ILASS Americas 20th Annual Conference on Liquid Atomization and Spray Systems, Chicago, IL, May 2007

A Hybrid Lagrangian-Eulerian Approach for Two-Phase Flows with Fully Resolved Interfaces

Ehsan Shams and Sourabh V. Apte*
Department of Mechanical 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. 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. A mesh-free, particle-based scheme for interface tracking (Hieber & Koumoutsakos 2005) is first integrated with a co-located grid based finite volume solver. A balanced force algorithm (Francois *et al.* 2006, Herrmann 2006), for accurate representation of surface tension forces, is used to solve the two-phase flow equations on a fixed background mesh. The accuracy of the particle-based scheme is first verified for standard test cases on interface tracking. The hybrid scheme is then applied to perform coupled two-phase flow simulations of stationary drop and rising bubbles.

*Corresponding Author: sva@engr.orst.edu

Introduction

Many problems in nature and engineering involve multiphase flows where a dispersed phase (solid, liquid, or vapor) exists, in the form of a cloud of heterogeneous size, in an ambient fluid (gas or liquid) undergoing time dependent and often turbulent motion. Examples include liquid fuel atomization in combustion chambers of propulsion systems, bubbly flows and drag reduction, cavitation and bubble capture over ship propellers, among others. These applications involve disparate length and time scales for common physical phenomena of mass, momentum, and energy transport across the interface between the dispersed phase and a continuum fluid. Numerical simulation of these flows requires modeling of the physics of deformation, breakup, collision/coalescence and the multiscale nature of two-phase turbulent flows with or without chemical reactions.

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) Lagrangian and (b) Eulerian approach.

Lagrangian methods use marker-points connected to each other representing the interface. The front is tracked by advecting the marker points [2, 3, 4] and the Navier-Stokes equations are solved on a fixed grid in an Eulerian frame. Although the accuracy of the method is very promising the change in topology of the droplets/bubbles is not handled automatically, resulting in increased complexity of the algorithm for three-dimensional reconstruction of the interface from marker points [1]. Another class of Lagrangian methods include *mesh-free* algorithms such as moving particle-methods [5], vortex-in cell methods [6, 7], and smoothed-particle hydrodynamics [8], where the interface is represented by Lagrangian points (LPs) and the flowfield 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 [6] and have been applied successfully to many two-phase flow problems [9, 10, 11]. 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 [6].

Eulerian approaches such as the volume-of-fluid (VOF) [12, 13] or the level-set method [14, 15, 16] are used extensively for two-phase flow computations. Both approaches 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 [17], adaptive mesh-refinement [18, 19], particle-level sets [20], refined level-set grid scheme [21, 22, 23] have been proposed for improved accuracy.

In addition, a variety of methods with boundary-fitted grids [24], and deforming grids [25] and boundary element methods, Lattice Boltzmann method [26, 27] have been developed, but are restricted to simplified geometries and flows.

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 [11]. 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) [28] 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.* [29, 30].

The Lagrangian points (LPs) in our interface calculations, are particles[1] distributed in a narrow band around the interface [31]. 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 ini-

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

tially uniform LPs. Regularization of the particles are performed by mapping the particles on a uniformly spaced lattice [11]. Values for particle properties at new LP locations are obtained through kernel mollification as done in Smoothed Particle Hydrodynamics [8] and remeshed-SPH [10]. 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 [22, 23, 32]. However, here the LPs move in space with the flow velocity as opposed to being fixed in the RLSG scheme. Thus 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 are described followed by description of the numerical scheme. The numerical approach is then applied to standard test cases to evaluate the accuracy of the scheme compared to other approaches. Finally, some preliminary results on rising bubble in a quiescent fluid are presented.

Governing Equations

In the Level Set method [33, 31] the incompressible, immiscible, two-fluid system is treated as a single fluid with strong variations in density and viscosity in the neighborhood of the interface. Let the two fluids be labeled as '1' and '2' and represented by a higher order signed distance function (SDF), Φ. The sign of the function Φ identifies which fluid is present whereas $\Phi = 0$ represents the interface. The evolution of the interface is given by the following scalar transport equation:

$$\frac{\partial \Phi}{\partial t} + \mathbf{u} \cdot \nabla \Phi = \frac{D\Phi}{Dt} = 0 \tag{1}$$

where $\frac{D}{Dt}()$ is the material derivative. Then the Navier-Stokes equations for the two-fluid system can be written as,

$$\nabla \cdot \mathbf{u} = 0 \tag{2}$$

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho} \nabla p + \frac{1}{\rho} \nabla \cdot (\mu(\nabla \mathbf{u} + \nabla^T \mathbf{u})) + \mathbf{g} + \frac{1}{\rho} \mathbf{F}_\sigma \tag{3}$$

