Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Computational Physics 213 (2006) 141–173
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

A balanced-force algorithm for continuous and sharp interfacial surface tension models within a volume tracking framework

Marianne M. Francois [a,*], Sharen J. Cummins [a], Edward D. Dendy [a], Douglas B. Kothe [a], James M. Sicilian [a], Matthew W. Williams [b]

[a] *Computer and Computational Sciences Division, Continuum Dynamics Group, Los Alamos National Laboratory, CCS-2 Continuum Dynamics Group, MS B296, Los Alamos, NM 87545, USA*
[b] *Applied Physics Division, Diagnostic Applications Group, Los Alamos National Laboratory, Los Alamos, NM 87545, USA*

Received 10 March 2005; received in revised form 7 June 2005; accepted 5 August 2005
Available online 19 September 2005

Abstract

A new balanced-force algorithm is presented for modeling interfacial flow with surface tension. The algorithm is characterized by a pressure-correction method with the interfaces represented by volume fractions. Within this flow algorithm, we devise a continuous (e.g., continuum surface tension model) and a sharp (e.g., a ghost fluid method) interface representation of the surface-tension-induced interfacial pressure jump condition. The sharp interface representation is achieved by temporarily reconstructing distance functions from volume fractions. We demonstrate that a flow algorithm designed to legislate force balance retains an exact balance between surface tension forces and the resulting pressure gradients. This balance holds for both continuous and sharp representations of interfacial surface tension. The algorithm design eliminates one of the elusive impediments to more accurate models of surface tension-driven flow, the remaining of which is accurate curvature estimation. To validate our formulation, we present results for an equilibrium (static) drop in two and three dimensions having an arbitrary density jump across the interface. We find that the sharp surface tension method yields an abrupt pressure jump across the interface, whereas the continuous surface tension method results in a smoother transition. Both methods, however, yield spurious velocities of the same order, the origin of which is due solely to errors in curvature. Dynamic results are also presented to illustrate the versatility of the method.

Keywords: Surface tension; Multiphase flow; Incompressible flow; Volume-of-fluid; Continuum surface force; Curvatures; Ghost fluid method; Drop dynamics

* Corresponding author. Tel.: +1 505 667 4237; fax: +1 505 665 4972.
E-mail address: mmfran@lanl.gov (M.M. Francois).

0021-9991/$ - see front matter © 2005 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2005.08.004

1. Introduction

Interfacial flows with surface tension are encountered frequently in industrial and engineering applications, a prototypical example being material processing. Accurate modeling of such flows is challenging because of the discontinuity in material properties across the interface and because of interfacial boundary conditions due to surface tension forces.

The continuum surface force (CSF) method of Brackbill et al. [1] has been employed extensively over the last 13 years to model surface tension in various fixed (Eulerian) mesh formulations for interfacial flows, in particular in the volume-of-fluid (VOF) [13,19,16], level-set (LS) [12,24] and front tracking (FT) [28,18] interface representation techniques. Surface tension forces acting on the interface are transformed to volume forces in regions near the interface via delta functions, leading to ideally discontinuous interfacial jump conditions being modeled as smooth.

Issues remain with CSF, however, despite its wide use. The CSF method has the propensity to generate unphysical flow (''spurious currents'') near the interface when surface tension forces are dominant. These spurious currents are best illustrated in the limiting case of an inviscid static drop in equilibrium without gravity where Laplace's formula applies. The major reason for the spurious currents is a numerical imbalance of the surface tension force and the associated pressure gradient. In the context of sharp interface representation techniques such as VOF and FT, several studies have proposed different ways to reduce these spurious currents by either improving curvature estimation [17,22,29], improving the flow algorithm [18,23,26], or by combining better algorithms with interface curvatures estimation [16,19]. Jamet et al. [11] have been able to eliminate these spurious currents from a minimal energy consideration, but only in the context of a diffuse interface representation technique.

Recently, ghost fluid methods (GFMs) have been proposed in [15] to impose sharper boundary conditions on embedded boundaries. Since GFM require knowledge of the distance from the interface, and since this information is naturally carried in LS methods, GFM have been applied successfully to model interfacial flow with surface tension [12] in conjunction with a LS technique. However, results in [12] also show a persistence of spurious currents as well as a loss of mass conservation.

In this paper, working within a volume tracking framework, we introduce a new balanced-force algorithm characterized by a pressure-correction method that leads to an exact balance of the pressure gradient with the surface tension force and therefore a commensurate reduction of the spurious currents. By temporarily reconstructing distance functions from volume fractions using the technique described in [6], we provide an alternative method to model surface tension forces within a mass-conservative volume tracking framework. Armed with an ability, then, to model surface tension forces with either a CSF model or a GFM within the same balanced-force algorithm, we then assess the relative strengths and weaknesses of each approach on appropriate (and difficult) test problems. Here we investigate the accuracy of the different methods and address whether or not there are advantages in using a sharp approach over the continuous one.

The structure of the paper is as follows. In Section 2, we present the governing equations and we describe the balanced-force flow algorithm in detail. In Section 3, we present the continuous and sharp surface tension models with two improved interfacial curvature models. In Section 4, we present the results for the case of a drop in static equilibrium and for the dynamic cases of an oscillating drop and a rising bubble by buoyancy.

2. Balanced-force formulation

2.1. Governing equations

In the VOF method, a single set of mass and momentum conservation equations is solved on a fixed grid and the volume fraction f of the fluids (ratio of fluid to total volume) is evolved with an advection equation.

In this work we assume the flow to be incompressible. The governing equations are the equations for the advection of the volume fraction f:

$$\frac{\partial f}{\partial t} + \mathbf{u} \cdot \nabla f = 0, \tag{1}$$

and mass and momentum conservation:

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

$$\frac{\partial(\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u}\mathbf{u}) = -\nabla P + \nabla \cdot (\mu(\nabla \mathbf{u} + \nabla^{\mathrm{T}} \mathbf{u})) + \mathbf{F}, \tag{3}$$

where $\mathbf{u}$ is the velocity field, P the total pressure, $\mathbf{F}$ any body force (such as the gravitational acceleration or surface tension), and ρ and μ are the fluid density and viscosity, respectively, defined as

$$\rho = \rho_1 f + \rho_2 (1 - f), \tag{4}$$

$$\mu = \mu_1 f + \mu_2 (1 - f). \tag{5}$$

Here the subscripts 1 and 2 denote fluid 1 and fluid 2, respectively. The expression for density in Eq. (4) results from mass conservation, whereas the expression for mixture viscosity in Eq. (5) is an approximation (others can be used, e.g., see [5]). In computational cells occupied with fluid 1, f is unity, and in cells occupied with fluid 2, f is zero. For cells containing the interface bounding fluid 1 and 2, f lies between zero and unity. The volume fractions of fluid 1 and 2 sum to unity everywhere.

2.2. Proper implementation of surface tension force within the flow algorithm

In this section, we present our balanced-force algorithm for incorporating the surface tension force, denoted as $\mathbf{F}$, within a pressure-correction flow algorithm [3]. This algorithm is developed in order to ensure that pressure gradient force is balanced exactly by the surface tension force (for static cases). The pressure-correction algorithm is characterized by collocated variables on the computational mesh, in that primary variables $\mathbf{u}$ and P reside at cell-centers, and to ensure discrete solenoidality a face-centered velocity is computed and used for advection.

To initialize the volume fractions f, we employ a recursive local mesh refinement technique. Here, we choose four levels of subdivisions and approximate the interface by linear segments at the finest level as in [6]. This recursive technique is simple and converges to second-order as shown in [6]. The global error is $O(10^{-6})$ for a circular drop on a uniform mesh of resolution $R/h = 10$, where R is the drop radius and h the mesh spacing.

The VOF advection equation, Eq. (1), is solved with a PLIC (piecewise linear interface calculation) algorithm [8,13,21] which consists of two steps. First, planar interfaces are reconstructed and then geometry-based fluid volumes fluxed across cell faces are computed:

$$\frac{f^{n+1} - f^n}{\Delta t} = -\mathbf{u}^n \cdot \nabla f^n. \tag{6}$$

Once the volume fractions are advected, fluid properties such as density are updated using Eq. (4) and f^{n+1}. The momentum equation is discretized first-order in time and is split into a "predictor" and a "corrector" step:

$$\frac{\rho^{n+1}\mathbf{u}^* - \rho^n \mathbf{u}^n}{\Delta t} = -\nabla \cdot (\rho \mathbf{u}\mathbf{u})^n - \nabla P^n + \mathbf{F}^n, \tag{7}$$

$$\frac{\rho^{n+1}\mathbf{u}^{n+1} - \rho^{n+1}\mathbf{u}^*}{\Delta t} = -\nabla \delta P^{n+1} + \mathbf{F}^{n+1} - \mathbf{F}^n, \tag{8}$$

where the superscripts n and $n+1$ represent the current and next time level, respectively, and the superscript * represent an intermediate level. The change or increment in pressure is $\delta P^{n+1} = P^{n+1} - P^n$.

The predictor step, Eq. (7), is first solved for an intermediate velocity at cell-centers:

$$\rho_c^{n+1}\mathbf{u}_c^* = \rho_c^n\mathbf{u}_c^n - \Delta t \sum_f \mathbf{u}_c^n \left(\sum_k \rho_k f_k \right) (\mathbf{u}_f \cdot \mathbf{n}_f) A_f - \Delta t \rho_c^{n+1} \left\langle \frac{\nabla P}{\rho_f} - \frac{\mathbf{F}_f}{\rho_f} \right\rangle_{f\to c}^n, \tag{9}$$

where $\mathbf{n}_f$ is the normal vector to the face, A_f is the face area, the subscript k is the material index, the subscripts c and f denotes cell-centered and face-centered quantities, respectively (illustrated in Fig. 1(a)). The operator $\langle\ \rangle_{f\to c}$ denotes an appropriate averaging of a face-centered quantity to a cell-centered quantity. For example, for the x component of the cell centered vector, $\vec{\psi}$, the operator is defined as:

$$\hat{x} \cdot \langle \vec{\psi} \rangle_{f\to c} = \frac{\sum_f^{n\,\text{faces}} (\vec{\psi}_f \cdot \hat{x}) |n_{x,f} A_f|}{\sum_f^{n\,\text{faces}} |n_{x,f} A_f|}. \tag{10}$$

Momentum advection, the second term on the RHS of Eq. (9), is performed in the same way as VOF-based mass advection in interface cells [2] namely by assigning momentum crossing cell faces as the product of VOF-based mass and the donor cell fluid velocity. The viscous term on the RHS of Eq. (9) has been omitted for simplicity, although it is implemented fully implicitly, semi-implicitly or fully explicitly in the numerical code used in this paper [25]. In order to design an exact balance between pressure and surface

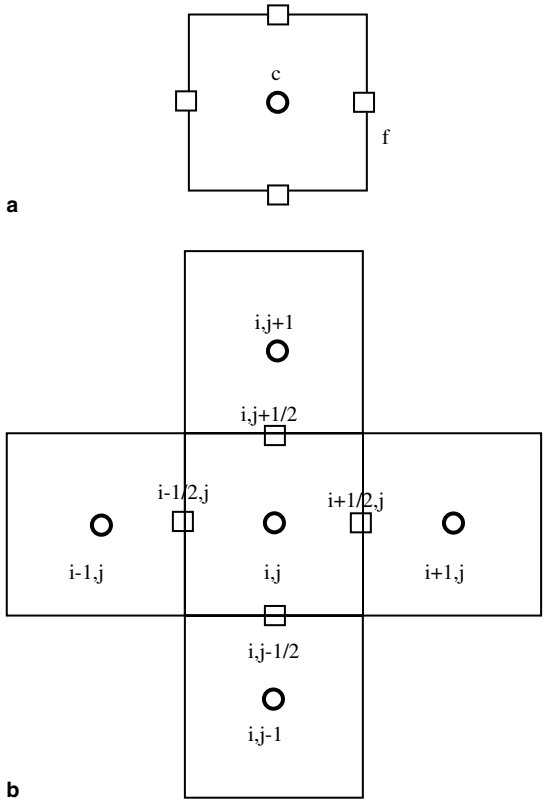


Fig. 1. Cell index notation referred to in the text, with (a) giving c and f index locations and (b) giving i, j index locations.

tension forces, the gradient operators in these two terms must be considered together as well as discretized identically and at the same location. As a result, in our formulation, since the gradient in pressure is estimated at cell faces, we must also estimate the surface tension force term at cell faces for consistency. This is a key point in our balanced-force algorithm, which has also been emphasized recently in other flow algorithms of [7,16,19].

Once the intermediate velocity $\mathbf{u}_c^*$ is estimated with Eq. (9), the intermediate face-centered velocity $\mathbf{u}_f^*$ is evaluated. In doing this, we take care to ensure that the vector quantity interpolated from cell to face centers (and vice versa) is not the velocity field, but rather the total vector field (prior to application of the projection correction) [4]. Proper projection correction occurs via application of pressure and surface tension forces at the required (face-centered) location, not by interpolating the forces from another location (cell centers) to the location of interest (face centers). This procedure can be also viewed as a modification of the Rhie–Chow [20] procedure to ensure pressure–velocity coupling. First, the intermediate velocity at cell-centers is negated by adding back the cell-centered approximation to the difference between the surface tension force and the pressure gradient, i.e., the total vector field is reacquired at cell faces. The total cell-centered vector field is then interpolated by simple averaging to the cell face. Finally, the difference between the surface tension force and pressure gradient is added to the cell face (the projection correction is reapplied at the location of interest), giving:

$$\mathbf{u}_f^* = \left\langle \mathbf{u}_c^* + \Delta t \left\langle \frac{(\nabla P)_f}{\rho_f} - \frac{\mathbf{F}_f}{\rho_f} \right\rangle_{f\to c}^{n} \right\rangle_{c\to f} - \frac{\Delta t}{\rho_f^{n+1}} [(\nabla P)_f^n - \mathbf{F}_f^{n+1}], \tag{11}$$

where the operator $\langle\ \rangle_{c\to f}$ is an interpolation of a cell-centered to a face-centered quantity. For example on a two-dimensional uniform Cartesian mesh, the interpolation of the cell-centered quantity ψ on the right face of cell (i,j) is given by a simple average:

$$\langle \psi \rangle_{c\to f} = \psi_{i+1/2,j} = \frac{1}{2}(\psi_{i,j} + \psi_{i+1,j}). \tag{12}$$

A discrete solenoidal condition is now imposed upon the face-centered velocity, yielding the following Poisson equation for the correction in pressure:

$$\nabla \cdot \left(\frac{(\nabla \delta P_c^{n+1})_f}{\rho_f^{n+1}} \right) = -\nabla \cdot \left(\frac{\mathbf{u}_f^*}{\Delta t} \right). \tag{13}$$

This pressure correction (Eq. (13)) is solved for the change in cell-centered pressure δP_c using a two-level preconditioned GMRES solution technique [25]. Given the new cell-centered pressure field, the velocity field is then corrected both at cell face and cell-center, respectively:

$$\mathbf{u}_f^{n+1} = \mathbf{u}_f^* - \Delta t \left(\frac{(\nabla \delta P_c^{n+1})_f}{\rho_f^{n+1}} \right), \tag{14}$$

$$\mathbf{u}_c^{n+1} = \mathbf{u}_c^* + \Delta t \left\langle \frac{(\nabla P)_f - \mathbf{F}_f}{\rho_f} \right\rangle_{f\to c}^{n} - \Delta t \left\langle \frac{(\nabla P)_f - \mathbf{F}_f}{\rho_f} \right\rangle_{f\to c}^{n+1}. \tag{15}$$

The density at faces (ρ_f) is taken to be a simple geometric average of the cell-center densities on either side of the face under consideration, which is consistent with continuity of acceleration across the cell face. For a uniform two-dimensional Cartesian mesh the face density between cell i and $i+1$ then becomes

$$\rho_f = \rho_{i+1/2,j} = \frac{1}{2}(\rho_{i,j} + \rho_{i+1,j}), \tag{16}$$

where the cell index notation (f,i,j) is illustrated in Fig. 1.

