INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2009; **00**:1–18 *Prepared using* fldauth.cls *[Version: 2002/09/18 v1.01]*

A super-sampled projection method for level-set construction in fluid-solid interaction problems

D. Gregory Tipton[1]* , Mark A. Christon[2], Marc S. Ingber[3]

[1] *Sandia National Laboratories, Albuquerque, New Mexico*
[2] *CTO Office, Dassault Sysèmes SIMULIA, Providence, Rhode Island*
[3] *Department of Mechanical Engineering, University of New Mexico, Albuquerque, New Mexico*

SUMMARY

This paper considers the issues central to fast level-set construction for the general treatement of moving interfaces in coupled fluid-solid interaction problems where the Lagrangian solid experiences large bulk motion. The central idea is based on a super-sampled $\mathcal{L}^2$ projection, that in conjunction with a Lagrangian interface position, permits rapid identification of the solid interface in the fluid mesh and enables the imposition of boundary conditions for the fluid. A series of convergence studies are presented in terms of numerical quadrature and mesh refinement to illustrate the effectiveness of the super-sampled projection on unstructured grids. The extraction of the interface location based on distance functions is compared to the super-sampled projection method. It is shown that the extraction of an interface location based on a zero level-set converges as $O(h)$ when compared to the exact interface location – suggesting that the availability of a Lagrangian interface description is always preferred. Copyright © 2009 John Wiley & Sons, Ltd.

KEY WORDS: Fluid-solid interaction; level-set; super-sampled projection; Galerkin; finite element; shock loading; interface conditions

1. INTRODUCTION

This work was motivated by the need for a fast level-set construction technique for use in fluid-solid interaction (FSI) problems where separate and distinct, but overlapping unstructured meshes for the fluid and solid domains are used. The distinct overlapping meshes for the fluid and solid domains permits the treatment of FSI problems where there is large relative motion between the fluid and solid, e.g., projectiles moving in a fluid. Although not the focus of this paper, the ultimate goal for this work was aimed at shock-body interactions where complex geometries dictate the use of unstructured grids.

Coupled fluid-solid systems interact at their common "wetted" interface where conjugate boundary conditions, that couple the fluid and solid, are applied. In computations where the

*Correspondence to: D. Gregory Tipton, Sandia National Laboratories, Albuquerque, NM. Email: dgtipto@sandia.gov

Received 15 June 2009
Revised xxxxxx
Accepted xxxxxx

solid exhibits large deformation and/or rigid-body motion, the fluid-solid interface, must be identified and tracked in order to enforce the proper interface conditions. If a Lagrangian description of the solid is used, the location of this interface is always known through the solution of the Lagrangian equations of motion. However, accurately locating the interface in the Eulerian reference frame on the fluid mesh is much more difficult – particularly when unstructured grids are used.

The importance of accurately locating the wetted interface and imposing boundary conditions is key to insuring energy conservation in FSI problems. Farhat and Lesoinne [8] demonstrate that the use of partitioned integration schemes can lead to errors in the exchange of kinetic energy between the fluid and the solid at the interface. Similarly, Michler *et al.* [16] and van Brummelen *et al.* [25] show that stability and accuracy of the solution can be degraded without careful implementation of the interface conditions.

Two general classes of computational techniques exist for treating FSI problems. In the first, fluid and solid meshes that conform at the wetted interface, possibly with mismatching mesh size, can be used. In these methods, the Eulerian fluid description must account for mesh movement. In the presence of, large Lagrangian interface displacements, the fluid mesh can become highly distorted requiring frequent and extensive remeshing with the concomitant remapping of fluid variables. The second class of methods uses distinct overlapping meshes for the fluid and solid. In these methods, a Lagrangian solid mesh moves independently of the Eulerian fluid mesh permitting large relative motion of the solid with respect to the fluid and without inducing distortion on the fluid mesh. However, the fluid-solid interface arbitrarily cuts the fluid mesh requiring some method to identify and/or track the wetted interface on the Eulerian fluid mesh.

Overlapping grid methods have been studied by Wang [26] and Tai *et al.* [24] who used a locally-refined overset grid around the solid to capture boundary effects. Jiang *et al.* [13, 12] focused on shock waves produced by a supersonic projectile discharged from a shock tube. In this work, the solid projectile is defined analytically as a rectangle on the fluid grid. The so-called "Cartesian-grid" method has been developed to compute flows about complex geometries by placing an analytic surface over a structured Eulerian fluid mesh. Boundary conditions are enforced at the interface in a fashion similar to the ghost-fluid method. However, the geometry of the cut-cells is altered based on the location of the interface. Marella *et al.* [15] combined the Cartesian-grid method with the level-set method to study flows with moving boundaries. Forrer and Jeltsch [11] studied interface treatment for static interfaces using the Cartesian grid method.

In contrast to the overlapping grid approach, level-set methods have been popularized as a means to implicitly describe and track interfaces on an Eulerian grid [17, 18, 21]. These methods have been applied to multi-material/multi-fluid problems with interfaces that evolve in an Eulerian reference frame. Strain [22, 23] has studied level-set methods for multi-phase flow problems. Kohno and Tanahashi [14] have used the level-set method with adaptive mesh refinement to study free surface flows. Enright *et al.* [7] investigated a hybrid particle level-set method.

Level-set methods have also been applied to fluid-solid interaction problems by a number of researchers. Fedkiw [9] used the level-set method on one and two-dimensional problems where the level-set was advected using the local velocity of the fluid-solid interface. Cirak [6] used the level-set method for coupled shell-fluid simulations of thin structures. Arienti *et al.* [2] used the level-set method for problems with large structural deformations relative to the fluid. Both Cirak and Arienti used an alternative to traditional level-set methods where the level-set function was reinitialized every time step based on the current location of the solid. In addition, both Cirak and Arienti applied the ghost-fluid method [10], where assumed state variables are specified in ghost elements and used to enforce boundary conditions at the fluid-solid interface.

Motivated by the need to simulate FSI problems where the Lagrangian solid exhibits large bulk motion relative to the fluid, we adopted the approach of using distinct and overlapping meshes for the fluid and solid domains. In the fluid-domain, we developed the techniques necessary to apply the ghost-fluid method on general, unstructured meshes. This approach requires methods for a) the fast construction of a level-set representing the Lagrangian interface on the unstructured fluid mesh, b) extracting the interface in a computationally efficient way, and c) applying interface conditions.

In this paper, we present a new super-sampled projection method for the fast construction of a level-set representing the intersection between a Lagrangian mesh immersed in a fluid mesh. Here, this interface is described analytically as circles and spheres on the backgound Eulerian mesh to permit error assessment of the level-set field and the implicitly defined interface. The use of the level-set in the identification of the interface and cut-cells in the Eulerian mesh is demonstrated. In §2 we present the basic ideas behind super-sampled projection. In §3 we discuss categorization of the elements based on the level-set data. In §4, it is shown that the extraction of an interface location based on a zero level-set converges as $O(h)$ when compared to the exact interface location – suggesting that the availability of a Lagrangian interface description is always preferred.

