ILASS Americas 21st Annual Conference on Liquid Atomization and Spray Systems, Orlando, FL, May 2008

A Hybrid Lagrangian-Eulerian Method (hLE) for Interface Tracking

Sourabh V. Apte*
School of Mechanical, Industrial, and Manufacturing Engineering
Oregon State University
204 Rogers Hall, Corvallis, OR 97331

Abstract

A hybrid Lagrangian-Eulerian (hLE) scheme, combining a particle-based, mesh-free technique with a finite-volume flow solver, is developed for direct simulations of two-phase flows. The approach uses marker points around the interface and advects the signed distance to the interface in a Lagrangian frame. The kernel–based derivative calculations typical of particle methods are used to extract the interface normal and curvature from unordered marker points. Connectivity between the marker points is not necessary. The fluid flow equations are solved on a background, fixed mesh using a co-located grid finite volume solver together with balanced force algorithm (Francois *et al.* JCP, 2006, Herrmann JCP, 2007) for surface tension force. The numerical scheme is first validated for standard test cases: (i) parasitic currents in a stationary spherical drop, (ii) small amplitude damped surface waves, (iii) capillary waves on droplet surface, (iv) Rayleigh-Taylor instability, and (v) gravity-driven bubble/droplet in a stationary fluid. Extension of the approach to three-dimensions is conceptually straight forward, however, poses challenges for parallel implementation. A domain-decomposition based on balancing the number of grid points per processor gives rise to load–imbalance due to uneven distribution of the marker points and advanced domain partitioning methods are needed for improved efficiency of the approach.

*Corresponding Author: sva@engr.orst.edu

Introduction

Numerical methods to accurately track/capture the interface between two fluids have been an area of research for decades. Tryggvason *et al.* [1] provide a detailed review on various methods used for direct simulation of multiphase flows. Broadly, these schemes can be classified into two categories: (a) front tracking and (b) front capturing methods.

Front tracking methods are Lagrangian in nature [2, 3, 4, 5], and the interface is tracked by a set of connected [1] or unconnected [6] marker points on the interface and the Navier-Stokes equations are solved on a fixed grid in an Eulerian frame. Another class of Lagrangian methods include *mesh-free* algorithms such as moving particle-methods [7], vortex-in cell methods [8, 9], and smoothed-particle hydrodynamics [10], where the interface is represented by Lagrangian points (LPs) and the flow-field is also evaluated on these points. Pure Lagrangian methods are promising as they avoid enormous memory requirements for a three-dimensional mesh. These methods automatically provide adaptive resolution in the high-curvature region [8] and have been applied successfully to many two-phase flow problems [11, 12, 13]. However, they exhibit other difficulties such as high cost of finding nearest neighbors in the zone of influence of a Lagrangian point, true enforcement of continuity (or incompressibility) conditions, and problems associated with accurate one-sided interpolations near boundaries [8].

In capturing methods the interface is not explicitly tracked, but captured using a characteristic function, which evolves using the advection equation. Representative capturing methods are: volume tracking [14, 15], level set [16, 17, 18] and phase field models. Both approaches (the VoF and level set) are straightforward to implement, however, level-set approach does not preserve volume of the fluids on either side of the interface. The VOF formulation on the other hand, conserves the fluid volume but lacks in the sharpness of the interface. Several improvements to these methods involving combination of the two [19], adaptive mesh-refinement [20, 21], particle-level sets [22], refined level-set grid scheme [23, 24, 25] have been proposed for improved accuracy.

In the present work, some of the limitations of the above schemes are addressed by *combining the two broad approaches* mentioned above. The basic idea is to merge the locally 'adaptive' mesh-free particle-based methods with the relative 'ease' of Eulerian finite-volume formulation in order to inherit the advantages offered by individual approaches. The interface between two fluids is represented and tracked using Lagrangian points or fictitious particles [13]. Unlike particle level set method [22] or the semi-Lagrangian methods [26], in the present approach the interface is represented by Lagrangian points (LPs) (or particles[1]) that are advanced in a Lagrangian frame. The motion of the interface is determined by a velocity field (interpolated to the particle locations) obtained by solving the Navier-Stokes equations on a *fixed* background mesh in an Eulerian frame. The interface location, once determined, identifies the region of the mesh to apply *jump-conditions* in fluid properties. In this sense, it is in the realm of Arbitrary Lagrangian-Eulerian (ALE) [27] schemes, wherein the computational grid deforms to conform to the shape of the dispersed phase. The potential advantage of the present hybrid method is that the background mesh could be of any kind: *structured, body-fitted, or arbitrary shaped unstructured* (hex, pyramids, tetrahedrons, prisms) and may be *stationary or changing in time (adaptive refinement)*. Here, we use a co-located grid, incompressible flow solver based on the energy conserving finite-volume algorithm developed by Mahesh *et al.* [28, 29].

The Lagrangian points (LPs) in our interface calculations, are particles distributed in a narrow band around the interface [30]. These LPs are initially uniformly spaced and carry information such as the signed distance to the interface (SDF) along the characteristic paths. Variations in flow velocities leads to an irregular distribution of the initially uniform LPs. Regularization of the particles are performed by mapping the particles on a uniformly spaced lattice [13]. Values for particle properties at new LP locations are obtained through kernel mollification as done in Smoothed Particle Hydrodynamics [10] and remeshed-SPH [12]. The novelty in our approach is that this mesh-free interface representation is integrated with a finite-volume solver where the governing equations for flow evolution are solved. The Lagrangian points provide sub-grid resolution and in this respect the method is similar to the Refined Level Set Grid (RLSG) approach [24, 25, 31]. However, here the LPs move in space with the flow velocity and different discretizations are necessary and we use high-order schemes based on mollification kernels.

The paper is arranged as follows. The governing equations and mathematical formulation is described followed by description of the numerical scheme. The numerical approach is then applied to standard test cases to evaluate the accuracy of the

[1] In this paper, the term 'particles' means Lagrangian points (LPs) that are used to represent the interface.

scheme compared to other approaches. Finally, some preliminary results on rising bubble in a quiescent fluid are presented.

