Implicit Surface Tension Formulation with a Lagrangian Surface mesh on an Eulerian Simulation Grid

Craig Schroeder*, Wen Zheng*, Ronald Fedkiw*

Stanford University, 353 Serra Mall Room 207, Stanford, CA 94305

Abstract

We present a method for applying implicit forces on a Lagrangian mesh to an Eulerian discretization of the Navier Stokes equations in a way that produces a sparse symmetric positive definite system. The resulting method has implicit and fully coupled viscosity, pressure, and Lagrangian forces. We apply our new framework for forces on a Lagrangian mesh to the case of surface tension force, which when treated explicitly leads to a tight $\Delta t = O(\Delta x^{3/2})$ time step restriction. By applying surface tension as an implicit Lagrangian force, the resulting method benefits from improved stability and the ability to take larger time steps. The resulting discretization is also able to maintain parasitic currents at low levels.

1. Introduction

Surface tension is a significant force in many phenomena, generally becoming more significant as features decrease in size. It plays a vital role in the dynamics of water droplets and bubbles, where the shape becomes increasingly dominated by surface tension forces as the size decreases. The foundations of surface tension simulation began with smeared δ-functions applied to interface maintained by front tracking [53], volume of fluid [4], and level sets [5, 46]. Another interesting early work on surface tension is [24]. Since then, a wide variety of different formulations have been investigated, and the difficulties posed by surface tension have been studied and increasingly understood.

Surface tension is subject to a tight $\Delta t = O(\Delta x^{3/2})$ time step step restriction, and a significant number number of approaches have been considered for alleviating it. The early approach of [23] started from a CSF formulation and worked out how the curvature changes in time to derive an implicit formulation of surface tension. This was followed up by [25], which used a simplified formulation that also added additional damping along the interface. The approaches of [43, 55, 56] (see also [10]) improve stability by adding and subtracting a Laplacian, effectively introducing additional viscosity to the fluid. There has also been work on improving stability with VOF-based surface tension schemes [45, 33].

The tendency for surface tension to produce parasitic currents is well known in the literature. [27] aimed to eliminate parasitic currents in the second gradient formulation of surface tension at the cost of momentum conservation. Methods based on delta-function require careful attention to produce accurate results, as even $O(1)$ errors may result if not implemented carefully, a point which has been noted for example in the context of length computation [44, 11, 49]. In [28], it was noted that parasitic currents when using a regularized delta function formulation were found to be three orders of higher than were obtained from a formulation handling surface tension as pressure jump across a sharp interface using the ghost fluid method. [15] showed that enforcing a balance between the pressure and the surface tension in a manner similar to the ghost fluid method was important. [16] also found that balancing the pressure with surface tension forces resulted in a more accurate formulation. [18], based on [19], proposed using XFEM to formulate a discontinuous pressure jump as well. See also [6, 7, 14]. Following these observations, we take a sharp-interface approach to applying

*{cas43,zhw,fedkiw}@cs.stanford.edu, Stanford University

surface tension, addressing accuracy through careful attention to the force balance between surface tension and pressure.

We develop a method for implicit surface tension based on a discretizing the Laplace Beltrami operator on a Lagrangian mesh, which we couple it to an otherwise Eulerian discretization. The resulting scheme is very similar in form to CSF schemes, but we are able to take advantage of the flexibility in mapping between degrees of freedom to reduce the severity of parasitic currents to levels comparable to those in [28]. While we formulate surface tension as a force, our surface tension discretization can also be formulated as a pressure jump.

We explore two approaches to maintaining our interface: level sets and front tracking. For a good review on front tracking, see [50]. Front tracking has the advantage that it is relatively easy to treat implicitly and is conveniently compatible with our treatment of surface tension through a Lagrangian surface mesh. See for other examples of implicit front tracking [32], which evolves an elastic membrane implicitly using and Jacobian-free Newton Krylov method solver, and [47], which uses the Jacobian-free Newton Krylov method to implicitly update control points.

2. Eulerian Navier Stokes Equations with Forces on a Lagrangian Mesh

2.1. Navier Stokes Equations

In this paper, we consider the Eulerian form of the incompressible Navier Stokes equations,

$$\rho\left(\frac{\partial \vec{u}}{\partial t} + (\vec{u}\cdot\nabla)\vec{u}\right) = -\nabla p + \mu\nabla^2\vec{u} + \vec{f_\rho} \tag{1}$$

subject to the incompressibility constraint

$$\nabla\cdot\vec{u} = 0. \tag{2}$$

Here, $\vec{u}$ is the fluid velocity, p is the pressure, μ is the dynamic viscosity, ρ is the density, and $\mathbf{f}_\rho$ is an additional force density. Later, $\mathbf{f}_\rho$ will be a surface tension force.

Our spatial discretization is based on a standard Marker and Cell (MAC, [20]) grid discretization. Chorin splitting [9] is used to separate the advection term and explicit forces from the pressure and implicit forces. We apply advection to obtain an intermediate velocity

$$\rho\vec{u}^\star = \rho\vec{u}^n - \Delta t\rho(\vec{u}\cdot\nabla)\vec{u} + \Delta t\vec{f}_{\rho 1}, \tag{3}$$

which is applied with 3$^{\text{rd}}$ order Runge Kutta [40] and 3$^{\text{rd}}$ order Hamilton Jacobi ENO [36]. The pressure and implicit forces are discretized as

$$\rho\vec{u}^{n+1} = \rho\vec{u}^\star - \Delta t\nabla p + \Delta t\mu\nabla^2\vec{u}^{n+1} + \Delta t\vec{f}_{\rho 2}. \tag{4}$$

It will be convenient to work in a fully-discretized and volume-weighted form, which is accomplished by first multiplying through by the cell volume V. The equations now take the form

$$\boldsymbol{\beta}\mathbf{u}^{n+1} = \boldsymbol{\beta}\mathbf{u}^\star - \Delta t\mathbf{G}\mathbf{p} - \Delta tV^{-1}\mu\tilde{\mathbf{G}}_\mu^T\tilde{\mathbf{G}}_\mu\mathbf{u}^{n+1} + \Delta t\mathbf{f}. \tag{5}$$

Here, $\mathbf{u}$ represents a vector of velocity degrees of freedom, $\boldsymbol{\beta} = V\rho$ is a diagonal matrix whose entries are the lumped fluid masses at faces, $\mathbf{G} = V\nabla$ is the discretized, volume-weighted pressure gradient, $\tilde{\mathbf{G}}_\mu = V\nabla$ is the discretized, volume-weighted velocity gradient, $\mathbf{p}$ the vector of pressures, and $\mathbf{f}$ is the vector of implicit forces $V\vec{f}_{\rho 2}$ an additional force. Finally, dividing by $\boldsymbol{\beta}$ and letting $\mathbf{G}_\mu = \sqrt{\Delta tV^{-1}\mu}\tilde{\mathbf{G}}_\mu$ and $\tilde{\mathbf{p}} = \Delta t\mathbf{p}$ for notational convenience,

$$\mathbf{u}^{n+1} = \mathbf{u}^\star - \boldsymbol{\beta}^{-1}\mathbf{G}\tilde{\mathbf{p}} - \boldsymbol{\beta}^{-1}\mathbf{G}_\mu^T\mathbf{G}_\mu\mathbf{u}^{n+1} + \Delta t\boldsymbol{\beta}^{-1}\mathbf{f}. \tag{6}$$

The corresponding incompressibility constraint is

$$\mathbf{G}^T\mathbf{u}^{n+1} = \mathbf{0}. \tag{7}$$

2.2. Structure Equations

Although an Eulerian framework is used for evolution, occasional references will be made of the evolution equations for structure as well. We use a traditional Lagrangian approach is used for structures, where we solve the equations $\dot{\hat{\mathbf{x}}} = \hat{\mathbf{v}}$ and $\hat{\mathbf{M}}\dot{\hat{\mathbf{v}}} = \hat{\mathbf{f}}$. Here, the quantities $\hat{\mathbf{x}}$ and $\hat{\mathbf{v}}$ are the vectors containing the position and velocity degrees of freedom for all of the Lagrangian particles, and $\hat{\mathbf{M}}$ is the mass matrix. It suffices for our purposes to consider elastic forces, where the force $\hat{\mathbf{f}}$ is defined in terms of a potential energy as $\hat{\mathbf{f}} = -\frac{\partial \phi}{\partial \hat{\mathbf{x}}}$, where ϕ is a potential energy function that is a function of only the positions $\hat{\mathbf{x}}$.

One particularly simple and stable way of evolving these equations is via backward Euler. In this case, one solves

$$\hat{\mathbf{x}}^{n+1} = \hat{\mathbf{x}}^n + \Delta t \hat{\mathbf{v}}^{n+1}, \tag{8}$$

$$\hat{\mathbf{v}}^{n+1} = \hat{\mathbf{v}}^n + \Delta t \hat{\mathbf{M}}^{-1} \hat{\mathbf{f}}^{n+1}. \tag{9}$$

Taking the first order approximation $\hat{\mathbf{f}}^{n+1} = \hat{\mathbf{f}}^n + \frac{\partial \hat{\mathbf{f}}^n}{\partial \hat{\mathbf{x}}}(\hat{\mathbf{x}}^{n+1} - \hat{\mathbf{x}}^n) + \frac{\partial \hat{\mathbf{f}}^n}{\partial \hat{\mathbf{v}}}(\hat{\mathbf{v}}^{n+1} - \hat{\mathbf{v}}^n) = \hat{\mathbf{f}}^n + \Delta t \frac{\partial \hat{\mathbf{f}}^n}{\partial \hat{\mathbf{x}}} \hat{\mathbf{v}}^{n+1} = \hat{\mathbf{f}}_e + \hat{\mathbf{D}}\hat{\mathbf{v}}^{n+1}$, since $\hat{\mathbf{f}}^{n+1}$ does not have explicit velocity dependence. $\mathbf{f}_e$ and $\hat{\mathbf{D}}\hat{\mathbf{v}}^{n+1}$ represent the explicit and implicit portions of the force, respectively. The backward Euler update now takes the form

$$\hat{\mathbf{x}}^{n+1} = \hat{\mathbf{x}}^n + \Delta t \hat{\mathbf{v}}^{n+1}, \tag{10}$$

$$\hat{\mathbf{v}}^{n+1} = \hat{\mathbf{v}}^n + \Delta t \hat{\mathbf{M}}^{-1}(\hat{\mathbf{f}}_e + \hat{\mathbf{D}}\hat{\mathbf{v}}^{n+1}). \tag{11}$$

$\hat{\mathbf{D}} = \Delta t \frac{\partial \hat{\mathbf{f}}^n}{\partial \hat{\mathbf{x}}} = -\frac{\partial^2 \phi}{\partial \hat{\mathbf{x}} \partial \hat{\mathbf{x}}}$ is symmetric since it is a second derivative. In practice, $\hat{\mathbf{D}}$ will often be negative definite. This is the case for the surface tension force considered in this paper. Note that $\hat{\mathbf{D}}$ contains the factor Δt, so that the damping forces due to the implicit force treatment vanish under temporal refinement.

2.3. Framework for Forces Discretized on a Lagrangian Mesh

Consider a single particle whose motion is computed from the velocities stored on an Eulerian grid. This particle does not have any real degrees of freedom, and they do not have mass. In general, these particles may be on the surface of a fluid region or in the interior. The Lagrangian velocities $\hat{\mathbf{v}}$, a vector containing the velocities for all Lagrangian particles, are defined in terms of the vector of all Eulerian velocities $\mathbf{u}$ by the relation

$$\hat{\mathbf{v}} = \mathbf{H}\mathbf{u}, \tag{12}$$

where $\mathbf{H}$ is some linear operator that interpolates Eulerian velocities to Lagrangian velocities. For clarity, use hats through this paper to indicate that quantities are defined on the Lagrangian degrees of freedom.

The force $\mathbf{f}$ in Eq. (6) is discretized over the Eulerian degrees of freedom. We would like to instead add a force $\hat{\mathbf{f}}$ which is discretized over the new Lagrangian degrees of freedom. Let P be the power (work per unit time) performed by applying a force $\mathbf{f}$ to fluid moving with velocity $\mathbf{u}$. That is, $P = \mathbf{f}^T\mathbf{u}$. In this case, we computed the power using the Lagrangian degrees of freedom. Computing it instead from the Lagrangian degrees of freedom produces $P = \hat{\mathbf{f}}^T\hat{\mathbf{v}} = \hat{\mathbf{f}}^T\mathbf{H}\mathbf{u} = \mathbf{f}^T\mathbf{u}$. Since $\hat{\mathbf{v}}$ and $\mathbf{f}$ could have been arbitrary, it follows that

$$\mathbf{f} = \mathbf{H}^T\hat{\mathbf{f}}. \tag{13}$$

Two variables whose inner product produces the work (or a related quantity like power) are said to be work conjugates. The above derivation can be considered an application of the principle of virtual work, a powerful tool that is describe in more detail in [3]. The operator $\mathbf{H}^T$ is a conservative force distribution operator. In general, whenever $\mathbf{H}$ is an interpolation operator from one set of degrees of freedom to another, the operator $\mathbf{H}$ can be used to distribute forces from the second set of degrees of freedom to the first.

Since the motion of the Lagrangian degrees of freedom is dictated by fluid motion, one could compute an *effective* mass for the Lagrangian particles. Observe that the velocity change of Eulerian degrees of freedom $\Delta\mathbf{u}$ due to force defined in terms of Lagrangian degrees of freedom is $\Delta\mathbf{u} = \Delta t \boldsymbol{\beta}^{-1}\mathbf{f} = \Delta t \boldsymbol{\beta}^{-1}\mathbf{H}^T\hat{\mathbf{f}}$. The velocity change of Lagrangian degrees of freedom $\Delta\hat{\mathbf{v}}$ due to force defined in terms of Lagrangian degrees of

freedom is $\Delta\hat{\mathbf{v}} = \mathbf{H}\Delta\mathbf{u} = \Delta t\mathbf{H}\boldsymbol{\beta}^{-1}\mathbf{H}^T\hat{\mathbf{f}} = \Delta t\hat{\mathbf{M}}^{-1}\hat{\mathbf{f}}$, so that $\hat{\mathbf{M}}^{-1} = \mathbf{H}\boldsymbol{\beta}^{-1}\mathbf{H}^T$ is the effective mass of the Lagrangian degrees of freedom.

In constitutive mechanics, Lagrangian forces for constitutive models have elastic and damping components. These components are typically assumed to take the form $\hat{\mathbf{f}} = \hat{\mathbf{f}}_e + \hat{\mathbf{D}}\hat{\mathbf{v}}$ [3]. Here, $\hat{\mathbf{f}}_e$ is the explicit force and $\hat{\mathbf{D}}$ is the symmetric negative semidefinite damping matrix. The non-advective portion of the Navier Stokes equations now takes on the form

$$\mathbf{u}^{n+1} = \mathbf{u}^\star - \boldsymbol{\beta}^{-1}\mathbf{G}\tilde{\mathbf{p}} - \boldsymbol{\beta}^{-1}\mathbf{G}_\mu^T\mathbf{G}_\mu\mathbf{u}^{n+1} + \Delta t\boldsymbol{\beta}^{-1}\mathbf{H}^T\hat{\mathbf{f}}_e + \Delta t\boldsymbol{\beta}^{-1}\mathbf{H}^T\hat{\mathbf{D}}\mathbf{H}\mathbf{u}^{n+1}. \tag{14}$$

Observe that the implicit component of the additional force has the same form as the viscosity term. The explicit portion of the surface tension force goes in $\vec{f}_{\rho 1}$, and the implicit part goes in $\vec{f}_{\rho 1}$. Eq. (3) and Eq. (4) can be written (in fully discretized form) as

$$\mathbf{u}^\star = \mathbf{u}^n - \Delta t\mathbf{A}(\mathbf{u}) + \Delta t\boldsymbol{\beta}^{-1}\mathbf{H}^T\hat{\mathbf{f}}_e \tag{15}$$

and

$$\mathbf{u}^{n+1} = \mathbf{u}^\star - \boldsymbol{\beta}^{-1}\mathbf{G}\tilde{\mathbf{p}} - \boldsymbol{\beta}^{-1}\mathbf{G}_\mu^T\mathbf{G}_\mu\mathbf{u}^{n+1} + \Delta t\boldsymbol{\beta}^{-1}\mathbf{H}^T\hat{\mathbf{D}}\mathbf{H}\mathbf{u}^{n+1}, \tag{16}$$

where $\mathbf{A}(\mathbf{u})$ is the discretized advective term. We have effectively constructed a methodology for formulating arbitrary forces on a set of Lagrangian degrees of freedom in a way that results in a purely Eulerian problem and such that the linearized force terms take on the form as the viscosity term. The framework of [37] allows one to construct a symmetric and positive definite system where viscosity is applied implicitly and coupled to the pressure projection. Since we have expressed the implicit portion of the Lagrangian forces in the same form, we can implicitly compute the pressure, viscosity, and Lagrangian forces in a way that is fully coupled and produces a symmetric and positive definite system. We assemble this system in Section 2.6.