2. THE SUPER-SAMPLED PROJECTION METHOD

In this section, a brief overview of the level-set method is presented. A new super-sampled projection method for the fast construction of level-set data suitable for identifying the wetted surface and for applying the requisite coupling conditions in FSI problems is outlined. Finally, the sensitivity of the numerical quadrature rule for the projection is explored.

As background, we begin with the topological description of unstructured grids, which is most easily presented in terms of its constituent vertices, edges, and faces (see Barth *et al.* [3, 4, 5]). An unstructured mesh can be described in terms of its primal and dual grids as shown in Figure 1. Here, the primal grid consists of the elements and nodes (i.e., vertices). Edges of the primal grid connect the vertices or nodes of the mesh. Edges of the dual grid connect the element centers, i.e., vertices of the dual-grid. The edges of the centroidal dual grid are straight lines connecting the cell centers. Edges of the median dual grid are not necessarily straight, but pass through the center of the face of the elements.

Edges are, in general, lines that connect a pair of vertices and constitute a primary data structure required for the efficient implementation of edge-based CFD solvers. For our FSI work, we used an edge-based discontinuous Galerkin formulation. However, we note that the work presented here is generally applicable to node and element-centered finite volume techniques, as well as finite element methods.

The coupling terms for overlapping Lagrangian-Eulerian grids requires information about the intersection between the Lagrangian interface and the Eulerian mesh. In our edge-based flow solver, the conserved variables are centered in the elements of the primal grid, and the

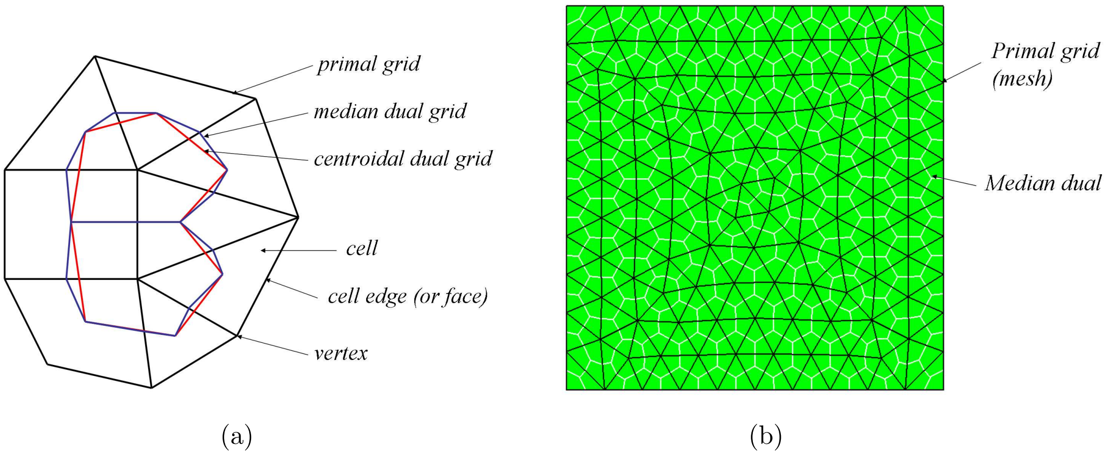


Figure 1. Examples of the (a) primal, median dual, and centroidal dual grids on a mixed mesh of quadrilateral and triangular elements, and (b) the primal grid (black) and median dual grid (white) on a square domain meshed with triangular elements.

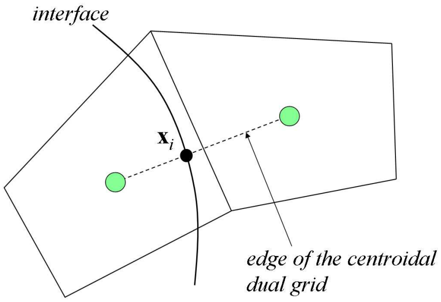


Figure 2. Intersection of the centroidal dual grid and the fluid-solid interface on a cut edge. The interface location (at the intersection) is denoted $\mathbf{x}_i$.

numerical flux is located on unique edges of the median dual grid. These numerical fluxes must be modified to enforce the fluid-solid interface conditions on edges of the dual grid cut by the interface. This requires knowledge of the intersection point between the edge of the centroidal dual grid and the interface as shown in Figure 2. In the following discussion, we describe a new level-set construction technique and the associated calculations for determining the intersection between the Lagrangian interface and unstructured fluid mesh.

2.1. The Level-Set Method

The level-set method was originally devised by Osher and Sethian[19] to evolve moving interfaces under the action of a velocity field. The level-set method implicitly embeds the location of the interface at the zero level-set in a higher-dimensional function. The level-set function, ϕ, has the following properties

$$\phi(\mathbf{x},t) = \begin{cases} < 0 & \text{outside the interface,} \\ > 0 & \text{inside the interface,} \\ = 0 & \text{on the interface.} \end{cases} \tag{1}$$

The motion of the interface is analyzed by convecting the level-set function with a velocity $\mathbf{v}$ according to

$$\frac{\partial \phi}{\partial t} + \mathbf{v} \cdot \nabla \phi = 0 \tag{2}$$

In FSI problems, the motion of the interface is dictated by the Lagrangian solid. The advection equation only needs to be solved in a narrow band around the fluid-solid interface [1, 20].

A common and convenient level-set definition specifies ϕ as

$$\phi(\mathbf{x},t) = \begin{cases} -\delta(\mathbf{x},t) & \text{outside the interface,} \\ +\delta(\mathbf{x},t) & \text{inside the interface,} \\ 0 & \text{on the interface.} \end{cases} \tag{3}$$

where $\delta(\mathbf{x},t)$ is the normal distance to the interface, i.e., a signed-distance function.

In many level-set implementations, the initialization of the signed-distance function is done at the beginning of the simulation, and equation 2 is solved to evolve the interface over time. The signed-distance function may be reinitialized periodically to correct for any errors that accumulate in the solution. However, the expense of periodically determining whether or not fluid mesh points are inside or outside the interface and their distance to the interface is not a major concern compared to the cost of solving equation 2. Narrow band methods can reduce the computational cost by reinitializing only nodes in the mesh that lie in close proximity to the interface.

It is computationally intensive to construct the signed-distance function at each time step in a fluid-solid interaction problem, especially in three-dimensional problems with complex geometry. Instead, it's desirable to use the known Lagrangian interface location to construct an interface-local Heaviside function that can be projected onto the fluid mesh. For our problem, we use a least-squares projection of a locally constructed Heaviside function to construct a nodal level-set function. We have found this approach to be a computationally expedient alternative to re-distancing operations for the generation of level-set data.