Mathematical Formulation and Governing Equations

Consider two immiscible fluids labeled as '1' and '2' forming an interface. The regions occupied by the two fluids can be represented by an indicator function Ψ which can be zero or unity representing one of the fluids. In a front tracking scheme [1], connected marker points are used to represent the interface surface. The marker points are embedded in a background computational mesh. Knowing the location of the marker points, a smoothly varying indicator function (or color function) is constructed on the computational mesh (Ψ_{cv}, where cv stands for the grid control volume). Torres and Brackbill [6] developed a point-set method where the connectivity between the marker points was not necessary. The present work is motivated by the point-set method. Instead of using marker points on the interface only, a uniformly distributed set of points (termed here as Lagrangian points, LP) are placed in a small band surrounding the interface. These marker points are assigned a signed distance function (SDF, Φ) to the interface, and thus represent the interface implicitly.

The main advantage of this approach is that the color function can be easily constructed from the marker points (as discussed in the following sections) and does not require a solution of the Laplace equation ($\bigtriangledown^2\Psi = 0$) as in the point-set method [6]. However, as the marker points are moved by the underlying flow, the flow strain can cluster the LPs in some region and spread them apart in other regions. Such non-uniform distribution may affect the interface representation (and interface properties such as surface normals, curvature etc.) and the LPs are periodically re-arranged to a uniform distribution through a systematic *remeshing* or *reconfiguration* procedure [13] typically used in remeshed Smooth Particle Hydrodynamics [8]. The formulation thus represents a combination of marker points, level set methods, and smooth particle hydrodynamics and is termed as hybrid Lagrangian Eulerian (hLE) approach. Note that this approach is different from the particle-level set method [22] where the level set function was advanced in an Eulerian frame and later corrected by subgrid particles. In the present work, the advection of the interface is performed solely in the Lagrangian frame using the motion of the LPs.

Hybrid Lagrangian-Eulerian (hLE) Scheme

Following Hieber & Koumoutsakos [13], the interface between two fluids is represented using uniformly spaced Lagrangian points (LPs) or fictitious particles in a narrow band around the interface. Each LP is associated with position $\mathbf{x}_p$, velocity $\mathbf{u}_p$, volume $\mathcal{V}_p$ and a scalar function Φ_p which represents the signed distance to the interface. The average spacing (h between the uniformly spaced LPs is related to the volume $\mathcal{V}_p$. In this work, we use cubic elements ($h = \mathcal{V}_p^{1/3}$). As the LPs move, they carry the SDF value along the characteristic paths and implicitly represent the motion of the interface. The evolution of the interface is calculated by solving level set equations in the Lagrangian form:

$$\frac{D\Phi_p}{Dt} = 0;\ \frac{D\mathcal{V}_p}{Dt} = \langle \nabla \cdot \mathbf{u} \rangle_p \mathcal{V}_p;\ \frac{D\mathbf{x}_p}{Dt} = \mathbf{u}_p, \qquad (1)$$

where p denotes the Lagrangian point or particle. For incompressible fluids, the velocity field is divergence free and theoretically, the change in volume of the LPs ($D\mathcal{V}_p/Dt$) is zero.

As is done in Smoothed Particle Hydrodynamics (SPH) and mesh-free methods, smoothed approximation of the level set function and its derivatives can be obtained by using a mollification operator with LPs as quadrature points. The localized mollification kernel ξ_ϵ generates a smooth continuous approximation of Φ around the particle at location $\mathbf{x}_p$ using SDF of other particles at locations $\mathbf{x}_q$:

$$\Phi_q = \sum_{p=1}^{N} \mathcal{V}_p \Phi_p \xi_\epsilon(\mathbf{x}_q - \mathbf{x}_p) \qquad (2)$$

where $\Phi_q = \Phi(\mathbf{x}_q)$ and $\sum_{p=1}^{N} \xi_\epsilon \mathcal{V}_p = 1$. Different mollification kernels have been proposed such as quartic spline and M_n splines [10]. In this paper, we use the quartic spline function given as:

$$\xi_\epsilon(\mathbf{x}) = \begin{cases} \frac{s^4}{4} - \frac{5s^2}{8} + \frac{115}{192}, & 0 \le s < \frac{1}{2} \\ -\frac{s^4}{6} + \frac{5s^3}{6} - \frac{5s^2}{4} + \frac{5s}{24} + \frac{55}{96} & \frac{1}{2} \le s < \frac{3}{2} \\ \frac{(2.5-s)^4}{24}, & \frac{3}{2} \le s < \frac{5}{2} \\ 0 & s \ge \frac{5}{2} \end{cases}$$

where $s = |\mathbf{x}|/\epsilon$. Here, ϵ is the radius of influence around the particle (or LP) and depends on the spacing between the LPs and the width of the mollification kernel. For all calculations in this work ϵ is set equal to the uniform spacing between the LPs. The surface normal and curvature calculations require derivatives of the scalar function Φ on the particles. These are approximated in a conservative

form by using derivatives of the mollification kernel [32]:

$$\langle \nabla \Phi \rangle_q = \sum_p \mathcal{V}_p \left(\Phi_p - \Phi_q \right) \nabla \xi_\epsilon (\mathrm{x}_q - \mathrm{x}_p), \qquad (3)$$

$$\langle \nabla^2 \Phi \rangle_q = \sum_p \mathcal{V}_p \left(\Phi_p - \Phi_q \right) \nabla^2 \xi_\epsilon (\mathrm{x}_q - \mathrm{x}_p). \qquad (4)$$

In the above equations, kernel (ξ_ϵ) and its first and second derivatives should be properly normalized such that corresponding non-zero moment conditions are satisfied (details are given in [32]).

Once the location of the LPs and the associated Φ_p values are obtained, a color function $\Psi(\mathbf{x})$ can be constructed. Following the definition of color function, finding Ψ on the LPs is straightforward: $\Psi = 0$ when $\Phi \geq 0$ and $\Psi = 1$ for $\Phi < 0$. Then the color function field can also be obtained on the background computational mesh by interpolating Ψ from the LPs. In order to obtain a smooth function, the M_4' kernel interpolation is used:

$$M_4' = \begin{cases} 1 - 5/2s^2 + 3/2s^3 & 0 \leq s < 1 \\ 1/2(1-s)(2-s)^2 & 1 \leq s < 2 \\ 0 & 2 \leq s \end{cases} \qquad (5)$$

where $s = x$ in one-dimension. The higher-dimensional interpolations are obtained by taking tensorial products of their one-dimensional counterparts.

