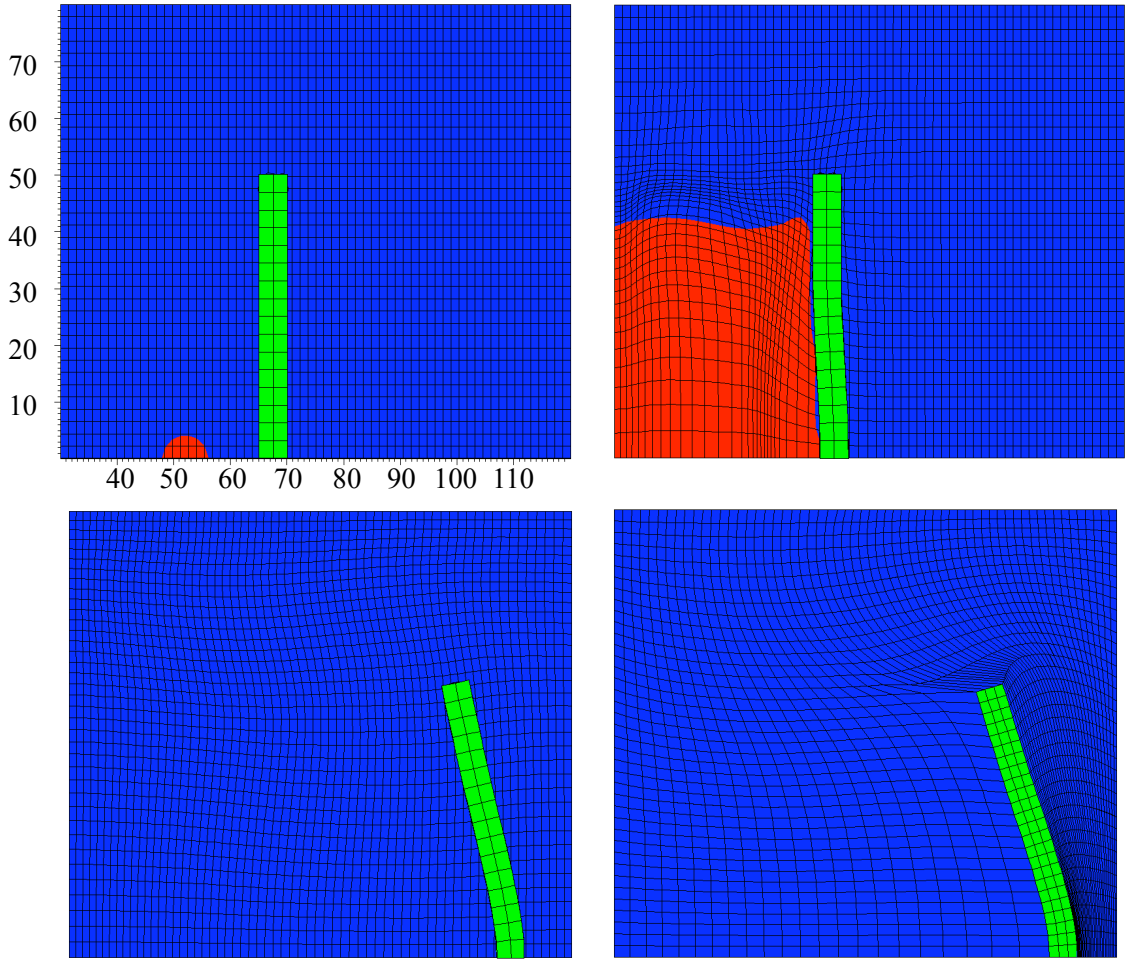

Figure 15: (a) Two dimensional coarsest mesh model of C4 placed in front of aluminum plate surrounded by air. Dimension on axes are in centimeters. (b) Coupled ALE-Lagrange result from constant multipliers at time $t = 120\,\mu$s. Final position ($t = 5000\,\mu$s) for (d) coupled ALE-Lagrange and (c) conforming mesh.