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 ($\Phi = 0$). Density and viscosity are related to the Φ value:

$$\rho(\Phi) = \rho_1 + (\rho_2 - \rho_1) H(\Phi) \tag{4}$$

$$\mu(\Phi) = \mu_1 + (\mu_2 - \mu_1) H(\Phi) \tag{5}$$

where subscripts 1 and 2 denote fluids with $\Phi < 0$ and $\Phi > 0$, respectively and H is the heaviside function [17]. The interface normal and curvature values are given as:

$$\mathbf{n} = \frac{\nabla \Phi}{|\nabla \Phi|}, \qquad \kappa = \nabla \cdot \frac{\nabla \Phi}{|\nabla \Phi|} \tag{6}$$

Hybrid Lagrangian-Eulerian (hLE) Scheme

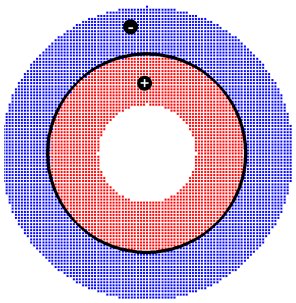

Figure 1: Uniformly distributed Lagrangian points (LPs) in a thin band around the interface (solid line). Each LP carries a signed distance Φ to the interface which identifies the type of fluid.

As mentioned earlier, in the present work, the locally 'adaptive' particle-based method for interface representation is merged with the Eulerian finite-volume formulation in order to inherit the advantages offered by individual approaches. Following Hieber & Koumoutsakos [11], the interface between two fluids is represented using uniformly spaced Lagrangian points (LPs) or fictitious particles in a narrow band around the interface as shown in Figure 1. 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 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. The important steps in this hybrid approach are described below.

Lagrangian Particle Level Set

The evolution of the interface is calculated by solving level set equations in the Lagrangian form:

$$\frac{D\Phi_p}{Dt} = 0; \;\; \frac{D\mathcal{V}_p}{Dt} = \langle \nabla \cdot \mathbf{u} \rangle_p \, \mathcal{V}_p; \;\; \frac{D\mathbf{x}_p}{Dt} = \mathbf{u}_p, \qquad (7)$$

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 $\mathbf{x}_q$:

$$\Phi_q = \sum_{p=1}^{N} \mathcal{V}_p \Phi_p \xi_\epsilon(\mathbf{x}_q - \mathbf{x}_p) \qquad (8)$$

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 [8]. 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 [34]:

$$\langle \nabla \Phi \rangle_q = \sum_p \mathcal{V}_p \left(\Phi_p - \Phi_q \right) \nabla \xi_\epsilon(\mathrm{x}_q - \mathrm{x}_p), \qquad (9)$$

$$\langle \nabla^2 \Phi \rangle_q = \sum_p \mathcal{V}_p \left(\Phi_p - \Phi_q \right) \nabla^2 \xi_\epsilon(\mathrm{x}_q - \mathrm{x}_p). \qquad (10)$$

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 [34]).

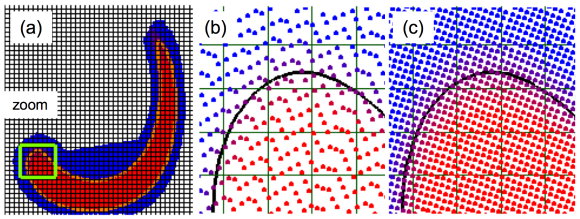

Figure 2: Reconfiguration procedure: (a) particle-based interface tracking, (b) distorted particle map, (c) reconfigured particles. The background grid, although not necessary for LP advection, is shown to elucidate the hLE approach.