This combination is particularly appealing in the case of surface tension. Both surface tension forces and pressure forces are stiff, and they are frequently in a delicate balance. This point is exemplified by the classical and important situation of a stationary circle, which is considered in Section 4.2. If surface tension is treated explicitly or in an implicit step separate from pressure, the surface tension force and the pressure force will be only weak coupling. The proposed method, by contrast, produces strong coupling between surface tension and pressure.

2.4. Fluid-Structure Interaction Equations

The derivation in Section 2.3 for applying forces on a Lagrangian mesh to an Eulerian simulation can also be obtained directly by considering the case of fluid-structure interaction where the Lagrangian forces are applied to a Lagrangian structure which is in turn coupled to an Eulerian fluid. The equations for fluid-structure interaction are constructed from the equations of fluid dynamics and structure mechanics by the addition of a contact constraint, which is enforced through a momentum exchange. The contact constraint ensures that the relative velocity between fluid and structure is zero at a set of constraint locations. In [37] these constraint locations were placed at fluid faces near the fluid-structure interface. The contact constraint can be written $\hat{\mathbf{J}}\hat{\mathbf{v}}^{n+1} = \mathbf{W}\mathbf{u}^{n+1}$, where $\hat{\mathbf{J}}$ is the matrix that interpolates structure velocities to constraint locations and $\mathbf{W}$ is the matrix that interpolates fluid velocities to constraint locations. The constraint is enforced by exchanging impulses $\boldsymbol{\lambda}$ between the structure and the fluid at the constraint locations. As noted in Section 2.3, since fluid velocities are interpolated to the constraint locations with $\mathbf{W}$, the impulses $\boldsymbol{\lambda}$ should be applied back to the fluid using $\mathbf{W}^T$. This produces the modified fluid equation,

$$\boldsymbol{\beta}\mathbf{u}^{n+1} = \boldsymbol{\beta}\mathbf{u}^\star - \mathbf{G}\tilde{\mathbf{p}} + \mathbf{W}^T\boldsymbol{\lambda}, \tag{17}$$

Where for simplicity the viscosity term is ignored. An equal and opposite impulse is applied to structure similarly using $\hat{\mathbf{J}}^T$. The modified structure equation is

$$\hat{\mathbf{M}}\hat{\mathbf{v}}^{n+1} = \hat{\mathbf{M}}\hat{\mathbf{v}}^\star + \Delta t\hat{\mathbf{f}}_e + \Delta t\hat{\mathbf{D}}\hat{\mathbf{v}}^{n+1} - \hat{\mathbf{J}}^T\boldsymbol{\lambda}. \tag{18}$$

Finally the fluid-structure interaction equations are

$$\begin{aligned}
\boldsymbol{\beta}\mathbf{u}^{n+1} &= \boldsymbol{\beta}\mathbf{u}^{\star} - \mathbf{G}\tilde{\mathbf{p}} + \mathbf{W}^T\boldsymbol{\lambda} && (19)\\
\hat{\mathbf{M}}\hat{\mathbf{v}}^{n+1} &= \hat{\mathbf{M}}\hat{\mathbf{v}}^{\star} + \Delta t\hat{\mathbf{f}} + \Delta t\hat{\mathbf{D}}\hat{\mathbf{v}}^{n+1} - \hat{\mathbf{J}}^T\boldsymbol{\lambda} && (20)\\
\mathbf{G}^T\mathbf{u}^{n+1} &= \mathbf{0} && (21)\\
\hat{\mathbf{J}}\hat{\mathbf{v}}^{n+1} &= \mathbf{W}\mathbf{u}^{n+1} && (22)
\end{aligned}$$

For more details see [37], noting that our definition of $\hat{\mathbf{J}}$ corresponds to their $\mathbf{W}\hat{\mathbf{J}}$.

2.5. Derivation from Fluid Structure Interaction

Consider the FSI coupling equations in Section 2.4, if $\hat{\mathbf{J}}$ was an invertible matrix, it would be possible to eliminate $\hat{\mathbf{v}}^{n+1}$ from Eq. (20) by solving Eq. (22). Normally, however, $\hat{\mathbf{J}}$ will not be an invertible matrix. In this case, it is still possible to express $\hat{\mathbf{v}}^{n+1}$ as the sum of two vectors, one ($\hat{\mathbf{J}}^T\boldsymbol{\psi}$) in the range of $\hat{\mathbf{J}}^T$ and one ($\hat{\mathbf{v}}'$) orthogonal to it. There will in general be many suitable choices for $\boldsymbol{\psi}$, since $\boldsymbol{\psi}$ could contain components in the nullspace of $\hat{\mathbf{J}}^T$. The vector $\boldsymbol{\psi}$ is selected to be the unique solution with minimum norm. The quantity $\boldsymbol{\psi}$ is defined at the constraint degrees of freedom. With this, $\hat{\mathbf{v}}^{n+1}$ can be written as

$$\hat{\mathbf{v}}^{n+1} = \hat{\mathbf{J}}^T\boldsymbol{\psi} + \hat{\mathbf{v}}' \qquad \hat{\mathbf{J}}\hat{\mathbf{v}}' = \mathbf{0}, \tag{23}$$

The vector $\boldsymbol{\psi}$ can now be eliminated. Substituting Eq. (23) into Eq. (22) produces $\hat{\mathbf{J}}\hat{\mathbf{J}}^T\boldsymbol{\psi} + \hat{\mathbf{J}}\hat{\mathbf{v}}' = \mathbf{W}\mathbf{u}^{n+1}$, which simplifies to

$$\hat{\mathbf{J}}\hat{\mathbf{J}}^T\boldsymbol{\psi} = \mathbf{W}\mathbf{u}^{n+1} \tag{24}$$

using Eq. (23). The minimum norm solution to Eq. (24) is obtained by taking the pseudoinverse, denoted by $\cdot^+$.

$$\boldsymbol{\psi} = (\hat{\mathbf{J}}\hat{\mathbf{J}}^T)^+\mathbf{W}\mathbf{u}^{n+1} \tag{25}$$

Substituting Eq. (25) into Eq. (23) produces

$$\hat{\mathbf{v}}^{n+1} = \hat{\mathbf{J}}^T(\hat{\mathbf{J}}\hat{\mathbf{J}}^T)^+\mathbf{W}\mathbf{u}^{n+1} + \hat{\mathbf{v}}' = \mathbf{H}\mathbf{u}^{n+1} + \hat{\mathbf{v}}', \tag{26}$$

where the definition

$$\mathbf{H} = \hat{\mathbf{J}}^T(\hat{\mathbf{J}}\hat{\mathbf{J}}^T)^+\mathbf{W} \tag{27}$$

has been used. The operator $\mathbf{H}$ maps fluid velocities to solid velocities and behaves as an interpolation operator. The operator $\mathbf{W}$ interpolates fluid velocities to constraint degrees of freedom, and $\hat{\mathbf{J}}$ interpolates solid velocities to constraint degrees of freedom. The operator $\hat{\mathbf{J}}^T(\hat{\mathbf{J}}\hat{\mathbf{J}}^T)^+$ is almost the inverse of the interpolation operator $\hat{\mathbf{J}}$. Indeed, $(\hat{\mathbf{J}}\hat{\mathbf{J}}^T)(\hat{\mathbf{J}}\hat{\mathbf{J}}^T)^+$ is as nearly an identity matrix as is possible.

Next, decompose $\boldsymbol{\lambda}$ using a similar orthogonal decomposition

$$\boldsymbol{\lambda} = \hat{\mathbf{J}}\hat{\boldsymbol{\xi}} + \boldsymbol{\lambda}' \qquad \hat{\mathbf{J}}^T\boldsymbol{\lambda}' = \mathbf{0}, \tag{28}$$

where this time $\hat{\boldsymbol{\xi}}$ is located at Lagrangian velocity degrees of freedom. As before, $\hat{\boldsymbol{\xi}}$ is taken to be of minimum norm, so that it is uniquely determined. Next, $\hat{\boldsymbol{\xi}}$ is eliminated. Substituting Eq. (28) into Eq. (20) produces

$$\hat{\mathbf{J}}^T\hat{\mathbf{J}}\hat{\boldsymbol{\xi}} + \hat{\mathbf{J}}^T\boldsymbol{\lambda}' = (\Delta t\hat{\mathbf{f}} + \hat{\mathbf{M}}\hat{\mathbf{v}}^{\star}) + (\Delta t\hat{\mathbf{D}} - \hat{\mathbf{M}})\hat{\mathbf{v}}^{n+1}, \tag{29}$$

which can be further simplified and solved using the pseudoinverse to produce

$$\hat{\boldsymbol{\xi}} = (\hat{\mathbf{J}}^T\hat{\mathbf{J}})^+((\Delta t\hat{\mathbf{f}} + \hat{\mathbf{M}}\hat{\mathbf{v}}^{\star}) + (\Delta t\hat{\mathbf{D}} - \hat{\mathbf{M}})\hat{\mathbf{v}}^{n+1}). \tag{30}$$

Substituting Eq. (30) into Eq. (28) produces

$$\boldsymbol{\lambda} = \hat{\mathbf{J}}(\hat{\mathbf{J}}^T\hat{\mathbf{J}})^+((\Delta t\hat{\mathbf{f}} + \hat{\mathbf{M}}\hat{\mathbf{v}}^{\star}) + (\Delta t\hat{\mathbf{D}} - \hat{\mathbf{M}})\hat{\mathbf{v}}^{n+1}) + \boldsymbol{\lambda}'. \tag{31}$$

Applying the identity $\hat{\mathbf{J}}(\hat{\mathbf{J}}^T\hat{\mathbf{J}})^+ = (\hat{\mathbf{J}}\hat{\mathbf{J}}^T)^+\hat{\mathbf{J}}$ produces

$$\boldsymbol{\lambda} = (\hat{\mathbf{J}}\hat{\mathbf{J}}^T)^+\hat{\mathbf{J}}((\Delta t\hat{\mathbf{f}} + \hat{\mathbf{M}}\hat{\mathbf{v}}^\star) + (\Delta t\hat{\mathbf{D}} - \hat{\mathbf{M}})\hat{\mathbf{v}}^{n+1}) + \boldsymbol{\lambda}'. \tag{32}$$

Finally, substituting Eq. (32) into $\mathbf{W}^T\boldsymbol{\lambda}$ and simplifying with the definition Eq. (27) produces

$$\mathbf{W}^T\boldsymbol{\lambda} = \mathbf{H}^T((\Delta t\hat{\mathbf{f}} + \hat{\mathbf{M}}\hat{\mathbf{v}}^\star) + (\Delta t\hat{\mathbf{D}} - \hat{\mathbf{M}})\hat{\mathbf{v}}^{n+1}) + \mathbf{W}^T\boldsymbol{\lambda}' \tag{33}$$

Finally, substitute Eq. (33) into Eq. (19),

$$\boldsymbol{\beta}\mathbf{u}^{n+1} = \boldsymbol{\beta}\mathbf{u}^\star - \mathbf{G}\tilde{\mathbf{p}} + \mathbf{H}^T((\Delta t\hat{\mathbf{f}} + \hat{\mathbf{M}}\hat{\mathbf{v}}^\star) + (\Delta t\hat{\mathbf{D}} - \hat{\mathbf{M}})\hat{\mathbf{v}}^{n+1}) + \mathbf{W}^T\boldsymbol{\lambda}' \tag{34}$$

Rearranging and converting from Lagrangian degrees of freedom to the Eulerian degrees of freedom using Eq. (26) produces

$$\boldsymbol{\beta}\mathbf{u}^{n+1} = \boldsymbol{\beta}\mathbf{u}^\star - \mathbf{G}\tilde{\mathbf{p}} + \mathbf{H}^T(\Delta t\hat{\mathbf{D}} - \hat{\mathbf{M}})\mathbf{H}\mathbf{u}^{n+1} + \mathbf{H}^T(\Delta t\hat{\mathbf{f}} + \hat{\mathbf{M}}\hat{\mathbf{v}}^\star) + \mathbf{H}^T(\Delta t\hat{\mathbf{D}} - \hat{\mathbf{M}})\hat{\mathbf{v}}' + \mathbf{W}^T\boldsymbol{\lambda}' \tag{35}$$

The quantities $\hat{\mathbf{v}}'$ and $\boldsymbol{\lambda}'$ are problematic, and it is desirable to choose a discretization so that they vanish. One way of doing this is to introduce one (independent) coupling constraint per solid degree of freedom. Then $\hat{\mathbf{J}}$ is an invertible matrix, and $\hat{\mathbf{v}}'$ and $\boldsymbol{\lambda}'$ vanish. This produces the lumped mass formulation

$$\boldsymbol{\beta}\mathbf{u}^{n+1} = \boldsymbol{\beta}\mathbf{u}^\star - \mathbf{G}\tilde{\mathbf{p}} + \mathbf{H}^T(\Delta t\hat{\mathbf{D}} - \hat{\mathbf{M}})\mathbf{H}\mathbf{u}^{n+1} + \mathbf{H}^T(\Delta t\hat{\mathbf{f}} + \hat{\mathbf{M}}\hat{\mathbf{v}}^\star) \tag{36}$$

This can be placed into a particularly appealing form using the simplifying assumption $\hat{\mathbf{v}}^\star = \mathbf{H}\mathbf{u}^\star$ and the lumped mass $\tilde{\boldsymbol{\beta}} = \boldsymbol{\beta} + \mathbf{H}^T\hat{\mathbf{M}}\mathbf{H}$

$$\tilde{\boldsymbol{\beta}}\mathbf{u}^{n+1} = \boldsymbol{\beta}\mathbf{u}^\star + \mathbf{H}^T\hat{\mathbf{M}}\hat{\mathbf{v}}^\star - \mathbf{G}\tilde{\mathbf{p}} + \mathbf{H}^T\Delta t\hat{\mathbf{f}} + \Delta t\mathbf{H}^T\hat{\mathbf{D}}\mathbf{H}\mathbf{u}^{n+1} \tag{37}$$

In the limit of a massless structure, $\hat{\mathbf{M}} = \mathbf{0}$, and $\tilde{\boldsymbol{\beta}} = \boldsymbol{\beta}$, so that what remains is left with a massless formulation

$$\boldsymbol{\beta}\mathbf{u}^{n+1} = \boldsymbol{\beta}\mathbf{u}^\star - \mathbf{G}\tilde{\mathbf{p}} + \Delta t\mathbf{H}^T\hat{\mathbf{f}} + \Delta t\mathbf{H}^T\hat{\mathbf{D}}\mathbf{H}\mathbf{u}^{n+1}. \tag{38}$$

This massless equation depends only on the ability to eliminate the velocity degrees of freedom using the contact constraints.

In summary, we have shown that if one there is one constraint per Lagrangian degree of freedom, so that $\hat{\mathbf{J}}$ is invertible, then the FSI problem in Eq. (19) through Eq. (22) can be entirely written in terms of the Eulerian degrees of freedom on the Eulerian mesh with the momentum in $\hat{\mathbf{v}}^\star$ transfered to fluid with $\mathbf{H}^T$. What remains is an Eulerian problem to solve. Eq. (22) is unchanged, but the other three FSI equations are replaced with Eq. (37). In the event that the Lagrangian mesh is massless, as we propose, then we are left with Eq. (22) and Eq. (38). This provides legitimacy to the the last two terms in Eq. (14). We can now revert back to Eq. (7), Eq. (15), and Eq. (16), which also include viscosity, advection, and other forces.