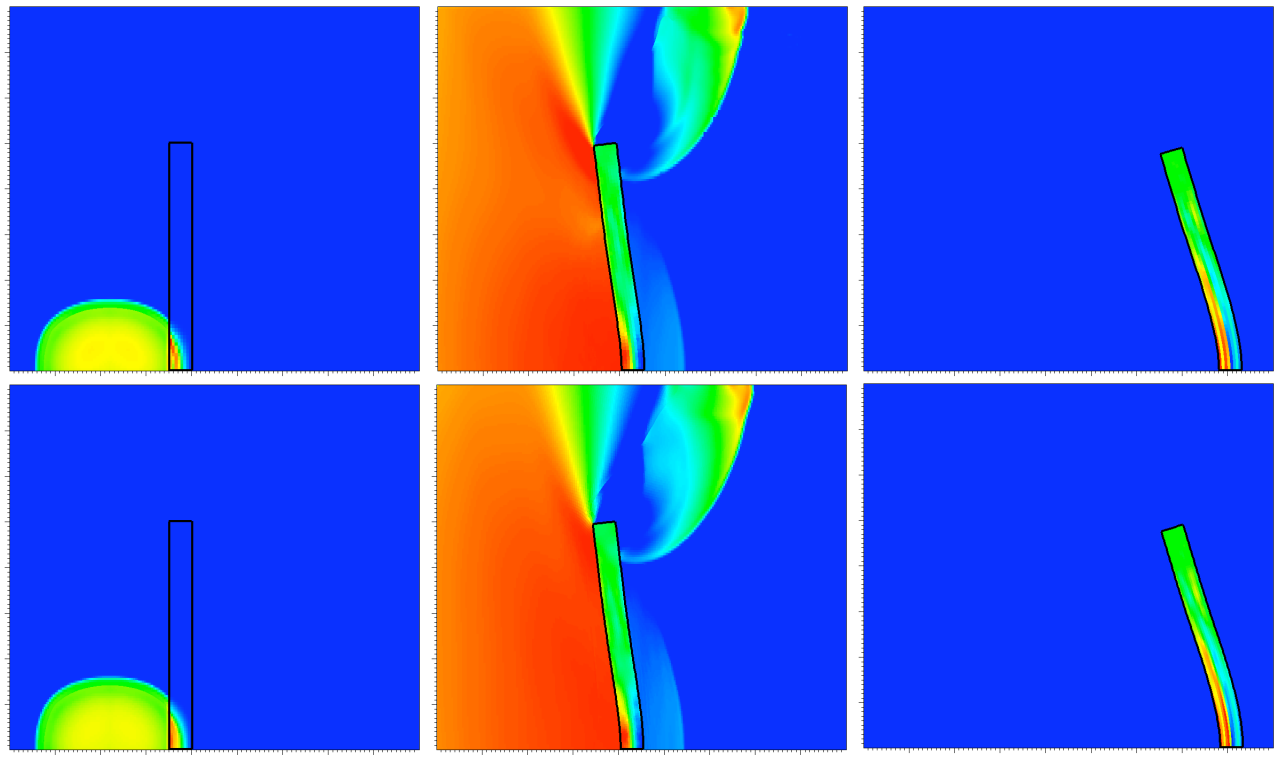

Figure 16: Pressure results from (top) coupled Euler-Lagrange with nodal multiplier and (bottom) ALE conforming mesh at different time states.