Particle Map Distortion, Reconfiguration and Reinitialization

Figure 2 shows potential problems with particle-based methods during simple advection of a spherical interface by a stationary vortical flow. This example is investigated in detail later. Initially all LPs are positioned with uniform spacings between them (Fig. 1). A background vortical flow distorts the initial spherical interface (Fig. 2a), and the LPs may overlap or grow apart (Fig. 2b). This LP distortion leads to a problem similar to the development of kinks when the level-set is evolved using a non-dissipative central differencing [14], representing 'crossing' of particle trajectories in a hyperbolic system. For the present particle-based method, the LPs should overlap in order to obtain the correct entropy solution. If the particle map gets highly distorted, the level set function will no longer be smooth and continuous. This is overcome by performing a consistent re-configuration of the particle locations, termed as *remeshing*, around the interface (Fig. 2c). 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 [6]. Remeshing removes any unphysical kinks in the interface and gives the 'entropy-satisfying viscous solution' [11]. It also eliminates unnecessary points away from the interface. For remeshing, we use the M_4' kernel to obtain the interpolated SDF values:

$$M_4' \;\; = \;\; \begin{cases} 1 - 5/2s^2 + 3/2s^3 & 0 \le s < 1 \\ 1/2(1-s)(2-s)^2 & 1 \le s < 2 \\ 0 & 2 \le s \end{cases} \qquad (11)$$

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

parts. 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. [35, 36] in which the following equation is solved on uniformly spaced LPs:

$$\frac{\partial \Phi}{\partial \tau} = \text{sign}(\Phi_0)(1 - |\nabla \Phi|) \tag{12}$$

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. [32].

Coupling to the Flow Solver

As mentioned earlier, the signed distance function defined on the Lagrangian points implicitly represent the interface. The velocity at the Lagrangian points is obtained from interpolation of the velocity field computed on a fixed background mesh. The coupling between the particle-based interface representation and the finite-volume flow solver occurs only through this interpolated velocity field. The background mesh resolution is assumed to be coarser than the particle resolution. Typically a factor of (≥ 3) is used between the two resolutions to fit the width of the quartic spline mollification kernel on all LPs within the control volume and its *immediate* neighbors.

Even though Lagrangian points provide sub-cell characterization of the interface, the resolution of the background mesh plays important role in coupling the pressure and velocity fields; which depends on the actual location of the interface and density variations between the two phases.

For efficient implementation, each LP or particle is first located on the background mesh and the corresponding control volume is identified. The background mesh connectivity is then used to identify the immediate neighbors of each particle that influence the mollification operation used for interpolations. Finding neighboring particles is the most time consuming part of the scheme and advanced techniques based on octree structures, Verlet lists [37] and linked-lists are required for efficient implementation [7]. In the present work, focus is placed on the feasibility of the hybrid approach and issues of numerical efficiency are deferred to follow-on work. For the cases considered here, use of the background mesh connectivity to search for the nearest neighbors is found sufficient. However, note that, LP resolutions of more than five-times finer than the background mesh results in significant slow-down in finding all neighbors of LPs within the kernel width. For present work we used a factor of three or four.

In order to solve the Navier Stokes equations on the background mesh, the interface represented on the LPs is used to evaluate a color function Ψ indicating the volume of each fluid within the computational cell and and the surface tension force. 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 mesh by interpolating Ψ from the LPs. In order to obtain a smooth function, the M_4' kernel interpolation (equation 11) is used. Once the color function Ψ is obtained at a control volume (cv), the density and viscosity are given as:

$$\rho_{cv} = \rho_1 + (\rho_2 - \rho_1)\Psi_{cv} \tag{13}$$

$$\mu_{cv} = \mu_1 + (\mu_2 - \mu_1)\Psi_{cv} \tag{14}$$

The surface tension force in the Navier Stokes equation also requires knowledge of the interface location. Following Brackbill *et al.* [38], the surface tension force is modeled as a continuum surface force (CSF):

$$\mathbf{F}_\sigma^{CSF} = \sigma \kappa \hat{\mathbf{n}} \delta(\Phi) \tag{15}$$

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, 18] due to inaccuracies in the discrete approximations to the surface-tension forces (equation 3). 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.* [39] indicated that the surface tension force must be evaluated at the faces of the control volumes as:

$$\mathbf{F}_{\sigma,f}^{CSF} = \sigma \kappa_f \left(\nabla \Psi\right)_f \tag{16}$$

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. This force reconstruction should be consistent with the pressure-gradient reconstruction typically required in co-located mesh algorithms. This procedure, termed as 'balanced force algorithm', was also developed earlier by Ham *et al.* [18] and recently by Herrmann [23] to obtain considerably reduced magnitudes of spurious currents. In the present work, a similar approach is

followed. First the gradients of the color function at the faces of the control volumes are obtained as:

$$\nabla\Psi_f = \frac{\Psi_{icv2} - \Psi_{icv1}}{|s_n|} \tag{17}$$

where $icv1$ and $icv2$ are the control volumes associated with a face and $|s_n|$ is the distance between the centroids of the two control volumes. With the balanced force algorithm it was shown by Francois *et al.* that any spurious velocity currents appear *solely* due to errors in curvature evaluations. Herrmann [23] later developed a procedure to compute the curvature accurately in the level-set framework. Here we follow a similar procedure for curvature evaluations at the LPs:

- First the curvature is evaluated using equations 10 on the Lagrangian points with $|\Phi| \leq 2h$, where h is the spacing between the LPs.