Once the color function Ψ is obtained at a control volume (cv), the density and viscosity are given as:

$$\rho_{cv} = \rho_1 + (\rho_2 - \rho_1)\Psi_{cv} \qquad (6)$$

$$\mu_{cv} = \mu_1 + (\mu_2 - \mu_1)\Psi_{cv} \qquad (7)$$

Then the flow field is computed on a background mesh (which could be structured or unstructured) by solving the Navier-Stokes equations for the two-fluid system:

$$\nabla \cdot \mathbf{u} = 0 \qquad (8)$$

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho} \nabla p + \frac{1}{\rho} \nabla \cdot (\mu (\nabla \mathbf{u} + \nabla^T \mathbf{u})) + \mathbf{g} + \frac{1}{\rho} \mathbf{F}_\sigma \qquad (9)$$

where $\mathbf{u}$ is velocity vector of fluid, p is pressure, ρ and μ are fluid density and viscosity (uniform inside each fluid), $\mathbf{g}$ body force, and $\mathbf{F}_\sigma$ is the surface tension force which is non-zero only at the interface location ($\Phi = 0$). Following Brackbill *et al.* [33], the surface tension force is modeled as a continuum surface force (CSF):

$$\mathbf{F}_\sigma^{CSF} = \sigma \kappa \hat{\mathbf{n}} \delta(\Phi) \qquad (10)$$

where σ is the surface tension coefficient (assumed constant in the present work), κ is the curvature, $\hat{\mathbf{n}}$ the interface normal, and $\delta(\Phi)$ a dirac-delta function. A common issue with numerical simulations involving surface tension force, is the development of *spurious currents* (unphysical velocity field) [1, 20] due to inaccuracies in the discrete approximations to the surface-tension forces (equation 9). In order to obtain a consistent coupling of the surface tension force with the pressure gradient forces in a finite-volume approach, Francois *et al.* [34] indicated that the surface tension force must be evaluated at the faces of the control volumes as:

$$\mathbf{F}_{\sigma,f}^{CSF} = \sigma \kappa_f \left(\nabla \Psi \right)_f \qquad (11)$$

where the subscript f stands for the face of the control volume. The surface tension force at the cv-centers can be obtained through reconstruction from the faces of each cv.

To compute the surface tension force, accurate estimation of the curvature of the interface is necessary. Any errors in the curvature calculation and surface tension force representation give rise to non-physical velocity fields that could be detrimental to the accuracy of the numerical solution [1, 6]. Herrmann [25, 35] developed a procedure to compute the curvature accurately in the level-set framework. Here we follow a similar procedure for curvature evaluations:

- The curvature at a point on the interface is given as:

$$\kappa = \nabla \cdot \frac{\nabla \Phi}{|\nabla \Phi|}; \qquad \mathbf{n} = \frac{\nabla \Phi}{|\nabla \Phi|}. \qquad (12)$$

  First the curvature and normal are evaluated at the LPs close to the interface ($|\Phi| \leq 2\Delta_{LP}$, where Δ_{LP} is the spacing between the LPs. The gradients in the curvature and surface normal computations are evaluated using using equations 4.

- For each of these LPs (with $|\Phi| \leq 2\Delta_{LP}$) , a point on the interface is obtained by projecting normals onto the interface [25]:

$$\mathbf{x}_{interface} = \mathbf{x}_{LP} - \frac{\Phi}{|\nabla \Phi|} \hat{\mathbf{n}} \qquad (13)$$

- Curvature on the interface point $\mathbf{x}_{interface}$ is evaluated by using curvature values on LPs in its neighborhood through M_4'-kernel based interpolation (equation 5).

- Once curvatures on all interface points are evaluated, these values are *assigned* to the corresponding LPs from which these interface points were obtained.

- Curvature at the background control volume cv is then computed by simply adding the curvatures of LPs that lie inside the control volume.
- Curvature at the faces of the control volume are evaluated by arithmetic average of the two control volumes associated with the face. Here, the average is taken only if the both cvs contain the interface, i.e. color function $0 < \Psi_{cv} < 1$, else κ_f is assigned the value of κ_{cv} containing the interface.

Finally, the motion of the LPs is determined by a velocity field obtained by solving the Navier Stokes equations on a fixed background mesh. The velocity of each LP is obtained through interpolation from the background mesh. The motion of the LPs, may distort the initially uniformly spaced particles and a reconfiguration step is necessary wherein the distorted LPs are mapped to a uniformly spaced Cartesian lattice as described below.

Particle Map Distortion, Reconfiguration and Reinitialization

For the present particle-based method of interface representation, the LPs should overlap in order to obtain an accurate solution and interface properties such as normals and curvature. If the particle map gets highly distorted, the color function obtained from the LPs will no longer be smooth and continuous. This is overcome by performing a consistent re-configuration of the LP locations, termed as *remeshing*, around the interface. Here the Lagrangian points are redistributed on a Cartesian lattice with uniform spacing. After new sets of Lagrangian points are generated the values of the signed-distance function are obtained from the old ones by using higher order interpolations [8]. Remeshing removes any unphysical kinks in the interface and gives the 'entropy-satisfying viscous solution' [13]. It also eliminates unnecessary points away from the interface. For remeshing, we use the M_4' kernel to obtain the interpolated SDF values. Although the reconfiguration procedure provides the entropy solution, it does not guarantee that Φ remains a signed-distance to the interface, which is crucial to obtain accurate curvature and interface normals. In this work, reinitialization is implemented according to the method suggested by Sussman et al. [36, 37] in which the following equation is solved on uniformly spaced LPs:

$$\frac{\partial \Phi}{\partial \tau} = \text{sign}(\Phi_0)(1 - |\nabla \Phi|) \tag{14}$$

where $\Phi(x, 0) = \Phi_0$ and $\text{sign}(\Phi_0) \equiv 2(H_\epsilon(\Phi) - 1/2)$ and $H_\epsilon(\Phi)$ is the Heaviside function. We apply redistancing in a two-layer narrow band around the interface and using the procedure described in Gomez et al. [31].