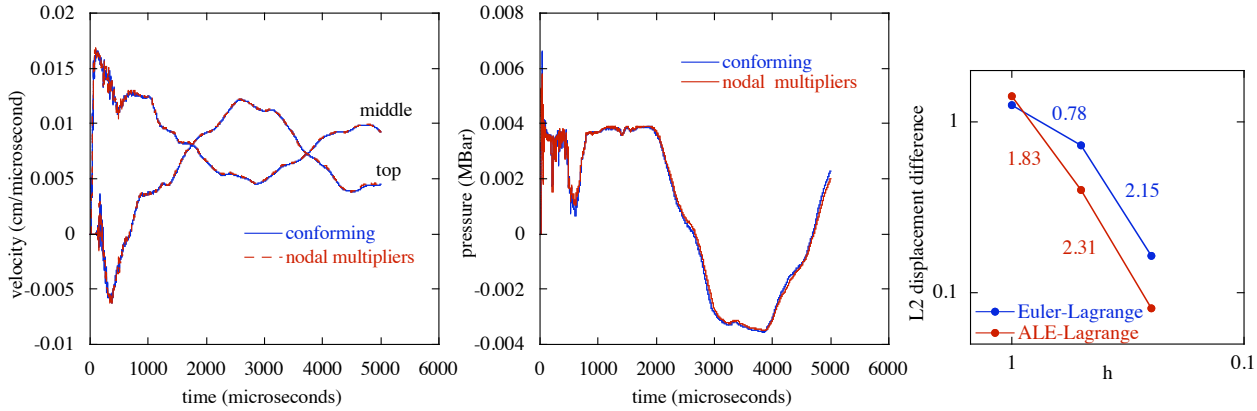

Figure 17: Time history results from coupled Euler-Lagrange with nodal multipliers compared to conforming on finest mesh for (a) displacement at top (left) and center (left) nodes and (b) for pressure at center-left element in plate. (c) L_2 norm of displacement difference for coupled Euler-Lagrange with nodal multipliers and coupled ALE-Lagrange with constant multipliers.

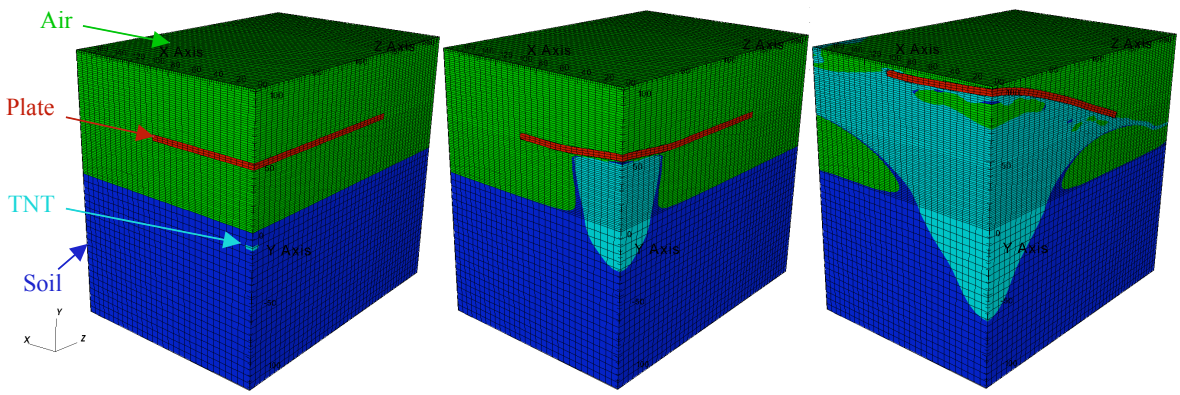

Figure 18: (a) Model of buried TNT mine in soil. (b) Deformation at $t = 1080\,\mu$s (c) Deformation at $t = 12000\,\mu s$.