The complete balanced-force flow algorithm from time step n to time step $n+1$ can now be summarized as follows:

1. advect the time n volume fractions f to obtain the volume fraction at time $n+1$, f^{n+1} (Eq. (1));
2. solve the predictor step for $\mathbf{u}_c^*$ (Eq. (9));
3. transfer the cell-center velocity $\mathbf{u}_c^*$ to face center velocity $\mathbf{u}_f^*$ (Eq. (11));
4. solve the pressure equation for δP_c^{n+1} (Eq. (13)), then obtain P_c^{n+1}; and
5. correct the face-centered velocity field $\mathbf{u}_f^{n+1}$ (Eq. (14)) and cell-centered velocity field $\mathbf{u}_c^{n+1}$ (Eq. (15)).

This balanced-flow algorithm is implemented in the *Truchas* software [25] as part of the Telluride project at Los Alamos National Laboratory. The *Truchas* software is the framework for all algorithm implementation and numerical results in the present paper.

In the results section, we compare the above face-centered formulation with a cell-centered formulation in which the surface tension term only appears in the predictor step:

$$\rho^{n+1}\mathbf{u}_c^* = \rho^n\mathbf{u}_c^n - \Delta t\sum_f \mathbf{u}_c^n\left(\sum_k \rho_k f_k\right)(\mathbf{u}_f\cdot\mathbf{n}_f)A_f - \Delta t\rho_c^{n+1}\left\langle\frac{\nabla P}{\rho_f}\right\rangle_{f\to c}^{n} - \Delta t\mathbf{F}_c^n, \tag{17}$$

which is the standard implementation for most previously published body force models for surface tension, i.e., through application of an explicit force at the center of the momentum control volume (which is cell-centered). As will be shown in the following, however, this implementation is markedly inferior to the face-to-cell centered formulation (balanced-flow algorithm given in Eq. (9)) in which a cell-centered surface tension force is derived from a consistent averaging of face-centered forces. It is demonstrated that this face-centered formulation gives an exact balance between the surface tension force and the pressure gradient for the case of a static drop when the curvature is imposed to its exact value. Here an "exact balance" for this equilibrium problem is deduced from the resulting velocity field, which is of the order of round-off. Therefore, the remaining task is an accurate evaluation of the interfacial curvature. In the next section, we present the continuous and sharp approaches to model the surface tension force. We also detail techniques that are employed to compute curvatures.

3. Surface tension model

3.1. Continuous approach

In the following, $\mathbf{F}^{CSF}$, denotes the continuum surface force (CSF) defined as [1]:

$$\mathbf{F}^{CSF} = \sigma\kappa\hat{\mathbf{n}}\delta, \tag{18}$$

where σ is the surface tension coefficient (assumed constant for this work), κ is the interfacial curvature, $\hat{\mathbf{n}}$ is the interface unit normal, δ is the Dirac Delta function. Here, we reformulate the CSF by simply replacing the product of the delta function and the unit normal with the gradient of the volume fraction. For a consistent coupling of the surface tension force with pressure gradient forces within the flow algorithm, $\mathbf{F}^{CSF}$ is estimated at faces as

$$\mathbf{F}_f^{CSF} = \sigma\kappa_f(\nabla f)_f. \tag{19}$$

The surface tension force $\mathbf{F}^{CSF}$ is non-zero only on faces where the face gradient of the volume fraction $(\nabla f)_f$ is non-zero. In discretized form on a two-dimensional Cartesian mesh, for example at the left face of cell (i,j), the x- and y-components of the surface tension force are approximated as

$$Fx^{\mathrm{CSF}}_{i-1/2,j} = \sigma\kappa_{i-1/2,j}\frac{f_{i,j} - f_{i-1,j}}{\Delta x} \quad \text{and} \quad Fy^{\mathrm{CSF}}_{i-1/2,j} = 0, \tag{20}$$

where the volume fractions are assumed to vary negligibly along (tangential to) faces ($\partial f/\partial y$ in this case). If this "stair-stepped" approximation of the gradient of the volume fractions were to be relaxed (which is easily done), the profile of the gradient will become more diffuse, losing as a result some interface locality desirable for the balance of the surface tension forces with the pressure gradient. The curvature κ in Eq. (18) is first computed at cell-centers and then interpolated at faces, with the interpolation techniques described in Section 3.4.

3.2. Sharp approach

In the GFM [15], an interfacial jump condition is applied as rigorously on the interface as the distance function ϕ allows. The distance function is zero on the interface, negative inside the interface (fluid 1) and positive outside the interface (fluid 2). To illustrate the GFM, we consider as an example the discretization of the equation $\nabla^2 P = 0$ in one-dimension with uniform points (separated by equal Δx spacing), where across the interface a jump condition in pressure ($[P] = A$) exists. In this example the interface is located between P_i and P_{i+1} where ϕ changes signs (between points i and $i+1$) as illustrated in Fig. 2. The discretized equations using central differences become:

$$\frac{P_{i+1} - 2P_i + P_{i-1}}{\Delta x^2} = \frac{A}{\Delta x^2}, \tag{21}$$

$$\frac{P_{i+2} - 2P_{i+1} + P_i}{\Delta x^2} = -\frac{A}{\Delta x^2}, \tag{22}$$

where the jump value A appears in the right-hand side of the equations.

Since in the present work surface tension is modeled within a VOF framework (where interfaces are represented by volume fractions), a temporary distance function must be "reconstructed" at each integration time step. Here the reconstructed distance function (RDF) technique proposed in [6] is used. First, distances in a direction normal from the piecewise linear interface segments (available from the PLIC VOF interface reconstructions) to all nearby (nearest neighbor) cells are computed using simple geometrical relations. The

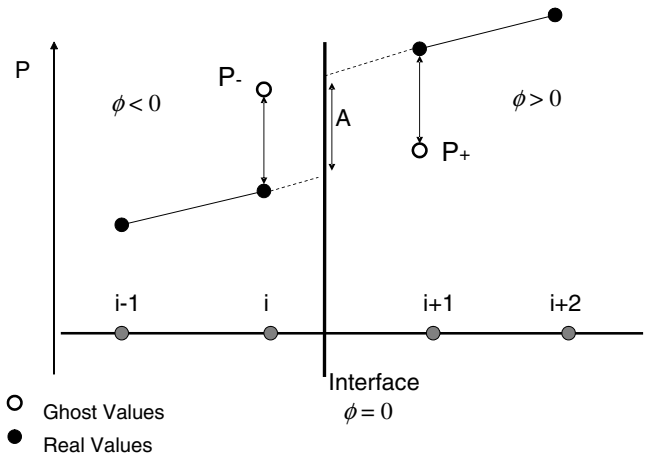


Fig. 2. Illustration of the GFM's ability to impose a sharp pressure jump condition due to the surface tension force via a specialized discretization of the pressure P across the interface.

piecewise linear segments are constructed based on volume fractions [21] and are part of the VOF technique. The normal distances from all nearby interface segments to the reference cell are accumulated with appropriate weights to obtain the distance function. The weights are function of the angle made by the interfacial normal and the vector defined by connecting the reference cell centroid to the centroid of the interfacial linear segment. The smaller the angle for a particular interface segment, the greater the contribution (weight) of the normal distance to that segment. The reconstructed distance function is second-order accurate as shown in [6].

We denote the present sharp approach by SSF (sharp surface tension force). The surface tension force term is non-zero only at faces across which the distance function ϕ changes sign. For example, if $\phi_{i,j} \leqslant 0$ and $\phi_{i-1,j} > 0$ across the $(i-1/2,j)$ face of cell (i,j), the surface tension term (on a uniform Cartesian mesh) becomes

$$Fx^{\mathrm{SSF}}_{i-1/2,j} = \frac{\sigma\kappa_{\mathrm{I}}}{\Delta x} \quad \text{and} \quad Fy^{\mathrm{SSF}}_{i-1/2,j} = 0. \tag{23}$$

If, on the other hand, $\phi_{i,j} > 0$ and $\phi_{i-1,j} \leqslant 0$, the surface tension term becomes

$$Fx^{\mathrm{SSF}}_{i-1/2,j} = -\frac{\sigma\kappa_{\mathrm{I}}}{\Delta x} \quad \text{and} \quad Fy^{\mathrm{SSF}}_{i-1/2,j} = 0, \tag{24}$$

where the subscript I denotes on the interface and Δx is the grid spacing across the face (distance between cell centroids). The interpolation of local cell curvature values to the interface is described in Section 3.4. Note that in the sharp approach, logical tests are necessary to identify the non-zero surface tension terms, which are not required in the continuous approach.

3.3. Comparison of continuous and sharp formulations

To illustrate the difference in formulation between the continuous and sharp approaches, consider a one-dimensional example, illustrated in Fig. 3. The interface location within the grid is given in the first row, with the corresponding volume fractions f and distance functions ϕ given in the second and forth row, respectively. The grid spacing Δx and $\sigma\kappa$ are assumed constant and equal to unity. Assuming the initial pressure is zero for this case, the surface tension force terms for the first time step iteration enter the right-hand side (RHS) of the pressure-correction equation (13) through $\mathbf{u}^*_{\mathrm{f}}$. The RHS for the continuous approach and for the sharp approach are given in the third and fifth row, respectively, of Fig. 3. For both formulations the non-zero terms sum to zero, with the difference between the two formulations being solely in the distribution of the terms. As expected, the continuous approach has a smooth RHS transition, whereas the sharp approach has a more abrupt transition with larger extrema.

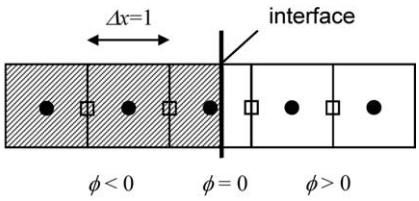
3.4. Curvature models

The interfacial curvatures are estimated using the time-advanced advected volume fractions (f^{n+1}). We consider here two different curvature approximations: a convolution technique and a height function method, described next.

3.4.1. Convolution technique

For the convolution technique, the work of Williams [29,30] is followed. The interfacial normal vector at cell centroids is obtained by convolving the volume fractions with the first derivatives of the kernel K_6:

$$\hat{\hat{\mathbf{n}}}_{\mathrm{c}} = (K_{6_x} * f, K_{6_y} * f, K_{6_z} * f), \tag{25}$$



f:	1	1	0.6	0	0
CSF:	0	-0.4	-0.2	0.6	0
ϕ:	-2.1	-1.1	-0.1	0.9	1.9
SSF:	0	0	-1	1	0

Fig. 3. One-dimensional example of the right-hand side of the pressure equation through a discrete interface at the first time step for the continuous (CSF) and sharp (SSF) interfacial surface tension models. Shown in the boxes are discrete values of the volume fraction (f) and resulting pressure for the CSF model and distance function (ϕ) and resulting pressure for the SSF model.

where $*$ denotes the convolution operation, defined as

$$K_{6x} * f = \int K_{6x}(\mathbf{x}' - \mathbf{x}) f(\mathbf{x}')\, \mathrm{d}\mathbf{x}', \tag{26}$$

with K_6 defined as:

$$K_6(r,d) = \begin{cases} A(d^2 - r^2)^3 & \text{if } \frac{r}{d} < 1, \\ 0 & \text{otherwise} \end{cases} \tag{27}$$

and its derivative with respect to x given by

$$K_{6x} = \frac{\partial K_6}{\partial x} = -6Ax(d^2 - r^2)^2, \tag{28}$$

where $r^2 = x^2 + y^2 + z^2$. Similar expressions follow for K_{6y} and K_{6z}. The constant A is chosen to normalize the kernel. The sixth-order kernel K_6 is chosen here over the eight-order kernel K_8 used in [30] because K_6 is not as highly peaked as K_8, therefore does not require as large a smoothing length (hence as much resolution) as K_8.

Given the cell-centered convolved unit normal estimated according to Eq. (21), the cell-centered curvature κ is evaluated directly with a discrete divergence operator,

$$\kappa_c = -\nabla \cdot \hat{\hat{\mathbf{n}}}_c. \tag{29}$$

Face-centered curvatures (required in our method for application of face-centered CSF forces) are estimated by interpolating the cell-centered curvatures computed in Eq. (29) with a simple arithmetic average. For example, on a two-dimensional Cartesian grid at face $(i + 1/2, j)$ of cell (i,j), the face curvature is estimated by

$$\kappa_{i+1/2,j} = \frac{1}{2}(\kappa_{i,j} + \kappa_{i+1,j}), \tag{30}$$

but only if volume fraction gradient at face $(i + 1/2, j)$ is non-zero (i.e., $\nabla f_f \neq 0$). Another method for estimating curvatures at faces is to directly evaluate its value at faces (without interpolation); this approach will be considered in future work.

For the sharp surface tension model, curvature on the interface is needed, hence it must be interpolated from cells in the proximity of the interface using distance function information. For example if the distance function changes sign between cell (i,j) and $(i + 1,j)$, the curvature at the interface is estimated by

$$\kappa_1 = \frac{\kappa_{i,j}|\phi_{i+1,j}| + \kappa_{i+1,j}|\phi_i|}{|\phi_{i,j}| + |\phi_{i+1,j}|}. \tag{31}$$

3.4.2. Height function technique

The height function [6,10,16,22,27] technique utilizes a geometrical method for estimating curvatures. A height function is reconstructed locally as an approximation of the interface location, and it is based on volume integrals (discrete sums) of volume fractions. The integration direction for the height function is taken to be in the direction of the largest component of the interface normal vector. For the height function integration volume, a 7×3 stencil is employed in two dimensions and a $7 \times 3 \times 3$ stencil (7 cells along the direction of the height function, with 3 cells above and below the reference cell) is employed in three dimensions. Seven cells are chosen based on relative accuracy comparisons with smaller stencils as illustrated in two-dimensions in Fig. 4(a). It should be noted, however, that this large stencil can be problematic (too large) in regions of high curvature where the radius of curvature is of the order of (or smaller) than the mesh spacing. Here the stencil must be reduced in proportion to the radius of curvature.