Numerical Algorithm

The governing equations are solved using a colocated grid finite-volume algorithm [28, 29]. Accordingly, all variables are stored at the control volume (cv) centers with the exception of a face-normal velocity, located at the face centers, and used to enforce the divergence-free constraint. The variables are staggered in time so that they are located most conveniently for the time advancement scheme. Denoting the time level by a superscript index, the velocities are located at time level t^n and t^{n+1}, and pressure, density, viscosity, the signed distance function, and the color function at time levels $t^{n-1/2}$ and $t^{n+1/2}$. A balanced force algorithm [34, 35] is used for discrete balance of surface tension force and pressure gradient in the absence of any flow and other external forces. The basic steps are summarized below:

1. Advance the LPs (from $t^{n-1/2}$ to $t^{n+1/2}$) according to equations (1) and using a velocity field interpolated to the LP location from the background mesh. In this work, we use the M_4'-kernel based interpolation. We use third-order Runge-Kutta scheme to solve the ordinary differential equations for each LP.

2. Remesh and reinitialize the particle-map if necessary. Remeshing of LPs is necessary only if the particles cease to overlap as they adapt to the flow map. This is indicated by the distortion index (DI) [13]:

$$DI = \frac{1}{N} \sum_p \frac{|H_p(t) - H_p(0)|}{H_p(0)}, \tag{15}$$

where $H_p(t) = \sum_q v_q(t)\xi_\epsilon(\mathbf{x}_p(t) - \mathbf{x}_q(t))$, N is the number of Lagrangian points, v_q the volume of each LP, and ξ_ϵ the quartic spline mollification kernel. By selecting a proper threshold for DI the remeshing procedure can be triggered. Reinitialization is only necessary after a few remeshing steps, thus making the hybrid approach attractive. Reinitialization is done on remeshed LPs so that standard 5^{th}-order WENO scheme [38] can be used.

3. Once the LPs are advanced, curvature κ_{LP} is evaluated using the procedure outlined in the previous section. M_4'-kernel based interpolations are performed from the LPs to the background mesh to obtain curvature ($\kappa_{cv}^{n+1/2}$). Similarly, $\Psi_{cv}^{n+1/2}$ is obtained through interpolations and $\rho_{cv}^{n+1/2}$, and $\mu_{cv}^{n+1/2}$ are calculated from equations (7). The face-

based surface tension force is then obtained as:

$$F_{\sigma,f}^{n+1/2} = \sigma \kappa_f^{n+1/2} \frac{\Psi_{icv2}^{n+1/2} - \Psi_{icv1}^{n+1/2}}{|s_n|} \quad (16)$$

where s_n is the vector joining the control volumes $icv1$ to $icv2$.

4. The remaining steps are a variant of the co-located fractional step method as described by Ham & Young [20]. We present the semi-descretization here for completeness. First, a projected velocity field $\hat{u}_i$ at the cv-centers is calculated:

$$\frac{\hat{u}_i - u_i^n}{\Delta t} = g_i + \frac{1}{\rho_{cv}^{n+1/2}} \left(-\frac{\partial p}{\partial x_i}\right)^{n-1/2} + F_{v,i}^{n+1/2} + F_{\sigma,i}^{n+1/2})$$

where $F_{v,i}$ represents the viscous, $F_{\sigma,i}$ the surface tension, and g_i the gravitational forces at the cv centroids. The viscous terms are treated implicitly using second order symmetric discretizations and the surface tension force is treated explicitly. The cv-based surface tension force is obtained from $F_{\sigma,f}$ using area weighted least-squares interpolation consistent with the pressure reconstruction scheme developed by Mahesh *et al.* [28]. This is the essence of the balanced force algorithm [20, 34, 25].

5. Subtract the old pressure gradient:

$$u_i^{*n+1} = \hat{u}_i + \Delta t \frac{1}{\rho_{cv}^{n+1/2}} \left(\frac{\delta p}{\delta x_i}\right)^{n-1/2} \quad (17)$$

6. Obtain an approximation for the face-based velocity:

$$U_f^{*n+1} = \overline{u_i^{*n+1}} - \Delta t \left(\overline{\frac{F_i^{n+1/2}}{\rho_{cv}^{n+1/2}}} - \frac{F_{\sigma,f}^{n+1/2}}{\rho_f^{n+1/2}} \right) \quad (18)$$

where $\rho_f^{n+1/2} = (\rho_{icv1}^{n+1/2} + \rho_{icv2}^{n+1/2})/2$ and the interpolation operator, $\overline{\eta} = n_{i,f}[\eta_{icv1} + \eta_{icv2}]/2$, yields a face-normal component from the adjacent cvs associated with the face and the normal $n_{i,f}$.

7. Solve the variable coefficient Poisson equation to obtain pressure:

$$\frac{1}{\Delta t} \sum_{\text{faces of cv}} U_f^{*n+1} A_f = \sum_{\text{faces of cv}} \frac{1}{\rho_f^{n+1/2}} A_f \left(\frac{\delta p}{\delta n}\right)^{n+1/2} \quad (19)$$

where A_f is the face area.

8. Update the face-normal velocities by imposing a divergence free constraint and update the cv-based velocities from the reconstructed pressure gradient at the cv-centers:

$$\frac{U_f^{n+1} - U_f^{*n+1}}{\Delta t} = -\frac{1}{\rho_f^{n+1/2}} \left(\frac{\partial p}{\partial n}\right)^{n+1/2}$$

$$\frac{u_i^{n+1} - u_i^{*n+1}}{\Delta t} = -\frac{1}{\rho_{cv}^{n+1/2}} \left(\frac{\partial p}{\partial x_i}\right)^{n+1/2}$$

where the pressure gradient at the cv-centers $(\partial p/\partial x_i)^{n+1/2}$is obtained from the face-normal gradient using the *same* area-weighted least-squares minimization approach [28] used for the surface tension force above.

9. Interpolate the velocity field $u_{i,cv}^{n+1/2}$ to the LP locations and advance the LPs to the next time level.

Results

In this section, some numerical examples of standard test cases using the hLE scheme are presented. First, the accuracy of the pure Lagrangian advection approach is evaluated by performing standard test cases such as the Zalesak disc rotation and the evolution of a circular interface in a deformation field [39]. These showed comparable results with published data [13] and are not shown here. The accuracy of the surface normal and curvature evaluation procedure for a circular interface is compared with analytical solution to show second-order convergence. Next, we test the balanced-force algorithm and curvature evaluations on a stationary bubble in a quiescent, zero-gravity environment to investigate the level of spurious currents obtained due to errors in surface tension force representation. A systematic grid-refinement study is performed. Test cases such as damped surface waves, oscillating liquid column, Rayleigh Taylor instability, and rise (or fall) of bubbles (or droplets) under gravity are also simulated to show good accuracy.