- For each of these LPs (with $|\Phi| \leq 2h$) , a point on the interface is obtained by projecting normals onto the interface [23]:

$$\mathbf{x}_{interface} = \mathbf{x}_{LP} - \frac{\Phi}{|\nabla\Phi|}\hat{\mathbf{n}} \tag{18}$$

- 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 11).

- 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.

With the above procedure, it will be shown later that accurate estimate of interface curvatures can be obtained. Below we briefly summarize the steps involved in the hybrid approach.

Numerical Algorithm

The governing equations are solved using a co-located grid finite-volume algorithm [29, 30]. 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 $t^{n+1/2}$. The basic steps are summarized below [18, 23]:

1. Advance the LPs (from $t^{n-1/2}$ to $t^{n+1/2}$) according to equations (7) 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) [11]:

$$DI = \frac{1}{N}\sum_p \frac{|H_p(t) - H_p(0)|}{H_p(0)}, \tag{19}$$

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 [40] 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 (14). The face-based surface tension force is then obtained as:

$$F_{\sigma,f}^{n+1/2} = \sigma\kappa_f^{n+1/2}\frac{\Psi_{icv2}^{n+1/2} - \Psi_{icv1}^{n+1/2}}{|s_n|} \tag{20}$$

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 [18]. We present the semi-descretization here for completeness. First, a projected velocity field $\hat{u}_i$ at the *cv*-centers is calculated:

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

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.* [29]. This is the essence of the balanced force algorithm [18, 39, 23].

5. Subtract the old pressure gradient:

$$u_i^{*n+1} = \hat{u}_i + \Delta t \frac{1}{\rho_{cv}^{n+1/2}} \frac{\delta p}{\delta x_i}^{n-1/2} \tag{21}$$

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

$$U_f^{*n+1} = \overline{u_i^{*n+1}} - \Delta t \left(\frac{\overline{F_i^{n+1/2}}}{\rho_{cv}^{n+1/2}} - \frac{F_{\sigma,f}^{n+1/2}}{\rho_f^{n+1/2}} \right) \tag{22}$$

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 *cv*s associated with the face and the normal $n_{i,f}$.

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

$$\frac{1}{\Delta t} \sum_{\text{faces of cv}} U_f^{*n+1} A_f = \sum_{\text{faces of cv}} \frac{1}{\rho_f^{n+1/2}} A_f \frac{\delta p}{\delta n}^{n+1/2} \tag{23}$$

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:

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

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

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 [29] 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, we verify our implementation of the particle-based interface representation scheme [11] wherein the velocity field is specified and not computed. 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. Finally, a preliminary calculation of a rising bubble against gravity is performed and compared with other results published in literature.

Zalesak's Slotted Disk

The first test case is rigid body rotation of a slotted disk. The slotted disk is initially located at (0.5,0.75) in a [1x1] domain and completes one revolution after 628 time steps with a specified velocity field:

$$\mathbf{v}(\mathbf{x}) = \frac{\pi}{314} \begin{bmatrix} 0.5 & -y \\ x & -0.5 \end{bmatrix} \tag{24}$$

Since the flow velocity represents a rigid-body rotation, the interface shape should remain unaltered. Any error in the area bounded by the interface will be related to the interface capturing scheme including time integration and calculation of the color function using mollification kernel. Table 1 shows the relative errors in area after one revolution with different particle spacings. The relative error in the area ($\mathcal{A}$) bounded by the interface is computed as: Error $= (\mathcal{A}_{\theta=0} - \mathcal{A}_{\theta=360})/\mathcal{A}_{\theta=0} \times 100$. Fig. 3 shows the zero level set at different stages during one revolution. As expected, the particle-based interface representation is *exact* for rigid-body motion. Any

Table 1: Relative error of area calculation on slotted disk after one revolution.