As an example of how the height function technique is used to estimate curvature in three-dimensions, consider a case where the largest interface normal component is in the z-direction (i.e., $|n_z| > |n_x|, |n_y|$). One can then locally construct nine height functions in the z-direction by summing the volume fractions in the z-direction (as shown in Fig. 4(b)):

$$H_{i,j,k} = \sum_{k-3}^{k+3} f_{i,j,k}\Delta z_k \quad \text{for} \quad i = i' - 1, i', i' + 1 \text{ and } j = j' - 1, j', j' + 1, \tag{32}$$

where $H_{i,j,k}$ represents the nine height functions along the z-direction. The curvature is then estimated from these height functions as:

$$\kappa = \left(\frac{f_z}{|f_z|}\right) \cdot \frac{H_{xx} + H_{yy} + H_{xx}H_y^2 + H_{yy}H_x^2 - 2H_{xy}H_xH_y}{(1 + H_x^2 + H_y^2)^{3/2}}, \tag{33}$$

where the derivatives with respect to x and y are estimated using a standard second-order finite-difference scheme. For example, the derivatives of H with respect to x (and similarly for y) on a uniform spaced mesh of spacing Δx in the x-direction are

$$H_x = \frac{H_{i+1,j,k} - H_{i-1,j,k}}{2\Delta x}, \tag{34}$$

$$H_{xx} = \frac{H_{i+1,j,k} - 2H_{i,j,k} + H_{i-1,j,k}}{\Delta x^2}, \tag{35}$$

and the cross derivative is:

$$H_{xy} = \frac{H_{i+1,j+1,k} - H_{i+1,j-1,k} - H_{i-1,j+1,k} + H_{i-1,j-1,k}}{2\Delta x \cdot 2\Delta y}. \tag{36}$$

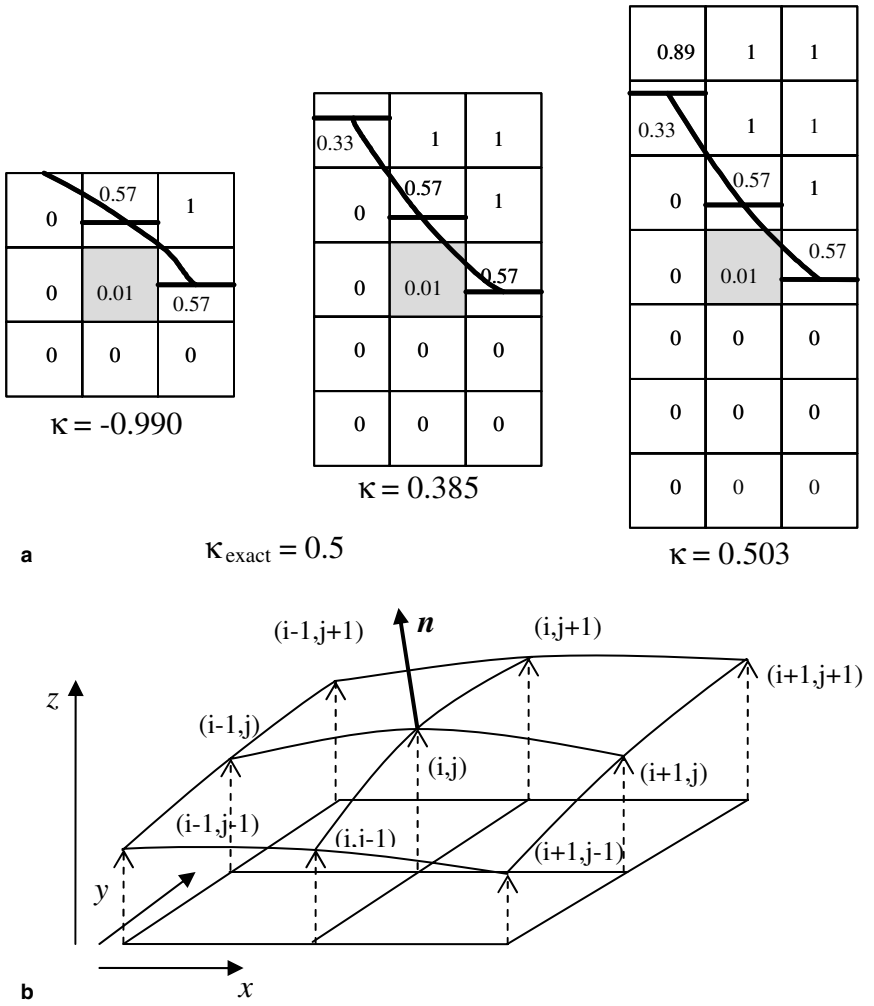


Fig. 4. (a) Illustration of the effect of stencil choice for the height function method in 2D estimates of curvature and (b) the nine height functions $z = z(x, y)$ for estimating curvature in 3D.

In two-dimensions curvatures are estimated for all interfacial cells, and in three-dimensions this estimate is obtained only in those cells actually containing the height function. Since curvature estimated from height functions is only defined at the centroid position within cells as specified by the "height" given by the height function value, one must interpolate the curvature to faces. The following interpolation is used:

$$\kappa_{i+1/2,j} = \begin{cases} \frac{1}{2}(\kappa_{i,j} + \kappa_{i+1,j}) & \text{if } H \text{ defined in cells } (i,j) \text{ and } (i+1,j), \\ \kappa_{i,j} & \text{if } H \text{ defined in cell } (i,j) \text{ and if } H \text{ not defined in cell } (i+1,j). \end{cases} \tag{37}$$

To estimate curvature on the interface, which is necessary in the sharp representation for surface tension, the face-valued curvature above is used. This is found to give a good approximation of the interfacial curvature κ_I, the key being to recognize in which cells H is defined and not defined.

Another way to interpolate curvatures to faces is to use a kernel K for the weighting coefficients:

$$\kappa_f = \frac{\sum \kappa_c K V_c}{\sum K V_c}, \tag{38}$$

where V_c is the cell volume. In the result section, we investigate the errors introduced by these various curvature models.

4. Numerical examples

We now look at applications of the above algorithm to three test cases: static drop in equilibrium (Section 4.1), oscillating drop (Section 4.2) and a rising bubble by buoyancy (Section 4.3).

4.1. Static drop in equilibrium

To validate our formulation, we consider the test case of a static drop in equilibrium without gravity. The exact jump in pressure across the drop in this case is given by:

$$\Delta P_{\text{exact}} = \sigma\kappa, \tag{39}$$

and the exact curvature is given by:

$$\kappa_{\text{exact}} = \begin{cases} 1/R & \text{in 2D,} \\ 2/R & \text{in 3D.} \end{cases} \tag{40}$$

The computational domain considered is a cube having side lengths of eight units. The drop is positioned at the center of the domain with a radius $R = 2$. The surface tension coefficient σ is taken to be 73, the density inside the drop is $\rho_1 = 1$ and the background fluid density ρ_2 is varied from 1 to 0. The exact pressure difference ΔP is 36.5 in two dimensions (2D) and 73 in three dimensions (3D). This case corresponds to the test case analyzed by Williams et al. [30]. Here, 2D and 3D computations are performed. Initially the velocity and pressure are zero. The CFL number limit in all these computations is 0.4. Velocity boundary conditions are free-slip, i.e., the normal component of velocity is zero. The computational mesh is uniform with $\Delta x = \Delta y = h$, with mesh resolution considered such that $R/h = 5$, 10, 20, and 40. The capillary time step constraint is taken as:

$$\Delta t \leqslant \left[\frac{\bar{\rho}(\Delta x)^3}{2\pi\sigma}\right]^{\frac{1}{2}} \tag{41}$$

with $\bar{\rho} = \frac{\rho_1+\rho_2}{2}$ as in [1].

In the computational results that follow, the error in the pressure jump and in the maximum velocity is investigated. The numerical jump in pressure is evaluated in three different ways:

(1) $\Delta P_{\text{total}} = P_{\text{in}} - P_{\text{out}}$ where the subscripts "in" denotes inside the drop (averaged for cells with $r \leqslant R$) and "out" outside the drop (averaged for cells with $r > R$);
(2) $\Delta P_{\text{partial}} = P_{\text{in}} - P_{\text{out}}$ where the subscripts "in" denotes inside the drop (averaged for cells with $r \leqslant R/2$) and "out" outside the drop (averaged for cells with $r \geqslant 3R/2$) to avoid considering the transition region; and
(3) $\Delta P_{\max} = P_{\max} - P_{\min}$ where the subscripts "max" denotes maximum and "min" minimum on the entire domain.

The relative pressure jump error is evaluated as:

$$E(\Delta P)_n = \frac{|\Delta P_n - \Delta P_{\text{exact}}|}{\Delta P_{\text{exact}}}, \tag{42}$$

where the subscript n denotes one of the three different evaluations (total, partial or max).

To measure the error in velocity, we employ the following L error norms:

$$L_1(\mathbf{u}) = \frac{\sum_{n=1}^{N} \|\mathbf{u}_n\|}{N}, \tag{43}$$

$$L_2(\mathbf{u}) = \frac{\sqrt{\sum_{n=1}^{N} \|\mathbf{u}_n\|^2}}{\sqrt{N}}, \tag{44}$$

$$L_\infty(\mathbf{u}) = |\mathbf{u}|_{\max} = \max(\|\mathbf{u}\|), \tag{45}$$

where $\| \ \|$ is the magnitude (norm) of the velocity vector $\mathbf{u}$.

4.1.1. Static drop: continuous and sharp surface tension models with exact curvature

First we test the coupling of the surface tension force with the pressure gradient in the new flow algorithm. For this test case, we impose the exact curvature given in Eq. (40) in the calculations. We consider a 2D drop on a mesh of resolution $R/h = 10$ for varying density ratios ranging from 1 to infinity, and impose a constant time step of 10^{-6} for all the cases, which is well within the limit of the capillary time step constraint defined in Eq. (41). Our convergence criteria for the pressure-correction equation (13) is 10^{-12}. In the previous works of Brackbill et al. [1] and Kothe et al. [13] a density scaling of the CSF is found to improve the CSF method's performance in high density ratio flows. Here we also investigate this issue. This density-scaled surface tension force is defined at cell face as:

$$\overline{\mathbf{F}}_{\text{f}}^{\text{CSF}} = \frac{\rho_{\text{f}}}{\bar{\rho}} \mathbf{F}_{\text{f}}^{\text{CSF}}. \tag{46}$$

We compare in Table 1 the results for the maximum velocity and pressure jump after one time step with the continuous CSF approach (described in Section 3.1), the density-scaled CSF given in Eq. (46), and the sharp SSF approach (described in Section 3.2). For the CSF method we compare both the cell-centered equation (17) and face-centered formulation equation (9). For the cell-centered formulation, three different implementations of the source term are considered: (1) original with the volume fraction gradient estimated at cell-centers, (2) density scaled and (3) original but applied only in interfacial cells (where $0 < f < 1$). For the original CSF and the case of density ratio of 1, the errors in pressure jump are comparable to the error with the face-centered CSF formulation, however, the maximum velocity is of order 10^{-5}, about 10 orders of magnitude greater than with the face-centered formulation, which clearly shows the improvements of the face-centered formulation over the cell-center formulation. For the other density ratios considered (10^3, 10^5 and infinity) we do not observe convergence of the pressure equation. Hence, we tested the cell-centered formulation with a density-scaled CSF and also by localizing the CSF only to interfacial cells instead to over a few grid points around the interface. With both these modified implementations, convergence is observed, but errors in the pressure jump are non-negligible due to resulting peaks in the solution. Later in Figs. 8 and 9 the cell-centered formulation only applied to interfacial cells is compared to the face-centered formulation when curvatures are computed.

Looking now to only the results of the face-centered formulation, from Table 1, we notice that the spurious currents are of the order of round off for the face-centered formulation, independent of the density ratio and the surface tension model. For the pressure jump, looking at its total error $E(\Delta P_{\text{total}})$, we observe $O(10^{-2})$ errors for the CSF and density-scaled CSF approach and $O(10^{-15})$–$O(10^{-20})$ errors with the SSF

Table 1
Error in velocity and pressure after one time step for the inviscid static drop in equilibrium when the exact curvature is specified

ρ_1/ρ_2	$\|\mathbf{u}\|_{\max}$	$E(\Delta P_{\text{total}})$	$E(\Delta P_{\text{partial}})$	$E(\Delta P_{\max})$
(a) CSF cell-centered				
1	2.28×10^{-5}	7.76×10^{-2}	1.03×10^{-14}	7.79×10^{-14}
10^3, 10^5, Infinity		Does not converge		
(b) Density-scaled CSF cell-centered				
1	2.28×10^{-5}	7.76×10^{-2}	1.03×10^{-14}	7.79×10^{-14}
10^3	1.33×10^{-4}	5.60×10^{-1}	5.78×10^{-1}	4.40×10^{-1}
10^5	1.33×10^{-4}	5.62×10^{-1}	5.81×10^{-1}	4.42×10^{-1}
Infinity	1.0×10^{-4}	6.30×10^{-1}	6.48×10^{-1}	4.72×10^{-1}
(c) CSF cell-centered (apply only where $0<f<1$)				
1	5.19×10^{-5}	4.99×10^{-1}	4.68×10^{-1}	3.97×10^{-1}
10^3	6.15×10^{-3}	1.06×10^{-1}	1.29×10^{-1}	1.82
10^5	6.91×10^{-3}	1.56×10^{-1}	1.77×10^{-1}	1.98
Infinity	4.71×10^{-3}	2.29×10^{-1}	2.52×10^{-1}	2.05
(d) CSF face-centered				
1	1.25×10^{-18}	2.89×10^{-2}	2.73×10^{-15}	8.72×10^{-14}
10^3	4.97×10^{-18}	2.89×10^{-2}	3.89×10^{-16}	3.89×10^{-16}
10^5	5.70×10^{-19}	2.89×10^{-2}	1.95×10^{-16}	7.79×10^{-16}
Infinity	3.05×10^{-19}	2.89×10^{-2}	4.28×10^{-15}	4.87×10^{-15}
(e) Density-scaled CSF face-centered				
1	1.25×10^{-18}	2.89×10^{-2}	2.73×10^{-15}	8.72×10^{-14}
10^3	1.67×10^{-17}	4.28×10^{-2}	1.95×10^{-15}	1.36×10^{-14}
10^5	1.69×10^{-17}	4.28×10^{-2}	7.79×10^{-16}	1.38×10^{-14}
Infinity	1.70×10^{-17}	4.28×10^{-2}	5.84×10^{-15}	1.34×10^{-14}
(f) SSF face-centered				
1	5.43×10^{-19}	1.36×10^{-15}	5.84×10^{-15}	1.63×10^{-14}
10^3	4.44×10^{-18}	1.95×10^{-16}	1.17×10^{-15}	3.11×10^{-15}
10^5	2.71×10^{-19}	3.89×10^{-16}	3.70×10^{-15}	4.87×10^{-15}
Infinity	1.14×10^{-19}	2.81×10^{-20}	0.0	7.79×10^{-16}

The fluid density inside the drop (ρ_1) is 1 and the fluid density outside (ρ_2) is allowed to vary. The drop of radius $R=2$ is centered in an 8×8 domain, the mesh is 40×40 ($R/h=10$), and the time step is constant and equal to 10^{-6}.

approach. To understand the difference in the pressure jump error between CSF and SSF, we plot the cell-centered pressure versus (x,y) for these two cases in Figs. 5(a) and (b), respectively, and the profiles along the x- and diagonal-directions in Fig. 6. From these profiles we clearly see that with the continuous approach there are intermediate pressure points in the interfacial transition region around the interface (Figs. 5(a) and 6) that are not present with the sharp approach (Figs. 5(b) and 6). Since all the grid points are considered in evaluating ΔP_{total}, the error measure $E(\Delta P_{\text{total}})$ reflects a transition region, whereas the error measure $E(\Delta P_{\text{partial}})$ does not include the transition region by definition.