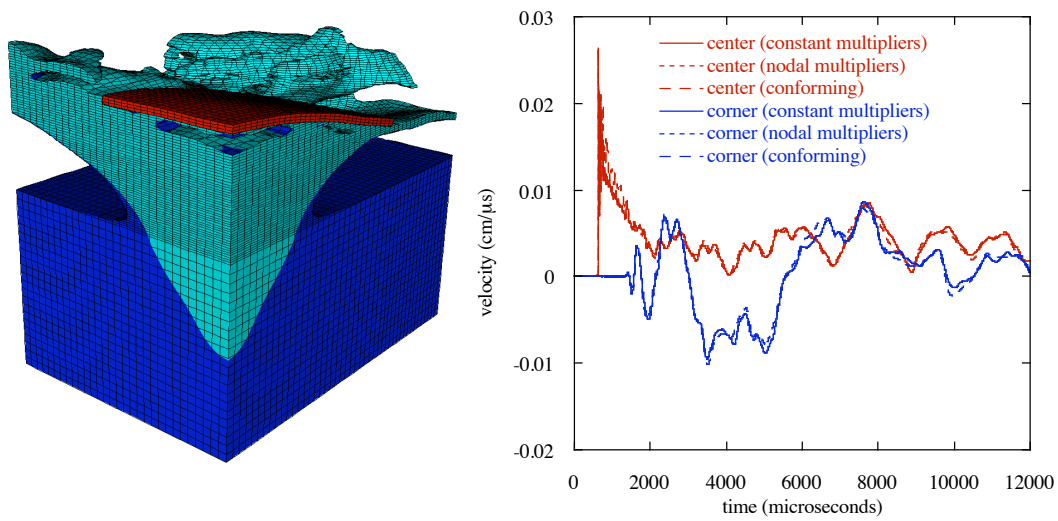

Figure 19: (a) Deformation at $t = 7240\,\mu$s with air (green) not shown. TNT gas flows to backside of plate. (b) Plot of velocity of center and far back corner of plate versus time for embedded grid multiplier methods and conforming mesh. Constant and nodal multiplier results ostensibly overlap.

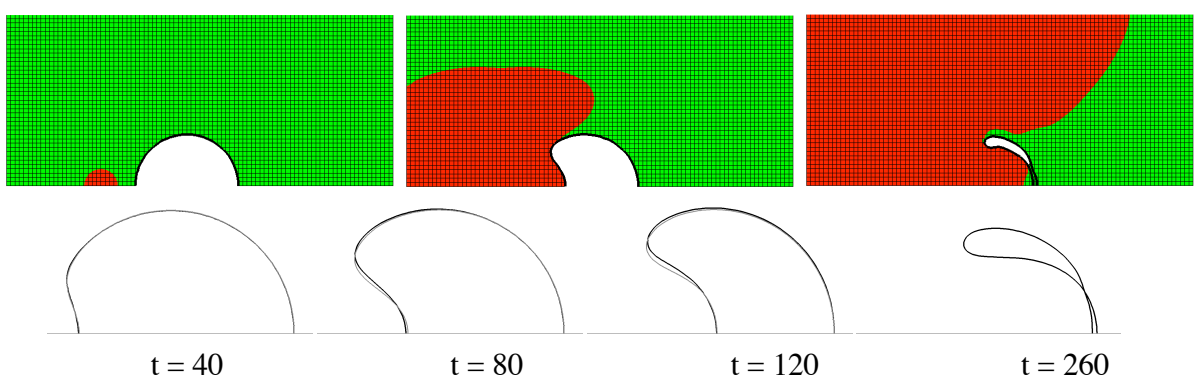

Figure 20: (top) Blast on coarse embedded mesh. (bottom) Pipe profile of finest embedded mesh in black and conforming mesh in grey. Conforming mesh analysis fails due to tangling soon after $t = 120\,\mu$s. No self-contact was used in this problem; hence, the penetration of the shell with itself.