Particle spacing	Error (%)	Particles
1/200	2	3109
1/400	0.2	12382
1/500	0.02	19387

errors obtained are due to the Runge-Kutta time-integration scheme and converge rapidly with refinement in the particle mesh. These results are consistent with those presented by Hieber and Koumoutsakos [11].

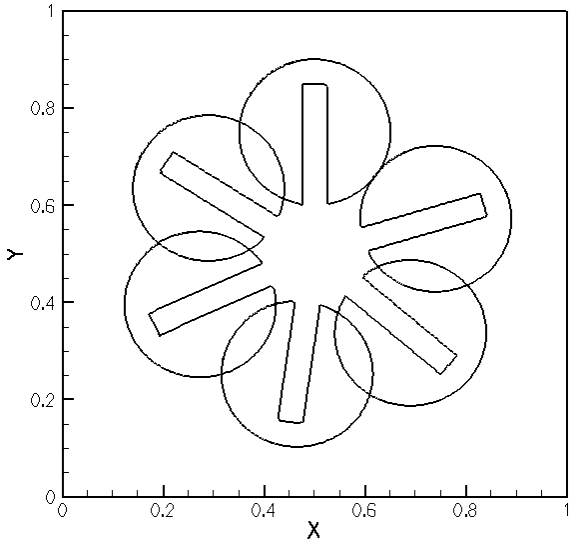

Figure 3: Zero level set (SDF) of the slotted disk in different angles.

Two-Dimensional Single Vortex Flow

The second test case involves advection of circular interface in a specified and steady deformation field introduced by Bell et al. [41] and is used as a benchmark in most of the interface tracking and capturing methods. The velocity field in this case causes extreme elongation of the circular region and usually is used to show the ability of methods on capturing narrow regions. Equation 25 shows the velocity field in 2D.

$$\mathbf{v}(\mathbf{x}) = 2 \begin{bmatrix} -\sin^2(\pi x)\sin(\pi y)\cos(\pi y) \\ -\sin^2(\pi y)\sin(\pi x)\cos(\pi x) \end{bmatrix} \qquad (25)$$

We calculate the error related to our implementation of remeshing and reinitialization methods based on the area inside the region with time. In the first part, the interface is evolved by the above velocity field up to $t = 3$ units with a time-step of $\Delta t = 1/30$ units (90 iterations). The distortion index (DI) defined by equation (19) is set to 10^{-7} resulting in remeshing at every time-step. The time evolution of the interface at different particle resolutions is shown in Figure 4. As the interface gets elongated, the tail of the interface becomes under resolved and some portions of the interface are lost. The relative error in the volume of the initially circular interface is shown

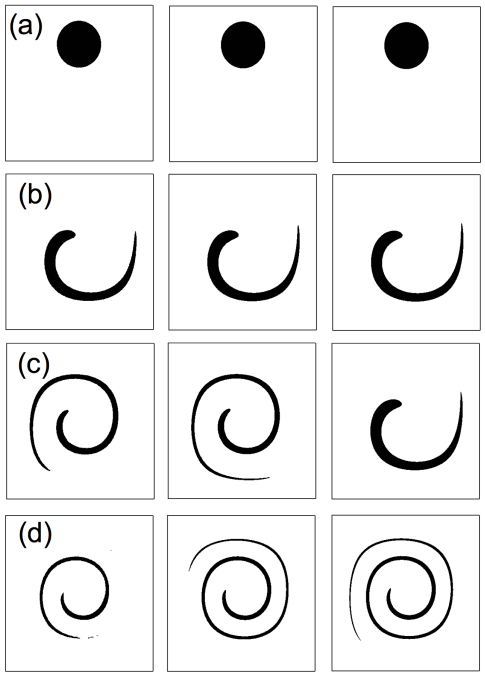

Figure 4: Time evolution of the interface at different particle-resolutions. Column 1: $h = 1/128$, Column 2: $h = 1/256$, Column 3: $h = 1/512$. Each row corresponds to a specific time: (a) $t = 0$, (b) $t = 1$, (c) $t = 2$, (d) $t = 3$.

in Table 2. With increased resolution, the interface is well resolved and the loss in volume is less than 2%. The results are consistent with those reported in the literature [11].

Table 2: Ratio of area calculation at $t = 2$ and $t = 3$ for different particle spacings.