2.6. SPD Linear System

A symmetric positive definite (SPD) system is formed from Eq. (16) and Eq. (7) following the approach of [37]. If $\hat{\mathbf{D}}$ is symmetric negative semidefinite, there will exist a matrix $\hat{\mathbf{C}}$ such that $\hat{\mathbf{C}}^T\hat{\mathbf{C}} = -\Delta t\hat{\mathbf{D}}$.

$$\mathbf{u}^{n+1} = \mathbf{u}^\star - \boldsymbol{\beta}^{-1}\mathbf{G}\tilde{\mathbf{p}} - \boldsymbol{\beta}^{-1}\mathbf{G}_\mu^T\mathbf{G}_\mu\mathbf{u}^{n+1} - \boldsymbol{\beta}^{-1}\mathbf{H}^T\hat{\mathbf{C}}^T\hat{\mathbf{C}}\mathbf{H}\mathbf{u}^{n+1}. \tag{39}$$

Rewriting this using block matrices produces

$$\mathbf{u}^{n+1} = \mathbf{u}^\star - \boldsymbol{\beta}^{-1}\begin{pmatrix}\mathbf{G}^T\\ \mathbf{G}_\mu\\ \hat{\mathbf{C}}\mathbf{H}\end{pmatrix}^T\begin{pmatrix}\tilde{\mathbf{p}}\\ \mathbf{G}_\mu\mathbf{u}^{n+1}\\ \hat{\mathbf{C}}\mathbf{H}\mathbf{u}^{n+1}\end{pmatrix}. \tag{40}$$

Eq. (40) can be written more concisely as

$$\mathbf{u}^{n+1} = \mathbf{u}^\star - \boldsymbol{\beta}^{-1}\mathbf{K}^T\mathbf{z}, \tag{41}$$

where

$$\mathbf{K} = \begin{pmatrix} \mathbf{G}^T \\ \mathbf{G}_\mu \\ \hat{\mathbf{C}}\mathbf{H} \end{pmatrix} \qquad \mathbf{z} = \begin{pmatrix} \tilde{\mathbf{p}} \\ \mathbf{G}_\mu\mathbf{u}^{n+1} \\ \hat{\mathbf{C}}\mathbf{H}\mathbf{u}^{n+1} \end{pmatrix}. \tag{42}$$

Left multiplying Eq. (41) by $\mathbf{K}$ and rearranging produces

$$\mathbf{K}\mathbf{u}^{n+1} + \mathbf{K}\boldsymbol{\beta}^{-1}\mathbf{K}^T\mathbf{z} = \mathbf{K}\mathbf{u}^\star. \tag{43}$$

This can be simplified by using Eq. (7) and noting that

$$\mathbf{K}\mathbf{u}^{n+1} = \begin{pmatrix} \mathbf{G}^T \\ \mathbf{G}_\mu \\ \hat{\mathbf{C}}\mathbf{H} \end{pmatrix}\mathbf{u}^{n+1} = \begin{pmatrix} \mathbf{0} \\ \mathbf{G}_\mu\mathbf{u}^{n+1} \\ \hat{\mathbf{C}}\mathbf{H}\mathbf{u}^{n+1} \end{pmatrix} = \begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{pmatrix}\begin{pmatrix} \tilde{\mathbf{p}} \\ \mathbf{G}_\mu\mathbf{u}^{n+1} \\ \hat{\mathbf{C}}\mathbf{H}\mathbf{u}^{n+1} \end{pmatrix} = \mathbf{P}\mathbf{z}, \tag{44}$$

where

$$\mathbf{P} = \begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{pmatrix} \tag{45}$$

and $\mathbf{I}$ is the identity matrix. Substituting Eq. (44) into Eq. (43) produces the final SPD system

$$(\mathbf{P} + \mathbf{K}\boldsymbol{\beta}^{-1}\mathbf{K}^T)\mathbf{z} = \mathbf{K}\mathbf{u}^\star. \tag{46}$$

This system is symmetric positive semidefinite since it is the sum of two symmetric positive semidefinite matrices. Note that to make this system SPD, the unknowns must be changed, and this requires that the damping forces factor. The viscosity factors in a straightforward way. For many force models, including the surface tension model proposed in this paper, the matrix $\hat{\mathbf{D}}$ factors in a straightforward way as well. Solving this system produces $\mathbf{z}$, and the desired result $\mathbf{u}^{n+1}$ can be obtained by substituting into Eq. (41).

It is possible to characterize its possible nullspace. To do this, assume $(\mathbf{P} + \mathbf{K}\boldsymbol{\beta}^{-1}\mathbf{K}^T)\mathbf{z} = \mathbf{0}$. Semidefiniteness of each part implies $\mathbf{P}\mathbf{z} = \mathbf{0}$ and $\mathbf{K}\boldsymbol{\beta}^{-1}\mathbf{K}^T\mathbf{z} = \mathbf{0}$. Since $\boldsymbol{\beta}^{-1}$ is nonsingular, it must be that $\mathbf{K}^T\mathbf{z} = \mathbf{0}$. The requirement $\mathbf{P}\mathbf{z} = \mathbf{0}$ leads to

$$\mathbf{z} = \begin{pmatrix} \tilde{\mathbf{p}} \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix}. \tag{47}$$

The second requirement implies that $\mathbf{G}\tilde{\mathbf{p}} = \mathbf{0}$, which may occur when all Neumann boundary conditions are imposed. That is, our new system will have a nullspace under the same circumstances as a one-phase incompressible simulation. Note that if $\mathbf{z}$ is a nullspace vector, then $\mathbf{z}^T\mathbf{K}\mathbf{u}^\star = \mathbf{0}$, so that Eq. (46) remains consistent even when the system is singular. That is, the right hand side is in the range of the system, so no projection of the right hand side is required.

2.7. Comments on Extensions to Hybrid Solids

The framework of [42] provides a means of applying forces at locations other than the locations of the actual degrees of freedom. It does this by introducing a notion of a hard bound particle, which is a particle whose position and velocity are defined implicitly as a linear combination of the actual degrees of freedom. This makes it possible, for example, to embed a detailed mesh for resolving collisions inside a coarser and better-conditioned simulation mesh. If $\mathbf{H}$ is the operator that interpolates from velocity degrees of freedom to the hard bound velocities then the force felt at velocity degrees of freedom due to forces on the hard bound particles is $\mathbf{H}^T\hat{\mathbf{f}} + \mathbf{H}^T\hat{\mathbf{D}}\mathbf{H}\hat{\mathbf{v}}$, where $\hat{\mathbf{v}}$ are their Lagrangian velocity degrees of freedom. In each case, the force is applied by interpolating velocities from real degrees of freedom to extra degrees of freedom using $\mathbf{H}$,

computing the force over the extra degrees of freedom, and then distributing forces back to the real degrees of freedom using $\mathbf{H}$. What is different in this case is that our real degrees of freedom are Eulerian rather than Lagrangian.

There is nothing special about defining Lagrangian degrees of freedom in terms of other degrees of freedom. Indeed, one could also define Eulerian velocity degrees of freedom in terms of other Eulerian degrees of freedom or in terms of Lagrangian degrees of freedom. One of the applications proposed in [42] for the purely Lagrangian form is to stitch together meshes with T-junctions. In the context of a purely Eulerian simulation, one could use it, for example, to simplify the discretization of adaptive methods. Thus, for an octree discretization, each refinement level could be discretized independently. Then, where different levels meet, the coarse face velocities can be defined implicitly in terms of the fine velocity degrees of freedom. The interpolation in this case can simply be chosen so that the flux measured by the coarse or fine degrees of freedom agree. This effectively decouples the problem of operator discretization from that of managing degrees of freedom. Symmetry and definiteness are obtained very easily.

Similarly, one might consider defining Eulerian degrees of freedom implicitly in terms of Lagrangian degrees of freedom. One might, for example, treat the particles of a fluid implicit particle (FLIP) scheme as being the real degrees of freedom, with the Eulerian velocities required by the pressure projection step defined implicitly in terms of the particle degrees of freedom. In this way, the composition of the fluid interpolation operator and the divergence operator effectively defines a discretization of a Lagrangian divergence operator over the particle degrees of freedom.

3. Spatial Discretization

Up to this point, we have been working in discrete form but have only described the temporal discretization. In this section, we address our spatial discretization. Evolving the proposed scheme requires the discretization of $\boldsymbol{\beta}$, $\mathbf{G}$, $\mathbf{H}$, $\hat{\mathbf{C}}$, and $\tilde{\mathbf{G}}_\mu$. We consider both one-phase and two-phase discretizations. Because the extension to two-phase is straightforward, the one-phase case is considered, and differences in the two-phase case are discussed as appropriate. In the one-phase case, the fluid region is bounded by a free surface with a constant exterior Dirichlet pressure boundary condition and a stress-free Neumann velocity boundary condition.

In the two-phase case, the velocity field is assumed to be continuous across the interface. This simplifies advection and the discretization of $\tilde{\mathbf{G}}_\mu$, neither of which are the focus of this paper. This assumption is not a limitation of the proposed technique, as the discretization of $\mathbf{G}$ very naturally handles a discontinuous velocity field, and the remaining operators do not depend on the velocity discretization.

Neither advection nor viscosity play an important role in the proposed algorithm, so their discretizations are not particularly important. Our advective term is discretized using 3^{rd} order Hamilton Jacobi ENO [21, 40, 41]. For viscosity in the one-phase case, a standard central differencing discretization of $\tilde{\mathbf{G}}_\mu$ is applied with a simple first order treatment of the Neumann boundary condition at the interface. In the two-phase case, the velocity field is continuous and standard central differencing is used throughout. The operators that remain to be discretized are $\boldsymbol{\beta}$ (Section 3.4), $\mathbf{G}$ (Section 3.4), $\mathbf{H}$ (Section 3.5), and $\hat{\mathbf{C}}$ (Section 3.2).

3.1. Lagrangian Degrees of Freedom

In this paper a Lagrangian discretization is used for the purpose of computing surface tension, and therefore require a discretization of the surface. The details of the surface mesh are not very important, though for simplicity in discretization both the fluid grid and surface mesh are assumed to be fine enough to resolve the interface. Two different approaches to maintaining this surface discretization are considered, each suited to a different underlying interface representation.

3.1.1. Front Tracking

The first approach is to use a front tracking surface mesh [50]. The positions of the interface particles are evolved using $\hat{\mathbf{x}}^{n+1} = \hat{\mathbf{x}}^n + \Delta t \hat{\mathbf{v}}^{n+1}$, where $\hat{\mathbf{v}}^{n+1} = \mathbf{H}\mathbf{u}^{n+1}$. In this case, the time evolution of the interface

closely resembles backward Euler. If advection and forces on the Eulerian degrees of freedom are ignored, the fluid update looks like

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t \boldsymbol{\beta}^{-1}\mathbf{H}^T\hat{\mathbf{f}}_e + \Delta t \boldsymbol{\beta}^{-1}\mathbf{H}^T\hat{\mathbf{D}}\mathbf{H}\mathbf{u}^{n+1}. \tag{48}$$

Multiplying by $\mathbf{H}$ produces

$$\mathbf{H}\mathbf{u}^{n+1} = \mathbf{H}\mathbf{u}^n + \Delta t \mathbf{H}\boldsymbol{\beta}^{-1}\mathbf{H}^T\hat{\mathbf{f}}_e + \Delta t \mathbf{H}\boldsymbol{\beta}^{-1}\mathbf{H}^T\hat{\mathbf{D}}\mathbf{H}\mathbf{u}^{n+1}. \tag{49}$$

Let $\hat{\mathbf{M}}^{-1} = \mathbf{H}\boldsymbol{\beta}^{-1}\mathbf{H}^T$ be the effective mass (inverse) of the structure. Making the approximation $\hat{\mathbf{v}}^n = \mathbf{H}\mathbf{u}^n$ (this is not strictly true since $\mathbf{H}$ changes each time step as the structure moves through the Eulerian grid) produces

$$\hat{\mathbf{v}}^{n+1} = \hat{\mathbf{v}}^n + \Delta t\hat{\mathbf{M}}^{-1}(\hat{\mathbf{f}}_e + \hat{\mathbf{D}}\hat{\mathbf{v}}^{n+1}) = \hat{\mathbf{v}}^n + \Delta t\hat{\mathbf{M}}^{-1}\hat{\mathbf{f}}^{n+1}. \tag{50}$$

This, along with $\hat{\mathbf{x}}^{n+1} = \hat{\mathbf{x}}^n + \Delta t\hat{\mathbf{v}}^{n+1}$, is a backward Euler update for a structure which has surface tension forces and an effective mass.

A level set is rebuilt from the front tracked surface mesh after each time step to simplify the discretization of other operators. Since only inside-outside tests will be required of the level set in this case, the signed distance is computed from the surface mesh exactly for the band $|\phi| < \Delta x$ to ensure consistency between the two representations. The remainder of the level set is computed using fast marching [51, 52, 38, 22, 39, 8].

In order to maintain a valid and good quality surface mesh, remeshing is performed whenever necessary. The remeshing in triggered when any segment has length bigger than $1.5\Delta x$ or smaller than $.5\Delta x$, where Δx is the size of an Eulerian grid cell. Then a long segment will be split into two segments by adding a new point through a third order stencil, $-\frac{1}{16}X_{-1} + \frac{9}{16}X_0 + \frac{9}{16}X_1 - \frac{1}{16}X_2$. And short segments will be merged into adjacent segments by deleting points in between until the resulting length is larger than the lower bound.

3.1.2. Level Set Tracked

The second approach considered for maintaining the surface mesh is to compute it from a level set representation of the interface at each time step. In this case, the particle level set method [12] is used to evolve the level set. The method used to compute the surface mesh from the level set is similar to marching cubes [34], but with careful attention paid to accuracy and surface mesh quality. Variations for computing surface meshes whose nodes are third and fourth order accurate are described. Since the surface mesh will be used to compute curvature, which is a second derivative, third order accuracy is required to obtain a first order accurate curvature. We use the fourth order discretization in all of our examples, but we demonstrate the accuracy of both variations in Section 4.1.

It is convenient to perform marching cubes such that the squares coincide with the fluid grid cells, since this will produce a mesh that is very convenient for the discretization of $\mathbf{G}$. Unfortunately, the fluid level set lives at cell centers and must be averaged to nodes. We use the standard fourth order stencil in Figure 1(a),

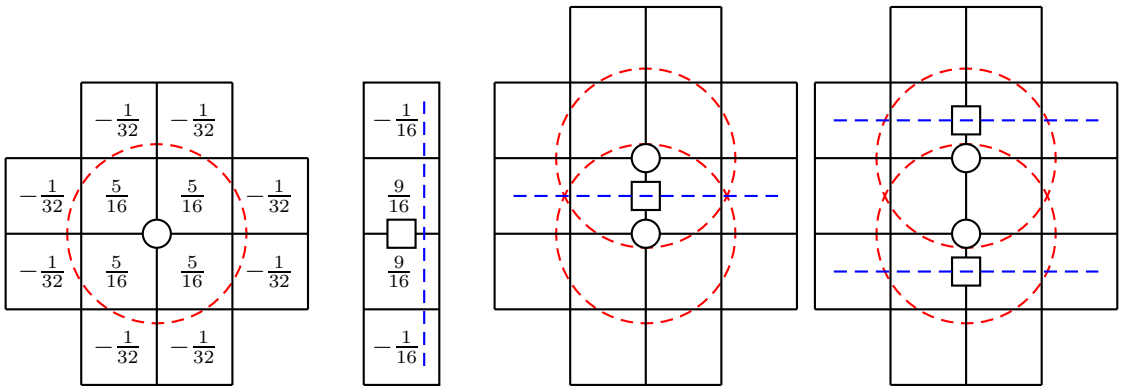
$$\begin{aligned}\phi_{j+\frac{1}{2},k+\frac{1}{2}} = &-\frac{1}{32}(\phi_{j-1,k} + \phi_{j+2,k} + \phi_{j-1,k+1} + \phi_{j+2,k+1} + \phi_{j,k-1} + \phi_{j+1,k-1} + \phi_{j,k+2} + \phi_{j+1,k+2}) \\ &+ \frac{5}{16}(\phi_{j,k} + \phi_{j+1,k} + \phi_{j,k+1} + \phi_{j+1,k+1}).\end{aligned} \tag{51}$$

These node-centered level set values are used to determine the location of level set crossings and also as part of the computation of the location of those crossings.

The location of the level set crossing could be computed by constructing an interpolating polynomial from the node-centered level set. This results in an unnecessarily large stencil, which we would like to avoid. Instead one can interpolate the cell-centered level set to grid faces using the stencil in Figure 1(b)

$$\phi_{j+\frac{1}{2},k} = -\frac{1}{16}(\phi_{j-1,k} + \phi_{j+2,k}) + \frac{9}{16}(\phi_{j,k} + \phi_{j+1,k}). \tag{52}$$

to provide additional samples for computing an interpolation polynomial.



(a) Node stencil (b) Face Stencil (c) Third order stencil (d) Fourth order stencil

Figure 1: Level set samples are located at cell centers. These stencils are used to interpolate the level set to nodes (indicated with circles) and faces (indicated with squares) with the weights shown in the first two diagrams. The other two diagrams show stencils suitable for computing the location of a level set crossing along a face. The signs at the circled nodes are used to determine whether a crossing exists. The level set values averaged to the circled and squared locations are used to construct an interpolating polynomial. The level set crossing is taken to be the zero of this polynomial. The dashed lines and dashed circles indicate the cells used to compute each interpolated value.