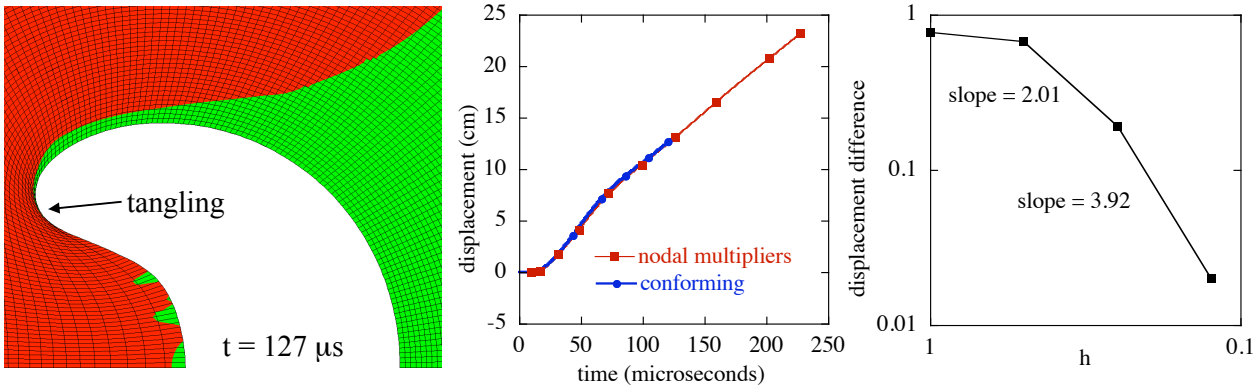

Figure 21: (a) Conforming mesh analysis terminates at time $t = 127\,\mu$s due to tangling. (b) Displacement of bottom, left shell node versus time for finest mesh. (c) Difference in displacement between embedded and conforming mesh at time $t = 120$ of bottom, left node.

7. Summary and Discussion

An embedded mesh method that uses Lagrange multipliers defined on the background discretization was proposed. Two multiplier approaches were considered and both provided nearly identical results in most problems. The stabilized, piecewise constant multiplier originally developed for the embedded Dirichelet problem in [26] was applied to the embedded mesh problem. The proposed nodal multiplier approach also performed well in the examples and the (near) orthogonalization in (20) accelerated convergence of the CG solution scheme. An approach that uses a single stress point (25) for the cut cells was proposed and issues regarding time steps of cut cells were addressed. Implementation within a two step ALE approach was detailed. An explicit time integration scheme with monolithic fluid-structure coupling necessitated a CG iterative solution for the Lagrange multipliers. Void region in cut cells was computed via computational geometry; this led to a modification of the Young's interface reconstruction scheme when multiple fluids are on the background mesh. Aspects involving modification of the ALE advection scheme were also discussed. A number of examples demonstrating convergence and robustness for both Lagrange-Lagrange, Euler-Lagrange and ALE-Lagrange coupling were presented. Overall, the proposed algorithms, novel in their application to embedded mesh methods, and the relatively extensive review constitute the original contributions of this work.

Additional future work is as follows. The nodal multiplier approach warrants further study, e.g. a numerical *inf-sup* test. For shell structures, the current implementation considered a single wetted surface, i.e. fluid on one side of the surface and void on the other. Implementation for a doubly wetted surface would require additional degrees of freedom similar to XFEM to capture jumps in pressure and velocity across the shell surface. Regarding the CG solver, the examples were performed with tight convergence tolerances but results using very loose tolerance appeared to be very good. Nonetheless, extension to staggered or some other partitioned solution scheme that obviates the iterative solve will be considered. Such schemes are standard for conforming mesh or mismatched mesh fluid/structure interation but are more difficult for the embedded mesh approach due to the cut cells.