For our purposes, we define our level-set function, ϕ, as

$$\phi(\mathbf{x}) = 1 - 2H(\mathbf{x} - \mathbf{x}_i) \tag{4}$$

where $\mathbf{x}_i$ is the location of the interface, and

$$H(\mathbf{x} - \mathbf{x}_i) = \begin{cases} 0 & \mathbf{x} < \mathbf{x}_i \\ \frac{1}{2} & \mathbf{x} = \mathbf{x}_i \\ 1 & \mathbf{x} > \mathbf{x}_i \end{cases} \tag{5}$$

Here, we introduce a local Heaviside function to capture the sharp interface in the element. Equation 5 permits a simple "in-out' test on the solid geometry. However, a naive approach to setting nodal values in this way can lead to a relatively poor representation of the interface unless a highly refined mesh is used locally at the interface. Figure 3(a) shows the exact signed

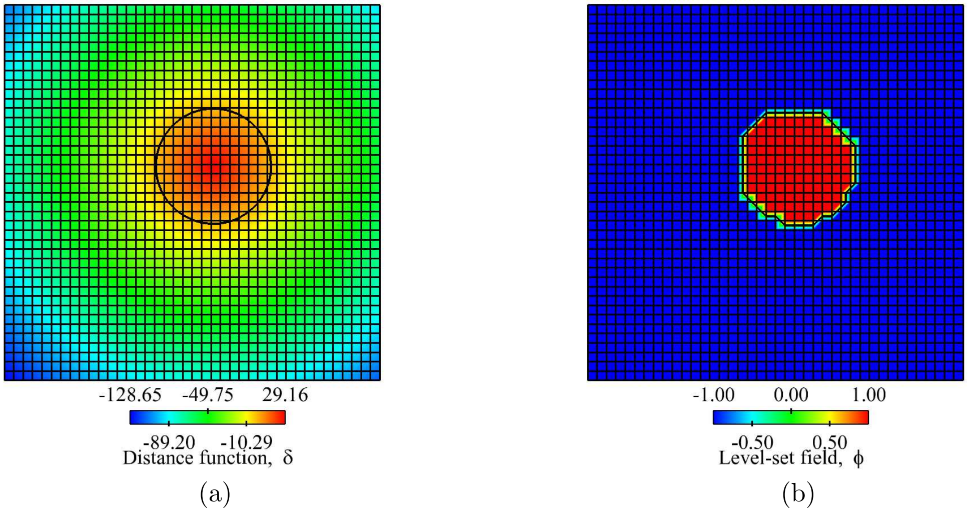


Figure 3. Example level-set functions representing a circle on a uniform background mesh. (a) Nodal signed-distance function. (b) Naive approach to constructing nodal "in-out" data with $\phi = \pm 1$.

distance function for a circular interface, and Figure 3(b) shows the corresponding level-set based on a simple in-out test. To avoid this difficulty, we project this Heaviside function onto the fluid mesh using a super-sampling technique that avoids the need for highly-refined meshes in the region around the interface.

2.2. The Super-Sampled Projection Method

In order to construct the super-sampled projection, we begin with a Galerkin finite element formalism and a least-squares projection

$$\int_{\Omega_F} w\left(\hat{\phi} - \phi\right) = 0 \tag{6}$$

where $\hat{\phi}$ are the desired nodal values of the level-set function, ϕ are level-set values obtained by a simple in-out test (equation 5), w is the weight function, and Ω_F is the fluid domain. The element level equations take the form

$$\int_{\Omega_F^e} N_I\left(N_J\hat{\phi}_J - \phi\right) = 0 \tag{7}$$

where N_I are the element shape functions. Rearranging yields,

$$\int_{\Omega_F^e} N_I N_J \hat{\phi}_J = \int_{\Omega^e} N_I \phi, \tag{8}$$

or alternatively, as an element-level system,

$$M_{IJ}^e \hat{\phi}_J = F_I^e \tag{9}$$

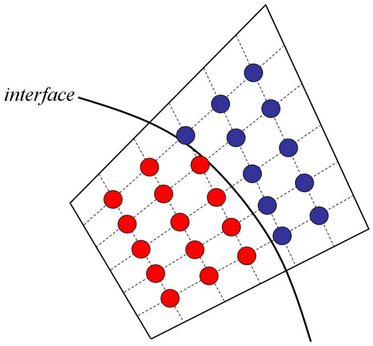


Figure 4. Super-sampling within a quadrilateral element using a 5 × 5 quadrature rule.

where M^e_{IJ} is the element unit mass matrix and F^e_I is the element load vector.

The element mass matrix and corresponding right-hand-side vector are evaluated using standard Gaussian quadrature rules as follows

$$M^e_{IJ} \approx \sum_{k=1}^{N_{QPT}} w_k N_I|_k N_J|_k det(J)|_k \tag{10}$$

$$F^e_I \approx \sum_{k=1}^{N_{QPT}} w_k N_I|_k \phi|_k det(J)|_k \tag{11}$$

where N_{QPT} is the number of quadrature points, w_k is the quadrature weighting, and $det(J)$ is the determinant of the Jacobian matrix.

There are a number of super-sampling techniques that could be used at the element-level to evaluate the level-set function. The most expedient super-sampling corresponds to increasing the number of quadrature points in the quadrature rule. This is depicted in Figure 4 with an interface within a quadrilateral element and a 5 × 5 quadrature rule. Here the quadrature points to the left of the interface are given a level-set value of 1.0 (inside the interface, colored red). The quadrature points to the right of the interface are given a value of -1.0 (outside the interface, colored blue). This approach to super-sampling permits simple in-out tests to be used at each quadrature point making the implementation fast, robust, and amenable to use with either continuous or discrete (faceted) interfaces.

Assembling the finite element equations over the entire domain yields

$$M_{ij}\hat{\phi}_j = F_i \tag{12}$$

where $1 \leq i, j \leq Nnp$, and Nnp is the number of nodes in the mesh.

Solving equation 12 for $\hat{\phi}_j$ is straight forward and yields the desired nodal values of the level-set function. The eigenvalues of the unit mass matrix are uniformly bounded, making the matrix problem amenable to simple iterative techniques, e.g., simple diagonal-scaled conjugate gradient techniques converge in $O(5 - 10)$ iterations. Alternatively, row-sum lumping the unit mass matrix is also acceptable resulting in a projection that only requires the assembly of two vectors and one vector divide.

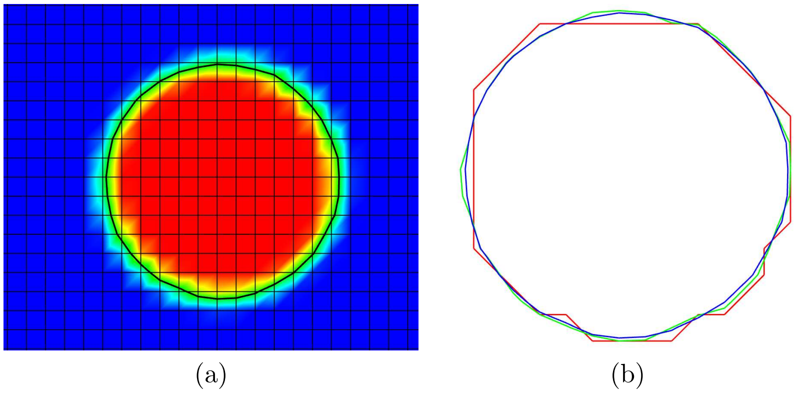


Figure 5. Level-set representation of a circle on a 40 × 40 structured mesh using the super-sampled projection. (a) Level-set field with $-1 \leq \phi \leq 1$ and the zero contour for a 10 × 10 quadrature rule. (b) Comparison of the zero level-set contours from nodal data (red), 2×2 quadrature (green), 10×10 quadrature (blue).