With the density-scaled CSF method, the pressure jump error $E(\Delta P_{\text{total}})$ is larger than with the CSF method as shown in Table 1 for density ratios greater than 1, and similarly for $E(\Delta P_{\text{partial}})$ and $E(\Delta P_{\max})$. This suggests that the density-scaled CSF results in a different pressure distribution in the transition region. Here we note that the definition of the pressure jump is important in the observations, and that (ΔP_{total}) is the most sensitive quantity for the pressure distributions between the different techniques. From this test case, we can conclude that:

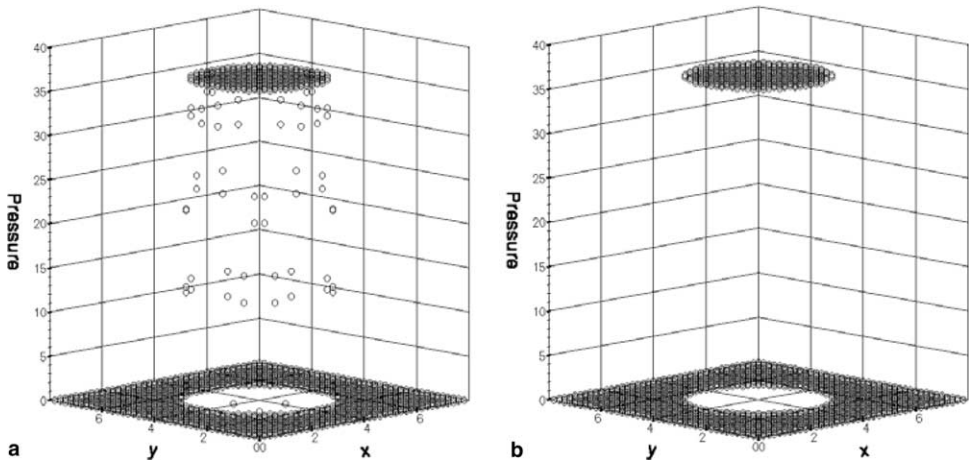


Fig. 5. Computed pressure obtained with (a) the CSF and (b) SSF models for the inviscid static drop case when the exact curvature is specified. The grid is uniform and of resolution $R/h = 10$. The drop of radius $R = 2$ is centered in an 8×8 domain. The surface tension coefficient is 73 and the density ratio is 10^3.

- spurious velocities are of the order of round-off error and the pressure jump is recovered independently of the density ratios and surface tension models; and
- the pressure distribution is sensitive to the surface tension model, but is independent of the density ratio across the interface.

4.1.2. Static drop: curvature errors

In Fig. 7, the L_∞ error norm for curvature, defined as $L_\infty = \max|\kappa_f - \kappa_{exact}|$, is shown for the different curvature models presented in Section 3 and for different methods for interpolating curvature to cell faces.

For the convolution techniques and using the simple interpolation (Eq. (30)), we consider different kernel smoothing lengths, d, that are functions of the mesh spacing h:$d = 3h$, and $d = 1.9h^{1/2}$ and we also consider a constant kernel smoothing length $d = 0.4$ independently of the mesh spacing chosen as $R/5$ based on the Williams study [29]. We observe:

- second-order convergence for the case of a constant smoothing length ($d = 0.4$), which corresponds to the $d = R/5$ as suggested in [29];
- first-order convergence when $d = 1.9h^{1/2}$, but with much lower error than with the constant case of $d = 0.4$; and
- *no* convergence when $d = 3h$.

We have also investigated different kernels and a 'direct' convolution method (where the volume fractions are convolved directly). All convolution techniques yield a similar error level and resulting behavior. Using the kernel interpolation in Eq. (38) to estimate face curvatures leads to slightly larger errors, which is not surprising since more surrounding points are taken into account as compared to the simple averaging interpolation.

The height function is second-order accurate. For a detailed study on curvatures, including a discussion of its limitation, we refer the reader to the study presented in [6]. When the curvatures are computed with the height function and when the kernel interpolation to estimate the curvatures at faces (Eq. (38)) is

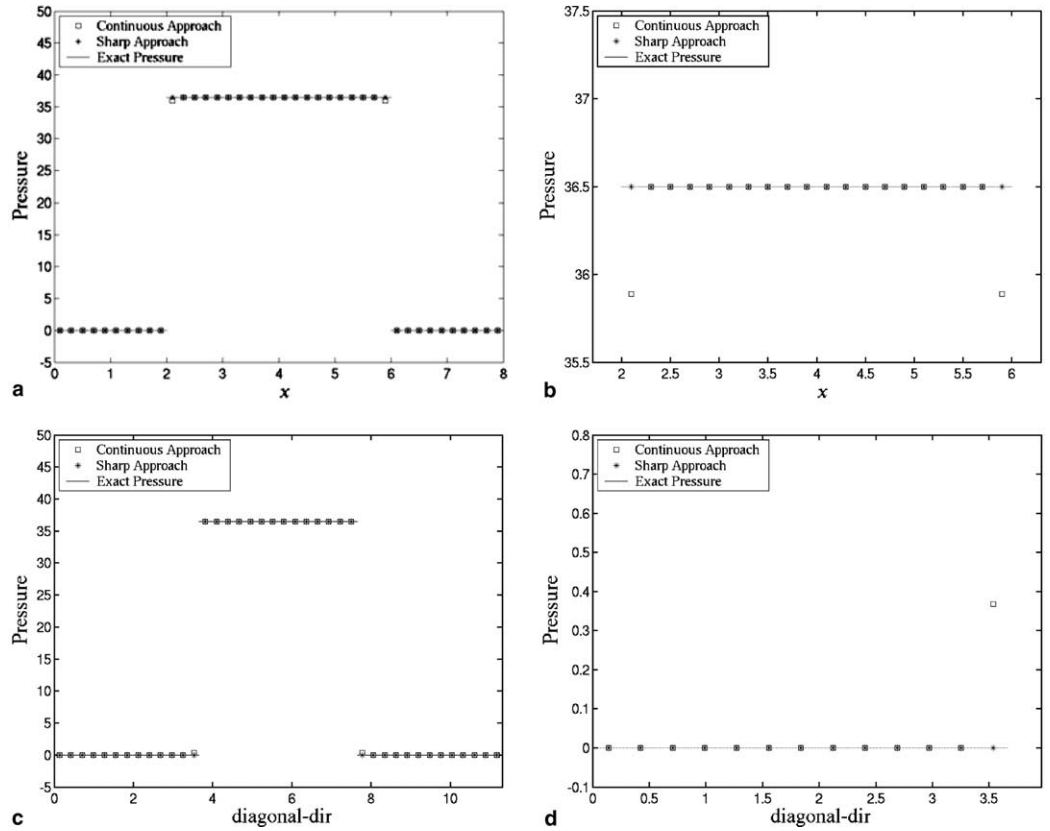


Fig. 6. Computed pressure along the x-direction at the mid-section $y = 4$ (a, b) and along the diagonal-direction (c, d) with the CSF and SSF models for the inviscid static drop when the exact curvature is specified. The grid is uniform and of resolution $R/h = 10$. The drop of radius $R = 2$ is centered in an 8×8 domain. The surface tension coefficient is 73 and the density ratio is 10^3. The time step is held constant at 10^{-6}.

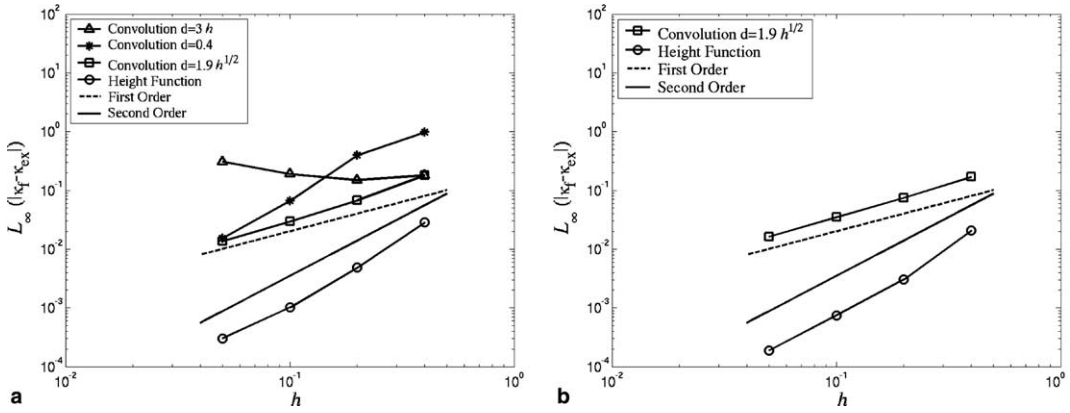


Fig. 7. L_∞ face curvature error norms for a circular interface using the convolution technique and the height function method when the curvatures at faces are interpolated using (a) a simple interpolation scheme (Eq. (30) for the convolution method and Eq. (37) for the height function method) and (b) the kernel interpolation technique of Eq. (38). The interface is a circle of radius $R = 2$ centered in an 8×8 domain. In the plot legends above, d is the smoothing length of the kernels and h is the grid spacing.

employed, slightly smaller errors result. This is primarily because the kernel interpolation helps to smooth the maximum errors relative to the simple interpolation (Eq. (37)).

In the following, we choose the simple averaging interpolation scheme for both the convolution and height function approach. This is computationally less expensive than the kernel interpolation scheme but still allows a meaningful comparison of the methods.

4.1.3. Static drop: continuous surface tension model

4.1.3.1. Comparison of cell-centered to face-centered formulations. First, we compare the results using the cell-centered (Eq. (17)) and face-centered (Eq. (9)) formulations, the latter being the present balanced-force algorithm. We perform the computations in 2D on a mesh of resolution $R/h = 5$ and use the convolution method to compute the curvatures with a smoothing length $d = 1.9h^{1/2}$ in the continuous approach for surface tension. The spurious currents and pressure profile through the drop for the two formulations are shown in Figs. 8 and 9, respectively. The face-centered formulation yields a maximum velocity about 36 times smaller than the cell-centered formulation. The face-centered formulation also yields a pressure profile that does not possess spikes as seen in the cell-centered formulation. Improvements gained with a balanced-force algorithm are clearly evident.

4.1.3.2. 3D static drop. In Table 2, results are presented for a 3D inviscid drop after one and 50 time steps using the convolution technique and the height function method for curvatures and compared to the results of Williams et al. [30]. Note that the BKZ method of Williams et al. [30] is closer to our convolution method, since we do not use delta functions as in method I and II. The balanced-force algorithm yields spurious velocities after one time step that are smaller than those obtained with the methods of Williams et al. [30]. After 50 time steps, using the convolution method, the order of the spurious currents is the same as the best case of Williams et al. [30], however the growth of the maximum spurious velocity is more pronounced. With the height function method to estimate curvatures, maximum velocities an order of magnitude lower than with the convolution technique are obtained. The results of Table 2 show that the

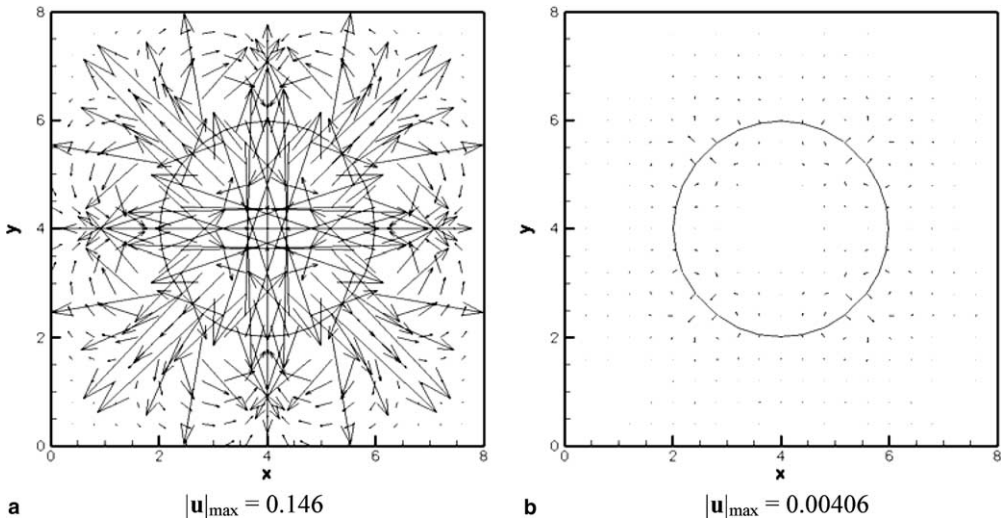


Fig. 8. Plot of the velocity field and drop shapes ($f = 1/2$ contour) for the inviscid static drop in equilibrium with (a) the cell-centered formulation of Eq. (17) and (b) the face-centered formulation of Eq. (9). The drop of radius $R = 2$ is centered in an 8×8 domain. The surface tension coefficient is 73 and the density ratio is 10. The results are shown after one time step with $\Delta t = 10^{-3}$ on a grid of resolution $R/h = 5$. The curvature is computed using the convolution model with a smoothing length of $d = 1.2$.

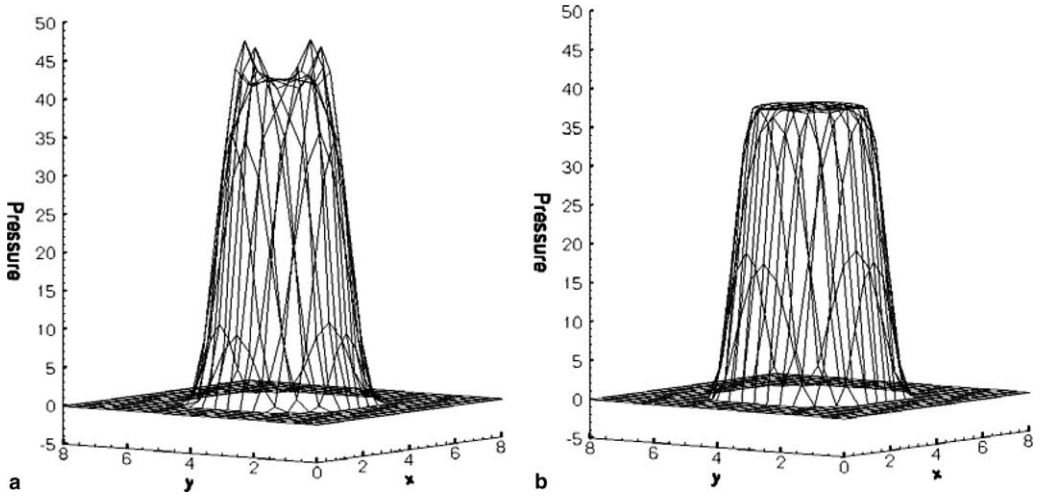


Fig. 9. Computed pressure for the inviscid static drop in equilibrium with (a) the cell-centered formulation of Eq. (17) and (b) the face-centered formulation of Eq. (9). The drop of radius $R = 2$ is centered in an 8×8 domain. The surface tension coefficient is 73 and the density ratio is 10. The results are shown after one time step with $\Delta t = 10^{-3}$ on a grid of resolution $R/h = 5$. The curvature is computed using the convolution model with a smoothing length of $d = 1.2$.

Table 2
Error in maximum velocity $|\mathbf{u}|_{\max}$ after one and 50 time steps for a 3D inviscid static drop in equilibrium with different curvature estimates

$\Delta t = 10^{-3}$	Balanced-force algorithm		Williams et al. [30]		
	Convolution	Height function	BKZ	Method I	Method II
$t = \Delta t$	4.87×10^{-3}	4.02×10^{-3}	3.49×10^{-1}	1.03×10^{-1}	8.55×10^{-2}
$t = 50\Delta t$	1.63×10^{-1}	4.02×10^{-2}	2.55	8.46×10^{-1}	3.86×10^{-1}

The drop of radius 2 is centered in an 8^3 domain, the density ratio is 10, and the time step is constant and equal to 10^{-3}. BKZ represents the results of the original CSF paper of Brackbill et al. [1] with a smooth volume fraction, Method I represents the Williams et al. [30] results with convolved curvatures and a step delta function, and Method II represents the Williams et al. [30] results with finite-differenced normals and a parabolic delta function.

effect of an accurate curvature model (i.e., the height function) is still extremely important even when a balanced-force algorithm is employed.