We provide an interpolating polynomial for third order and fourth order. The third order accurate crossing is obtained from the quadratic formed by interpolating the three points shown in Figure 1(c)

$$\left(0, \phi_{j+\frac{1}{2},k-\frac{1}{2}}\right) \qquad \left(\frac{1}{2}, \phi_{j+\frac{1}{2},k}\right) \qquad \left(1, \phi_{j+\frac{1}{2},k+\frac{1}{2}}\right). \tag{53}$$

The third order crossing is obtained by locating a root θ of the interpolating polynomial in the interval $[0, 1]$. The crossing occurs at the position $(1-\theta)\hat{\mathbf{x}}_{j+\frac{1}{2},k-\frac{1}{2}} + \theta\hat{\mathbf{x}}_{j+\frac{1}{2},k+\frac{1}{2}}$. The fourth order accurate crossing is a obtained from the cubic formed by interpolating the four points shown in Figure 1(d)

$$\left(-\frac{1}{2}, \phi_{j+\frac{1}{2},k-1}\right) \qquad \left(0, \phi_{j+\frac{1}{2},k-\frac{1}{2}}\right) \qquad \left(1, \phi_{j+\frac{1}{2},k+\frac{1}{2}}\right) \qquad \left(\frac{3}{2}, \phi_{j+\frac{1}{2},k+1}\right). \tag{54}$$

The location of the crossing is similarly obtained by locating a root of the interpolating polynomial.

Although two node-centered and three face-centered level set values are available, they cannot be used to construct a fifth order accurate estimate of the level set crossing since the interpolation of the cell-centered level set to nodes and faces is only fourth order accurate. There are more face-centered, interpolated level set values available than are needed, so the stencils are chosen to be symmetric for simplicity and to avoid possible biasing. Note that using the two averaged node values in constructing the interpolating polynomial guarantees that a polynomial root will be found whenever the sign tests indicate that a face is crossed by the interface. A bracketed secant method was used to compute the interpolating polynomial roots.

The resulting mesh is not quite suitable for use with our curvature discretization. The surface may contain sliver elements, which the curvature discretization is sensitive to, a point which is discussed in more detail in Section 3.2.1. This is handled by eliminating surface elements whose length is less than $0.1\Delta x$. Such elements are merged with a neighboring element by removing an endpoint.

3.2. Lagrangian Surface Tension

Let $\hat{\mathbf{x}}_{k-1}$, $\hat{\mathbf{x}}_k$, and $\hat{\mathbf{x}}_{k+1}$ be three consecutive vertices on the Lagrangian surface mesh. These points conceptually lie on a smooth curve $\boldsymbol{\gamma}(s)$. The curvature of this curve is then given by $\boldsymbol{\gamma}'' = \kappa\mathbf{n}$. Let $\ell_{k-\frac{1}{2}} = \|\hat{\mathbf{x}}_k - \hat{\mathbf{x}}_{k-1}\|$ and $\ell_{k+\frac{1}{2}} = \|\hat{\mathbf{x}}_{k+1} - \hat{\mathbf{x}}_k\|$ be the lengths of the edges adjacent to $\hat{\mathbf{x}}_k$. It will also be convenient to associate some portion of the surface with each of the vertices on the Lagrangian surface mesh.

We adopt the convention that a length $\ell_k = \frac{\ell_{k-\frac{1}{2}} + \ell_{k+\frac{1}{2}}}{2}$ of the surface mesh is associated with point $\hat{\mathbf{x}}_k$. With these definitions, the curvature can be discretized using

$$(\kappa\hat{\mathbf{n}})_k = (\boldsymbol{\gamma}'')_k = \frac{\frac{\hat{\mathbf{x}}_{k+1} - \hat{\mathbf{x}}_k}{\ell_{k+\frac{1}{2}}} - \frac{\hat{\mathbf{x}}_k - \hat{\mathbf{x}}_{k-1}}{\ell_{k-\frac{1}{2}}}}{\ell_k} \tag{55}$$

This quantity is also known as the surface Laplacian. Since surface tension results in a pressure jump $[p] = \sigma\kappa$ across the interface, the force due to surface tension will be the pressure times the surface area, or

$$\hat{\mathbf{f}}_k = \sigma\ell_k(\kappa\hat{\mathbf{n}})_k = \sigma\left(\frac{\hat{\mathbf{x}}_{k+1} - \hat{\mathbf{x}}_k}{\ell_{k+\frac{1}{2}}} - \frac{\hat{\mathbf{x}}_k - \hat{\mathbf{x}}_{k-1}}{\ell_{k-\frac{1}{2}}}\right) \tag{56}$$

Let $\hat{\mathbf{f}} = \hat{\mathbf{D}}\hat{\mathbf{x}}$, where $\hat{\mathbf{f}}$ and $\hat{\mathbf{x}}$ are the vectors of forces and positions and $\hat{\mathbf{D}}$ is the matrix given by

$$\hat{\mathbf{D}}_{k,k} = -\sigma\left(\ell^{-1}_{k+\frac{1}{2}} + \ell^{-1}_{k+\frac{1}{2}}\right)\mathbf{I} \qquad \hat{\mathbf{D}}_{k,k+1} = \hat{\mathbf{D}}_{k+1,k} = \sigma\ell^{-1}_{k+\frac{1}{2}}\mathbf{I} \tag{57}$$

with $\hat{\mathbf{D}}_{i,j} = 0$ otherwise. Note that $\hat{\mathbf{D}}$ depends on $\hat{\mathbf{x}}$ through edge lengths. The matrix $\hat{\mathbf{D}}$ is a symmetric, and one can see that $\hat{\mathbf{D}}$ is in fact negative semidefinite by noting that it can be written as the sum of force contributions from individual segments,

$$\begin{pmatrix} \hat{\mathbf{f}}_k \\ \hat{\mathbf{f}}_{k+1} \end{pmatrix} = -\sigma\ell^{-1}_{k+\frac{1}{2}} \begin{pmatrix} \mathbf{I} & -\mathbf{I} \\ -\mathbf{I} & \mathbf{I} \end{pmatrix} \begin{pmatrix} \hat{\mathbf{x}}_k \\ \hat{\mathbf{x}}_{k+1} \end{pmatrix}. \tag{58}$$

These matrices are symmetric negative semidefinite, so their sum $\hat{\mathbf{D}}$ must also be.

This per-element form of the force is particularly convenient. In the remainder of this section, we focus on a single surface segment and drop indices where no confusion will result. Let ϕ be the potential energy due to surface tension for the element. Recalling that $\ell = \|\hat{\mathbf{x}}_{k+1} - \hat{\mathbf{x}}_k\|$, one may define a tangential direction for the element $\mathbf{t} = \ell^{-1}(\hat{\mathbf{x}}_{k+1} - \hat{\mathbf{x}}_k)$. The forces due to this segment are related to the potential by $\hat{\mathbf{f}}_k = -\frac{\partial\phi}{\partial\hat{\mathbf{x}}_k}$ and $\hat{\mathbf{f}}_{k+1} = -\frac{\partial\phi}{\partial\hat{\mathbf{x}}_k}$. Finally, the potential energy and forces can be written and computed very concisely as

$$\phi = \sigma\ell \qquad \hat{\mathbf{f}}_{k+1} = -\hat{\mathbf{f}}_k = -\sigma\mathbf{t} \tag{59}$$

That is, the potential energy due to surface tension is just a constant times the surface area. As one would expect, this potential energy is bounded from below. It is interesting to note that the 2D surface tension force has a form very similar to that of a zero-rest-length spring, whose potential energy is proportional to its length squared. The force derivatives are given by

$$\frac{\partial\hat{\mathbf{f}}_k}{\partial\hat{\mathbf{x}}_k} = \frac{\partial\hat{\mathbf{f}}_{k+1}}{\partial\hat{\mathbf{x}}_{k+1}} = -\frac{\partial\hat{\mathbf{f}}_{k+1}}{\partial\hat{\mathbf{x}}_k} = -\frac{\partial\hat{\mathbf{f}}_k}{\partial\hat{\mathbf{x}}_{k+1}} = -\frac{\sigma}{\ell}(\mathbf{I} - \mathbf{t}\mathbf{t}^T) = -\frac{\sigma}{\ell}\mathbf{n}\mathbf{n}^T, \tag{60}$$

where $\mathbf{n}$ is the normal vector defined to be orthogonal to $\mathbf{t}$. This resulting system is symmetric and negative definite. The force derivatives project out and tangential velocity component before computing damping, and the resulting force acts only in the normal direction. Interestingly, a frozen-coefficient treatment of edge lengths threats $\hat{\mathbf{D}}$ as constant. This leads to $\hat{\mathbf{D}}$ as an approximation to the force derivatives, which will damp motion in the tangential direction and apply forces with both normal and tangential components. Though not the correct force derivatives, this extra damping may actually be desirable in practice. Since our scheme for handling surface tension uses pressures (see Section 3.5 for more details on why this is so) and thus effectively discards tangential velocity. This prevents the Lagrangian surface from experiencing tangential velocity, so tangential damping is unnecessary. As such, the correct force derivatives are used in our discretization.

To summarize, a fully implicit treatment and a fully explicit treatment would both compute $\hat{\mathbf{f}}_e$ from Eq. (15) as

$$(\hat{\mathbf{f}}_e)_k = \sigma \mathbf{t}_{k+\frac{1}{2}} - \sigma \mathbf{t}_{k-\frac{1}{2}} \qquad \ell_{k+\frac{1}{2}} = \|\hat{\mathbf{x}}_{k+1} - \hat{\mathbf{x}}_k\| \qquad \mathbf{t}_{k+\frac{1}{2}} = \frac{\hat{\mathbf{x}}_{k+1} - \hat{\mathbf{x}}_k}{\ell_{k+\frac{1}{2}}}. \tag{61}$$

A fully implicit treatment would compute $\hat{\mathbf{D}}$ from Eq. (16) as

$$\hat{\mathbf{D}}_{k,k} = -\frac{\Delta t \sigma}{\ell_{k-\frac{1}{2}}} \mathbf{n}_{k-\frac{1}{2}} \mathbf{n}_{k-\frac{1}{2}}^T - \frac{\Delta t \sigma}{\ell_{k+\frac{1}{2}}} \mathbf{n}_{k+\frac{1}{2}} \mathbf{n}_{k+\frac{1}{2}}^T \qquad \hat{\mathbf{D}}_{k,k+1} = \hat{\mathbf{D}}_{k+1,k} = \frac{\Delta t \sigma}{\ell_{k+\frac{1}{2}}} \mathbf{n}_{k+\frac{1}{2}} \mathbf{n}_{k+\frac{1}{2}}^T \tag{62}$$

where

$$\mathbf{n}_{k+\frac{1}{2}} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \mathbf{t}_{k+\frac{1}{2}}. \tag{63}$$

A fully implicit treatment would compute $\hat{\mathbf{C}}$ from Section 2.6 as

$$\hat{\mathbf{C}}_{k,k} = -\hat{\mathbf{C}}_{k,k+1} = \Delta t \sqrt{\frac{\sigma}{\ell_{k+\frac{1}{2}}}} \mathbf{n}_{k+\frac{1}{2}}^T \tag{64}$$

A fully explicit treatment would compute $\hat{\mathbf{D}} = \mathbf{0}$ and $\hat{\mathbf{C}} = \mathbf{0}$, making it less stable.

3.2.1. Discretization Accuracy

Consider again the three points on the smooth curve $\boldsymbol{\gamma}(s)$, where s is taken to be the arc length parameter. Then, $\|\boldsymbol{\gamma}'\| = 1$ and $\kappa = |\boldsymbol{\gamma}''|$. Assume that $\hat{\mathbf{x}}_{k-1} = \boldsymbol{\gamma}(-\delta)$, $\hat{\mathbf{x}}_k = \boldsymbol{\gamma}(0)$, and $\hat{\mathbf{x}}_{k+1} = \boldsymbol{\gamma}(\beta\delta)$. Here, β is fixed and the refinement $\delta \to 0$ is considered. This corresponds roughly to choosing edge lengths which differ by a factor of β. The first few terms of the Taylor series expansion of the discretization in Eq. (55) are

$$\begin{aligned}(\kappa\hat{\mathbf{n}})_k \;\approx\; & \kappa\mathbf{n} + \frac{\beta-1}{12}(4\dot{\kappa}\mathbf{n} + 3\kappa^2\mathbf{t})\delta + \frac{\beta^2-\beta+1}{12}(2\dot{\kappa}\kappa\mathbf{t} - \ddot{\kappa}\mathbf{n})\delta^2 \\ & + \left(\frac{\beta-1}{576}\big(3(\beta-1)^2\kappa^4 + 8(\beta^2+1)(3\kappa\ddot{\kappa} + 2\dot{\kappa}^2)\big)\mathbf{t} - \frac{\beta-1}{360}\big((4\beta^2-5\beta+4)\dot{\kappa}\kappa + 6(\beta^2+1)\dddot{\kappa}\big)\mathbf{n}\right)\delta^3,\end{aligned}$$

where that $\mathbf{n}$ and $\mathbf{t}$ are the normal and tangential directions. There are a few useful observations to make from this Taylor series. The discretization is first order accurate. When $\beta = 1$, so that the surface mesh is uniformly spaced, the first and third order terms vanish, and the discretization is second order accurate. In particular, the error coefficient will be reduced if the discretization is approximately uniform.

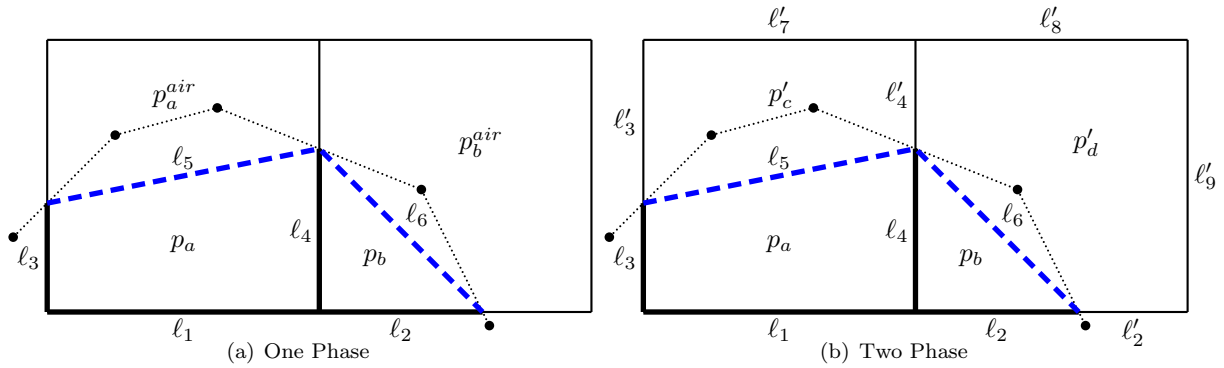
The special case of a circle is quite interesting. In that case, κ is constant, so that $\dot{\kappa} = \ddot{\kappa} = \dddot{\kappa} = 0$. The series simplifies significantly to

$$(\kappa\hat{\mathbf{n}})_k \approx \kappa\mathbf{n} + \frac{\beta-1}{4}\kappa^2\delta\mathbf{t} + \frac{(\beta-1)^3}{192}\kappa^4\delta^3\mathbf{t}. \tag{65}$$

If the discretization is uniform, these error terms vanish. What is more, the errors are purely in the tangential direction. Since our discretization of $\mathbf{H}$ does not transmit (discretized) tangential impulses, the resulting tangential force errors do not feed significantly into the rest of the simulation. This additional accuracy in the case of a circle is observed numerically, and our curvature estimates for a circle correct to roundoff error for a completely uniform discretization whose vertices lie exactly on the circle.

Next the sensitivity of the proposed discretization to errors in particle locations is considered. Errors made along the surface mesh correspond approximately to choosing a different β. Since no attempt is made to keep $\beta = 1$, such errors are not a significant problem. This leaves only the sensitivity of the curvature computation to errors in the normal direction. Consider that three points are chosen on a circle, but the middle point is displaced slightly in the normal direction. That is

$$\hat{\mathbf{x}}_{k-1} = \begin{pmatrix} r\cos\theta \\ -r\sin\theta \end{pmatrix} \qquad \hat{\mathbf{x}}_k = \begin{pmatrix} r(1+\epsilon) \\ 0 \end{pmatrix} \qquad \hat{\mathbf{x}}_{k+1} = \begin{pmatrix} r\cos\phi \\ r\sin\phi \end{pmatrix}. \tag{66}$$



(a) One Phase (b) Two Phase

Figure 2: Two cells cut by a Lagrangian surface mesh. A face is constructed in each cut cell (shown as a dashed line) connecting the point where the surface mesh enters the cell and the point where it leaves. These two faces are referred to as coupling faces and have lengths ℓ_5 and ℓ_6. There are three cut faces with lengths ℓ_2, ℓ_3, and ℓ_4. Finally, there is one interior face with length $\ell_1 = \Delta x$. Each cut cell has a pressure, which in this case are labeled p_a and p_b. In the one phase case, there are Dirichlet pressure boundary conditions p_a^{air} and p_b^{air} outside the coupling faces. In the two phase case, there are additional pressure degrees of freedom in the outside portion of each cut cell instead, labelled p_c and p_d, and there are additionally three exterior faces with lengths $\ell_7' = \ell_8' = \ell_9' = \Delta x$.