This method yields smoothly varying level-set data near the interface, and preserves the bounds imposed on the level-set, i.e., ±1. However, we note that the same quadrature rule is required for both the unit mass matrix and the right-hand-side. A failure to maintain equivalent quadrature rules for the unit mass and right-hand-side leads to nodal level-set values that exhibit grid bias on unstructured meshes.

Next, we consider the effects of super-sampling via quadrature points with a series of computational experiments conducted with analytical definitions of circular and spherical interfaces and with both structured and unstructured meshes. Curved interfaces such as these are nice because the interface normal continuously changes orientation with respect to the background mesh, testing a large variety of configurations. It should be noted that relatively coarse meshes were used to obtain the results presented here. This was done intentionally because we would like to permit the mesh size to be driven primarily by the physics in the fluid field, not the accuracy requirements for the level-set construction.

Additionally, for the problems presented here, the super-sampled projection was done on the entire fluid mesh. However, it would be more computationaly expedient to perform this projection just in the narrow band around the interface. This method is easily extened to a narrow band approach.

The first test mesh consisted of a structured arrangement of 40 × 40 4-node quadrilateral elements on a square 100 × 100, domain. A circle of radius 30.74 was located at $(11.3, 14.1)$ intentionally to avoid any beneficial artifacts associated with mesh alignment. The complete mesh can be seen in Figure 3.

Figure 5(a) shows the level-set field obtained with the super-sampled projection method with a 10 × 10 quadrature rule. Here, the level-set field is naturally bounded to $-1 \leq \phi \leq 1$, and the the zero-contour indicates the approximation to the circular interface. Figure 5(b) shows the zero level-set contours overlaid for nodal data that samples the local Heaviside function, and the super-sampled projection results for 2×2, and 10×10 quadrature rules. The influence of super-sampling via the quadrature rule, even with a fixed mesh resolution, is apparent.

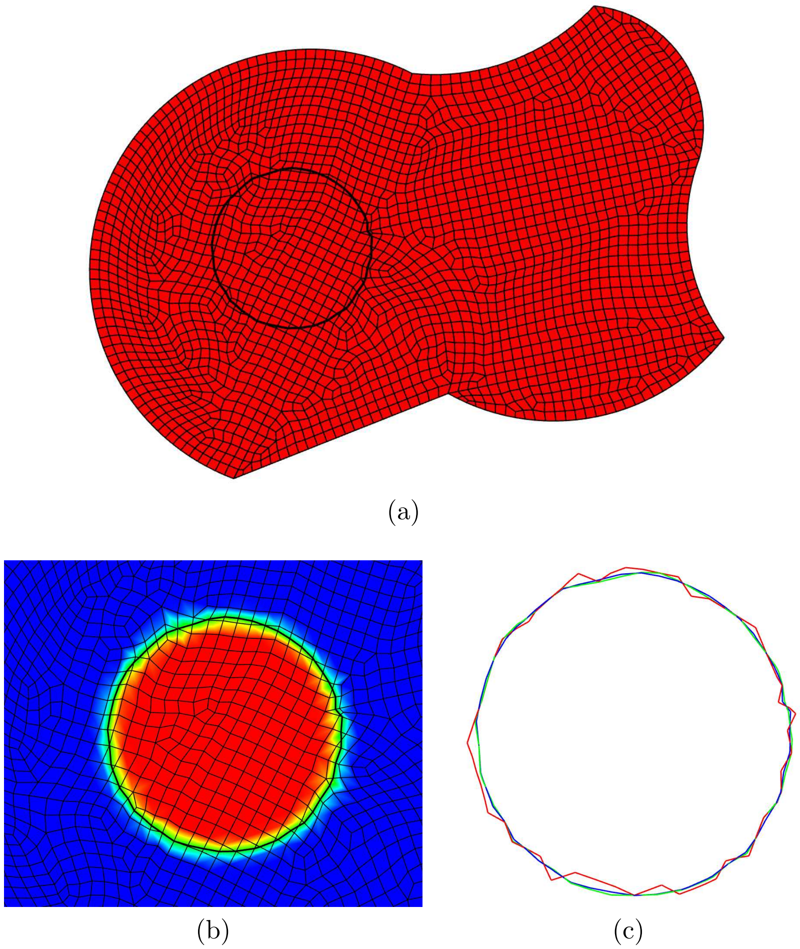
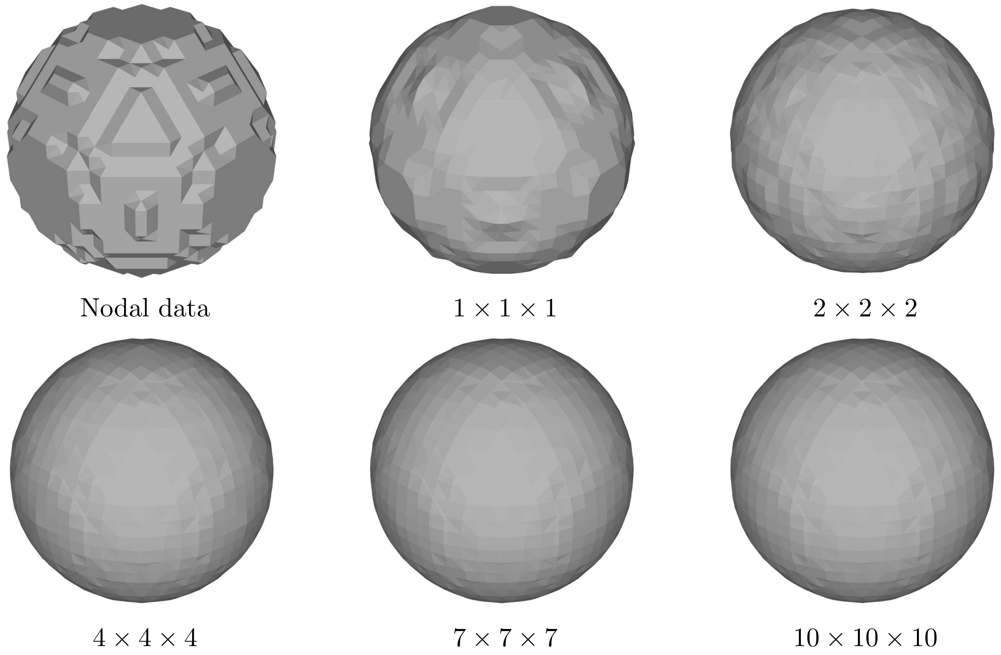


Figure 6. Level-set representation of a circle on an unstructured mesh using the super-sampled projection. (a) Mesh and zero contour obtained with a 10×10 quadrature rule, (b) Level-set field with $-1 \leq \phi \leq 1$ and the zero contour for a 10×10 quadrature rule. (c) Comparison of the zero contours from nodal data (red), 2×2 quadrature (green), 10×10 quadrature (blue).