4.1.4. Static drop: continuous/sharp surface tension model comparison

4.1.4.1. Comparison of the continuous and sharp approaches. Pressure profiles through the drop are shown in Fig. 10 for continuous and sharp surface tension models when curvatures are estimated with either the convolution or height function methods. It is evident from these plots that the continuous method results in a pressure transition that is not present in the more desirable abrupt jump generated with the sharp method. We note, however, that oscillations and an overprediction of the pressure inside the drop that is more pronounced with the sharp approach when the convolution method is used for estimating curvatures. This behavior is also evident in the pressure profiles along the x-direction ($y = 4$) and along the diagonal-direction ($y = x$) in Figs. 11 and 12 for the continuous and sharp approach, respectively. To better understand this pressure distribution, curvature errors are plotted versus (x, y) in Fig. 13. The plots show the error located along the perimeter of the drop, and indicate that errors are smaller with the height function relative to those obtained with the convolution method. This is also evident in error trends displayed in Fig. 7. Re-

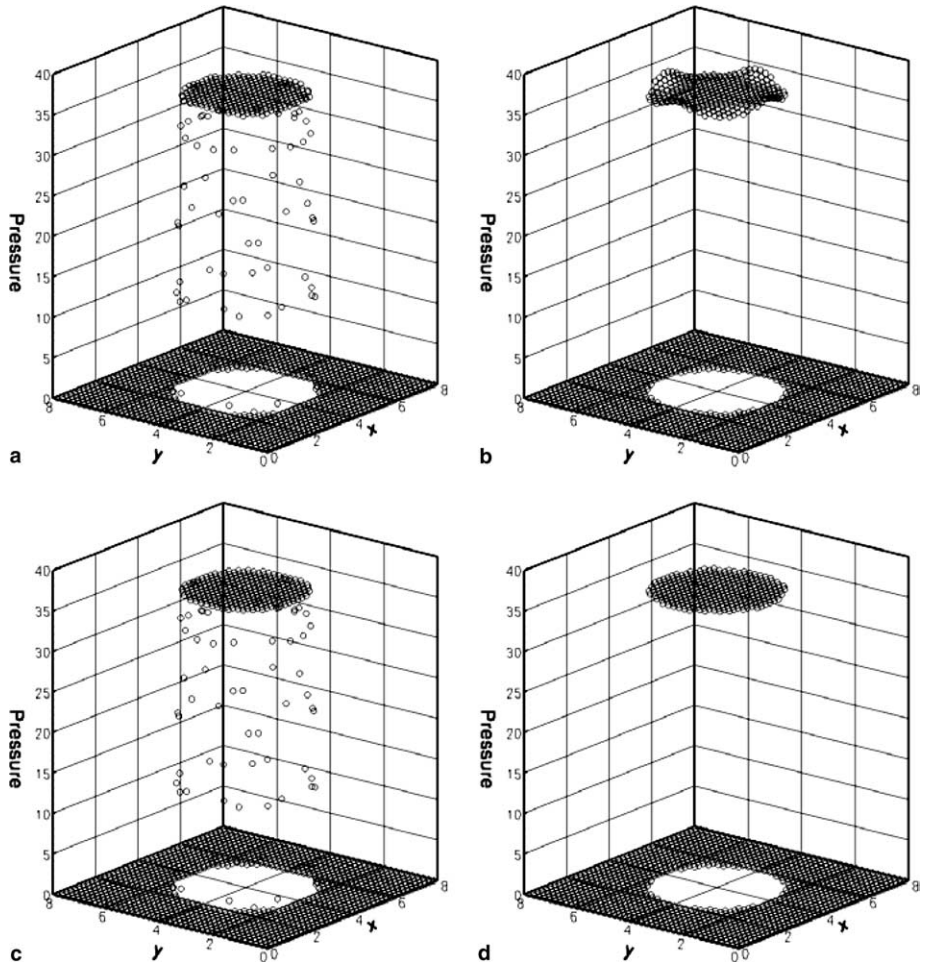


Fig. 10. Computed pressure for the inviscid static drop for the CSF and SSF models using both the convolution and height function techniques to estimate curvatures. Shown is (a) CSF with the convolution method, (b) SSF with the convolution method, (c) CSF with the height function method, and (d) SSF with the height function method. The grid is uniform and of resolution ($R/h = 10$). The drop of radius $R = 2$ is centered in an 8×8 domain. The surface tension coefficient is 73 and the density ratio is 10^3.

call that with the convolution technique, curvatures are estimated at every cell centroid, but the height function method only generates curvature estimates within interfacial cells (those having $0 < f < 1$). With convolution-based curvature estimates, errors are largest in the interior of the drop and along the x-direction (and y-direction by symmetry), which explains the high pressure peaks along the x-direction in Fig. 10(b). Along the diagonal-direction (not mesh-aligned), curvature errors resulting from the convolution method are smaller relative to mesh-aligned (e.g., x or y) directions. On the other hand, curvature errors resulting from height function estimates are largest along non-mesh-aligned directions, e.g., diagonal directions, corresponding to a 45° normal angle with equal components. The locations of highest curvature error also result in higher spurious current velocities, seen in Fig. 17(b). With the sharp model for surface tension, the drop pressure distribution is more sensitive to the distribution of the error in curvature than with the continuous approach.

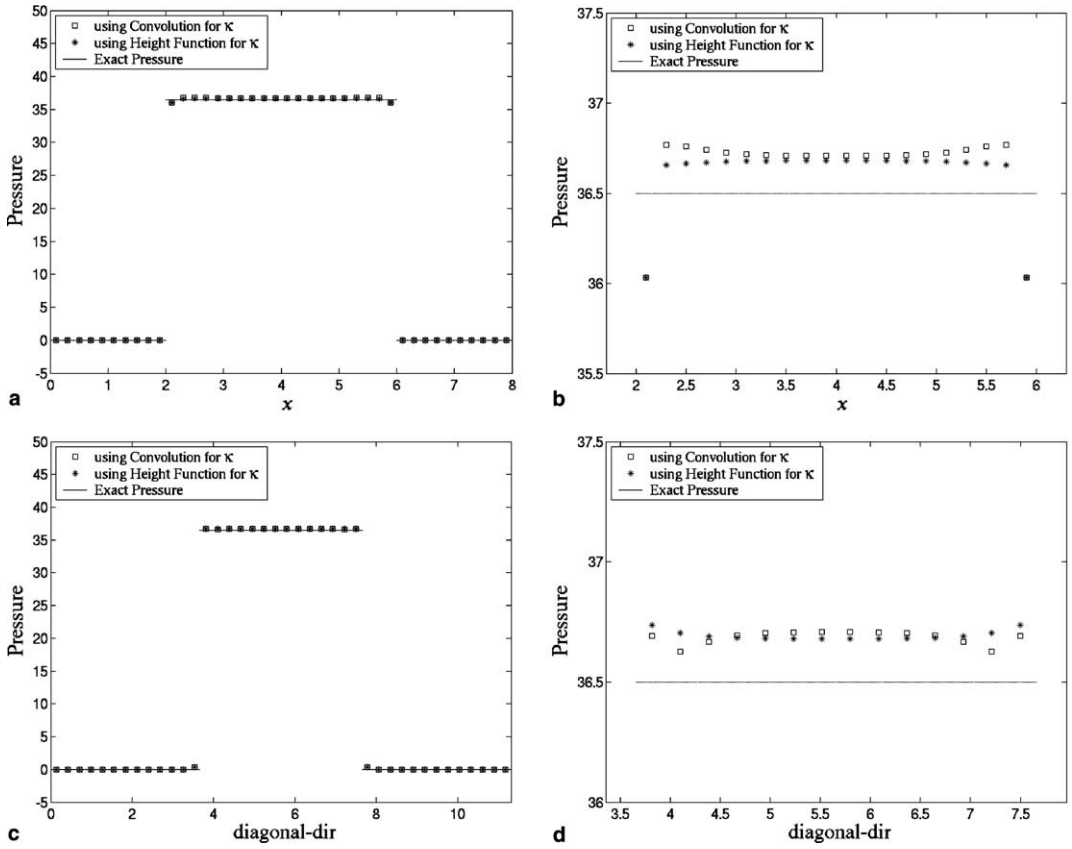


Fig. 11. Pressure along the x-direction at the mid-section $y = 4$ (a, b) and along the diagonal-direction (c, d) with the CSF model for the inviscid static drop when curvature is computed with the convolution and height function methods. The grid is uniform and of resolution $R/h = 10$. The drop of radius $R = 2$ is centered in an 8×8 domain. The surface tension coefficient is 73 and the density ratio is 10^3. The time step is 10^{-6}.

Next, we consider the performance of these methods under mesh refinement by first investigating the error in velocity. This error, as shown in Fig. 14 is of same order of magnitude for both the continuous and sharp surface tension models. With convolution-based curvature estimates, velocity errors are an order of magnitude greater than those generated with height function curvatures estimates. The maximum velocity errors are given in quantitative detail in Tables 3–9. From the L_1 and L_2 velocity error norms in Figs. 14(a) and (b), we note that convolution-based curvature estimates exhibit a first-order convergence with the sharp approach, but less than first-order with the continuous approach. With height function curvature estimates, on the other hand, second-order convergence is realized with both the continuous and sharp approaches. For the L_∞ error norms (Fig. 14(c)), convolution-based curvature estimates exhibit no convergence with the continuous approach, and less than first-order convergence with the sharp approach. With height function curvature estimates, the same convergence behavior is observed with both the continuous and sharp models. For the height function, we note that better convergence results when face curvatures are interpolated with the kernel approach of Eq. (38) rather than the simple interpolation of Eq. (37) (Fig. 14(d)). The interpolation acts like a filter, smoothing high-frequency errors.

Pressure jump errors, shown in Fig. 15, exhibit first-order convergence with the continuous surface tension model and second-order convergence with the sharp model as measured by $E(\Delta P_{\text{total}})$ (Fig. 15(a)). Sec-

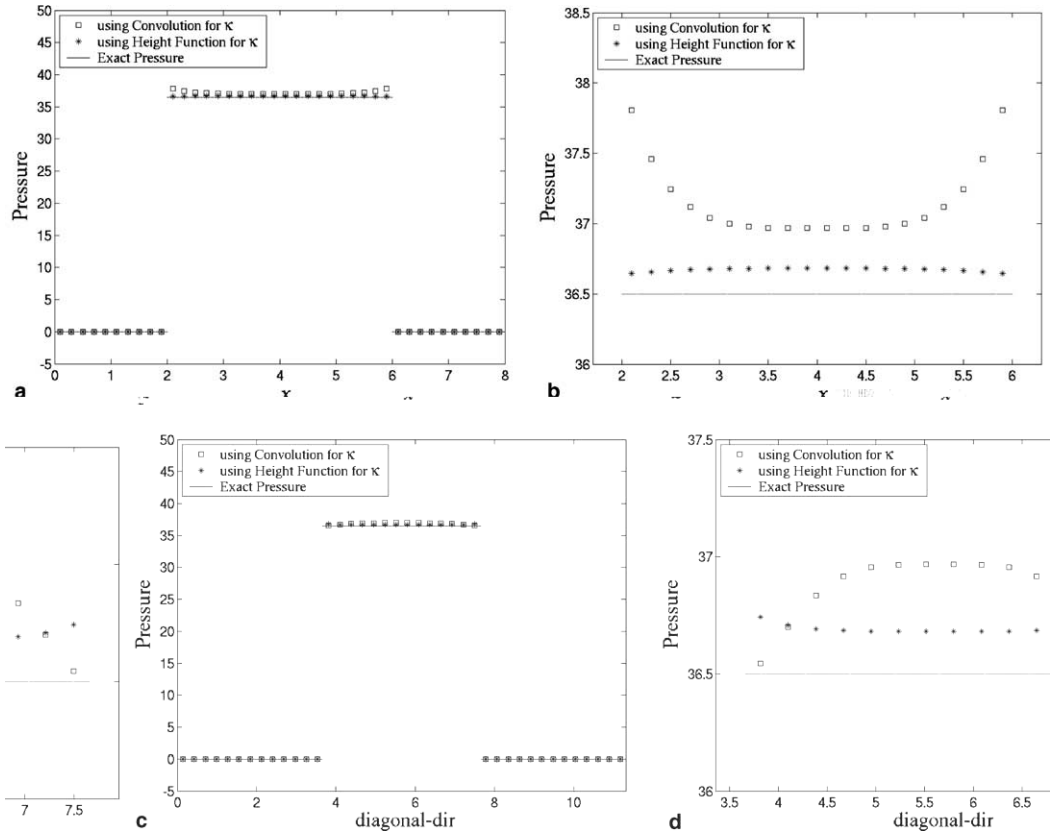


Fig. 12. Pressure along the x-direction at the mid-section $y = 4$ (a, b) and along the diagonal-direction (c, d) with the SSF model for the inviscid static drop when curvature is computed with the convolution and height function methods. The grid is uniform and of resolution $R/h = 10$. The drop of radius $R = 2$ is centered in an 8×8 domain. The surface tension coefficient is 73 and the density ratio is 10^3. The time step is 10^{-6}.

ond-order convergence in pressure jump error is realized for both methods as measured by $E(\Delta P_{\text{partial}})$ (Fig. 15(b)). The first point in the sharp/convolution results of Fig. 15(a) is not aligned with others due to intrinsic error in the reconstructed distance function (owing to errors in the interface normal estimates) [6]. When performing the same computation using an exact distance function, all points in Fig. 15(a) align with a slope of two (for second-order convergence).

From this static drop comparison of surface tension models (sharp or continuous) and curvature estimates (height function or convolution), we conclude that:

- the order of magnitude of the spurious velocities depend on the curvature method *and not* on the surface tension model; and
- the error in pressure jump and distribution depends primarily on the surface tension model.