Particle spacing	$t = 2$	$t = 3$
1/64	0.3133	0.0428
1/128	0.8796	0.5210
1/256	0.9750	0.9162
1/512	0.9931	0.9887

As a second test, the circular interface is evolved by a modified velocity field which reverses direction after $t = 4$. The velocity field v(x) in equation 25 is multiplied by $\cos(\pi t/T)$, with the period $T = 8$. Accordingly, the circular interface should elongate and deform up to $t = 4$ and then contract back to the original circular shape (at $t = 8$). The calculation is then performed up to $t = 8$ and the area under the interface is compared with the original area at $t = 0$ as shown in Table 3. The results are consistent with

those reported in the literature [11].

Table 3: Ratio of area for the reverse vortex case at $t = 8$ and different particle resolutions.

Particle spacing	$t = 8$
1/96	0.0831
1/128	0.5651
1/256	0.9796
1/320	0.9892

Estimation of Curvature on a Circular Interface

To test the accuracy of the curvature calculation by using the procedure described before we calculated curvature for 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 (10). Only those LPs are considered where $|\Phi| \leq 2h$, where h is the LP-spacing. For all these LPs, corresponding points on the interface are calculated using the normals and signed distance function (Φ) (equation 18). The 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. The corresponding L_1-errors are plotted at different LP-resolutions in Fig. 5a, showing second order convergence similar to Herrmann [23].

Error calculations based on the κ values at the LPs are also shown, indicating only a first-order convergence. Since the LPs are located in a band around the interface, the κ values at their locations are far from $1/R$, the exact value, indicating the importance of finding the point on the interface for each LP value. Also the actual error distribution is shown along the circumference of the circle as a scatter plot in Fig. 5b, indicating relatively small scatter in kappa values.

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 media 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

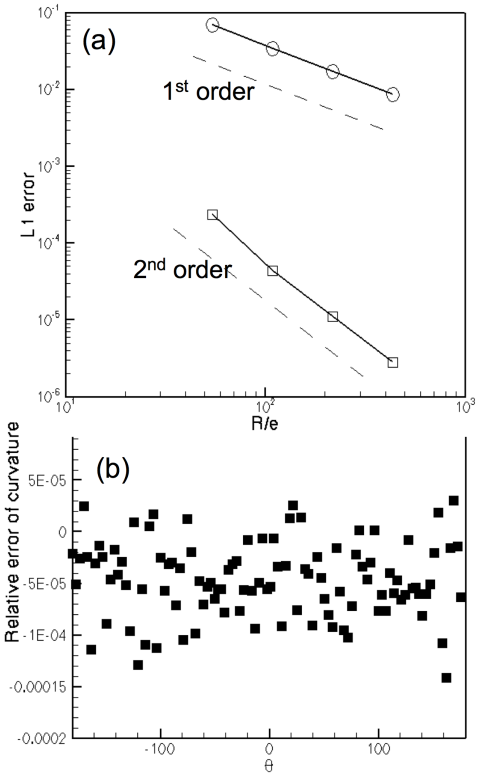

Figure 5: L_1-error in curvature for a circular interface: (a) convergence with refinement in LP-resolution. Circles indicate error based on κ at LP location, and squares indicate error based on κ on the interface, (b) distribution of error around the circumference.

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.* [39].

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 $R/h = 45, 60, 75$. Figure 6a shows the variation of the pressure drop across the centerline of the droplet after one time-step of $\Delta t = 10^{-3}$ for the particle resolution of $R/h = 45$. The pressure drop is very well predicted and is smeared across the interface over one background mesh resolution. The

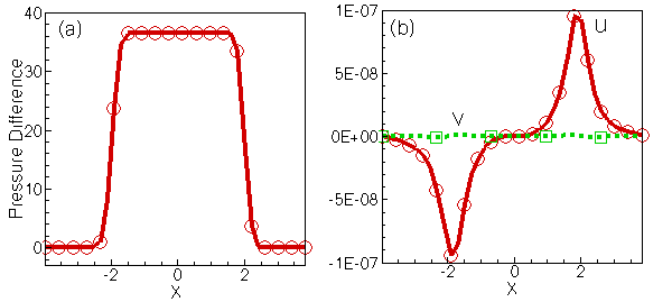

Figure 6: Variation of pressure difference (a) and velocity fields (b) along the x-axis passing through the center of the bubble.