Estimation of Surface Normal and Curvature

The accuracy of the surface normal and curvature calculation by using the procedure described before is tested on a circular interface. The Lagrangian points (LPs) are uniformly distributed in a narrow band around the interface and initialized by *exact* signed distance function. The surface normals and curvatures are first calculated on the LPs using the equations (4). Only those LPs are considered where $|\Phi| \leq 2\Delta_{LP}$, where Δ_{LP} is the LP-spacing. The average relative error in surface normal calculation is shown in Figure 1 indicating second order convergence. For all these LPs, corresponding points on the interface are calculated using the normals and signed distance function (Φ) (equation 13). The interface-projected curvature values at the interface points are evaluated using the M_4'-kernel based interpolation from the neighboring LP values. These curvatures are then compared with the exact curvature $\kappa_{exact} = 1/R$ for a two-dimensional interface.

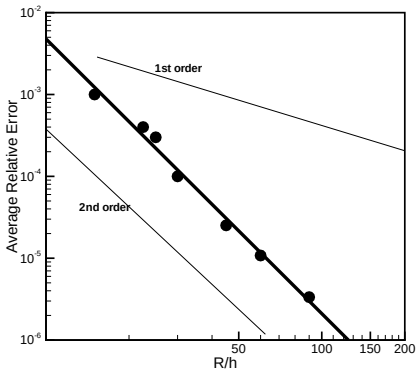

Figure 1: Average relative error (L_2) in surface normal for a circular interface.

The corresponding L_1-errors are plotted at different LP-resolutions in Figure 2, showing second order convergence similar to Herrmann [25]. Error calculations based on the κ values at the LPs are also plotted, showing only a first-order convergence, indicating the importance of the interface projected curvature calculation.

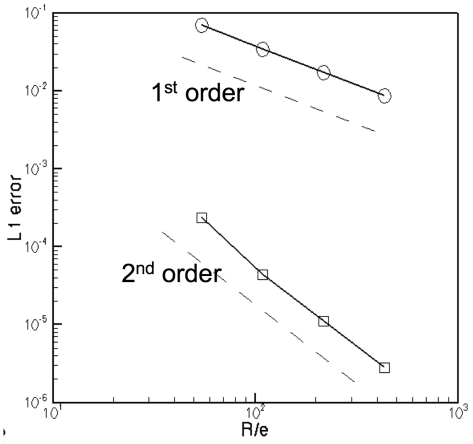

Figure 2: L_1-error in curvature for a circular interface: (a) convergence with refinement in LP-resolution. Circles indicate error based on κ at LP location, and squares indicate error based on κ on the interface.

Static Drop in Equilibrium

To validate the curvature and surface tension force calculation in hLE, we consider the test of static drop in a quiescent medium with zero gravity. The pressure gradient across the interface balances the surface tension force resulting a zero velocity and static drop at all times. Errors in representation of the surface tension and curvature at the interface, however, lead to non-zero velocity, or the so called 'spurious currents'. The exact solution for the pressure jump across the interface for a circular two-dimensional drop is: $\Delta P_{exact} = \sigma\kappa_{exact}$ where $\kappa_{exact} = 1/R$ and R is the radius of the drop. We consider a square domain having sides of eight units. A drop of radius $R = 2$ is placed at the center of the domain. The surface tension coefficient σ is taken to be 73, the drop density is 1 and the surrounding fluid density is 0.1. Accordingly, the pressure jump across the interface should be $\Delta P = 36.5$ units. All parameters are in SI units and correspond to the test case simulated by Francois *et al.* [34]. The background grid consists of uniform Cartesian elements with resolution of $R/\Delta = 10$. The resolution of the Lagrangian points is refined successively to have $R/\Delta_{LP} = 45, 60, 75$. The time step is fixed at $\Delta t = 10^{-3}$. The interface remains a perfect circle after $t = 0.5$ with low magnitudes of spurious currents. Table 1 shows the convergence of L_1-error in total kinetic energy at $t = 0.5$, indicating larger than second order convergence. Remeshing and reinitialization are *suppressed* in the above calculations, and the overall spurious current magnitudes are comparable to those reported by Herrmann [25].

Table 1: L_1 error in the total kinetic energy for a static drop in zero gravity $\frac{\Delta}{\Delta_{LP}} = 4.5$.

Δ	Error
0.044	4.5×10^{-8}
0.033	1.27×10^{-8}
0.0266	4.09×10^{-9}

Damped Surface Waves

Small amplitude damped surface wave between two immiscible fluids is investigated by comparing the numerical solution to the theoretical solution of the initial value problem obtained by Prosperetti [40]. Initially the interface between the two fluids inside a box $[0\ 2\pi] \times [0\ 2\pi]$ is perturbed by a sinusoidal wave disturbance of wavelength $\lambda = 2\pi$ and amplitude $A_0 = 0.01\lambda$ [35]. Periodic boundary conditions are used in the x direction, and slip conditions are used in the y-direction. The case analyzed consists of two fluids of equal density $\rho_1 = \rho_2 = 1$, and equal kinematic viscosities $\nu = 0.006472$. For $\sigma = 2$ and $\Delta t = 0.02$, the time-evolution of the amplitude of the surface is plotted in figure 3. Two different grid resolutions 16^2 and 32^2 are used for

this test case with the $\frac{\Delta}{\Delta_{LP}} = 4$. The coarse grid solution shows large errors in period and amplitude, however, with grid refinement, converging results are obtained.

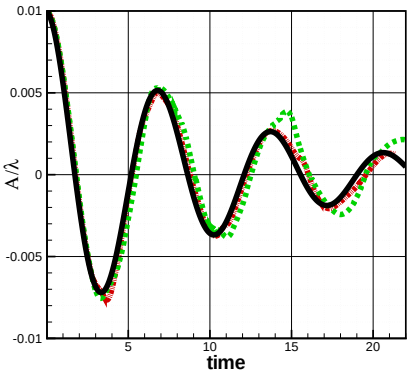

Figure 3: Time evolution of a damped surface wave for two grid resolutions $\Delta = 16^2$ (dashed) and 32^2 (dash dot) compared with theoretical result (solid). The LP resolution is held fixed at $\frac{\Delta}{\Delta_{LP}} = 4$.