In the recent work of Kang et al. [12], the authors compare surface tension models using either a ghost fluid (sharp) or delta function (continuous) method within a level-set formulation, also using an equilibrium static drop for the comparison framework. They found that the GFM resulted in lower spurious velocities relative to the delta function formulation. This basic result is inconsistent with those observed here, namely

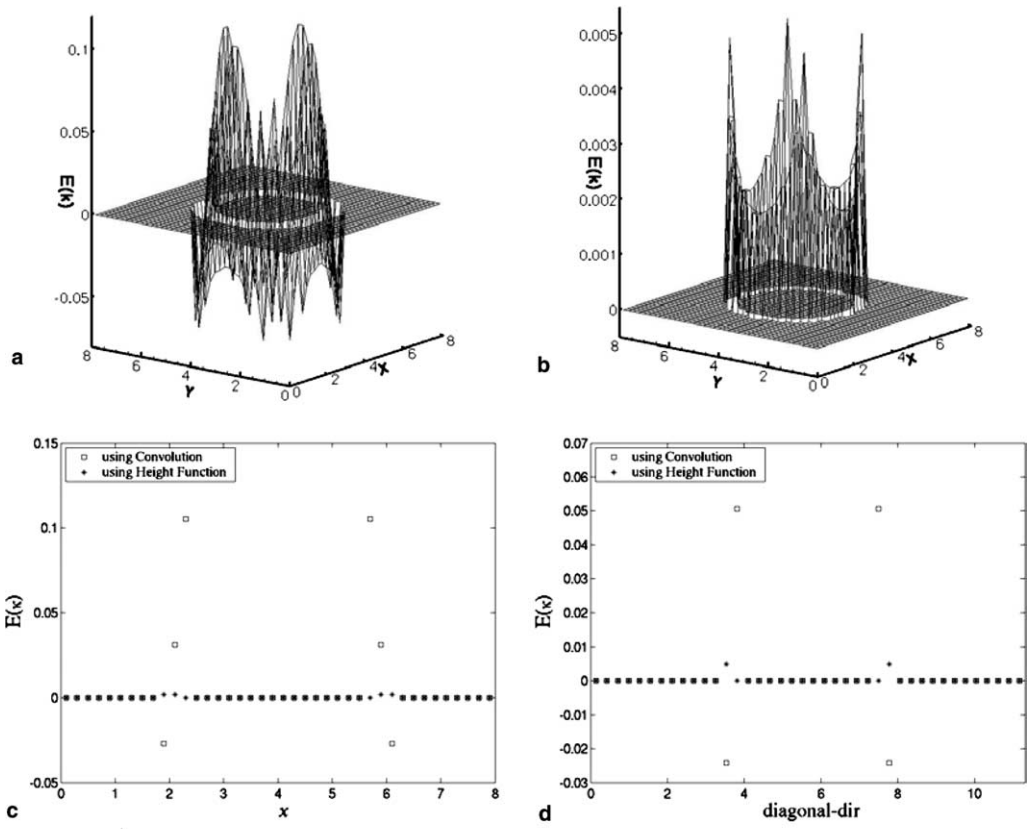


Fig. 13. Plots of the error in curvatures versus position for a circular drop of radius (R) equal to 2 centered in an 8×8 domain. The grid is uniform and of resolution $R/h = 10$. Shown is error in curvature versus (x, y) using (a) the convolution method and (b) the height function. Shown in (c) is error in curvature along the x-direction at $y = 4$ and (d) error in curvature along the diagonal-direction.

spurious velocities of same order for a given curvature model are generated with the continuous and sharp approaches. A possible reason for this difference is that level-set flow algorithm is not adequately designed to give force balance (between surface tension and pressure gradient forces) using the delta formulation. Also, in using a delta function (continuous) method within a level-set formulation, the specified Heaviside function acts to spread fluid properties around the interface more relative to our present volume tracking framework (where fluid properties are only averaged on interfacial cells). Another notable difference in their GFM is their consideration of sharp property jumps across the interface that is not considered in our sharp approach. In our case, to facilitate the sharp/continuous comparison, we have employed an identical, force-balanced flow algorithm where fluid properties are only averaged in cells containing the interface (for both the continuous and sharp approach). We also believe that fluid densities should not affect the pressure jump, owing to the fact that the term $1/\rho_f$ appears on both sides of the pressure equation and hence should vanish.

Next we investigate the effect of the fluid properties and the time step on the spurious currents.

4.1.4.2. Effect of fluid properties and integration time step. For this section, the results for the error in maximum velocity are presented in Tables 3–9. In Table 3, we investigate results integrated to the same time

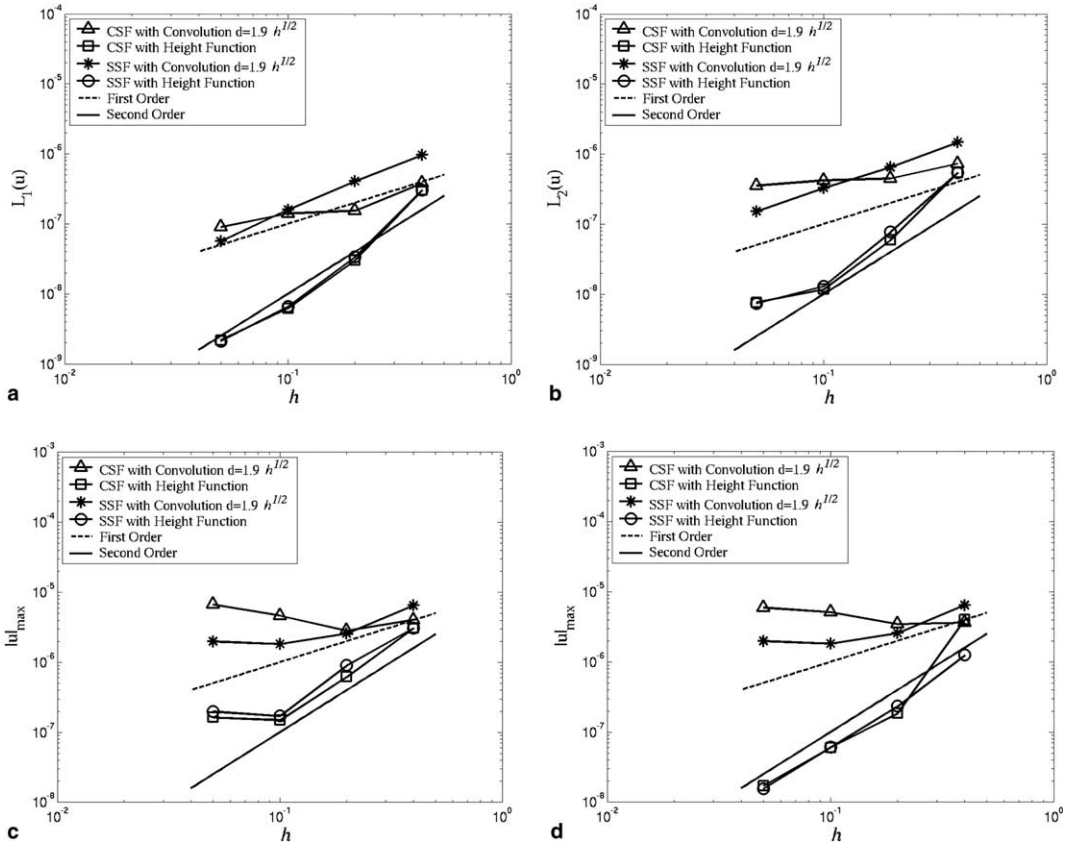


Fig. 14. Error in velocity after one time step for the inviscid static drop when the curvature is computed using the convolution technique and the height function method for different mesh sizes of spacing h. The drop of radius $R = 2$ is centered in an 8×8 domain, the density ratio is 10^3, and the time step is held constant at 10^{-6}. (a) and (b) L_1 and L_2 velocity error norm with the simple interpolation to estimate curvatures at faces (Eq. (30) for the convolution method and Eq. (37) for the height function method). (c) L_∞ velocity error norm with the simple interpolation to estimate curvatures at faces (Eq. (30) for the convolution method and Eq. (37) for the height function method). (d) L_∞ velocity error norm with the kernel interpolation to estimate curvatures at faces (Eq. (38) for both the convolution the height function methods).

Table 3
Effect of the time step on the error in maximum velocity $|\mathbf{u}|_{\max}$ at $t = 0.001$ for the inviscid static drop in equilibrium using the CSF and SSF models

Δt	Number of time steps	CSF		SSF	
		Convolution	Height function	Convolution	Height function
10^{-3}	1	1.94×10^{-3}	4.35×10^{-4}	2.58×10^{-3}	6.28×10^{-4}
10^{-4}	10^1	1.90×10^{-3}	3.92×10^{-4}	2.53×10^{-3}	5.68×10^{-4}
10^{-5}	10^2	1.84×10^{-3}	3.64×10^{-4}	2.56×10^{-3}	4.99×10^{-4}
10^{-6}	10^3	1.81×10^{-3}	3.53×10^{-4}	2.50×10^{-3}	4.79×10^{-4}

The drop of radius $R = 2$ is centered in an 8×8 domain, the fluid density ratio is 10, and the mesh is 40×40 ($R/h = 10$).

($t = 0.001$) using different time steps to investigate the dependency on the integration time step. With smaller time steps, the velocity at $t = 0.001$ are slightly smaller and more pronounced with the height function method.

Table 4
Effect of the fluid density ratio on the error in maximum velocity $|\mathbf{u}|_{\max}$ after one time step for the viscous static drop in equilibrium using the CSF and SSF models

ρ_1/ρ_2	CSF		SSF	
	Convolution	Height function	Convolution	Height function
10	1.94×10^{-6}	4.40×10^{-7}	2.30×10^{-6}	6.29×10^{-7}
10^3	2.84×10^{-6}	6.22×10^{-7}	2.59×10^{-6}	9.07×10^{-7}
10^5	2.89×10^{-6}	6.25×10^{-7}	2.59×10^{-6}	9.12×10^{-7}

The drop of radius $R=2$ is centered in an 8×8 domain and the mesh is 40×40 ($R/h=10$). The density inside the drop (ρ_1) is 1 and the density outside (ρ_2) is allowed to vary. The time step is constant and equal to 10^{-6}. The fluid viscosity ratio is 10, with $\mu_1=10^{-2}$ and $\mu_2=10^{-3}$.

Table 5
Effect of the viscosity ratio on the error in maximum velocity $|\mathbf{u}|_{\max}$ after 100 time steps for the viscous static drop in equilibrium using the CSF and SSF models

μ_1/μ_2	CSF		SSF	
	Convolution	Height function	Convolution	Height function
1	2.57×10^{-4}	4.82×10^{-5}	2.73×10^{-4}	6.63×10^{-5}
10	2.57×10^{-4}	4.83×10^{-5}	2.73×10^{-4}	6.64×10^{-5}
100	2.58×10^{-4}	4.83×10^{-5}	2.73×10^{-4}	6.64×10^{-5}

The drop of radius $R=2$ is centered in an 8×8 domain, the fluid density ratio is 10^3 with the density inside (ρ_1) and outside (ρ_2) the drop equal to 1 and 10^{-3}, respectively, and the mesh is 40×40 ($R/h=10$). The time step is constant and equal to 10^{-6}. The fluid viscosity inside the drop is $\mu_1=10^{-2}$ and outside the viscosity is allowed to vary.

Table 6
Effect of the fluid density on the error in maximum velocity $|\mathbf{u}|_{\max}$ after one time step for the viscous static drop in equilibrium using the CSF and SSF models

ρ_1	ρ_2	CSF		SSF	
		Convolution	Height function	Convolution	Height function
1	10^{-3}	2.84×10^{-6}	6.22×10^{-7}	2.59×10^{-6}	9.07×10^{-7}
10^1	10^{-2}	2.84×10^{-7}	6.22×10^{-8}	2.59×10^{-7}	9.07×10^{-8}
10^2	10^{-1}	2.84×10^{-8}	6.22×10^{-9}	2.59×10^{-8}	9.07×10^{-9}
10^3	1	2.84×10^{-9}	6.22×10^{-10}	2.59×10^{-9}	9.07×10^{-10}

The drop of radius $R=2$ is centered in an 8×8 domain, the fluid viscosity ratio is 10 with the viscosity inside (μ_1) and outside (μ_2) the drop equal to 10^{-2} and 10^{-3}, respectively, and the mesh is 40×40 ($R/h=10$). The density inside (ρ_1) and outside (ρ_2) the drop is allowed to vary, but the fluid density ratio is constant and equal to 10^3. The time step is constant and equal to 10^{-6}.

In Tables 4 and 5, we investigate the effect of the density and viscosity ratios, respectively. The properties of fluid 1 (the drop fluid) are kept constant. We notice as the density ratio increases a slight increase in spurious velocities occurs. As the viscosity ratio increases, we observe negligible difference in the results.

In Tables 6–9, the effects of the drop density, surface tension coefficient, time step and drop viscosity are given, respectively, for a constant density ratio of 10^3 and viscosity ratio 10^2. For the parameters studied here after one time step, we observe that spurious velocities are directly proportional to the surface tension coefficient and the time step, inversely proportional to the drop density, and independent of the drop viscosity. Previously, we also demonstrated that spurious velocities depend on curvature errors. If a simple dimensional analysis is performed, we have $u\sim\sigma\Delta t E(\kappa)^2/\rho_1$, with $E(\kappa)$ the error in curvature.

Table 7
Effect of the surface tension coefficient on the error in maximum velocity $|\mathbf{u}|_{\max}$ after one time step for the viscous static drop in equilibrium using the CSF and SSF models

σ	CSF		SSF	
	Convolution	Height function	Convolution	Height function
0.73	2.84×10^{-8}	6.22×10^{-9}	2.59×10^{-8}	9.07×10^{-9}
7.3	2.84×10^{-7}	6.22×10^{-8}	2.59×10^{-7}	9.07×10^{-8}
73	2.84×10^{-6}	6.22×10^{-7}	2.59×10^{-6}	9.07×10^{-7}
730	2.84×10^{-5}	6.22×10^{-6}	2.59×10^{-5}	9.07×10^{-6}

The drop of radius $R=2$ is centered in an 8×8 domain, the fluid density ratio is 10^3 with the density inside (ρ_1) and outside (ρ_2) the drop equal to 1 and 10^{-3}, respectively, and the mesh is 40×40 ($R/h=10$). The time step is constant and equal to 10^{-6}. The viscosity ratio is 10 with $\mu_1=10^{-2}$ and $\mu_2=10^{-3}$.

Table 8
Effect of the time step magnitude on the error in maximum velocity $|\mathbf{u}|_{\max}$ after one time step for the viscous static drop in equilibrium using the CSF and SSF models

Δt	CSF		SSF	
	Convolution	Height function	Convolution	Height function
10^{-3}	2.84×10^{-3}	6.22×10^{-4}	2.59×10^{-3}	9.07×10^{-4}
10^{-4}	2.84×10^{-4}	6.22×10^{-5}	2.59×10^{-4}	9.07×10^{-5}
10^{-5}	2.84×10^{-5}	6.22×10^{-6}	2.59×10^{-5}	9.07×10^{-6}
10^{-6}	2.84×10^{-6}	6.22×10^{-7}	2.59×10^{-6}	9.07×10^{-7}

The drop of radius $R=2$ is centered in an 8×8 domain, the fluid density ratio is 10^{-3} with the density inside (ρ_1) and outside (ρ_2) the drop equal to 1 and 10^{-3}, respectively, and the mesh is 40×40 ($R/h=10$). The viscosity ratio is 10 with $\mu_1=10^{-2}$ and $\mu_2=10^{-3}$.

Table 9
Effect of fluid viscosity on the error in maximum velocity $|\mathbf{u}|_{\max}$ after (a) 100 and (b) 1000 time steps for the viscous static drop in equilibrium using the CSF and SSF models

μ_1	μ_2	CSF		SSF	
		Convolution	Height function	Convolution	Height function
(a) After 100 times steps					
1	10^{-1}	2.50×10^{-4}	4.73×10^{-5}	2.73×10^{-4}	6.42×10^{-5}
10^{-2}	10^{-3}	2.57×10^{-4}	4.83×10^{-5}	2.73×10^{-4}	6.64×10^{-5}
10^{-4}	10^{-5}	2.58×10^{-4}	4.83×10^{-5}	2.73×10^{-4}	6.65×10^{-5}
(b) After 1000 times steps					
1	10^{-1}	2.22×10^{-3}	4.28×10^{-4}	2.82×10^{-3}	5.57×10^{-4}
10^{-2}	10^{-3}	2.41×10^{-3}	4.56×10^{-4}	2.87×10^{-3}	6.22×10^{-4}
10^{-4}	10^{-5}	2.41×10^{-3}	4.58×10^{-4}	2.83×10^{-3}	6.26×10^{-4}

The drop of radius $R=2$ is centered in an 8×8 domain, the fluid density ratio is 10^3 with the density inside (ρ_1) and outside (ρ_2) the drop equal to 1 and 10^{-3}, respectively, and the mesh is 40×40 ($R/h=10$). The time step is constant and equal to 10^{-6}. The viscosity ratio is constant and equal to 10.