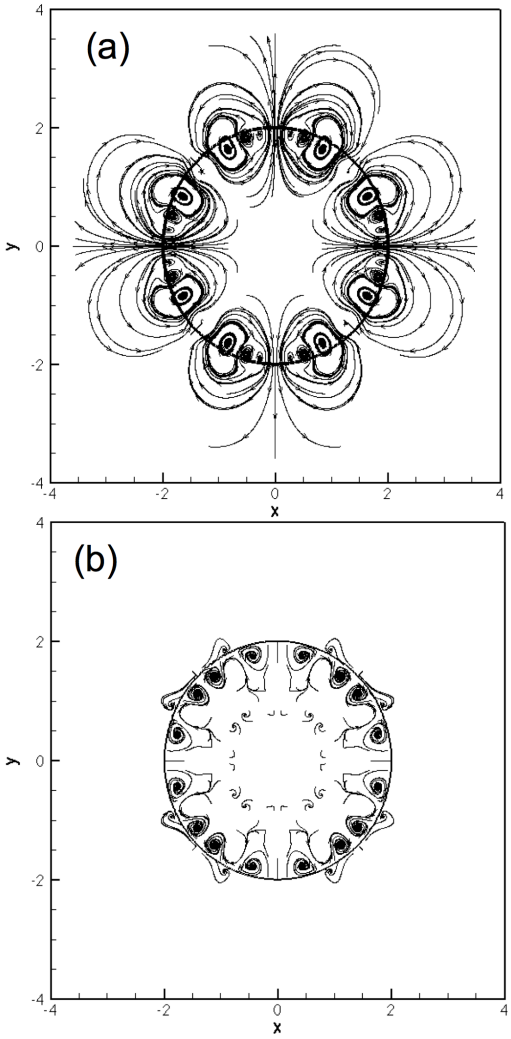

Figure 7: Streamlines showing symmetric pattern of spurious currents for the static bubble case: (a) after one time-step, (b) at $t = 0.5$.

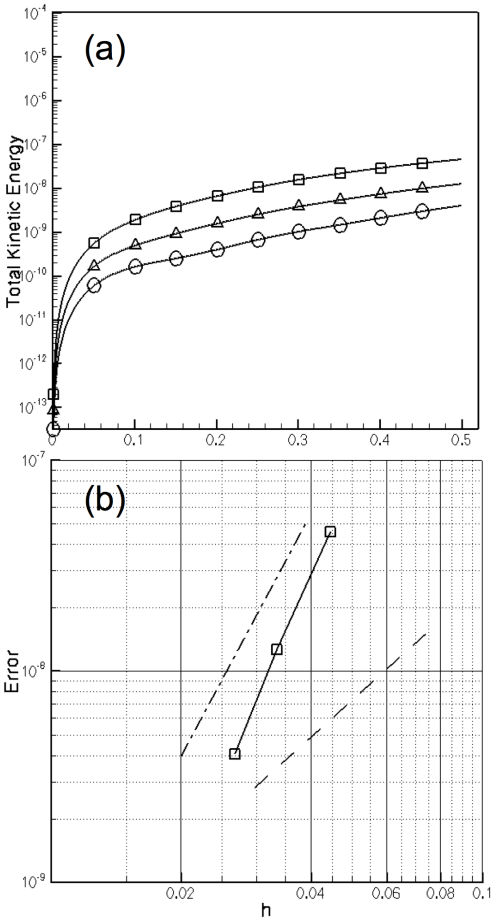

Figure 8: L_1-error in total kinetic energy for the stationary bubble case on a fixed background grid of $[40\times40]$: (a) time evolution of error for particle resolutions of $h = 0.044$ (squares),$h = 0.033$ (triangles), and $h = 0.0266$ (circles), (b) the convergence of error at $t = 0.5$ with particle resolutions. Also shown are lines of second (dashed) and third (dashed-dot) order convergence.

corresponding spurious currents are also shown in Figure 6b, indicating a very low magnitude. Figure 7a-b show the streamlines obtained at $t = 10^{-3}$ and $t = 0.5$, respectively. Figure 7a is for particle resolution of $R/h = 45$ whereas figure 7b is for $R/h = 60$. Both figures show a symmetric pattern of vortical structures around the interface. Similar patterns have been reported in the literature [18, 39]. The interface remains a perfect circle after $t = 0.5$ with low magnitudes of spurious currents. Figure 8a shows the time evolution of the total kinetic energy in this calculation for the three particle resolutions. The corresponding convergence of the L_1-error at $t = 0.5$ is shown in Fig. 8b, indicating larger than second order convergence. However, the oscillatory

behavior of the kinetic energy as shown by Francois *et al.* [39] and Herrmann [23] is not observed in our calculations. This is probably because, here the particle distribution, although uniform, is not symmetric with respect to the background mesh. Also the particle distribution changes with time as they are moved by the spurious currents. Remeshing and reinitialization are *suppressed* in the above calculations, yet the overall spurious current magnitudes are similar to those reported by Herrmann [23].