Droplet Oscillation

Simulation of oscillating droplet or liquid column due to perturbations on the surface under zero-gravity conditions are performed to analyze the accuracy of the solver for capillary waves. A cylindrical liquid column with the radius perturbed according to:

$$r = r_0 + \alpha \cos(n\theta), \tag{20}$$

has a frequency of oscillation given by

$$\omega_n^2 = \frac{(n^3 - n)\sigma}{(\rho_d + \rho_e) r_0^3} \tag{21}$$

where ρ_d and ρ_e are the density, interior and exterior to the liquid column, respectively [6]. The cases considered include $\sigma = 1$, $\alpha = 0.1$ (this is 10% larger than the perturbation considered in [6]), $\rho_d = 1$, $\rho_e = 0.01$, $r_0 = 2$ in a $[-10, 10]^2$ doubly connected computational domain. The grid resolution is 32^2 and the Lagrangian particle resolution is fixed at $\frac{\Delta}{\Delta_{LP}} = 4.5$. The second, third, and fourth modes are simulated. Figure 4 shows the time evolution of an initially perturbed column over one period for the second mode. The relative error in period of oscillation $E_{period} = |T_{numerical}\omega_n/2\pi - 1|$ is given in Table 2. The errors are comparable to those reported in [6], even for ten times larger perturbation used in the present study.

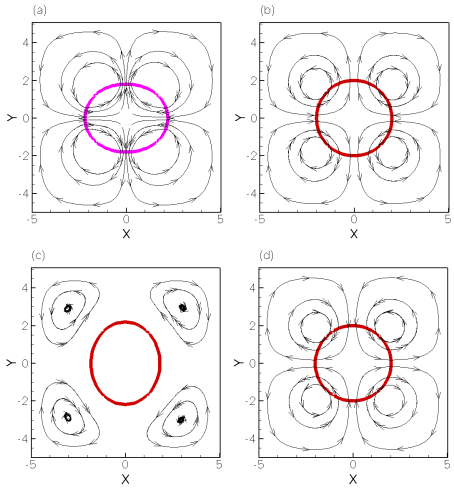

Figure 4: Time evolution of an oscillating cylindrical column over the initial period together with the streamline pattern: (a) $t = T/200$ (or close to initial), (b)$t = T/4$, (c) $t = T/2$, (d) $t = 3T/4$.

Table 2: Error in the predicted period of an oscillating liquid column for different modes on a 32^2 grid with $\frac{\Delta}{\Delta_{LP}} = 4.5$.

Mode	Predicted ω	Theoretical ω_n	E_{period}
2	0.844	0.862	0.020
3	1.671	1.726	0.0328
4	2.58	2.719	0.0519

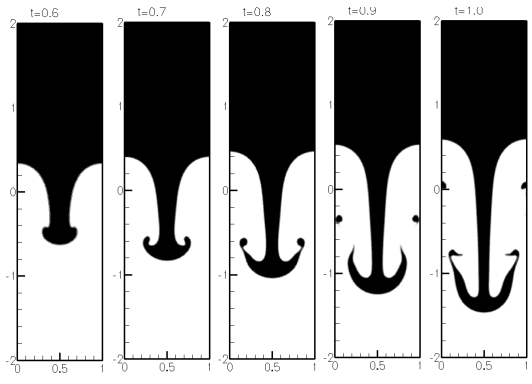

Figure 5: Time sequence of Rayleigh Taylor instability of $\Delta = 1/64$ grid.

Rayleigh Taylor Instability

The common test of heavier fluid over a lighter fluid giving rise to Rayleigh-Taylor instability is performed to evaluate the accuracy of the scheme ([41, 31, 35]. The simulation parameters are: $\rho_1 = 1.225$, $\mu_1 = 0.00313$ (heavy fluid), $\rho_2/\rho_1 = 0.1383$, $\frac{\mu_2}{\mu_1} = 1$, $g = 9.81$, $\sigma = 0$ in a computational domain of size $[1, 4]$. Initially the interface between the two fluids (at the center of the domain) is perturbed by a cosine wave of amplitude 0.05. The heavier fluid falls due to gravity, giving rise to a Rayleigh-Taylor instability of the interface perturbation. The surface tension forces are neglected. The boundaries in the x direction are assumed periodic whereas slip conditions are employed in the y direction. The time step is fixed at 5×10^{-4} and the flow evolutions for different grid resolutions are compared at certain time instances. The grid resolutions used are $\Delta = 1/64$, $1/128$, and $1/288$. The LP resolution relative the grid is fixed at $\frac{\Delta}{\Delta_{LP}} = 4.5$. The figures (5,6,7) show the instantaneous snapshots obtained from these grid resolutions. Loss of interface in high deformation regions

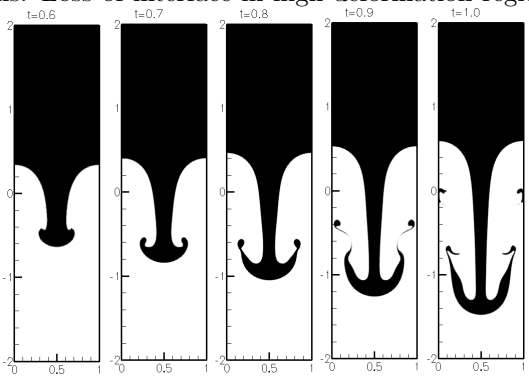

Figure 6: Time sequence of Rayleigh Taylor instability of $\Delta = 1/128$ grid.

is noticeable for the coarse grid ($1/64$ at $t = 0.9$ and 1). Increased resolution captures the thin filaments properly at $t = 0.9$ and the results are comparable to those by Herrmann [35] for respective resolutions.