As before, it is assume that $\phi = \beta\theta$, where β is held fixed and θ and ϵ are both very small. Then

$$(\kappa\hat{\mathbf{n}})_k \approx \left(\frac{1}{r} + \frac{\epsilon}{\beta r}\left(\frac{2}{\theta^2} + \frac{\beta^2 - 12\beta + 1}{12} + \frac{(7\beta^4 + 10\beta^2 + 7)\theta^2}{2880}\right)\right)\mathbf{n} + \frac{(1-\beta)(1-\epsilon)\theta}{192r}\left(48 + (1-\beta)^2\theta^2\right)\mathbf{t}$$

Of particular importance is the term $\frac{2\epsilon}{\beta r\theta^2}$. Convergence is only possible if this term vanishes under refinement. Since the surface is a circle of radius r, the angle θ is related to the Eulerian grid spacing Δx by $r\theta \approx \Delta x$. Then, the error term takes the form $\frac{2\epsilon r}{\beta \Delta x^2}$. Since we require the error to be of order $O(\Delta x)$ or better for convergence, it is necessary for $\epsilon = O(\beta \Delta x^3)$. This is the level of sensitivity to Δx that one would expect from a second derivative. The tolerable error is also sensitive to the irregularity β of the discretization, and very small elements will result in poor accuracy. For this reason, it is important to avoid sliver elements in our boundary mesh.

Our discretization of curvature can be obtained from the potential energy in Eq. (59). The Taylor series expansion of the potential energy discretization along the portion of the curve between $\boldsymbol{\gamma}(0)$ and $\boldsymbol{\gamma}(s)$ is

$$\phi \approx \sigma s - \frac{1}{24}\sigma\kappa^2 s^3 - \frac{1}{24}\sigma\kappa\dot{\kappa}s^4. \tag{67}$$

Since $\boldsymbol{\gamma}(s)$ is assumed to be parameterized in terms of its arc length, the exact length of this portion of the curve is s. Thus, the estimate of surface area is second order accurate, which leads to a first order curvature force. This suggests that if a more accurate curvature force computation is desired, a third or higher order approximation of the surface area should be sought.

3.3. Eulerian Degrees of Freedom

Our Eulerian discretization is based on a standard MAC grid discretization, but it is altered slightly near the interface. We refer to any cell through which the interface passes as a cut cell, and we refer to any face through which the interface passes as a cut face. We refer to cells/faces entirely inside/outside the fluid as interior/exterior cells/faces. In addition to this, additional fluid degrees of freedom are defined that lie approximately on the interface, and we will refer to these as coupling faces. These degrees of freedom are illustrated in Figure 2.

One Phase. In the one phase case, each interior cell and each cut cell contains a pressure degree of freedom, and each interior face (Figure 2, ℓ_1), cut face (Figure 2, ℓ_2, ℓ_3, and ℓ_4), and coupling face (Figure 2, ℓ_5 and ℓ_6) contain one velocity degree of freedom. In the case of coupling faces, the velocity degree of freedom is taken to be directed in the outward normal direction. Outside the coupling faces, there is a Dirichlet pressure.

Two Phase. In the two phase case, each cut cell contains an inside pressure and an outside pressure. All other cells contain one pressure degree of freedom. Every grid face and every coupling face contains one velocity degree of freedom. Note that one may allow the velocity field to support a discontinuity in tangential velocity by simply providing each cut face with an inside velocity and an outside velocity in much the same way as is done for pressure. Since surface tension will be applied to the coupling faces, there will be a pressure jump across this face. Thus, one cannot assume a continuous pressure field, and cut cells must contain separate inside and outside pressures. As in the one phase case, the velocity degree of freedom for a coupling face is taken to be directed in the outward normal direction.

3.4. Gradient and Fluid Mass

The volume weighted gradient $\mathbf{G}$ and the fluid mass β are required for our discretization. First consider the two phase case. The mass for a interior or exterior faces is just ρV, where V is the cell volume and ρ is the density at that face. For a cut face with inside length and density ℓ and ρ and outside length and density ℓ' and ρ' the mass is $\beta = (\rho\ell + \rho'\ell')\Delta x$. For a coupling face with length ℓ, the mass will be $\beta = \frac{1}{2}\ell\Delta x(\rho + \rho')$. The one phase case is obtained by letting the exterior be massless, so that $\rho' = 0$. Intuitively, each face has a dual cell associated with it. Some portion of that dual cell may be in the inside region, and some portion may be in the outside region. The masses above correspond to dividing the dual cell up into inside and outside portions, simplifying things by making all the pieces rectangular. This discretization of mass is only first order accurate, since it double-counts mass on the boundary and treats the boundary as linear rather than computing the mass based on the curvature. A higher order discretization of mass will require a more detailed and careful treatment.

The weights on the volume-weighted gradient stencil are just the lengths of the faces. The stencil entry G_{fc} is positive if the velocity u_f represents flow into cell c and negative otherwise. One can think of this as corresponding to a constant pressure in each (possibly cut) cell. Alternatively, one can think of this instead in terms of the divergence operator, where the faces are now used to compute the flux into the (possibly cut) cell.

As an example in the one-phase case, consider the discretization in Figure 2(a), where faces and cells are labeled with the index of the length or pressure at that location in the figure. Then, $\beta_{55} = \frac{1}{2}\ell_5\rho_5\Delta x$, $\beta_{66} = \frac{1}{2}\ell_6\rho_6\Delta x$, and $\beta_{kk} = \ell_k\rho_k\Delta x$ for $1 \le k \le 4$. The volume-weighted gradient stencil entries are $G_{1a} = \ell_1 = \Delta x$, $G_{3a} = \ell_3$, $G_{4a} = -\ell_4$, $G_{5a} = -\ell_5$, $G_{2b} = \ell_2$, $G_{4b} = \ell_4$, and $G_{6b} = -\ell_6$.

As a two-phase example, consider the discretization in Figure 2(b). Here, primes indicate quantities corresponding to the outside phase and subscripts indicate velocity or pressure degrees of freedom. Then, $\beta_{55} = \frac{1}{2}\ell_5(\rho_5 + \rho'_5)\Delta x$, $\beta_{66} = \frac{1}{2}\ell_6(\rho_6 + \rho'_6)\Delta x$, $\beta_{11} = \ell_1\rho_1\Delta x$, $\beta_{kk} = (\ell_k\rho_k + \ell'_k\rho'_k)\Delta x$ for $2 \le k \le 4$, and $\beta_{kk} = \ell'_k\rho'_k\Delta x$ for $7 \le k \le 9$. The gradient stencil entries are $G_{1a} = \ell_1 = \Delta x$, $G_{3a} = \ell_3$, $G_{4a} = -\ell_4$, $G_{5a} = -\ell_5$, $G_{2b} = \ell_2$, $G_{4b} = \ell_4$, $G_{6b} = -\ell_6$, $G_{3c} = \ell'_3$, $G_{4c} = -\ell'_4$, $G_{5c} = \ell'_5$, $G_{7c} = -\ell'_7 = -\Delta x$, $G_{2d} = \ell'_2$, $G_{4d} = \ell'_4$, $G_{6d} = \ell'_6$, $G_{8d} = -\ell'_8 = -\Delta x$, and $G_{9d} = -\ell'_9 = -\Delta x$.

3.5. Eulerian-Lagrangian Interpolation

The role of the velocity interpolation operator $\mathbf{H}$ is to interpolate Eulerian velocity degrees of freedom to Lagrangian degrees of freedom, nothing that $\mathbf{H}^T$ serves the complementary role of distributing force from the Lagrangian degrees of freedom back to the Eulerian degrees of freedom. In principle, any interpolation operator could be used for this purpose, such as one constructed from delta functions. In practice, using such a discretization with surface tension produces large parasitic currents, and we are lead to formulate $\mathbf{H}$ with the special needs of surface tension in mind. If there are multiple forces on a Lagrangian mesh, each force can utilize its own discretization of $\mathbf{H}$, so formulating $\mathbf{H}$ specifically for one force is not problematic.

Formulating a discretization of surface tension that eliminates parasitic currents entirely is quite difficult. This task is well beyond the scope of interest, as acceptable results can be obtained even with parasitic currents, provided their magnitude is kept at a manageable level. There are many sources for parasitic currents. One source of parasitic currents is simply an inaccurate curvature computation [15]. While our curvature computation is in general only first order accurate, we have found it to produce acceptable results.

A more serious source of parasitic currents is related to the ability of the pressure to discretely balance out surface tension, and this is the issue we seek to address. Consider the case of a static circle, where a force $\mathbf{H}^T\hat{\mathbf{f}}$ is produced by surface tension, and a force $-\mathbf{G}\tilde{\mathbf{p}}$ is produced by the resulting pressure field to resist it. The net force on the fluid is $\mathbf{H}^T\hat{\mathbf{f}} - \mathbf{G}\tilde{\mathbf{p}}$. Since the pressure projection step can be formulated as a kinetic energy minimization problem [1], if the fluid is at rest and a suitable pressure field $\tilde{\mathbf{p}}$ exists such that $\mathbf{H}^T\hat{\mathbf{f}} = \mathbf{G}\tilde{\mathbf{p}}$, then the result of the pressure projection will be to choose such a pressure. The result would be to produce a zero velocity field. Such a $\tilde{\mathbf{p}}$ exists precisely when $\mathbf{H}^T\hat{\mathbf{f}}$ lies in the column space of $\mathbf{G}$. This suggests a formulation of the form $\mathbf{H}^T = \mathbf{G}\tilde{\mathbf{H}}^T$, for some operator $\tilde{\mathbf{H}}$, and this is the approach that was taken. In particular, the force distribution operator $\mathbf{H}^T$ is formulated directly, leaving the velocity interpolation operator to be defined by its transpose.

Since the Lagrangian force will be applied to the Eulerian degrees of freedom using the volume-weighted gradient operator $\mathbf{G}$, the Lagrangian force must be translated to a pseudo-pressure via $\tilde{\mathbf{H}}^T$. In fact, it is more straightforward to construct the gradient times pseudo-pressure directly on coupling faces. Our first step, then, is to define a force density everywhere on the surface. In Section 3.2, the particle $\hat{\mathbf{x}}_k$ is associated with a length of the surface equal to half its neighboring segments, so that the force density at the surface can be defined by

$$\hat{\mathbf{f}}_k^\rho = \frac{\hat{\mathbf{f}}_k}{\ell_k} = \sigma(\kappa\hat{\mathbf{n}})_k, \tag{68}$$

where as before

$$\ell_k = \frac{\ell_{k-\frac{1}{2}} + \ell_{k+\frac{1}{2}}}{2} = \frac{\|\hat{\mathbf{x}}_{k+1} - \hat{\mathbf{x}}_k\| + \|\hat{\mathbf{x}}_k - \hat{\mathbf{x}}_{k-1}\|}{2}. \tag{69}$$

The force $\hat{\mathbf{f}}_k$ was defined in Eq. (56) and $(\kappa\hat{\mathbf{n}})_k$ was defined in Eq. (55). Note that $\hat{\mathbf{f}}_k^\rho$ is a vectorial quantity rather than a pseudo-pressure. This force density $\hat{\mathbf{f}}^\rho$ is extended from the Lagrangian degrees of freedom to the entire surface mesh by linear interpolation.

The next step requires the per-cell coupling faces illustrated in Figure 2 as ℓ_5 and ℓ_6. Let $\mathbf{n}_i$ be the area-weighted surface normal for the coupling face in cell i. For cell i, let S_i be the list of segments k of the Lagrangian mesh (with endpoints $\hat{\mathbf{x}}_k$ and $\hat{\mathbf{x}}_{k+1}$) that intersect cell i, and let $\alpha_{k,i}$ and $\beta_{k,i}$, where $0 \le \alpha_{k,i} \le \beta_{k,i} \le 1$, be the barycentric coordinates on the segment k indicating the portion of the segment contained within the cell i. That is, if $\alpha_{k,i} = 0$, then $\hat{\mathbf{x}}_k$ is inside cell i. Otherwise, $\hat{\mathbf{x}}_k + \alpha_{k,i}(\hat{\mathbf{x}}_{k+1} - \hat{\mathbf{x}}_k)$ is on the boundary of cell i. Similarly, if $\beta_{k,i} = 1$, then $\hat{\mathbf{x}}_k$ is inside cell i. Otherwise, $\hat{\mathbf{x}}_{k+1} + \beta_{k,i}(\hat{\mathbf{x}}_k - \hat{\mathbf{x}}_{k+1})$ is on the boundary of cell i. With these barycentric weights, the coupling face normal $\mathbf{n}_i$ and its unit normal are easily computed as the sum of a subset of the individual area weighted normals from the Lagrangian mesh as

$$\mathbf{n}_i = \sum_{k \in S_i} (\beta_{k,i} - \alpha_{k,i})\mathbf{R}(\hat{\mathbf{x}}_{k+1} - \hat{\mathbf{x}}_k) \qquad \mathbf{R} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \qquad \tilde{\mathbf{n}}_i = \frac{\mathbf{n}_i}{\|\mathbf{n}_i\|}. \tag{70}$$

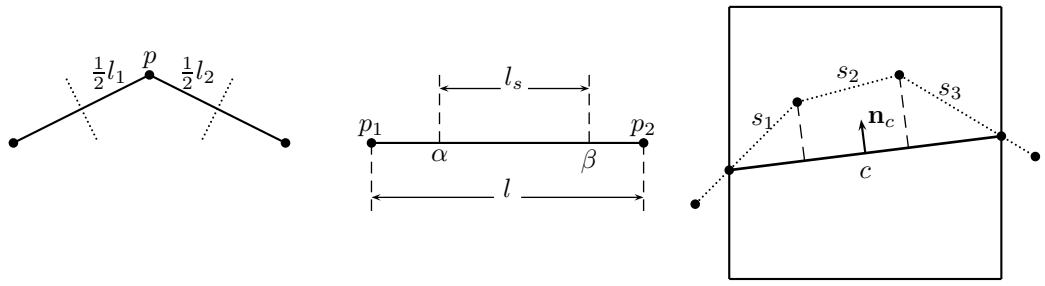
Next, a force $\hat{\mathbf{f}}_i$ is computed for the cell by integrating the surface force density over the portion of the surface in the cell i

$$\hat{\mathbf{f}}_i = \sum_{k \in S_i} \int_{\alpha_{k,i}}^{\beta_{k,i}} \hat{\mathbf{f}}_\rho \, dA. \tag{71}$$

The volume-weighted gradient of pseudo-pressure $G_{kc}p_i$ in cell i would then be

$$G_{kc}p_i = \tilde{\mathbf{n}}_i \cdot \hat{\mathbf{f}}_i. \tag{72}$$

Here, the dot product with $\tilde{\mathbf{n}}_i$ removes the tangential components of the force.



(a) Distribution (b) Integration (c) Projection and summation

Figure 3: Interpolation from Lagrangian mesh to Eulerian mesh. Force defined on particle p is first distributed onto half length of its neighboring segments (a). Then the force density will be integrated over every cut segment given relative length ratio, α and β, of its two cut locations (b). Finally, the integrated force on segments s_1, s_2 and s_3 within a cut cell will be projected onto the coupling face c given its unit normal $\mathbf{n}_c$, and get summed up (c).

For convenience, the explicit form of $(\mathbf{H}^T)_{i,k}$, the operator such that $(\mathbf{H}^T)_{i,k}\hat{\mathbf{f}}_k$ is the force at the coupling face in cell i from particle k, is provided as

$$\mathbf{H}_{k,i} = \left(\frac{\beta_{k-1,i}+\alpha_{k-1,i}}{2}\right)\frac{(\beta_{k-1,i}-\alpha_{k-1,i})\ell_k}{\ell_{k-\frac{1}{2}}}\tilde{\mathbf{n}}_i + \left(1-\frac{\beta_{k,i}+\alpha_{k,i}}{2}\right)\frac{(\beta_{k,i}-\alpha_{k,i})\ell_k}{\ell_{k+\frac{1}{2}}}\tilde{\mathbf{n}}_i, \tag{73}$$

where the convention that $\alpha_{k,i} = \beta_{k,i} = 0$ for $k \notin S_i$ has been used. The first term in Eq. (73) is the contribution from the edge between $\hat{\mathbf{x}}_{k-1}$ and $\hat{\mathbf{x}}_k$, and the second term is the contribution from the edge between $\hat{\mathbf{x}}_k$ and $\hat{\mathbf{x}}_{k+1}$. This formula can be interpreted more intuitively. The first term corresponds to a segment that touches the cell i over some fraction of its length. The vector $\tilde{\mathbf{n}}_i$ is the direction over which the force is applied, and it is determined by the orientation of the Eulerian coupling face. The average $\frac{\beta_{k-1,i}+\alpha_{k-1,i}}{2}$ is the location of the middle of this partial segment. If the portion of the segment in cell i is near the node k, then $\beta_{k-1,i}$ and $\alpha_{k-1,i}$ will be larger. If the portion in cell i is closer to the other endpoint, then the average will be smaller. This distributes more of the pressure force to the segment onto the endpoint of the segment that is closer. The factor $\beta_{k-1,i} - \alpha_{k-1,i}$ scales down the area over which the pressure is applied, and thus the magnitude of the final force, based on the fraction of the segment over which the pressure from cell i applies. The remaining term is the ratio $\frac{\ell_k}{\ell_{k-\frac{1}{2}}}$, which is effectively a correction for when the segments adjacent to particle k are not the same length.