Rising Bubble

Lastly, we consider the rise of an air bubble column in water by buoyancy forces. The characteristic non-dimensional numbers for this situation are the Bond number, or the ratio of gravitational to surface tension forces, and the Reynolds number:

$$Bo = \frac{\rho_\ell g D^2}{\sigma}; \quad Re^* = \frac{\rho_\ell g^{1/2} D^{3/2}}{\mu_\ell}, \qquad (26)$$

where ρ_ℓ is the density of liquid, μ_ℓ dynamic viscosity of the liquid, σ the surface tension coefficient, and D is the initial diameter of the bubble column. For the present simulations, we use the small-bubble problem of Kang *et al.* [42]. Accordingly, a bubble column of non-dimensional diameter $D/L_{ref} = 2/3$ is placed in a domain of non-dimensional lengths $[-1, 1] \times [0, 3]$. Here the reference length scale (L_{ref}) is taken as 0.01 m. No-slip condition is used on all sides of the domain. The bubble is initially placed at $(0, 1)$. The background mesh is uniform Cartesian with 60×90 grid points. Uniformly spaced sub-cell Lagrangian points are used with $1/4^{th}$ resolution of the background mesh. 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. The Bond and Reynolds numbers are 5.736 and 1500, respectively.

Figure 9 shows the time evolution of the initially circular bubble. The area loss at $t = 0.05$ s is less than 2% of the initial area for the resolution used. The results qualitative agree with those of Kang *et al.* [42]. A systematic grid-refinement study needs to be performed to estimate the convergence of the hLE scheme applied to the rising bubble problem.

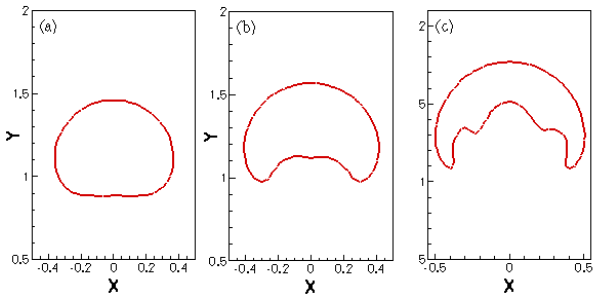

Figure 9: A cylindrical air-bubble rising against gravity in water: (a) t=0.02, (b) t=0.035, (c) t=0.049 s

Summary

A new hybrid Lagrangian-Eulerian (hLE) scheme, combining a particle-based, mesh-free 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 [11] 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 [23, 39] 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) and coupled simulations for stationary drop and rising buoyant bubble with encouraging results. The efficiency of the scheme can be considerably enhanced by using dynamic-load balancing of the Lagrangian particles and advanced methods for finding the nearest neighbors within the mollification kernels [7].

Acknowledgments

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

SVA is grateful to Dr. Marcus Herrmann for several useful discussions on level set methods, curvature calculations and his RLSG approach. Past interactions with Dr. Frank Ham and Prof. Krishnan Mahesh on co-located mesh, variable density algorithms are also acknowledged.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

[23] M. Herrmann. *Center for Turbulence Research*, 2006.

[24] G. Ryskin and LG Leal. *J. Fluid Mech. Digital Archive*, 148:19–35, 2006.

[25] E.S. Oran and J.P. Boris. *Numerical simulation of reactive flow*. Elsevier New York, 1987.

[26] AJC Ladd and R. Verberg. *J. Statistical Phys.*, 104(5):1191–1251, 2001.

[27] A.T.E.N. CATE, J.O.S.J. DERKSEN, L.M. PORTELA, and H.E.A. VAN DEN AKKER. *J. Fluid Mech.*, 519:233–271, 2004.

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

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

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

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

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

[33] S. Osher and J.A. Sethian. *J. Comput. Phys.*, 79(1):12–49, 1988.

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

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

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

[37] L. Verlet. *Phys. Rev*, 159(1):98–103, 1967.

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

[39] 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.

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

[41] J.B. Bell, P. Colella, and H.M. Glaz. *J. Comput. Phys.*, 85(2):257–283, 1989.

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