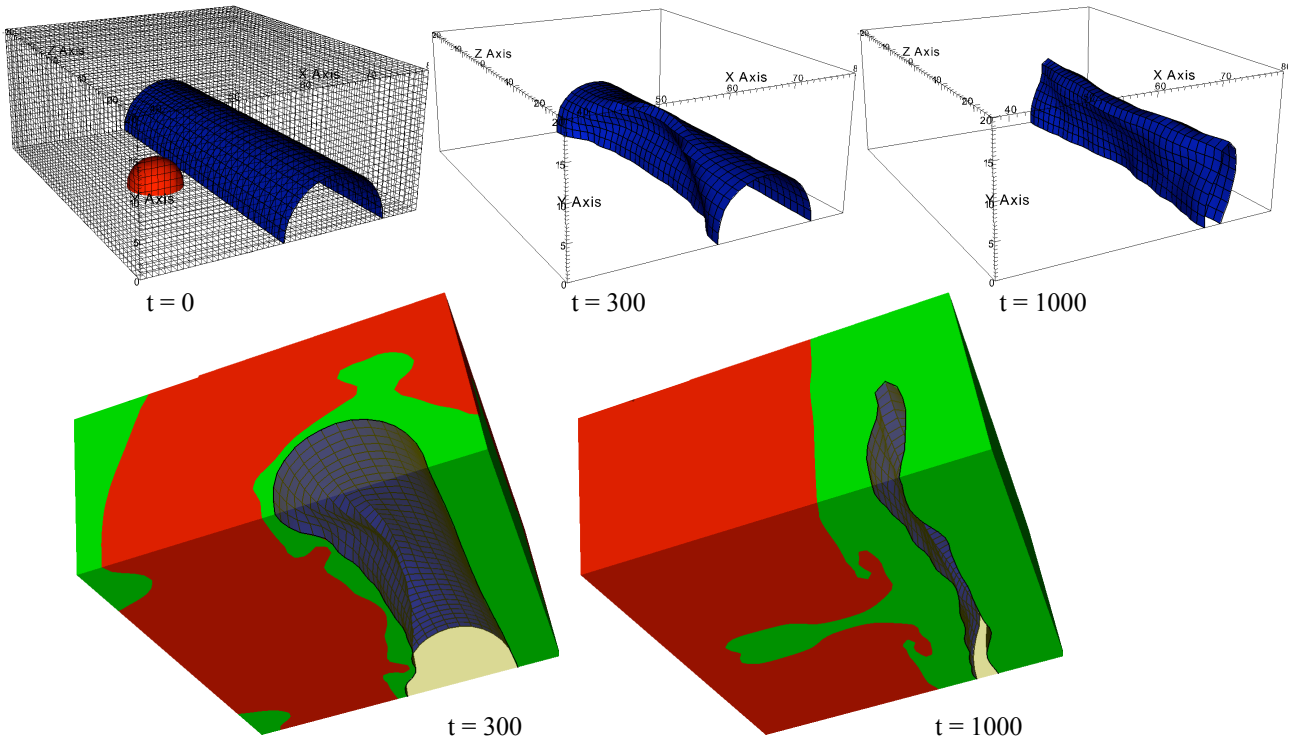

Figure 22: (top) Pipe with spherical charge and sequence of deformation states (bottom) View of same deformation from bottom with constituents: TNT (red), air (green) and void (transparent yellow).