Note that $\tilde{\mathbf{n}}_i \cdot \hat{\mathbf{f}}_i$ guarantees that only the normal component of the force from the Lagrangian mesh will be transfered back to the fluid, a property entirely consistent with a description of surface tension as a pressure jump. Substituting $\mathbf{H}^T = \mathbf{G}\tilde{\mathbf{H}}^T$ into Eq. (14) produces

$$\mathbf{u}^{n+1} = \mathbf{u}^\star - \boldsymbol{\beta}^{-1}\mathbf{G}(\tilde{\mathbf{p}} - \Delta t\tilde{\mathbf{H}}\hat{\mathbf{f}}_e - \Delta t\tilde{\mathbf{H}}\hat{\mathbf{D}}\mathbf{H}\mathbf{u}^{n+1}) - \boldsymbol{\beta}^{-1}\mathbf{G}_\mu^T\mathbf{G}_\mu\mathbf{u}^{n+1}. \tag{74}$$

Written in this way, the surface tension force takes the form of a jump in pressure near the interface. Since $\mathbf{H}^T$ does not transfer tangential forces, the interpolation operator $\mathbf{H}$ will not transfer tangential velocities. This property is also consistent with surface tension, which is sensitive to the interface location but not motion tangential to the interface.

3.6. Note on Second Order Neumann Poisson Discretization

Our discretization of gradient and mass in the one phase case is very closely related to the Neumann Poisson discretization of [35]. Let ℓ_f be the length of face f and $s_{fc} = 1$ if the velocity u_f represents flow into cell c, $s_{fc} = -1$ if the velocity u_f represents flow out of cell c, and $s_{fc} = 0$ otherwise. Let $r_f = \frac{1}{2}$ if f is a coupling face and $r_f = 1$ otherwise. With these, our gradient is $G_{fc} = \ell_f s_{fc}$, and our mass is

$\beta_{ff} = r_f \ell_f \rho_f \Delta x$. In the absence of surface tension or viscosity, the system that must be solved and the subsequent pressure application are

$$-\mathbf{G}^T \boldsymbol{\beta}^{-1} \mathbf{G} \tilde{\mathbf{p}} = -\mathbf{G}^T \mathbf{u}^\star \qquad \mathbf{u}^{n+1} = \mathbf{u}^\star - \boldsymbol{\beta}^{-1} \mathbf{G} \tilde{\mathbf{p}} \tag{75}$$

In terms of components, this is

$$-\sum_{f,c} G_{fc'} \beta_{ff}^{-1} G_{fc} \hat{p}_c = -\sum_f G_{fc'} u_f^\star \qquad u_f^{n+1} = u_f^\star - \sum_c \beta_{ff}^{-1} G_{fc} \hat{p}_c \tag{76}$$

Substituting in the discretization produces

$$-\sum_{f,c} s_{fc'} s_{fc} \frac{\ell_f}{\rho_f r_f} \frac{\hat{p}_c}{\Delta x} = -\sum_f s_{fc'} \ell_f u_f^\star \qquad u_f^{n+1} = u_f^\star - \sum_c \frac{s_{fc}}{\rho_f r_f} \frac{\hat{p}_c}{\Delta x} \tag{77}$$

Our discretization corresponds to a Dirichlet boundary condition. The boundary condition can be converted into a Neumann condition by prescribing fixed values for the coupling faces, so that those values move to the right hand side. Assume for simplicity that these velocities are zero. Since only cut and interior faces remain, $r_f = 1$, and the discretization simplifies to

$$-\sum_{f,c} s_{fc'} \frac{\ell_f}{\rho_f} \frac{\sum_c s_{fc} \hat{p}_c}{\Delta x} = -\sum_f s_{fc'} \ell_f u_f^\star \qquad u_f^{n+1} = u_f^\star - \sum_c \frac{s_{fc}}{\rho_f} \frac{\hat{p}_c}{\Delta x} \tag{78}$$

The Poisson discretization is the same as equation (2) in [35]. The pressure application equation is the same as that used by [35]. In some sense, our Dirichlet pressure discretization is compatible with the Neumann pressure discretization in [35]. This also demonstrates that the Neumann discretization can be written in the form of Eq. (75). This form is particularly useful, since it allows the discretization of [35] to be used with the FSI formulations of [37]. Given our rather simple choice of the mass at coupling faces, our discretization will only be first order accurate as mentioned in Section 3.4. We leave it for future work to try to extend our Dirichlet discretization to second order.

3.7. Note on Second Order Dirichlet Poisson Discretization

The second order Dirichlet Poisson discretization of [17] is used for the sake of comparison. As with with the proposed scheme, it is implemented within the framework of [37], which requires the discretization to factor as in Eq. (75). It also requires the application of nonzero Dirichlet boundary conditions, a task not addressed in [37]. The omission is straightforward to address by adding $-\boldsymbol{\beta}^{-1} \mathbf{G}_{bc} \tilde{\mathbf{p}}_{bc}$ to $\mathbf{u}^\star$. Here $\mathbf{G}_{bc}$ is a continuation of the gradient stencil to include pressures other than the degrees of freedom, and $\tilde{\mathbf{p}}_{bc}$ are the corresponding Dirichlet pressures. One may obtain this result by replacing the gradient and pressure with extended versions that include both degrees of freedom and ghost pressures. Then, the ghost pressure terms are moved to the right hand side, where it is found that they can be folded into $\mathbf{u}^\star$ as described. It is important to note that this correction to $\mathbf{u}^\star$ occurs on the system right hand side as well as in the pressure update equation. The next issue to address is that the pressure jump conditions are not imposed at cells but rather at the interface location. This is addressed by simply multiplying the appropriate pressure jump by each entry in $\mathbf{G}_{bc}$ rather than using the same pressure jump for all faces adjacent to the Dirichlet cell as would occur if ghost pressure jumps were computed, followed by a matrix vector multiply.

The final issue to address is the factorization of the second order Dirichlet Poisson operator. The appropriate discretizations are $G_{fc} = s_{fc} \Delta x$ and $\beta_{ff} = \ell_f \rho_f \Delta x$, where $\ell = \theta \Delta x$. Indeed, the Poisson operator $\mathbf{G}^T \boldsymbol{\beta}^{-1} \mathbf{G}$ simplifies to $\sum_f \frac{s_{fc'} s_{fc}}{\theta_f \rho_f}$, which is just equation (21) in [17], except that our discretization is volume weighted and thus scaled by Δx^2.

Implementing this discretization in the context of [37] has an interesting additional benefit. This framework allows us to couple the pressure and viscosity systems, something that could not be done by [28].

Third Order Interface								
N	$L^\infty(\epsilon_\Gamma)$	order	$L^1(\epsilon_\Gamma)$	order	$L^\infty(\epsilon_\kappa)$	order	$L^1(\epsilon_\kappa)$	order
40	$1.299 \cdot 10^{-5}$	-	$3.214 \cdot 10^{-6}$	-	$3.375 \cdot 10^{-2}$	-	$9.626 \cdot 10^{-3}$	-
80	$1.478 \cdot 10^{-6}$	3.14	$2.838 \cdot 10^{-7}$	3.50	$2.063 \cdot 10^{-2}$	0.71	$4.931 \cdot 10^{-3}$	0.97
160	$1.783 \cdot 10^{-7}$	3.05	$2.637 \cdot 10^{-8}$	3.43	$1.065 \cdot 10^{-2}$	0.95	$2.350 \cdot 10^{-3}$	1.07
320	$2.230 \cdot 10^{-8}$	3.00	$2.827 \cdot 10^{-9}$	3.22	$5.258 \cdot 10^{-3}$	1.02	$1.127 \cdot 10^{-3}$	1.06
Fourth Order Interface								
N	$L^\infty(\epsilon_\Gamma)$	order	$L^1(\epsilon_\Gamma)$	order	$L^\infty(\epsilon_\kappa)$	order	$L^1(\epsilon_\kappa)$	order
40	$1.295 \cdot 10^{-5}$	-	$3.012 \cdot 10^{-6}$	-	$2.876 \cdot 10^{-2}$	-	$8.497 \cdot 10^{-3}$	-
80	$8.227 \cdot 10^{-7}$	3.98	$1.848 \cdot 10^{-7}$	4.03	$1.425 \cdot 10^{-2}$	1.01	$4.318 \cdot 10^{-3}$	0.98
160	$5.179 \cdot 10^{-8}$	3.99	$1.166 \cdot 10^{-8}$	3.99	$7.636 \cdot 10^{-3}$	0.90	$2.043 \cdot 10^{-3}$	1.08
320	$3.431 \cdot 10^{-9}$	3.92	$7.468 \cdot 10^{-10}$	3.96	$3.788 \cdot 10^{-3}$	1.01	$9.842 \cdot 10^{-4}$	1.05

Figure 4: Interface error ϵ_Γ and curvature error ϵ_κ for third and fourth order interfaces computed from the function $y = \sin x$.

4. Examples

4.1. Interface and Curvature Accuracy

The purpose of this example is to demonstrate the accuracy of our method for computing the interface location and for computing the curvature. We use a dimensionless domain $[0, 8] \times [-2, 2]$ with resolution $2N \times N$. The fluid region is taken to be the portion of the domain such that $y \geq sin(x)$. We only consider the portion of the fluid interface that is at least $8\Delta x$ from the domain boundary to avoid boundary artifacts. The initial level set is computed computed numerically to be exactly a signed distance function. Then, the interface is computed as in Section 3.1.2. In doing so, a number of points (x, y) are for the Lagrangian surface approximation. We compute the error for each of these points as $\epsilon_\Gamma = |y - \sin x|$. Then, we compute the curvature times normal $(\kappa\hat{\mathbf{n}})_k$ from the resulting Lagrangian mesh using Eq. (55). We compute the error estimate as $\epsilon_\kappa = \left| \|(\kappa\hat{\mathbf{n}})_k\| - |\sin x|(1 + \cos^2 x)^{-\frac{3}{2}} \right|$. The L^∞ and L^1 errors for ϵ_Γ and ϵ_κ are shown in Figure 4 for both the third and fourth order interface computations.

4.2. Stationary Circle

Perhaps the most important analytic test that exists for surface tension is that of a sphere in equilibrium. This example is considered in [25, 27, 45, 28], and we use the parameters of [25]. Our domain is $[0, 1] \times [0, 1]$. Our fluid has the unitless parameters $\rho = 10^4$, $\mu = 1$, $\sigma = 1$. The circle radius is $r = 0.25$. The analytic solution is to have a constant pressure $p = \frac{\sigma}{r}$, where the outside pressure is taken to be zero. The velocity should be zero everywhere, but in practice numerical methods generate nonzero parasitic currents. Both the L^1 and L^∞ norms of the velocity error are shown. The L^1 error is defined to be $\frac{1}{N}\sum_f |u_f|$, where f runs over all faces, u_f is the velocity component stored at that face, and $N = \sum_f 1$. The L^∞ error is defined to be $\max_f |u_f|$. In each case, the simulation is run until time $t = 5$ using a time step $\Delta t = 0.2\Delta x$, and all errors are computed at the final time. The results are shown in Figure 5 along with convergence order estimates. Note that the pressure jump scheme that we have chosen for comparison is known to produce parasitic currents as much as three orders of magnitude lower than would be achieved from a delta function formulation [28].

4.3. Oscillating Circle

In this example we consider the problem of a deformed circle oscillating under surface tension. This problem has been considered elsewhere [45, 13]. The initial configuration consists of a deformed circle described in polar coordinates by $r = s(1 + \epsilon \cos m\theta)$, where $m \geq 2$ is an integer indicating the oscillation mode, s is the unperturbed radius, and ϵ is the magnitude of the initial perturbation. The approximate frequency and velocity field in the absence of viscosity (see [31]) are

$$\omega^2 = \frac{\sigma(m^2 - 1)m}{\rho s^3} \qquad \mathbf{u} = -\epsilon\omega s\left(\frac{r}{s}\right)^{m-1} \sin \omega t \begin{pmatrix} \cos(1-m)\theta \\ \sin(1-m)\theta \end{pmatrix}. \tag{79}$$

	One Phase						Two Phase			
	Pressure Jump		Level Set		Front Tracking		Level Set		Front Tracking	
Δx^{-1}	L^1	L^∞	L^1	L^∞	L^1	L^∞	L^1	L^∞	L^1	L^∞
20	$9.40\cdot10^{-5}$	$3.20\cdot10^{-4}$	$1.07\cdot10^{-4}$	$3.70\cdot10^{-4}$	$1.83\cdot10^{-7}$	$9.59\cdot10^{-7}$	$8.57\cdot10^{-5}$	$2.48\cdot10^{-4}$	$2.76\cdot10^{-7}$	$8.32\cdot10^{-7}$
40	$5.66\cdot10^{-6}$	$4.73\cdot10^{-5}$	$5.39\cdot10^{-6}$	$2.64\cdot10^{-5}$	$4.41\cdot10^{-8}$	$3.28\cdot10^{-7}$	$7.56\cdot10^{-6}$	$3.33\cdot10^{-5}$	$3.72\cdot10^{-8}$	$1.60\cdot10^{-7}$
80	$4.39\cdot10^{-7}$	$6.56\cdot10^{-6}$	$2.38\cdot10^{-7}$	$2.37\cdot10^{-6}$	$4.16\cdot10^{-9}$	$3.11\cdot10^{-8}$	$1.15\cdot10^{-7}$	$1.31\cdot10^{-6}$	$4.90\cdot10^{-9}$	$1.54\cdot10^{-8}$
160	$1.58\cdot10^{-7}$	$1.76\cdot10^{-6}$	$1.74\cdot10^{-8}$	$2.71\cdot10^{-7}$	$5.20\cdot10^{-10}$	$5.54\cdot10^{-9}$	$1.04\cdot10^{-8}$	$1.58\cdot10^{-7}$	$1.25\cdot10^{-10}$	$1.33\cdot10^{-9}$
Order	3.1	2.5	4.2	3.5	2.9	2.6	4.5	3.6	3.6	3.1

Figure 5: Parasitic current magnitudes for a stationary circle.

We note that this is only an approximate solution, and its accuracy improves as $\epsilon \to 0$. Following [13], we use a dimensionless $[0,1] \times [0,1]$ domain with dimensionless parameters $s = \frac{1}{3}$, $m = 2$, $\rho = 27$, $\sigma = \frac{2}{3}$, and $\mu = 0$. Several choices of ϵ are considered. The (approximate) analytic period with these parameters is π. The first test was run with $\epsilon = 0.05$, where the period of the first oscillation and the x component of the fluid velocity were determined at the location $(0.75, 0.5)$ at time $t = 0.5$ to establish convergence. In the second test, we consider the refinement of ϵ with the resolution fixed to demonstrate convergence to the analytic analytic solution the solution itself becomes more accurate. Here we determine the period of the first oscillation, which does not depend on ϵ, and the relative error at time $t = 0.5$. The relative error is taken to be the velocity error, computed for each velocity degree of freedom, divided by the (approximate) maximum analytic velocity magnitude $\epsilon\omega s|\sin\omega t|$. Note that the solution scales down with ϵ, but the relative velocity does not, so that a decreasing relative velocity error indicates actual convergence.

The results of both tests are shown in Figure 6. The first simulation is a pressure jump formulation of surface tension applied using a second order accurate Dirichlet boundary condition which we use for comparison purposes. The second scheme is the proposed method with a surface mesh constructed from the level set, and the third scheme is the proposed method with a front tracked surface mesh. In the refinement test, the velocities in all three cases appear to be converging first order to a value around $-2.063 \cdot 10^{-2}$. Thus, the three methods are all consistent, at least with each other. Moreover, the pressure jump scheme is consistently more accurate. This is not particularly surprising, since it uses a second order curvature and applies it with a second order boundary condition. This test does not indicate convergence to the analytic solution, as the analytic solution is only approximate. Note that the periods of all three methods are near the (approximate) analytic value of π, but these values appear to be noisy in all three schemes and no clear order of convergence can be deduced in any of the schemes. The periods produced for the pressure jump scheme are very close to the values predicted in [13], which uses essentially the same scheme.

The second test effectively refines the analytic solution to make it more accurate. This allows us to test that the schemes are converging to the correct results. The decreasing velocity errors indicate that the analytic solution and the numerical solutions are approaching each other. The velocity errors are approximately first order for the pressure jump formulation but appear to be less than first order for the proposed method, suggesting that the proposed schemes are not yet fully in the convergence regime. This fact is even more evident from the periods. The periods for the pressure jump are converging first order, but they remain noisy in the proposed schemes. We also observed noisy periods when we tested the pressure jump formulation with a first order accurate discretization of the irregular Dirichlet boundary condition.