Gravity Driven Motion of a Single Bubble and Droplet Column

Lastly, we consider the rise of an air bubble column in water and fall of water column in air under gravity [42]. Two cases involving small bubble/droplet columns are simulated. For small bubble/droplet, the computational domain is $[-0.01\ 0.01] \times [0\ 0.03]$ in the $x - y$ direction with no-slip wall conditions on the edges. A bubble (or droplet) of diameter $2/300$ is initially released from

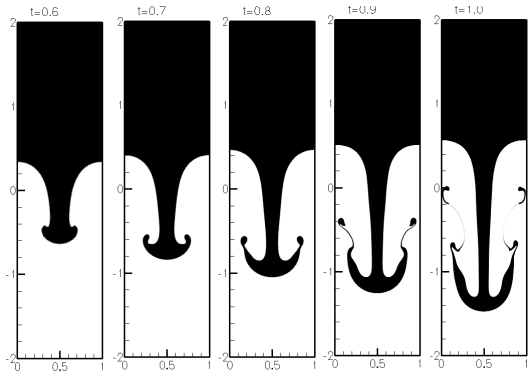

Figure 7: Time sequence of Rayleigh Taylor instability of $\Delta = 1/288$ grid.

a distance of 0.01 from the bottom wall (or top wall for droplet). The fluid properties of air and water are: $\rho_a = 1.226$, $\rho_\ell = 1000$, $\mu_a = 1.78 \times 10^{-5}$, $\mu_\ell = 1.137 \times 10^{-3}$, $\sigma = 0.0728$ and $g = 9.81$ in SI units.

Time evolution of the small bubble and droplet columns are shown in figures 8,9, respectively. The simulation was performed on a 80×120 grid with $\frac{\Delta}{\Delta_{LP}} = 4$. This corresponds to the finest grid used by Kang *etal.* [42] and the results are in good agreement with [42, 34]. The total volume loss at $t = 0.05$ was less than 0.8% for the bubble and 0.4% for the droplet.

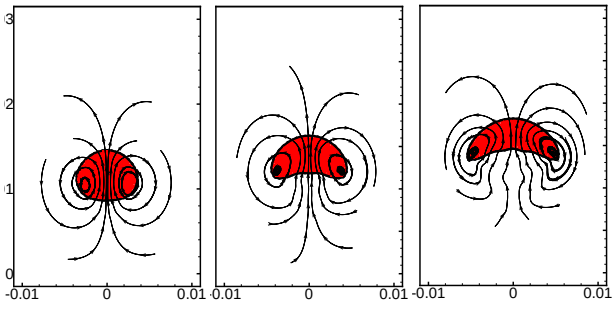

Figure 8: Time evolution of a small bubble rising under gravity: (a) t=0.02, (b) t=0.035, (c) t=0.049 s

A large bubble column (of size 100 times the above case) is also simulated in order to test the capability of the solver. Here the surface tension forces are too small to influence the inherent Kelvin-Helmholtz and Rayleigh-Taylor instabilities. The computational domain is $[-1\ 1] \times [0\ 3]$ and the bubble of diameter $2/3$ is released from unit distance from the bottom wall. Figure 10 shows the time evolutions obtained on a 80×120 grid with $\frac{\Delta}{\Delta_{LP}} = 4$. In this case, the deformation of the bubble is larger,

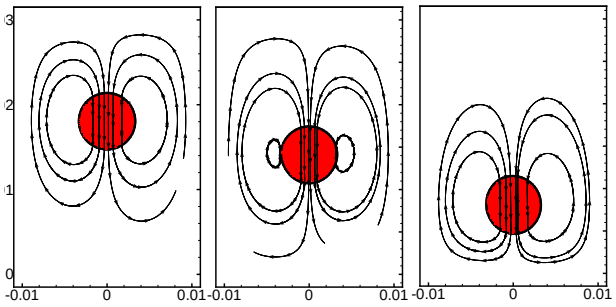

Figure 9: Time evolution of a small droplet column falling under gravity: (a) t=0.02, (b) t=0.035, (c) t=0.049 s

and the volume loss obtained is on the order of 1.5%. Results are in agreement with the terminal shapes reported by Kang *et al.* [42].

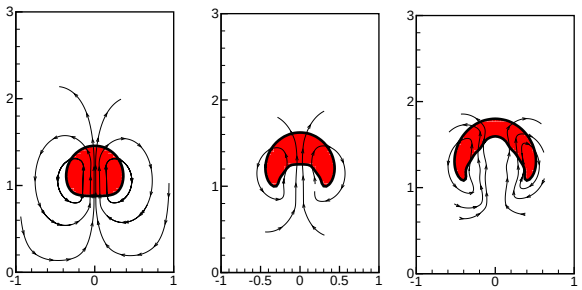

Figure 10: Time evolution of a large bubble rising under gravity: (a) t=0.2, (b) t=0.35, (c) t=0.5 s

Summary

A new hybrid Lagrangian-Eulerian (hLE) scheme, combining a mesh-free Lagrangian technique with a finite-volume flow solver, has been developed for direct simulations of two-phase flows with fully resolved interfaces. This approach merges the naturally adaptive nature of particle-based schemes, for efficient representation of the interface between two media, with the relative flexibility offered by grid-based solvers for complex flows. In hLE, a mesh-free, particle-based scheme for interface tracking [13] is integrated with a co-located grid based finite volume solver. The potential advantage of the hLE method is that the background mesh could be of any kind: *structured, body-fitted, or arbitrary shaped unstructured* (hex, pyramids, tetrahedrons, prisms) and may be *stationary or changing in time (adaptive refinement)*. In this work, we used uniform Cartesian grids for the background mesh. A balanced force algorithm [25, 34] for accurate representation of surface tension forces and considerably reduced magnitudes of spurious currents, is used to solve the two-phase flow equations. The accuracy of the hLE scheme is first verified for standard test cases on interface tracking including passive advection by a specified velocity field (Zalesak's disk, vortex in a box [39]). Test cases presented include (i) parasitic currents for a static drop in equilibrium, (ii) damped surface waves, (iii) capillary waves on oscillating droplet column, (iv) Rayleigh Taylor instability, and (v) rise (fall) of bubble (droplet) under gravity. Good predictive capability compared to other schemes is obtained.