The second test used an unstructured mesh, i.e. “paved”, with 2312 4-node quadrilateral elements, and is shown in Figure 6(a) with the level-set field obtained with the super-sampled projection. The arbitrary domain shape was created to ensure the generation of an unstructured mesh for testing the super-sampled projection. Figure 6(b) shows the level-set field with the zero level-set contour. Figure 6(c) shows the interface level-set contours for the nodal data, and the super-sampled projection results with 2×2, and 10×10 quadrature rules. Again, the influence of super-sampling on the regularity of the interface is apparent.

The structured mesh used for the 3-D study consisted of 125,000 8-node hexahedral elements, i.e, a $50 \times 50 \times 50$ mesh on a $10 \times 10 \times 10$ cube. A sphere of radius 2.0 was located at the



Nodal data

1 × 1 × 1

2 × 2 × 2

4 × 4 × 4

7 × 7 × 7

10 × 10 × 10

Figure 7. Zero contour of a level-set function representing a sphere on a structured 3D mesh. The effect of increasing the quadrature rule is clearly visible. The level-set function is defined with $-1 \leq \phi \leq 1$.

center of the domain. Figure 7 shows the results obtained with the super-sampled projection method. Also shown in the figure are the zero level-set contours for all of the quadrature rules tested. The benefits of increasing the quadrature rule can be clearly seen.

Similar tests were conducted for the sphere using the unstructured mesh shown in Figure 8(a). Figure 8(b) shows the zero level-set contour obtained using a $10 \times 10 \times 10$ quadrature rule. Some faceting can be seen, albeit for a mesh far more coarse than would be required to solve a realistic fluid flow problem.

3. ELEMENT CATEGORIZATION

The super-sampled projection produces nodal values for the level-set function, $\hat{\phi}$. Since this is continuous data in each element, values at the element-centroid can be obtained by evaluating

$$\phi^e = \sum_{i=1}^{N_{NPE}} N_i \hat{\phi}_i \tag{13}$$

where N_i are the shape functions for the element, and N_{NPE} is the number of nodes in the element. Here, the shape functions are evaluated at the center of the element.

We use both the nodal values and the element values to categorize the elements. The categories determine the role the elements play in the solution of the fluid equations. The

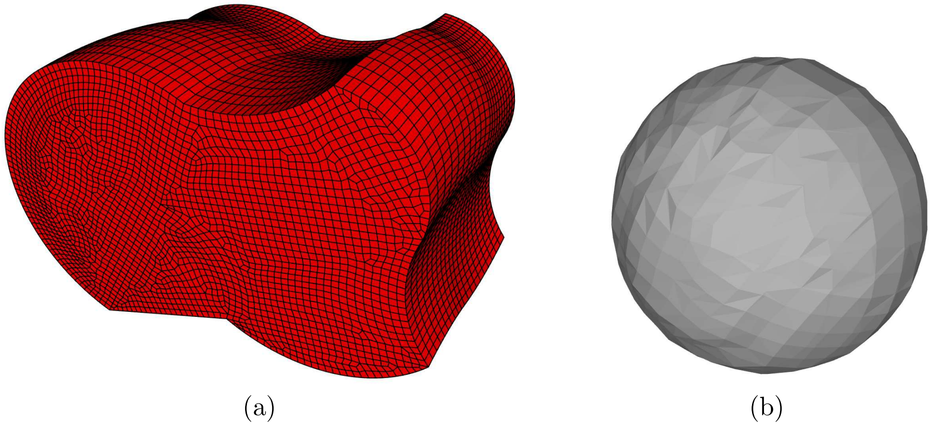


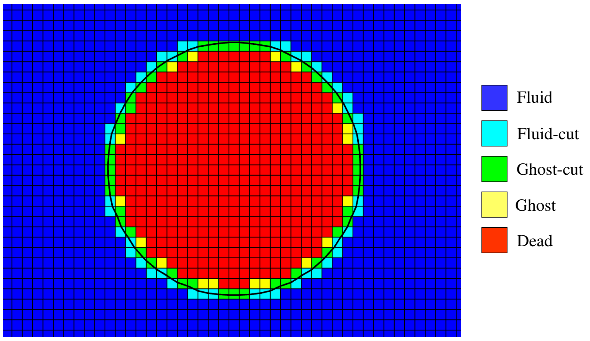
Figure 8. Mesh and zero contour of a level-set function representing a sphere on an unstructured 3D mesh. The level-set function is defined with $-1 \leq \phi \leq 1$. A $10 \times 10 \times 10$ quadrature rule was used.

elements are categorized as fluid, fluid-cut, ghost-cut, ghost, and dead. An example for a circular interface is shown in Figure 9(a), and a summary of the element categories is presented in Figure 9(b).

Elements completely outside the interface are marked as fluid elements. These elements have an element value of ϕ^e such that $-1 \leq \phi^e < 0$. In addition, all nodes are outside the interface such that $-1 \leq \hat{\phi} < 0$ at every node. Elements marked as fluid will be treated as usual in the flow solver and updated each time step. Fluid elements close to the interface may have one or more neighbors marked as ghost or ghost-cut. Ghost data specified in these neighbors is used to enforce velocity boundary conditions at the interface.

Figure 9(b) shows the definitions of each of the element categories. Elements marked as fluid-cut are intersected by the interface, but the cell center is in the fluid. Fluid-cut elements are updated each time step, but one or more of the neighboring elements is a ghost or a ghost-cut element. Ghost elements are completely contained within the interface. Ghost-cut elements are intersected by the interface, but their cell center is inside. Ghost and ghost-cut elements are not updated, but data is specified in these elements to enforce velocity boundary conditions at the interface. Elements marked as dead are omitted during the solution procedure for a given time step. However, dead elements do not share a common face with any fluid elements or fluid-cut elements like ghost elements do.

In our edge-based CFD solver, numerical fluxes are calculated at the unique edges of the dual-grid of the mesh in order to update the state in each fluid or fluid-cut element every time step. Edges of the dual-grid that are cut by the fluid-solid interface need to be identified so fluxes along these cut edges can be modified to properly enforce the fluid-solid interface conditions. The level-set data is used to identify these cut edges. Only certain combinations of element types can exist on either side of a cut edge. These combinations are fluid/ghost, fluid/ghost-cut, fluid-cut/ghost, and fluid-cut/ghost-cut. All other combinations signify a regular, or uncut, edge.