4.4. Rising Bubble–Stability

In this example, we test the stability of our method concerning a rising bubble in a heavier fluid, as also considered in [25]. We use the same setting as in [25]. The domain sizes $[0\,m, 1\,m] \times [0\,m, 2\,m]$. The bubble is given an initial radius of $r = 0.2$ and is placed at $[0.5\,m, 0.5\,m]$. The density of heavier fluid is $\rho_{water} = 10^4\,kg\,m^{-2}$, and the density of the lighter fluid is $\rho_{air} = 10^3\,kg\,m^{-2}$. The viscosity of both fluids is $\mu = 1\,kg\,s^{-1}$. The surface tension coefficient is $\sigma = 0.5\,kg\,s^{-2}$. We use a gravity of $g = -8 \times 10^{-4}\,m\,s^{-2}$. And we perform computations on a single 80×160 mesh. As discussed in [25], the capillary time step restriction is $\Delta t = 0.056\,s$, and they achieved stable results up to $\Delta t = 2.0\,s$. We run the same test for both

Refinement Test						
	Pressure Jump		Level Set		Front Tracking	
N	T	u	T	u	T	u
50	3.188	$-2.074 \cdot 10^{-2}$	3.277	$-1.916 \cdot 10^{-2}$	3.222	$-1.945 \cdot 10^{-2}$
100	3.192	$-2.069 \cdot 10^{-2}$	3.289	$-1.992 \cdot 10^{-2}$	3.224	$-2.002 \cdot 10^{-2}$
200	3.185	$-2.066 \cdot 10^{-2}$	3.171	$-2.027 \cdot 10^{-2}$	3.198	$-2.032 \cdot 10^{-2}$
Epsilon Test						
	Pressure Jump		Level Set		Front Tracking	
ϵ	T	$L^\infty(\mathbf{u})$	T	$L^\infty(\mathbf{u})$	T	$L^\infty(\mathbf{u})$
0.05	3.185	$5.487 \cdot 10^{-3}$	3.171	$1.021 \cdot 10^{-2}$	3.198	$9.712 \cdot 10^{-3}$
0.025	3.162	$2.277 \cdot 10^{-3}$	3.166	$5.718 \cdot 10^{-3}$	3.180	$6.412 \cdot 10^{-3}$
0.0125	3.152	$8.749 \cdot 10^{-4}$	3.203	$4.697 \cdot 10^{-3}$	3.168	$5.532 \cdot 10^{-3}$

Figure 6: A deformed circle oscillates under surface tension. The maximum deformation as a fraction of the radius is ϵ. In the refinement test, the spatial and temporal resolution are increased while ϵ is held fixed to demonstrate convergence of the numerical methods. In the epsilon test, the deformation is decreased, demonstrating that as the approximate analytic solution becomes more accurate it approaches the numerical results. The analytic period T in all cases is π.

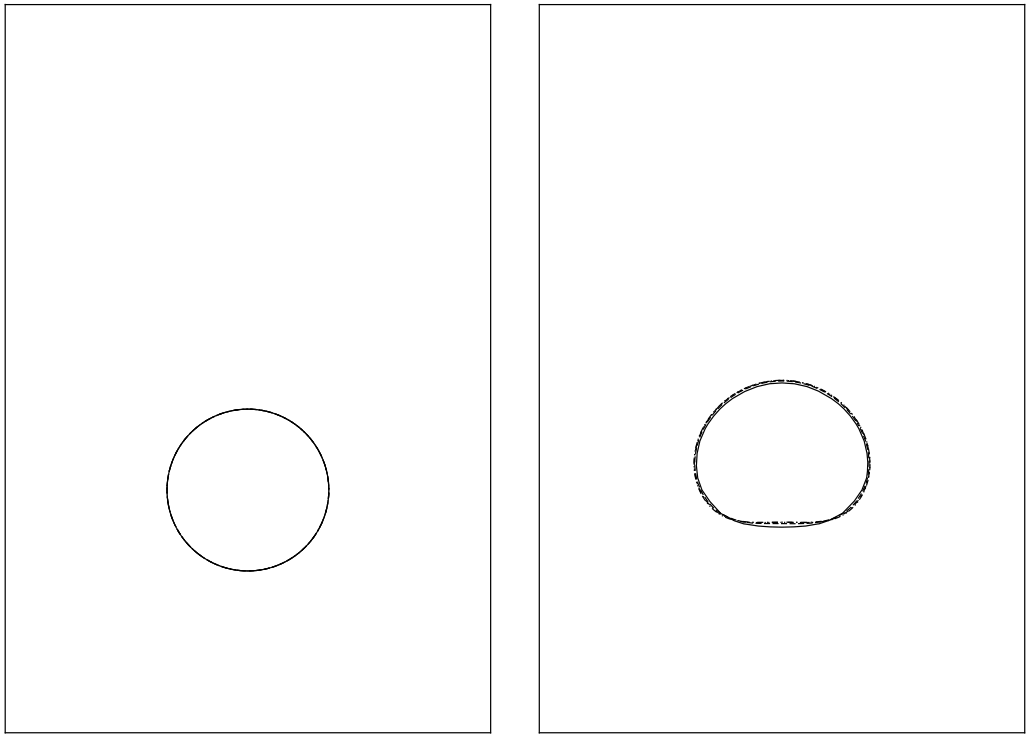
the level set version and the front tracking version of our method. The results demonstrate good stability up to $\Delta t = 3.0\,s$ for both the level set version and the front tracking version. We note that the stability we attained here is not just limited by our surface tension model but also by the CFL condition imposed by advection. To demonstrate that, we ran another set of tests using the semi-Lagrangian advection, and we attain stable results up to $\Delta t = 7.0\,s$ for both the level set version and the front tracking version.

4.5. Rising Bubble–Convergence

This example tests the convergence behavior under grid refinement concerning a rising bubble. We use a similar setting as in [28]. The $[-.01\,m, .01\,m] \times [-.02\,m, .01\,m]$ domain is initially filled with water except for a circular air bubble of radius $\frac{1}{300}\,m$ centered at the origin. The parameters are: $g = -9.8\,m\,s^{-2}$, $\sigma = .0728\,kg\,s^{-2}$, $\rho_{water} = 1000\,kg\,m^{-2}$, $\rho_{air} = 1.226\,kg\,m^{-2}$. We set the viscosity of both water and air to be $\mu = 1.78 \times 10^{-5}\,kg\,s^{-1}$, since the method presented in this paper cannot handle various viscosity or viscosity jump in an implicit manner, and also the viscosity plays a negligible role in affecting the dynamics of this example. The computation was carried out on meshes of size 40×60, 80×120, 160×240 and 320×480. Figure 7 and Figure 8 show snapshot results at different times for the level set version and the front tracking version respectively. We tested the convergence of the interface location at the top middle and bottom middle and in each case observed convergence consistent with first order.

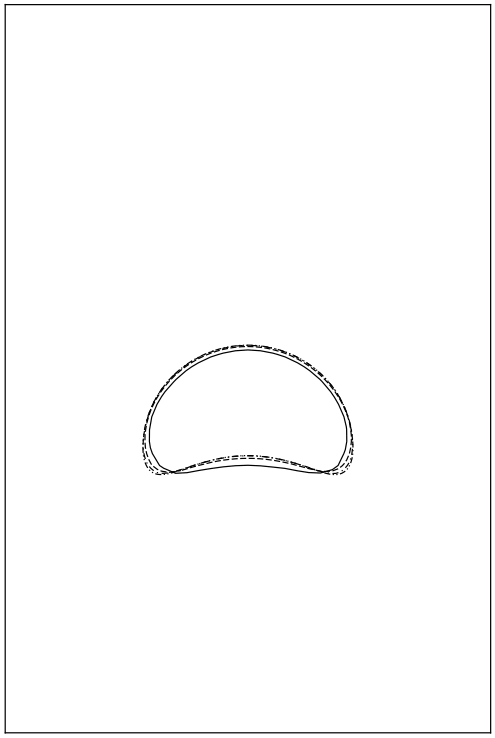
4.6. Stability

In this example, we probe the stability characteristics of different integration schemes to determine the effectiveness of our formulation for implicit surface tension at improving stability. In each case, a $[0\,m, 0.04\,m] \times [0\,m, 0.04\,m]$ domain was used. The fluid of density is $1000\,kg\,m^{-2}$, and the surface tension coefficient is $.0728\,kg\,ms^{-2}$. Since the purpose of this test is to stress the stability of each scheme, we do not use viscosity. The fluid starts in a circle of radius $0.01\,m$ centered within the domain. We test five schemes. The first scheme is a pressure jump scheme based on [28] which we use for comparison purposes. The other four are variants on the schemes proposed here. Two have a front tracked interface, and the other two have a level set interface. Each interface type is run explicitly (without force derivatives) and implicitly as discussed in Section 3.2. Each scheme is run with a wide range of Δx and Δt choices, and in each case it was noted whether the resulting simulation was stable or unstable when run for 500 time steps. The results are shown in Figure 9. The pressure jump scheme and the explicit versions of the proposed scheme have similar stability characteristics. The implicit level set version is noticeably more stable than the explicit schemes allowing a Δt approximately 2-3 times larger. The implicit front tracked version is far more stable, allowing a Δt approximately an order of magnitude larger. Unlike the implicit level set scheme, the position

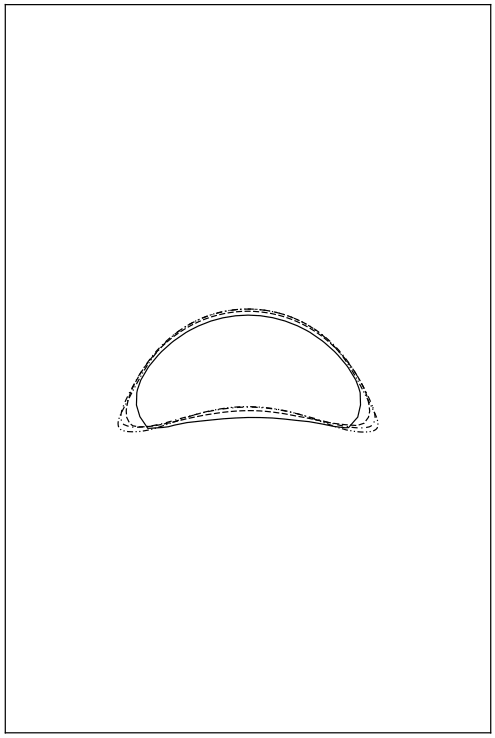


(a) t=0

(b) t=.02

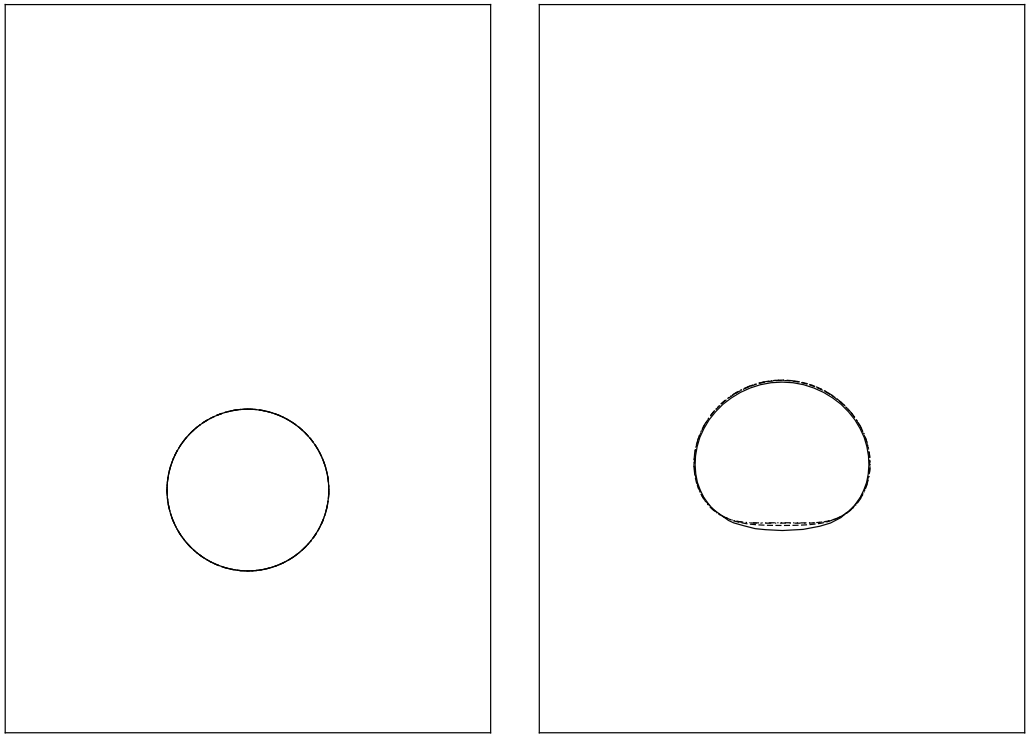


(c) t=.035



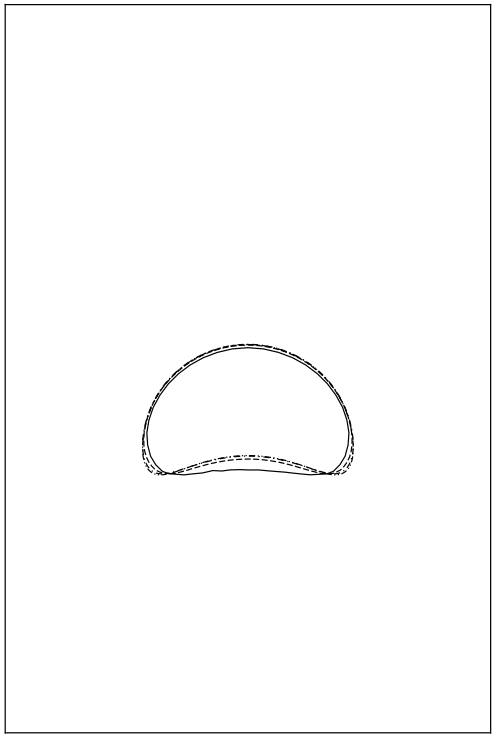
(d) t=.05

Figure 7: Rising bubble tests for convergence under grid refinement using the level set method. Grid resolutions: solid line: 40×60; dash line: 80×120; dash-dot line: 160×240; dash-dot-dot line: 320×480.

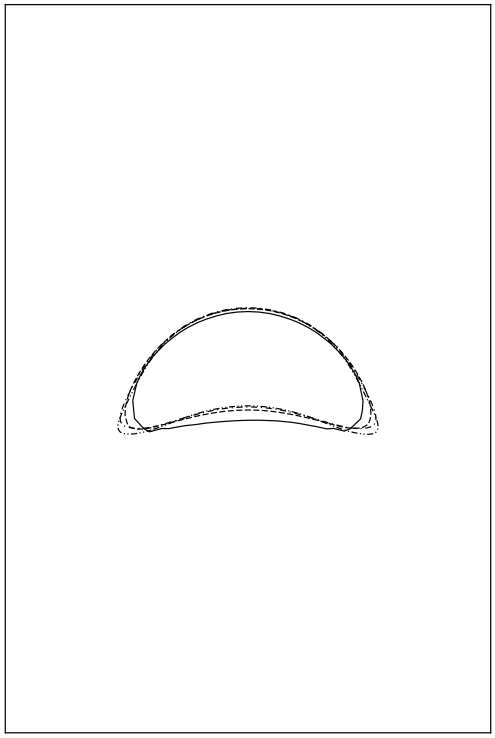


(a) t=0

(b) t=.02

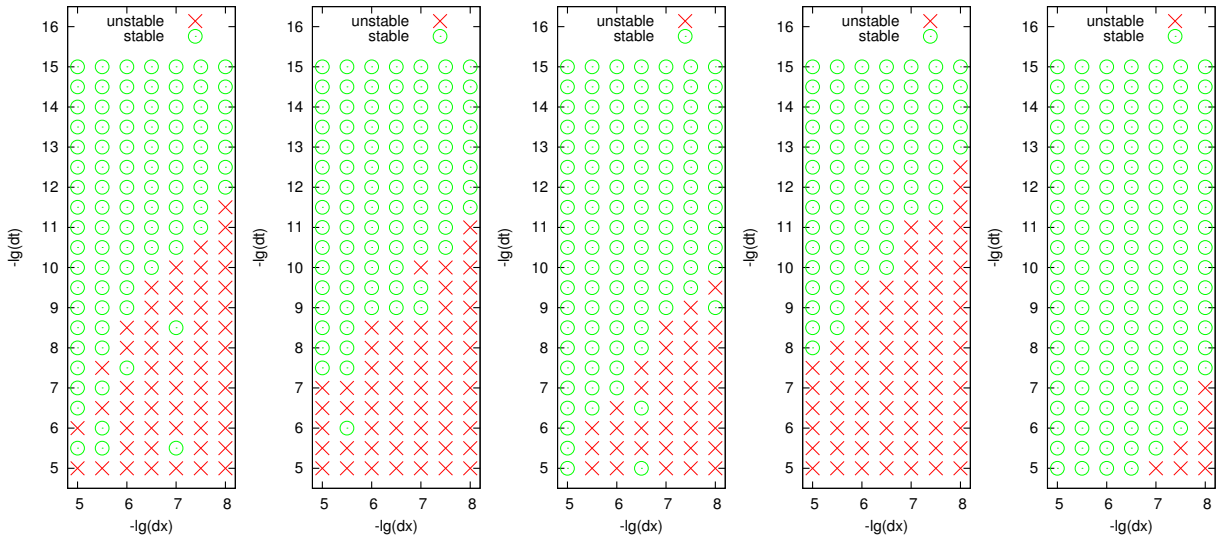


(c) t=.035



(d) t=.05

Figure 8: Rising bubble tests for convergence under grid refinement using the front tracking method. Grid resolutions: solid line: 40×60; dash line: 80×120; dash-dot line: 160×240; dash-dot-dot line: 320×480.