Presence of the interface (and hence Lagrangian points) only on some processors partaking in the simulation, however, gives rise to load imbalance, especially if the computational domain is partitioned based on the background mesh. Dual-constraint partitioning that optimize the number of grid cells and the number of Lagrangian points are necessary to balance the load and improve the efficiency of the scheme. These dynamic-load balancing strategies together with advanced methods for finding the nearest neighbors within the mollification kernels will make the current approach feasible for full three-dimensional simualtions [9].

Acknowledgments

This work is supported by the Office of Naval Research (ONR) grant number N000140610697, under the supervision of Dr. Ki-Han Kim.

SVA acknowledges Marcus Herrmann (Arizona State University) for several useful discussions on level set methods, curvature calculations and his RLSG approach.

References

[1] G. Tryggvason, B. Bunner, A. Esmaeeli, N. Al-Rawahi, W. Tauber, J. Han, YJ Jan, D. Juric, and S. Nas. *J. Comput. Phys.*, 169(2):708–759, 2001.

[2] F.H. Harlow and J.E. Welch. *Phy. Fluids*, 8:2182–2190, 1965.

[3] JU Brackbill, DB Kothe, and HM Ruppel. *Computer Physics Communications*, 48(1):25–38, 1988.

[4] W.J. Rider and D.B. Kothe. *AIAA Paper*, 95:17, 1995.

[5] S.O. Unverdi and G. Tryggvason. *J. Comput. Phys.*, 100(1):25–37, 1992.

[6] Torres D.J. and J.U. Brackbill. *J. Comp. Physics*, 165(2):620–644, 2000.

[7] S. Koshizuka, A. Nobe, and Y. Oka. *Int. J. Numer. Meth. Fluids*, 26(7):751–769, 1998.

[8] P. Koumoutsakos. *Annu. Rev. Fluid Mech.*, 37(1):457–487, 2005.

[9] IF Sbalzarini, JH Walther, M. Bergdorf, SE Hieber, EM Kotsalis, and P. Koumoutsakos. *J. Comput. Phys.*, 215(2):566–588, 2006.

[10] JJ Monaghan. *Reports on Progress in Physics*, 68(8):1703–1759, 2005.

[11] HY Yoon, S. Koshizuka, and Y. Oka. *Int. J. Multiphase Flow*, 27(2):277–298, 2001.

[12] JH Walther, T. Werder, RL Jaffe, and P. Koumoutsakos. *Physical Review E*, 69(6):62201, 2004.

[13] S.E. Hieber and P. Koumoutsakos. *J. Comput. Phys.*, 210(1):342–367, 2005.

[14] R. Scardovelli and S. Zaleski. *Annu. Rev. Fluid Mech.*, 31(1):567–603, 1999.

[15] WF Noh and P. Woodward. *Int. Conf. Numer. Meth. Fluid Dynamics, 5th, Enschede, Netherlands, June 28-July 2, 1976,*, 1976.

[16] JA Sethian. *J. Comput. Phys.*, 169(2):503–555, 2001.

[17] S. Osher and R.P. Fedkiw. *J. Comput. Phys.*, 169(2):463–502, 2001.

[18] M. Sussman, P. Smereka, and S. Osher. *J. Comput. Phys.*, 114(1):146–159, 1994.

[19] M. Sussman. *J. Comput. Phys.*, 187(1):110–136, 2003.

[20] F. Ham and YN. Young. *Annu. Research Briefs, 2003: Center for Turbulence Research*, 2003.

[21] F. Ham, Y.N. Young, S. Apte, and M. Herrmann. *2 nd Int. Conf. Comput. Meth. Multiphase Flow*, pp. 313–322, 2003.

[22] D. Enright, R. Fedkiw, J. Ferziger, and I. Mitchell. *J. Comput. Phys.*, 183(1):83–116, 2002.

[23] M. Herrmann. *Annu. Research Briefs, 2004: Center for Turbulence Research*, pp. 15–29, 2004.

[24] M. Herrmann. *Annu. Research Briefs, 2005: Center for Turbulence Research*, 2005.

[25] M. Herrmann. *Annu. Research Briefs, 2004: Center for Turbulence Research*, 2006.

[26] J. Strain. *J. Comp. Phy.*, 170:373–394, 2001.

[27] CW Hirt, AA Amsden, and JL COOK. *J. Comput. Phys.*, 14(3):227–253, 1974.

[28] K. Mahesh, G. Constantinescu, and P. Moin. *J. Comput. Phys.*, 197(1):215–240, 2004.

[29] K. Mahesh, G. Constantinescu, S. Apte, G. Iaccarino, F. Ham, and P. Moin. *J. Applied Mech.*, 73:374, 2006.

[30] D. Peng, B. Merriman, S. Osher, H. Zhao, and M. Kang. *J. Comput. Phys.*, 155(2):410–438, 1999.

[31] P. Gómez, J. Hernández, and J. López. *Int. J. Numer. Meth. Eng.*, 2005.

[32] Jeff D. Eldredge, Anthony Leonard, and Tim Colonius. *J. Comput. Phys.*, 180(2):686–709, 2002.

[33] JU Brackbill, DB Kothe, and C. Zemach. *J. Comput. Phys.*, 100(2):335–354, 1992.

[34] M.M. Francois, S.J. Cummins, E.D. Dendy, D.B. Kothe, J.M. Sicilian, and M.W. Williams. *J. Comput. Phys.*, 213(1):141–173, 2006.

[35] M. Herrmann. *J. Comp. Physics*, 227(4):2674–2706, 2008.

[36] E. Fatemi and M. Sussman. *SIAM J. Sci. Comput.*, 158(1):36–58, 1995.

[37] M. Sussman, E. Fatemi, P. Smereka, and S. Osher. *Computers & Fluids*, 27(5):663–680, 1998.

[38] GS Jiang and D. Peng. *SIAM J. Sci. Comput.*, 21(6):2126–2143, 2000.

[39] Shams E. and Apte S.V. *ILASS America's* 20^{th} *Annual Conference on Liquid Atomization and Spray Systems, Chicago, IL, May 2007*, 30, 2007.

[40] A. Prosperetti. *Phy. Fluids*, 24(7):1217–1223, 1981.

[41] Popinet S. and S. Zaleski. *Int. J. Numer. Methods Fluids*, 30:775–793, 1999.

[42] M. Kang, R.P. Fedkiw, and X.D. Liu. *J. Sci. Comput.*, 15(3):323–360, 2000.