(a)

Region	Category	ϕ^e	$\hat{\phi}$
Outside the Interface	Fluid	$-1 \leq \phi^e < 0$	$-1 \leq \hat{\phi} < 0$
Cut by the Interface	Fluid-Cut	$-1 < \phi^e \leq 0$	$-1 < \hat{\phi} < 1$
Cut by the Interface	Ghost-Cut	$0 < \phi^e < 1$	$-1 < \hat{\phi} < 1$
Inside the Interface	Ghost	$0 < \phi^e < 1$	$0 < \hat{\phi} < 1$
Inside the Interface	Dead	$0 < \phi^e \leq 1$	$0 < \hat{\phi} \leq 1$

(b)

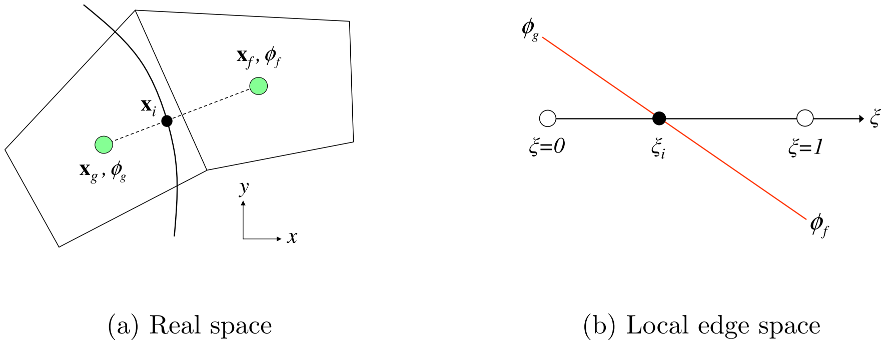
Figure 9. (a) Circular interface on a uniform mesh showing the element categories, (b) element categories and level-set values.

4. INTERFACE EXTRACTION

Enforcement of the fluid-solid interface conditions on a cut edge requires knowlege of the interface location on the cut edge, $\mathbf{x}_i$ (see Figure 2). We refer to this as "interface extraction". Two methods of interface extraction were investigated. First, the level-set data was used directly to extract the location of the interface on each cut edge. The second method used the analytical surface directly for the interface location.

The level-set data at each edge vertex, corresponding to the the level-set function at the cell centers, ϕ^e, can be interpolated along the edge to find the interface location where $\phi^e = 0$. To accomplish this, the data from a cut edge in physical space is mapped into a local, one-dimensional computational space as shown in Figure 10. Here the convention is adopted that the ghost cell center, $\mathbf{x}_g$, corresponds to the origin of the local system, $\xi = 0$. The fluid cell center, $\mathbf{x}_f$, corresponds to the point $\xi = 1$. In the local system, the interface location is computed as

$$\xi_i = \frac{-\phi_g}{\phi_f - \phi_g} \tag{14}$$



(a) Real space (b) Local edge space

Figure 10. Mapping from physical space into the local computational edge space.

where ϕ_g and ϕ_f are the cell centered values of the level-set function at the ghost and fluid elements respectively.

The interface location in physical space is calculated from ξ_i as

$$\mathbf{x}_i = \sum_{k=1}^{2} N_k(\xi_i)\mathbf{x}_k \tag{15}$$

where the N_k are the shape functions for the local element given by $N_1 = 1 - \xi$ and $N_2 = \xi$ which are evaluated at ξ_i.

To improve accuracy, this process is carried out iteratively. Each iteration, the value of the level-set function, ϕ_i, is calculated at the extracted interface location using

$$\phi_i = \sum_{k=1}^{Nnodes} N_k \hat{\phi}_k \tag{16}$$

where the shape functions, N_k, are evaluated at the interface location, and $\hat{\phi}_k$ are the nodal values of the level-set function in the element containing the interface. Iterations continue until ϕ_i is less than a desired tolerance. A tolerance of 10^{-8} was used and the solution typically required only two to three iterations.

Error in the extracted interface location was assessed using an $\mathcal{L}_1$-norm defined as

$$e = \frac{\sum |R - \overline{R}|}{|\overline{R}|} \tag{17}$$

where R is the extracted interface location and $\overline{R}$ is the interface location from the analytic surface (see Figure 11). The sum is performed over all cut edges in the mesh. The values of $\overline{R}$ and $R - \overline{R}$ are determined as

$$R - \overline{R} = \sqrt{(x - \overline{x})^2 + (y - \overline{y})^2 + (z - \overline{z})^2} \tag{18}$$

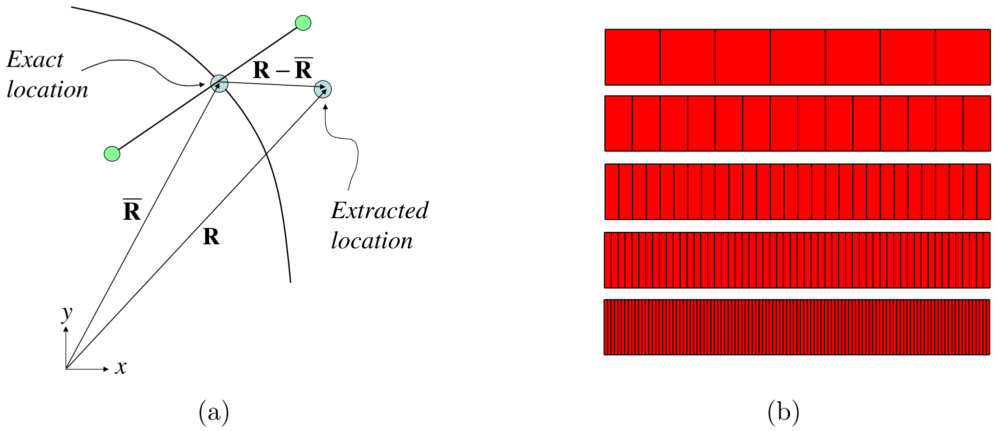


Figure 11. (a) $\mathcal{L}_1$ error in the extracted interface location, (b) patch of seven elements used to study interface extraction error (top). Four mesh refinements are shown here.

where $\overline{R} = \sqrt{\overline{x}^2 + \overline{y}^2 + \overline{z}^2}$.

Error in the extracted interface location was considered on a 7×1 domain initially discretized with 7 elements as shown in Figure 11(b). The interface was located a distance of 3.38 units from the left side. Error in the interface location was calculated using equation 17. Error was computed as a function of the quadrature rule used for the super-sampled projection as well as a function of the mesh size. We note here that the interface extraction problem in two and three dimensions reduces to the problem presented here, making the results of this test representative of a broad range of geometrical situations.