In the results of the viscous cases after 100 and 1000 time steps shown in Table 9, the effect of the drop dynamic viscosity on the spurious currents is found to be small compared to the other parameters. We have also performed two computations with the CSF method and the convolution method to estimate curvatures

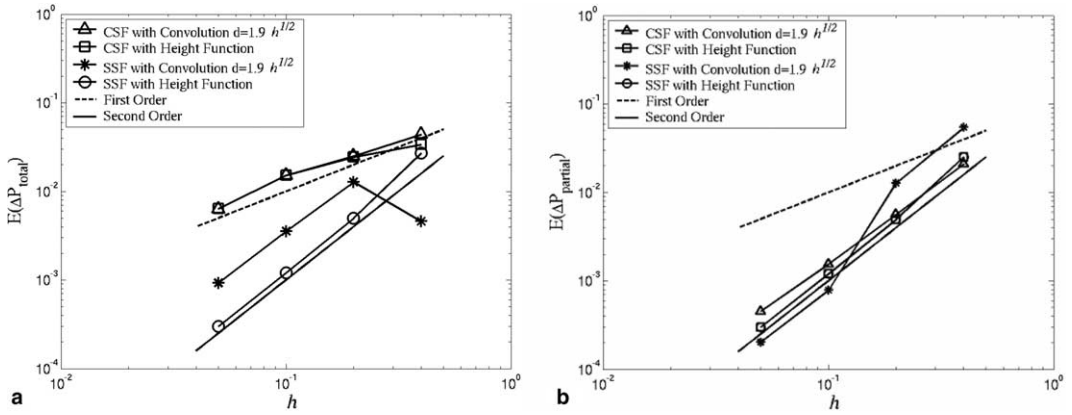


Fig. 15. Error in pressure jump after one time step for the inviscid static drop when the curvature is computed using the convolution technique and height function method for different mesh sizes of spacing h. The drop of radius $R = 2$ is centered in an 8×8 domain, the density ratio is 10^3, and the time step is constant equal to 10^{-6}. (a) Error ΔP_{total}, (b) error $\Delta P_{\text{partial}}$.

with different viscosities with a time step of $\Delta t = 10^{-5}$ and for 5000 time steps for a drop of density 1 and density ratio 10^3 and surface tension coefficient of 73. We obtain $|u|_{\max} = 1.19 \times 10^{-1}$ with $\mu_1 = 10^{-4}$ and $\mu_2 = 10^{-5}$ and $|u|_{\max} = 4.38 \times 10^{-2}$ with $\mu_1 = 1$ and $\mu_2 = 10^{-1}$. For this case, the effect of viscosity on the spurious currents is more pronounced which indicates that if the computations were to be continued we may observe the same relation $u \sim \sigma/\mu$ as in previous work [14,19,23,17,28].

4.1.4.3. Total kinetic energy evolution for 2D drop. A quantitative measure of the total spurious velocity is the total kinetic energy (TKE) of the flow. Examining the evolution of the TKE is an indication of the growth of the spurious velocities over time. The TKE, defined as

$$\text{TKE} = \frac{1}{2} \sum_{n\,\text{cells}} \left(\sum_k \rho_k f_k \right) V_c \mathbf{u}_c \cdot \mathbf{u}_c, \tag{47}$$

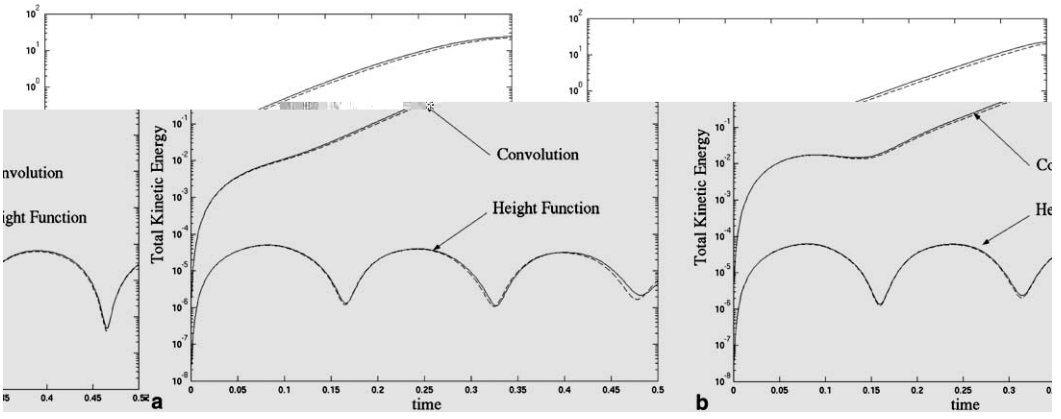


Fig. 16. Plot of the total kinetic energy versus time (for 500 time steps) for the inviscid (solid line) and viscous (dashed line) static drop in equilibrium using (a) the CSF and (b) SSF approaches. The drop of radius $R = 2$ is centered in an 8×8 domain on a grid of resolution ($R/h = 10$). The surface tension coefficient is 73, the density ratio is 10 with $\rho_1 = 1$ and $\rho_2 = 0.1$, the time step is constant equal to 10^{-3}. The y-axis is in log scale. For the viscous drop, the viscosity ratio is 10 with $\mu_1 = 10^{-2}$ and $\mu_2 = 10^{-3}$.

where V_c is the cell volume and the subscript k denotes material, is shown in Fig. 16 as a function of time for 2D inviscid and viscous static drops having a density ratio 10. Results are generated for both the (a) continuous and (b) sharp approach. The integration time step is kept constant at $\Delta t = 10^{-3}$ and is done over 500 time steps. The mesh resolution is such that $R/h = 10$. With height function curvatures, the total kinetic energy remains small and decrease slightly with time, whereas with convolution-based curvatures, the total kinetic energy increases rapidly. There is little difference between the inviscid and viscous cases, indicating again that the viscosity has a negligible effect on the problem relative to other parameters. Note

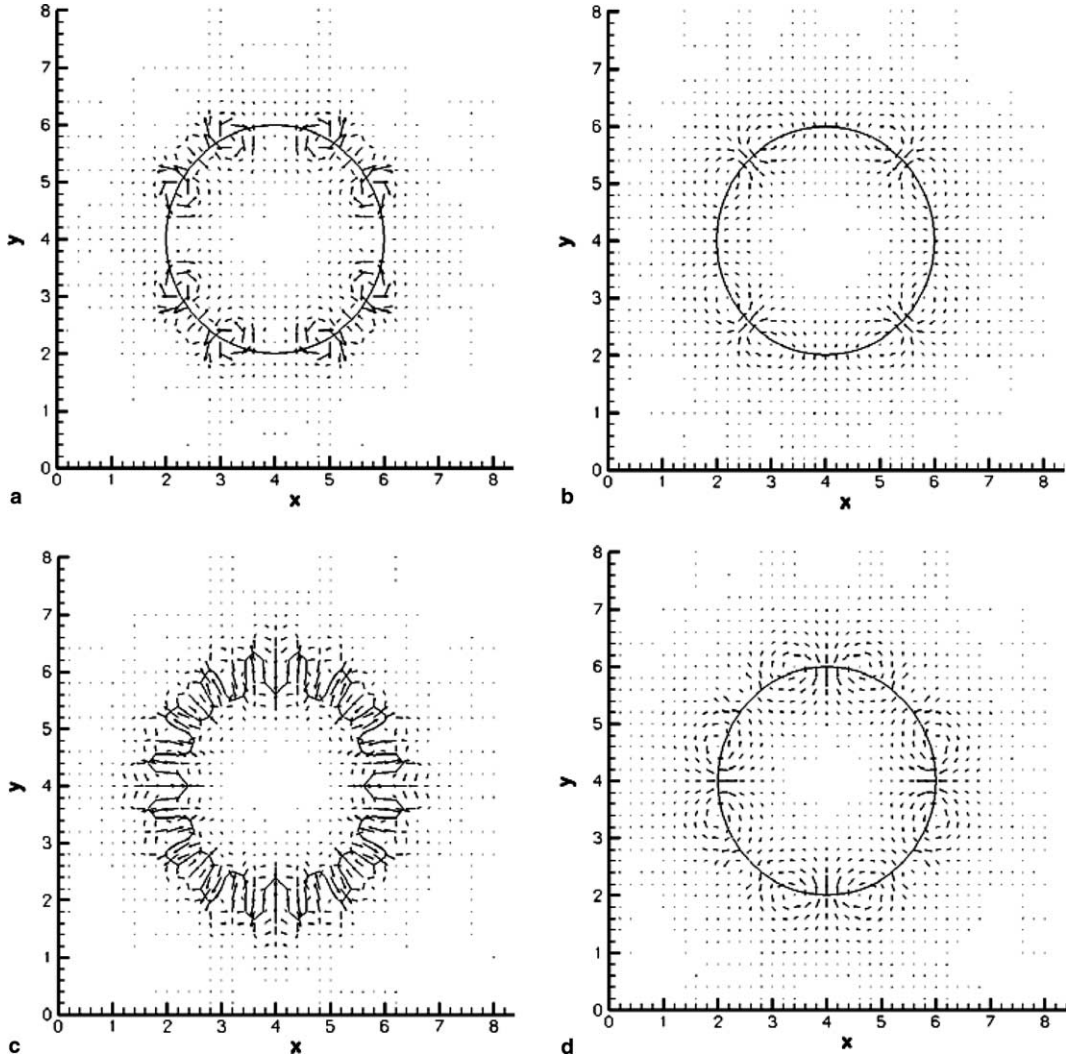


Fig. 17. Plot of the velocity vector field and drop shapes ($f = 1/2$ contour) of the volume fractions at times $t = 0.001$ and $t = 0.5$ for the inviscid static drop. Initially, the drop of radius $R = 2$ is centered in an 8×8 domain on a grid of resolution $R/h = 10$. The surface tension coefficient is 73, the density ratio is 10, and the time step is constant equal to 10^{-3}. (a) Velocity vectors magnified 300 times at time $t = 0.001$ using convolution for κ, (b) velocity vectors magnified 1000 times at time $t = 0.001$ using height function for κ, (c) velocity vectors not magnified (reference) at time $t = 0.5$ using convolution for κ, (d) velocity vectors magnified 1000 times at time $t = 0.5$ using height function for κ.

also that differences between the continuous and sharp surface tension approaches are not significant, suggesting again that the surface tension model choice within our balanced-force algorithm has little effect on the spurious velocity. As a baseline, in performing the same computations while imposing the exact curvature, the total kinetic energy is observed to remain constant at exactly zero (round off), hence further evidence that our balanced-force algorithm retains an exact balance between pressure gradient and surface tension forces. We can conclude, then, that spurious velocities in this framework result from error in curvature estimations, and that those induced by height function curvature estimates are lower.

In Fig. 17, the volume fraction half contour ($f = 1/2$) of the drop and the velocity field are plotted at times of $t = 0.001$ and $t = 0.5$. It is interesting to note that after the first time step ($t = 0.001$), the structure and magnitude of the velocity field are already different for the convolution and height function methods. With the height function method the magnitude of the vortices is lower than with the convolution method. These vortices are a direct result of inaccuracies in the curvature calculation.

At a much later time ($t = 0.5$), one can observe the effect of the increase in the velocity field and associated vortical structures. With the convolution method for estimating curvature, the drop has lost its circular shape and shows a regular (symmetric) deformation. With the height function method for estimating curvatures, the drop remains circular and velocities remain constant relative to the first time step. With the height function, the TKE oscillates with time. When looking at the vortices, we observe a change in direction as seen in the relative differences of Figs. 17(b) and (d). The velocity vectors with the height function seem to be in a 'stable' mode, whereas with the convolution they appear to be in an unstable regime, owing possibly to nonlinear advection effects.

Next, to illustrate dynamic cases, we briefly present results for an oscillating drop and a rising bubble.

4.2. Oscillating drops

Oscillating drops represent a standard test case for surface tension models. We consider the same test case as in [26]. Initially, the drop interface is an ellipse given by the equation $x^2/9 + y^2/4 = 1$. The drop is placed in a computational domain of size 20×20, partitioned with a 64×64 mesh. Gravitational forces are absent. The fluid properties are: $\rho_1 = 1$, $\rho_2 = 0.01$, $\mu_1 = 0.01$, $\mu_2 = 5 \times 10^{-5}$ and $\sigma = 1$. Due to the surface tension force, the droplet oscillates.

For this problem, the computed evolution of the total kinetic energy is shown in Fig. 18 for the continuous and the sharp approaches using both the convolution and height function estimates for curvature. For

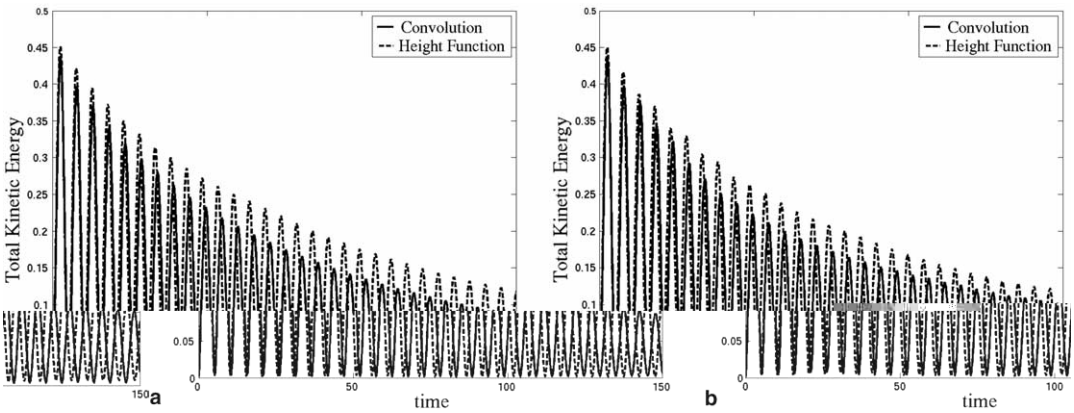


Fig. 18. Plot of the total kinetic energy versus time for the oscillating drop case using (a) the CSF and (b) the SSF approaches. The initial drop is an ellipse centered in a 20×20 domain on a 64×64 mesh.

the convolution method a smoothing length of $d = 0.8$ is used. For reasons similar to the static drop problem, we note again that very little difference is observed between the continuous and sharp approaches, with the main differences owing to the two curvature models. When the convolution method is used for curvature estimation, the total kinetic energy is smaller (relative to the height function) and the period slightly greater. We observe similar frequency of the drop oscillations as in [26] and first TKE peak. However, with our volume tracking method we observe a smaller decay than the point-set (front tracking) method of Torres and Brackbill [26], with the difference here likely due to the kinematical differences of the interface tracking methods. The volume tracking method is a purely Eulerian method, reconstructing the interface from volume fractions, whereas the front tracking method tracks the interface in a Lagrangian way through a set of marker points. The latter can be more precise in the computations of interfacial geometry such as curvature when the topology is not complex (broken up) as in this case.

4.3. Rising of air bubble in water

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

$$Bo = \frac{\rho g D^2}{\sigma}, \tag{48}$$

where D is the bubble diameter. The computational domain for this test problem is chosen to be $[-1,1]\times[-1,2]$. A drop of radius $R = 1/3$ is initially placed at $(0,0)$. The mesh is 40×60 ($R/h = 6.667$).