(a) Pressure jump (b) Level set, explicit (c) Level set, implicit (d) Front track, explicit (e) Front track, implicit

Figure 9: Stability tests for five schemes on stationary circle. The x axis indicates the Δx used, with more refined simulations on the right. The y axis indicates the Δt used, with smaller time steps at the top. A circle indicates a stable simulation, and an X indicates an unstable simulation.

update used on the front tracked interface corresponds to a backward Euler update, whereas the level set version has a level set update that differs significantly from the update suggested by the force derivatives. Thus, it is not surprising to see better stability from the front tracked scheme. This opens up an avenue of research to produce an implicit level set scheme, which may be used to improve the stability of the proposed method when the interface is tracked with a level set.

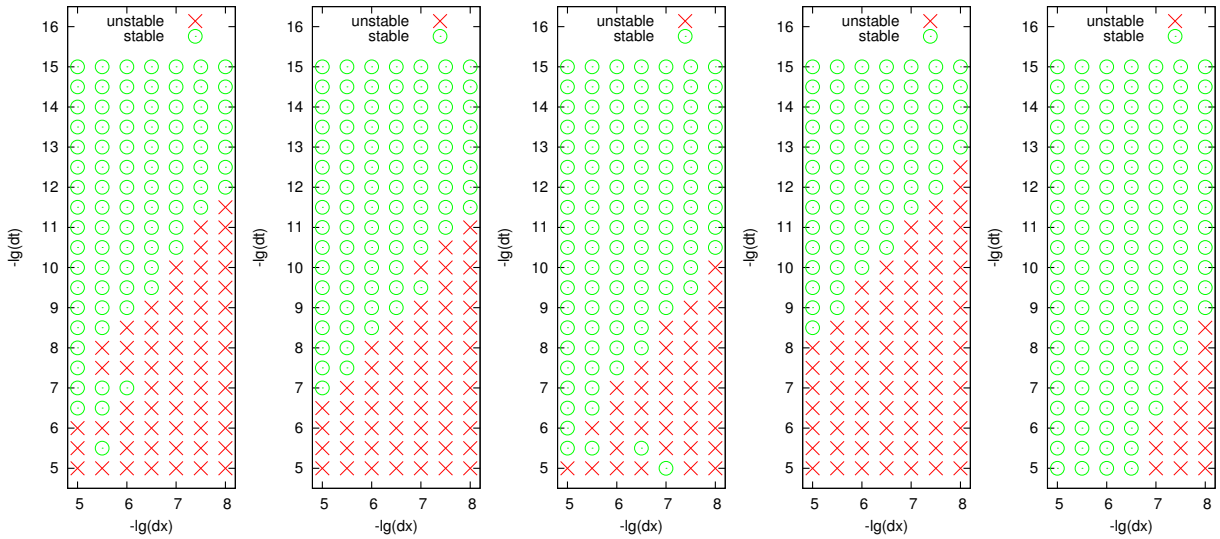
We repeated this on a dynamic configuration. This setup is identical to the first, except that the initial fluid configuration is an ellipse with major axis $0.011\,m$ and minor axis $0.009\,m$. The results are shown in Figure 10. The results are quite similar to the stationary circle case, though the dynamic nature of the simulation and the special circle configuration do appear to make a noticeable difference for the implicit front tracking scheme. Since this example has a nonzero velocity as part of its correct behavior, there will be a CFL restriction for advection, but all of the methods became unstable before this CFL restriction was reached.

5. Conclusions and Future Work

We have presented a method for applying implicit forces on a Lagrangian mesh to an Eulerian discretization of the Navier Stokes equations. The method leverages the SPD FSI framework of [37] to produce a sparse symmetric positive definite system. The resulting method has implicit and fully coupled viscosity, pressure, and Lagrangian forces.

We apply our new framework for forces on a Lagrangian mesh to the case of surface tension force. We describe two approaches for constructing and maintaining a suitable Lagrangian surface mesh with sufficient accuracy for accurate second derivatives at the surface. We construct a discretization that is able to reduce the magnitude of surface tension parasitic currents down to levels comparable to other state of the art schemes. Finally, we demonstrate the accuracy and enhanced stability of our new method.

Our surface tension discretization is only first order. Nevertheless, many parts are already at second order, and other parts share many attributes with second order methods, including the use of subcell information. We leave it for future work to construct a fully second order accurate discretization.



(a) Pressure jump (b) Level set, explicit (c) Level set, implicit (d) Front track, explicit (e) Front track, implicit

Figure 10: Stability tests for five schemes on an ellipse. The x axis indicates the Δx used, with more refined simulations on the right. The y axis indicates the Δt used, with smaller time steps at the top. A circle indicates a stable simulation, and an X indicates an unstable simulation.

The stability results in Section 4.6 demonstrate drastic improvement in stability for the front tracking interface but only modest improvement for the level set based interface. This is likely because we update the front tracked interface in a way that is implicit and similar to backward Euler, whereas the level set version is updated instead using an explicit particle level set evolution. This suggests that significant stability improvements could be made by making the level set update implicit in a way compatible with the Lagrangian surface velocities obtained from the coupled system, and we leave this for future work.

There are many interesting avenues for extending the proposed method, which we leave for future work. One may extend the method to treat surfactants [48, 30, 26] and foam [29]. Another potential extension is to problems where a jump in pressure is caused by phenomena other than surface tension, such as an electric potential [54]. Another potential application is to solving partial differential equations on a surfaces as in [2].

Acknowledgments

Research supported in part by ONR N00014-09-1-0101, ONR N00014-11-1-0027, ONR N00014-06-1-0505, ONR N00014-05-1-0479, for a computing cluster, NSF IIS-1048573, and King Abdullah University of Science and Technology (KAUST) 42959. C.S. was supported in part by a Stanford Graduate Fellowship.

References

[1] C. Batty, F. Bertails, and R. Bridson. A fast variational framework for accurate solid-fluid coupling. ACM Trans. Graph. (SIGGRAPH Proc.), 26(3):100, 2007.

[2] M. Bertalmio, L.T. Cheng, S. Osher, and G. Sapiro. Variational problems and partial differential equations on implicit surfaces: the framework and examples in image processing and pattern formation, 2000.

[3] J. Bonet and R. Wood. Nonlinear continuum mechanics for finite element analysis. Cambridge University Press, Cambridge, 1997.

[4] J. U. Brackbill, D. B. Kothe, and C. Zemach. A continuum method for modelling surface tension. J. Comput. Phys., 100:335–353, 1992.

[5] Y.C. Chang, T.Y. Hou, B. Merriman, and S. Osher. A level set formulation of eulerian interface capturing methods for incompressible fluid flows. J. Comput. Phys., 124(2):449–464, 1996.

[6] J. Chessa and T. Belytschko. An enriched finite element method and level sets for axisymmetric two-phase flow with surface tension. Int. J. Num. Meth. Eng., 58:2041–2064, 2003.

[7] J. Chessa and T. Belytschko. An extended finite element method for two-phase fluids. ASME J. Appl. Mech., 70:10–17, 2003.

[8] D. Chopp. Some improvements of the fast marching method. SIAM J. Sci. Comput., 223:230–244, 2001.

[9] A. Chorin. A numerical method for solving incompressible viscous flow problems. J. Comput. Phys., 2:12–26, 1967.

[10] J. Douglas Jr and T. Dupont. Alternating-direction Galerkin methods on rectangles. Numer. Sol. of Part. Diff. Eqns., 2:133–214.

[11] B. Engquist, A.-K. Tornberg, and R. Tsai. Discretization of Dirac delta functions in level set methods. J. Comput. Phys., 207(1):28–51, 2005.

[12] D. Enright, R. Fedkiw, J. Ferziger, and I. Mitchell. A hybrid particle level set method for improved interface capturing. J. Comput. Phys., 183:83–116, 2002.

[13] D. Enright, D. Nguyen, F. Gibou, and R. Fedkiw. Using the particle level set method and a second order accurate pressure boundary condition for free surface flows. In Proc. 4th ASME-JSME Joint Fluids Eng. Conf., number FEDSM2003–45144. ASME, 2003.

[14] P. Esser, J. Grande, and A. Reusken. An extended finite element method applied to levitated droplet problems. Intl. J. for Num. Meth. in Engng., 2010.

[15] M.M. Francois, S.J. Cummins, E.D. Dendy, D.B. Kothe, J.M. Sicilian, and M.W. Williams. A balanced-force algorithm for continuous and sharp interfacial surface tension models within a volume tracking framework. J. of Comput. Phys., 213(1):141–173, 2006.

[16] M.M. Francois, D.B. Kothe, and S.J. Cummins. Modelling Surface Tension Using a Ghost Fluid Technique within a Volume of Fluid Formulation. Technical report, Los Alamos National Laboratory, 2004.

[17] F. Gibou, R. Fedkiw, L.-T. Cheng, and M. Kang. A second-order-accurate symmetric discretization of the Poisson equation on irregular domains. J. Comput. Phys., 176:205–227, 2002.

[18] S. Gross and A. Reusken. An extended pressure finite element space for two-phase incompressible flows with surface tension. J. of Comput. Phys., 224(1):40–58, 2007.

[19] S. Gross and A. Reusken. Finite Element Discretization Error Analysis of a Surface Tension Force in Two-Phase Incompressible Flows. SIAM Journal on Numerical Analysis, 45(4):1679–1700, 2007.

[20] F. Harlow and J. Welch. Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluid with Free Surface. Phys. Fluids, 8:2182–2189, 1965.

[21] A. Harten, B. Enquist, S. Osher, and S. Chakravarthy. Uniformly high-order accurate essentially non-oscillatory schemes III. J. Comput. Phys., 71:231–303, 1987.

[22] J.J. Helmsen, E.G. Puckett, P. Colella, and M. Dorr. Two new methods for simulating photolithography development in 3D. In Proceedings of SPIE, volume 2726, page 253, 1996.

[23] J. Hochstein and T. Williams. An implicit surface tension model. AIAA Meeting Papers., 599, 1996.

[24] T.Y. Hou, J.S. Lowengrub, and M.J. Shelley. Removing the stiffness from interfacial flows with surface tension. J. of Comput. Phys., 114(2):312–338, 1994.

[25] S. Hysing. A new implicit surface tension implementation for interfacial flows. International Journal for Numerical Methods in Fluids, 51(6):659–672, 2006.

[26] A.J. James and J. Lowengrub. A surfactant-conserving volume-of-fluid method for interfacial flows with insoluble surfactant. J. of Comput. Phys., 201(2):685–722, 2004.

[27] D. Jamet, D. Torres, and JU Brackbill. On the theory and computation of surface tension: the elimination of parasitic currents through energy conservation in the second-gradient method. J. of Comput. Phys., 182(1):262–276, 2002.

[28] M. Kang, R. Fedkiw, and X.-D. Liu. A boundary condition capturing method for multiphase incompressible flow. J. Sci. Comput., 15:323–360, 2000.

[29] Y. Kim, M.C. Lai, and C.S. Peskin. Numerical simulations of two-dimensional foam by the immersed boundary method. J. of Comput. Phys., 229(13):5194–5207, 2010.

[30] M.C. Lai, Y.H. Tseng, and H. Huang. An immersed boundary method for interfacial flows with insoluble surfactant. J. of Comput. Phys., 227(15):7279–7293, 2008.

[31] H. Lamb. Hydrodynamics. Cambridge Univ Pr, 1997.

[32] D.V. Le, J. White, J. Peraire, K.M. Lim, and B.C. Khoo. An implicit immersed boundary method for three-dimensional fluid-membrane interactions. J. of Comput. Phys., 228(22):8427–8445, 2009.

[33] P. Liovic, M. Francois, M. Rudman, and R. Manasseh. Efficient simulation of surface tension-dominated flows through enhanced interface geometry interrogation. J. of Comput. Phys., 2010.

[34] W. Lorensen and H. Cline. Marching cubes: A high-resolution 3D surface construction algorithm. Comput. Graph. (SIGGRAPH Proc.), 21:163–169, 1987.

[35] Y-T. Ng, C. Min, and F. Gibou. An efficient fluid–solid coupling algorithm for single–phase flows. J. of Comp. Phys., 228:8807–8829, 2009.

[36] S. Osher and C.-W. Shu. High order essentially non-oscillatory schemes for Hamilton-Jacobi equations. SIAM J. Num. Anal., 28:902–921, 1991.

[37] A. Robinson-Mosher, C. Schroeder, and R. Fedkiw. A symmetric positive definite formulation for monolithic fluid structure interaction. J. Comput. Phys., 230:1547–66, 2011.

[38] J. Sethian. A fast marching level set method for monotonically advancing fronts. Proc. Natl. Acad. Sci., 93:1591–1595, 1996.

[39] J. Sethian. Fast marching methods. SIAM Review, 41:199–235, 1999.

[40] C.-W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock capturing schemes. J. Comput. Phys., 77:439–471, 1988.

[41] C.-W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock capturing schemes II (two). J. Comput. Phys., 83:32–78, 1989.

[42] E. Sifakis, T. Shinar, G. Irving, and R. Fedkiw. Hybrid simulation of deformable solids. In Proc. of ACM SIGGRAPH/Eurographics Symp. on Comput. Anim., pages 81–90, 2007.

[43] P. Smereka. Semi-implicit level set methods for curvature and surface diffusion motion. J. Sci. Comput., 19(1):439–456, 2003.

[44] P. Smereka. The numerical approximation of a delta function with application to level set methods. J. Comput. Phys., 211:77–90, 2006.

[45] M. Sussman and M. Ohta. A stable and efficient method for treating surface tension in incompressible two-phase flow. J. Sci. Comput, 31(4):2447–2471, 2009.

[46] M. Sussman, P. Smereka, and S. Osher. A level set approach for computing solutions to incompressible two-phase flow. J. Comput. Phys., 114:146–159, 1994.

[47] Z. Tan, D. Wang, and Y. Wang. A Jacobian-free-based IIM for incompressible flows involving moving interfaces with Dirichlet boundary conditions. Intl. J. for Num. Meth. in Engng., 83:508–536, 2010.

[48] K.E. Teigen, P. Song, J. Lowengrub, and A. Voigt. A diffuse-interface method for two-phase flows with soluble surfactants. J. of Comput. Phys., 230:375–393, 2010.

[49] A.-K. Tornberg and B. Engquist. Numerical approximations of singular source terms in differential equations. J. Comput. Phys., 200(2):462–488, 2004.

[50] G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, S. Nas, and Y. J. Jan. A front-tracking method for the computations of multiphase flow. J. Comput. Phys., 169:708–759, 2001.

[51] J. Tsitsiklis. Efficient algorithms for globally optimal trajectories. In Proc. 33rd Conf. on Decision and Control, pages 1368–1373, 1994.

[52] J. Tsitsiklis. Efficient algorithms for globally optimal trajectories. IEEE Trans. on Automatic Control, 40:1528–1538, 1995.

[53] S. O. Unverdi and G. Tryggvason. A front-tracking method for viscous, incompressible, multifluid flows. J. Comput. Phys., 100:25–37, 1992.

[54] B.P. Van Poppel, O. Desjardins, and J.W. Daily. A ghost fluid, level set methodology for simulating multiphase electrohydrodynamic flows with application to liquid fuel injection. J. of Comput. Phys., 229:7977–7996, 2010.

[55] J.J. Xu and H.K. Zhao. An Eulerian formulation for solving partial differential equations along a moving interface. J. of Sci. Comput., 19(1):573–594, 2003.

[56] W. Zheng, J.-H. Yong, and J.-C. Paul. Simulation of bubbles. In SCA '06: Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation, pages 325–333, 2006.