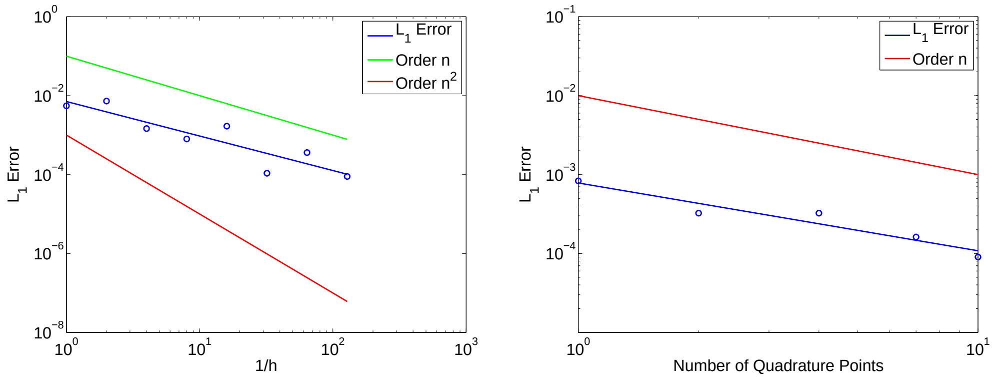
First, the mesh was doubled seven times in the "long" direction. Only four of these refinement levels are shown in Figure 11(b). For each mesh, the quadrature rule was held constant (10×10). Next, the quadrature rule was varied on the finest mesh (896 elements in the long direction). Figure 12 shows the results. Suprisingly, the error in the extracted interface location converges linearly with both the mesh size and the quadrature rule used for the super-sampled projection.

The error in the extracted interface location can be attributed to two sources. First, there is some error in the interface extraction itself. This can be measured by looking at the value of the level-set function at the extracted interface location which should be zero. For the cases run, this value was always on the order of 10^{-16}, i.e., roughly 64-bit machine precision. Thus the error in the extraction algorithm was considered small.

The second source of error is in the level-set function. Since this data is used directly to establish the interface location, these directly affect the extracted interface location. This error can be measured with respect to an exact grid function defined as

$$\Phi_{GF} = \frac{1}{V} \int_{\Omega^e} \phi d\Omega^e \tag{19}$$

where the value of the level set function, $\phi = \pm 1$, is exactly integrated over the element with



(a) (b)

Figure 12. $\mathcal{L}_1$ error in the extracted interface location. (a) Error versus mesh size. (b) Error versus quadrature rule.

domain, Ω^e, and volume, V. Using this definition for the exact grid function, the $\mathcal{L}_1$ error in the level-set function can be calculated as

$$e = \frac{\sum |\phi^e - \Phi_{GF}|}{|\Phi_{GF}|} \tag{20}$$

where ϕ^e are the cell centered valued of the level set function. The sum is carried out over all the elements in the mesh.

Error in the level-set function was investigated on the domain presented above. Results are shown in Figure 13 both with respect to element size and quadrature rule. Error in the level-set function decreases linearly with element size. However, the results did not show a significant decrease in error with quadrature rule.

The second method of interface extraction investigated used the exact geometry of the interface and the cut edges, both of which are known. Using the interface and edge geometry directly proved to be computationally simpler than using the level-set data and iterating to find the interface as described above. Here, both flat and curved interface geometry are considered.

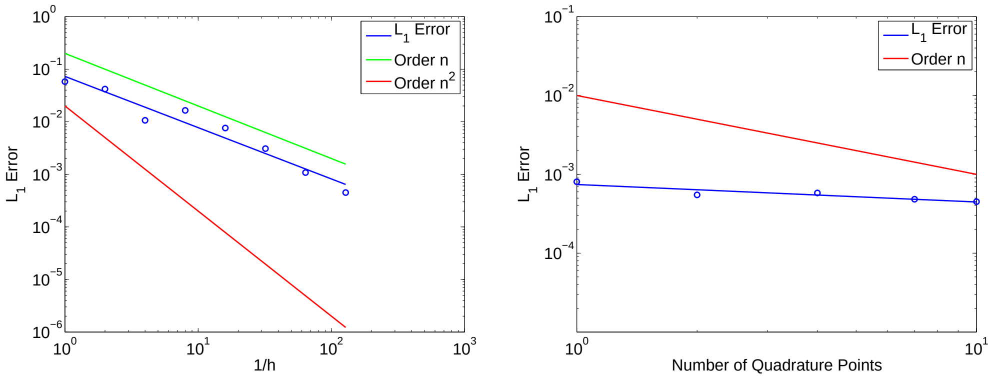
For the case of a rectangular interface in two or three dimensions, interface extraction requires calculation of the intersection of a line and a plane. The line represents the cut edge, and the plane represents the appropriate face of the rectangle or box.

Figure 14(a) shows a plane intersected by a line. The intersection point is denoted (x, y, z). The plane is defined by a point on the plane, (x_0, y_0, z_0), and the unit normal to the plane, $\mathbf{n}$. The line is defined along vector $\mathbf{V}$ between the two end points (X_0, Y_0, Z_0) and (X_1, Y_1, Z_1).

The intersection point can be found along the line as $\mathbf{R} = \mathbf{r}_0 + t\mathbf{V}$ where t is a scalar parameter. The vector between points (x, y, z) and (x_0, y_0, z_0) lies in the plane and is orthogonal to $\mathbf{n}$. Thus, the equation for the plane can be written as $(\mathbf{R} - \mathbf{R}_0) \cdot \mathbf{n} = 0$. These two equations can be solved for the unknown scale factor, t, such that

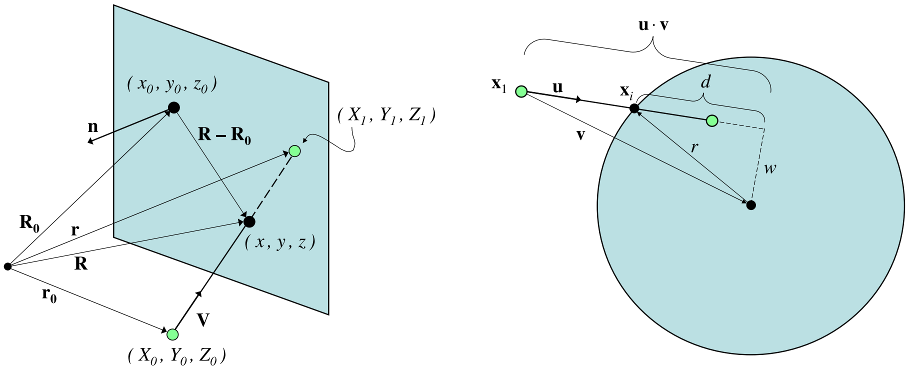
$$t = \frac{(\mathbf{R}_0 - \mathbf{r}_0) \cdot \mathbf{n}}{\mathbf{V} \cdot \mathbf{n}}. \tag{21}$$

yielding the intersection $\mathbf{R}$.



(a) (b)

Figure 13. $\mathcal{L}_1$ error in the level-set function measured with respect to an exact grid function. (a) Error versus mesh size. (b) Error versus quadrature rule.



(a) (b)

Figure 14. Definitions and geometry for calculating the intersection of (a) a line and a plane, (b) a line and a circle or sphere.