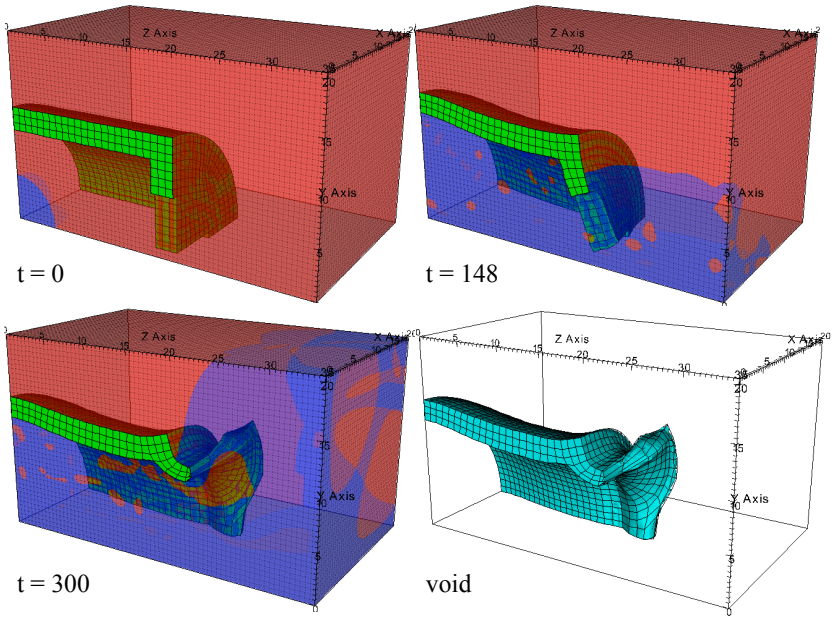

Figure 23: (a-c) Eighth symmetry model of cylinder with slot on ends where gas escapes.(d) Void background material with solid mesh superposed. Here it is seen that the void on the background grid matches well with the interior of the solid foreground mesh.

References

[1] WF Noh. A time-dependent, two space dimensional, coupled eulerian-lagrangian code. *Methods in Computational Physics*, 1964.

[2] DJ Benson. An efficient, accurate, simple ale method for nonlinear finite element programs. *Computer Methods in Applied Mechanics and Engineering*, 72:305–350, 1989.

[3] R Glowinski, TW Pan, TI Hesla, and DD Joseph. A fictitious domain approach to the direct numerical simulation of incompressible viscous flow pas moving rigid bodies: application to particulate flow. *Journal of Computational Physics*, 169:363–426, 2001.

[4] A Hansbo, P Hansbo, and MG Larson. A finite element method on composite grids based on Nitsche's method. *ESAIM: Mathematical Modeling and Numerical Analysis*, 37:209–225, 2003.

[5] L Zhang, A Gerstenberger, XD Wang, and WK Liu. Immersed finite element method. *Computer Methods in Applied Mechanics and Engineering*, 193(21-22):2051–2067, 2004.

[6] DJ Benson and Okazawa S. Contact in a multi-material eulerian finite element formulation. *Computer Methods in Applied Mechanics and Engineering*, 193:4277–4298, 2004.

[7] N Aquelet, M Souli, and Olovsson L. Euler Lagrange coupling with damping effects: Application to slamming problems. *Computer Methods in Applied Mechanics and Engineering*, 195:110–132, 2006.

[8] A Legay, J Chessa, and T Belytschko. An eulerian-lagrangian method for fluid-structure interaction based on level sets. *Computer Methods in Applied Mechanics and Engineering*, 195:2070–2087, 2006.

[9] H Wang and T Belytschko. Fluid-structure interaction by the discontinous-galerkin method for large deformations. *International Journal for Numerical Methods in Engineering*, 77:30–49, 2009.

[10] A Gerstenberger and WA Wall. An embedded Dirichlet formulation for 3D continua. *International Journal for Numerical Methods in Engineering*, 82:537–563, 2010.

[11] JD Sanders and MA Puso. An embedded mesh method for treating overlapping finite element meshes. *International Journal for Numerical Methods in Engineering*, 2011. in press.

[12] *LS-DYNA users's manual - version 971.* Livermore Software Technology Corp. (LSTC), Livermore, CA, 2007.

[13] *ABAQUS user's manual - version 6.10.* Dessault Systèmes Simulia Corp., Pawtucket, RI, 2009.

[14] AJ Lew and GC Buscaglia. A discontinuous-galerkin based immersed boundary method. *International Journal for Numerical Methods in Engineering*, 76(4):427–454, 2008.

[15] E Bechét, N Moës, and B Wohlmuth. A stable Lagrange multiplier space for stiff interface conditions within the extended finite element method. *International Journal for Numerical Methods in Engineering*, 78:931–954, 2009.

[16] T Belytschko, B Moran, and WK Liu. *Nonlinear Finite Elements for Continua and Structures.* John Wiley & Sons, 2000.