For the case of a circular or spherical interface, interface extraction requires calculation of the intersection between a line and a sphere as shown in Figure 14(b). Consider the unit vector $\mathbf{u}$ between the end points of the line, and the vector $\mathbf{v}$ between the external point and the center of the sphere. The radius of the circle is given by r. Inspection of Figure 14(b) shows two triangles for which $||\mathbf{v}||^2 = (\mathbf{u} \cdot \mathbf{v})^2 + w^2$, and $r^2 = d^2 + w^2$. Solving these two equations for the distance d, $d^2 = r^2 - ||\mathbf{v}||^2 + (\mathbf{u} \cdot \mathbf{v})^2$. The intersection location can then be determined as

$$\mathbf{x}_i = \mathbf{x}_1 + (\mathbf{u} \cdot \mathbf{v} - d)\mathbf{u}. \tag{22}$$

These equations, particularly equation 21, are important not just because they can be used for analytic interface geometry as developed here. But they are applicable to the case of discretized geometry as well. With overlapping meshes, the edges of the dual grid will be cut by external faces of the Lagrangian solid mesh. These faces are represented as planes, and equation 21 can be used to compute the intersection of the Eulerian dual grid and external faces of the Lagrangian mesh.

5. SUMMARY & CONCLUSIONS

This work was motivated by the need to simulate FSI problems using distinct and overlapping meshes for the fluid and solid domains. This approach requires a method to identify and extract the fluid-solid interface on the fluid mesh in a computationally efficient manner. To this end, we propose a new super-sampled projection method to initialize a level-set function on the fluid grid. We use the known Lagrangian interface location to construct an interface-local Heaviside function that we project onto the fluid mesh in a least-squares sense. Super sampling was implemented by increasing the quadrature rule used for the projection. This allieviated the need to refine the mesh locally around the interface. We found this approach to be a expedient alternative to re-distancing operations for the generation of level-set data.

Error analyses showed that the $\mathcal{L}_1$ error in the interface location varied linearly with mesh size and quadrature rule used for the super-sampled projection. Similar convergence rates can be expected for any choice of level-set function. This convergence rate was considered less than acceptable for fluid-solid interaction problems where the exact geometry of the interface is known *a priori*.

Based on these results, we chose to use the level-set data to identify which edges of the dual grid were cut by the interface as well as for interface visualization. However, we chose to use the Lagrangian interface geometry directly to determine the exact location of the interface/edge intersections in the mesh that are needed to enforce appropriate boundary conditions at the fluid-solid interface. This was determined to be an acceptable trade-off between the Eulerian level-set description and the Lagrangian interface description, since either could be used to indentify and track the interface.

ACKNOWLEDGEMENTS

This work was supported by Sandia National Laboratories and the ASC hydro methods effort at Los Alamos National Laboratory. Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

REFERENCES

1. D. Adalsteinsson and J. A. Sethian, *A fast level set method for propagating interfaces*, Journal of Computational Physics, 118 (1995), pp. 269–277.
2. M. Arienti, P. Hung, E. Morano, and J. Shepherd, *A level set approach to Eulerian-Lagrangian coupling*, Jounal of Computational Physics, 185 (2003), pp. 213–251.
3. T. J. Barth, *Recent developments in high order k-exact reconstruction on unstructured meshes*, AIAA-93-0668, January 1993.
4. T. J. Barth and P. O. Frederickson, *Higher order solution of the Euler equations on unstructured grids using quadratic reconstruction*, AIAA-90-0013, January 1990.

5. T. J. Barth and D. C. Jespersen, *The design and application of upwind schemes on unstructured meshes*, AIAA-89-0366, January 1989.
6. F. Cirak and R. Radovitzky, *A Lagrangian–Eulerian shell–fluid coupling algorithm based on level sets*, Computers and Structures, 83 (2005), pp. 491–498.
7. D. Enright, R. Fedkiw, J. Ferzinger, and I. Mitchell, *A hybrid particle level set method for improved interface capturing*, Journal of Computational Physics, 183 (2002), pp. 83–116.
8. C. Farhat and M. Lesoinne, *Two efficient staggered algorithms for the serial and parallel solution of three-dimensional nonlinear transient aeroelastic problems*, Computer Methods in Applied Mechanics and Engineering, 182 (2000), pp. 499–515.
9. R. P. Fedkiw, *Coupling an Eulerian fluid calculation to a Lagrangian solid calculation with the ghost fluid method*, Journal of Computational Physics, 175 (2002), pp. 200–224.
10. R. P. Fedkiw, T. Aslam, B. Merriman, and S. Osher, *A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method)*, Jounal of Computational Physics, 152 (1999), pp. 457–492.
11. H. Forrer and R. Jeltsch, *A higher–order boundary treatment for Cartesian–grid methods*, Journal of Computational Physics, 140 (1998), pp. 259–277.
12. Z. Jiang, Y. Huang, and K. Takayama, *Shocked flows induced by supersonic projectiles moving in tubes*, Computers and Fluids, 33 (2004), pp. 953–966.
13. Z. Jiang, K. Takayama, and B. W. Skews, *Numerical study on blast flowfields induced by supersonic projectiles discharged from shock tubes*, Physics of Fluids, 10 (1998), pp. 277–288.
14. H. Kohno and T. Tanahashi, *Numerical analysis of moving interfaces using a level set method with adaptive mesh refinement*, International Journal of Numerical Methods in Fluids, 45 (2004), pp. 921–944.
15. S. Marella, S. Krishnan, H. Liu, and H. S. Udaykumar, *Sharp interface Cartesian grid method I: An easily implemented technique for 3D moving boundary calculations*, Journal of Computational Physics, 210 (2005), pp. 1–31.
16. C. Michler, E. H. van Brummelen, S. J. Hulshoff, and R. de Borst, *The relavance of conservation for the stability and accuracy of numerical methods for fluid-structure interaction*, Computer Methods in Applied Mechanics and Engineering, 192 (2003), pp. 4195–4215.
17. S. Osher and R. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces*, vol. 153 of Applied Mathematical Sciences, Springer-Verlag, 1st ed., 2003.
18. S. Osher and R. P. Fedkiw, *Level set methods: An overview and some recent results*, Journal of Computational Physics, 169 (2001), pp. 463–502.
19. S. Osher and J. A. Sethian, *Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi equations*, Journal of Computational Physics, 79 (1988), pp. 12–49.
20. D. Peng, B. Merriman, S. Osher, H.-K. Zhao, and M. Kang, *A PDE-based fast local level set method*, Journal of Computational Physics, 155 (1999), pp. 410–438.
21. J. A. Sethian, *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*, vol. 3 of Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, 2nd ed., 1999.
22. J. Strain, *Tree mehods for moving interfaces*, Journal of Computational Physics, 151 (1999), pp. 616–648.
23. ———, *A fast modular semi-Lagrangian method for moving interfaces*, Journal of Computational Physics, 161 (2000), pp. 512–536.
24. C. H. Tai, Y. Zhao, and K. M. Liew, *Parallel computation of unsteady incompressible viscous flows around moving rigid bodies using an immersed object method with overlapping grids*, Journal of Computational Physics, 207 (2005), pp. 151–172.
25. E. H. van Brummelen, S. J. Hulshoff, and R. de Borst, *Energy conservation under incompatibility for fluid-structure interaction problems*, Computer Methods in Applied Mechanics and Engineering, 192 (2003), pp. 2727–2748.
26. Z. J. Wang, *A fully conservative interface algorithm for overlapped grids*, Journal of Computational Physics, 122 (1995), pp. 96–106.