[17] RP Fedkiw, Aslam T, Merriman B, and Osher S. A non-oscillatory eulerian approach to interfaces in multimaterial flows (the ghost fluid method. *Journal of Computational Physics*, 152:457–492, 1999.

[18] M Arienti, Hung P, Morano E, and Shepherd JE. A level set approach to eulerian-lagrangian coupling. *Journal of Computational Physics*, 185:213–251, 2003.

[19] R Deiterding, R Radovitzky, SP Mauch, L Noels, JC Cummings, and DI Meiron. A virtual test facility for the efficient simulation of solid material response under strong shock and detonation wave loading. *Engineering with Computers*, 22:325–347, 2006.

[20] AB Wardlaw, JA Luton, JR Renzi, KC Kiddy, and RM McKeown. The Gemini coupled hydrocode. Technical Report IHTR 2500, Indian Head Division Naval Surface Warfare Center, 2003.

[21] CS Peskin. Numerical analysis of blood flow in the heart. *Journal of Computational Physics*, 25:220–252, 1977.

[22] JL Steger and JA Benek. On the use of composite grid schemes in computational aerodynamics. *Computer Methods in Applied Mechanics and Engineering*, 64:301–320, 1987.

[23] SW Attaway, RL Bell, CT Vaughan, SP Goudy, and KB Morrill. Experiences with coupled codes for modeling explosive blast interacting with a concrete building. *Proceedings of the Workshop on Coupled Eulerian and Lagrangian Methodologies*, 2001.

[24] GC Bessette, CT Vaughan, and RL Bell. Zapotec: A coupled Euler-Lagrange program for modeling earth penetration. Technical Report SAND2002-3679C, Sandia National Laboratories, 2001.

[25] J Nitsche. Über ein Variationsprinzip zur Lösung von Dirichlet-problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, 36, 1971.

[26] E Burman and P Hansbo. Fictitious domain finite element methods using cut elements: I. a stabilized lagrange multiplier method. *Computer Methods in Applied Mechanics and Engineering*, 199:2680–2686, 2010.

[27] FPT Baaijens. A fictitious domain/mortar element method for fluid-structure interaction. *International Journal for Numerical Methods in Engineering*, 35:743–761, 2001.

[28] DL Youngs. D. l. youngs. time-dependent multi-material flow with large fluid distortion. In *Numerical Methods for Fluid Dynamics*, pages 273–285. Academic Press, 1982.

[29] JD Sanders, TA Laursen, and MA Puso. A Nitsche embedded mesh method. *Computational Mechanics*, 2011. in press.

[30] NJ Carpenter, RL Taylor, and MG Katona. Lagrange constraints for transient finite element surface contact. *International Journal for Numerical Methods in Engineering*, 32:103–128, 1991.

[31] Martin Hautefeuille, Chandrasekhar Annavarapu, and John E. Dolbow. Robust imposition of dirichlet boundary conditions on embedded surfaces. *International Journal for Numerical Methods in Engineering*, pages n/a–n/a, 2011.

[32] B.I. Wohlmuth. A mortar finite element method using dual spaces for the Lagrange multiplier. *Society for Industrial and Applied Mathematics*, 38:989–1012, 2000.

[33] DJ Benson. Computational methods in lagrangian and eulerian hydrocodes. *Computer Methods in Applied Mechanics and Engineering*, 99:235–394, 1991.

[34] A. Nichols. Users manual for ale3d. Technical Report UCRL-SM-404490, Lawrence Livermore National Laboratory, 2009.

[35] RJ Leveque. *Finite Volume Methods for Hyperbolic Problems*. Cambridge University Press, New York, NY, 2002.

[36] DJ Benson. Momentum advection on a staggered mesh. *Journal of Computational Physics*, 100:143–162, 1992.

Prepared by LLNL under Contract DE-AC52-07NA27344