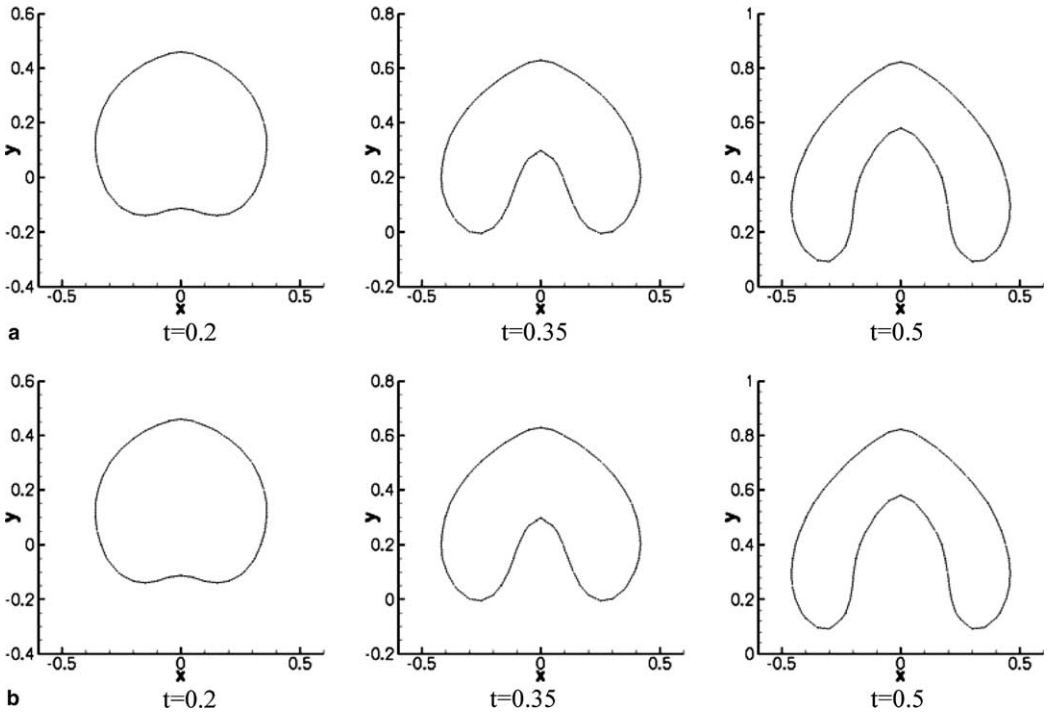


Fig. 19. Computed shapes ($f = 1/2$ contour) at times $t = 0.2$, 0.35, and 0.5 for the rising bubble with Bond numbers $Bo_1 = 73.4$ and $Bo_2 = 59.83\times 10^3$ (low surface tension) when curvatures are computed using the convolution technique with smoothing length $d = 0.15 = 3/2$ (dotted lines) and the height function method (solid lines). (a) Continuous approach (CSF), (b) sharp approach (SSF).

For the first case, the fluid properties of air (fluid 1) and water (fluid 2) are: $\rho_1 = 1.226$, $\rho_2 = 1000$, $\mu_1 = 1.137$, $\mu_2 = 1.78 \times 10^{-5}$ and $\sigma = 0.0728$. The downward-acting gravitational acceleration is $g = 9.8$. We note that a more realistic computation can be realized by modifying the boundary conditions to simulate the air-bubble injection into the water tank. At the bottom of the domain, air can be injected by specifying a material inflow boundary condition through a few cells. However, to ensure the incompressibility condition, it requires the addition of some outflow boundary condition cells.

The first case corresponds to a Bond number of $Bo_1 = 73.4$ and $Bo_2 = 59.83 \times 10^3$. For the second case, the only parameter changed is the surface tension coefficient, which is increased to $\sigma = 728$, corresponding to smaller Bond numbers of $Bo_1 = 7.34 \times 10^{-3}$ and $Bo_2 = 5.983$. The first case corresponds to the large bubble problem considered by Kang et al. [12].

Computed bubble shapes (measured by the $f = 1/2$ contour) are shown in Figs. 19 and 20 at times of $t = 0{,}0.2$, 0.35 and 0.5 for the case of $Bo_1 = 73.4$ and $Bo_2 = 59.83 \times 10^3$ and $Bo_1 = 7.34 \times 10^{-3}$ and $Bo_2 = 5.983$, respectively. For case of $Bo_1 = 73.4$ and $Bo_2 = 59.83 \times 10^3$, observable differences in the shapes are not evident from the use of different curvature estimation methods (convolution or height function) or different surface tension models (continuous or sharp approach). This is because in this case the surface tension force is not dominant, i.e., the flow is driven primarily by buoyancy forces.

In the second case, with smaller Bond numbers ($Bo_1 = 7.34 \times 10^{-3}$ and $Bo_2 = 5.983$), we observe less deformations in the bubble shapes since surface tension is more dominant relative to the first case. However, we observe a difference in the shapes for different curvature and surface tension models. Smaller deformations are observed with the continuous approach relative to the sharp approach, and smaller differences

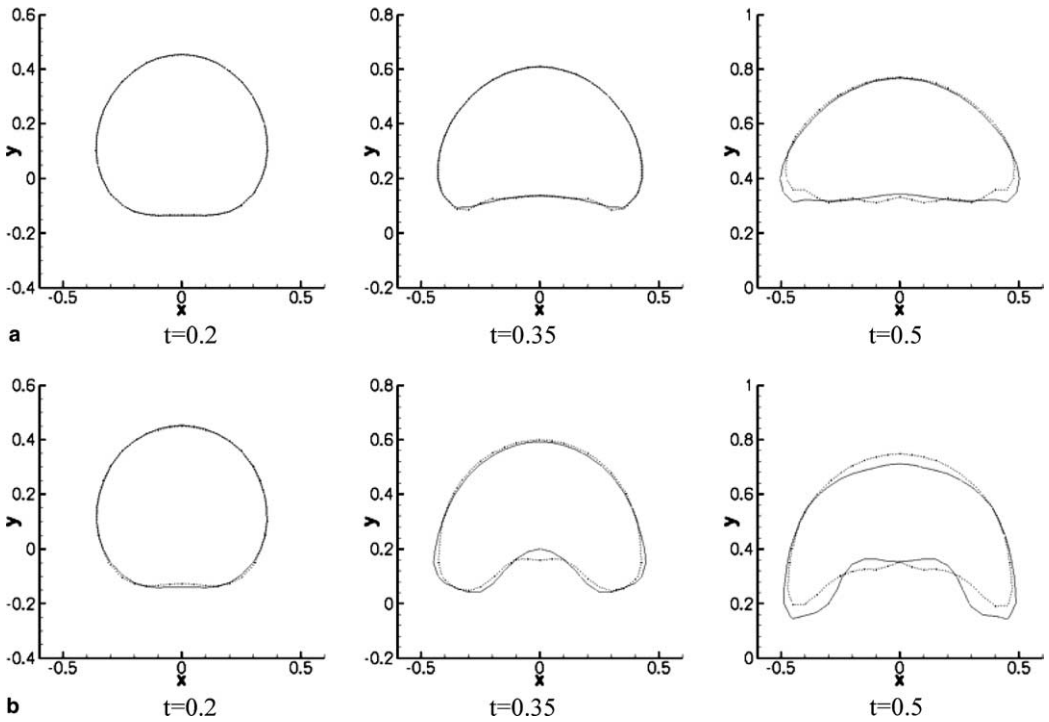


Fig. 20. Computed shapes ($f = 1/2$ contour) at times $t = 0.2$, 0.35, and 0.5 for the rising bubble with Bond numbers $Bo_1 = 7.34 \times 10^{-3}$ and $Bo_2 = 5.983$ (large surface tension) on a coarse mesh (40×80) when curvatures are computed using the convolution technique with smoothing length $d = 0.15 = 3h$ (dotted lines) and the height function method (solid lines). (a) Continuous approach (CSF), (b) sharp approach (SSF).

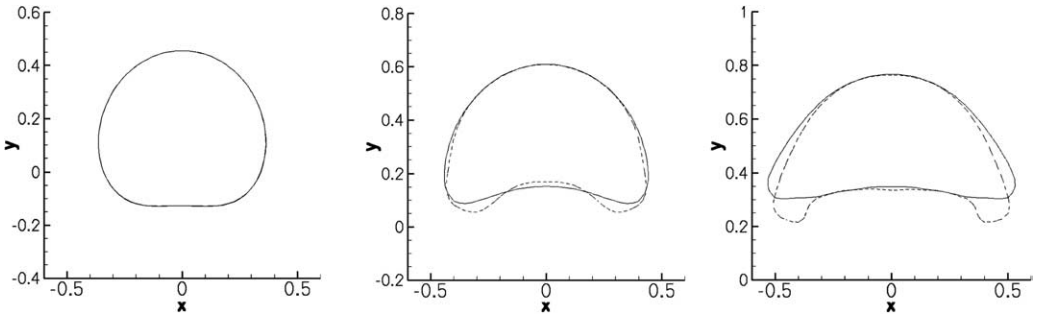


Fig. 21. Computed shapes ($f = 1/2$ contour) at times $t = 0.2$, 0.35, and 0.5 for the rising bubble with Bond numbers $Bo_1 = 7.34 \times 10^{-3}$ and $Bo_2 = 5.983$ (large surface tension) on a fine mesh (80×120) using the continuous approach (solid line) and sharp approach (dashed line) when curvatures are computed using the height function method.

in shapes are observed between the two curvature estimations for a given surface tension model. The resulting shapes with the convolution model are not as smooth as with the height function, especially at time $t = 0.5$. With the convolution estimation of curvature, a smoothing length d is kept constant throughout the calculation. A change to the convolution method that would likely improve these results would be to have an adaptive smoothing length d in different regions along the interface (depending upon local curvature), which is something to be considered in future work.

Next, we perform computations on a finer grid (80×120) using the height function method to compute curvatures with both the continuous and sharp approaches. The computed bubble shapes are shown in Fig. 21. Here we observe better convergence of the results between the continuous and sharp approaches. The main difference observed between the two approaches is that the sharp approach yields more bubble deformation on the side trailing edges, suggesting the importance of the pressure distribution in the bubble dynamics. Also, as opposed to the level-set results of Kang et al. [12], our results with the VOF method conserve mass exactly.

5. Summary and conclusions

The motivation for this work was straightforward: the improved modeling of surface tension-driven (low Weber/Bond number) flows within a volume tracking framework. The metric for this improvement is the reduction of numerical inaccuracies such as false velocity fields (“spurious currents”) in standard test problems. The initial plan was to tackle this problem primarily through the incorporation of a sharp interface representation (e.g., ghost fluid), as opposed to a continuous (CSF-like) approach. We anticipated that this would achieve our goals based on previous published successes in the literature. The ultimate path taken, however, was not what was initially expected. This path led us to three key conclusions and recommendations, as detailed in this paper, which we again summarize here.

First, we find that the most important consideration for modeling surface tension-driven flows is the formulation of an overall flow algorithm whose inherent property (by design) is force balance. This is a subtle (and perhaps obvious) point that is often overlooked in unsteady flow algorithms published today. The best test for force balance within a time-unsteady algorithm is achieving (and holding) steady state. We have devised such an algorithm within a cell-centered framework and demonstrated that this algorithm can achieve an exact (to within round-off) balance of surface tension and pressure gradient forces when interfacial curvatures are known accurately. This important litmus test also underscores an important finding, namely that this algorithm indeed legislates force balance when the forces can be accurately estimated.

Second, we find that a sharp (ghost fluid) representation for surface tension forces (SSF) can indeed be formulated within a "non-LS" (volume tracking) framework. This is achieved by temporarily reconstructing a distance function following the technique of Cummins et al. [6] – a technique that does not require solving a distance function evolution equation as in standard LS techniques. Given this sharp model for surface tension (SSF) and a pre-existing continuous surface tension model (CSF), both embodied within the same balanced-force flow algorithm, an objective comparison of these two models can be made for the first time within a volume tracking framework. In performing this comparison, we find that the only difference between the continuous and sharp approaches is in the spatial distribution of the pressure jump across the interface. The CSF yields a continuous pressure jump and is first-order accurate in pressure, whereas the SSF yields a sharp pressure jump and is second-order accurate in pressure. However, both CSF and SSF yield spurious currents of the same order and same convergence behavior for a given curvature model.

Third, we find that the origin of spurious currents within our balanced-force flow algorithm, regardless of the surface tension model employed, is errors in curvature estimates. We are therefore confident that by devising more accurate methods for computing curvatures from volume fractions (e.g., the spline-interpolant technique of Ginzburg and Wittum [9]) further reduction in spurious velocities (relative to this work) is achievable.

Finally, dynamic test cases have been performed on simple yet difficult flow problems (rising bubble, oscillating drop) to illustrate the versatility (robustness and accuracy) of our overall method. By having a balanced-force flow algorithm plus a choice of surface tension models and curvature estimate techniques, the relative merits of each individual model and model combination can be explored in detail. This has helped to narrow the search for cause and effect in algorithmic flaws.

As a result of this study, models for surface tension-driven flows are becoming increasingly more accurate and reliable. This evolution will rely upon better and better curvature estimate techniques, as shown in this work, but will also benefit improved models for interface kinematics, e.g., hybrid approaches utilizing both volume and front tracking methodologies.

Acknowledgments

This work is supported by the United States Department of Energy Advanced Simulation and Computing (ASC) Program. The authors thank Bryan Lally of the Los Alamos National Laboratory and Markus Bussmann of the University of Toronto for useful discussions and assistance.

References

[1] J.U. Brackbill, D.B. Kothe, C. Zemach, A continuum method for modeling surface tension, J. Comput. Phys. 100 (1992) 335–354.
[2] M. Bussmann, D.B. Kothe, J.M. Sicilian, Modeling high density ratio incompressible interfacial flows, Technical Report FEDSM2002-31125, in: Proceedings of the Sixth ASME/JSME Joint Fluids Engineering Conference, July 14–18, Montreal, Canada, 2002.
[3] A.J. Chorin, Numerical solution of Navier–Stokes equations, Math. Comput. 22 (1968) 745–762.
[4] A.J. Chorin, J.E. Marsden, A Mathematical Introduction to Fluid Mechanics, Springer, New York, 1993.
[5] A.V. Coward, Y.Y. Renardy, M. Renardy, J.R. Richards, Temporal evolution of periodic disturbances in two-layer Couette flow, J. Comput. Phys. 132 (1997) 346–361.
[6] S.J. Cummins, M.M. Francois, D.B. Kothe, Estimating curvature from volume fractions, Comput. Struct. 83 (2005) 425–434.
[7] M.M. Francois, D.B. Kothe, E.D. Dendy, J.M. Sicilian, M.W. Williams, Balanced force implementation of the continuum surface tension force method into a pressure correction algorithm, Technical Report FEDSM2003-45175, in: Proceedings of the Seventh ASME/JSME Joint Fluids Engineering Conference, July 6–11, Honolulu, HI, 2003.

[8] S.H. Garrioch, B.R. Baliga, A multidimensional advection technique for PLIC volume-of-fluid methods, in: Proceedings of the Fifth Annual Conference of the Computational Fluid Dynamics Society of Canada, University of Victoria, Victoria, BC, 1997, pp. 11(3)–11(8).
[9] I. Ginzburg, G. Wittum, Two-phase flows on interface refined grids modeled with VOF, staggered finite volumes, and spline interpolants, J. Comput. Phys. 66 (2001) 302–335.
[10] J. Helmsen, P. Colella, E.G. Puckett, Non-convex profile evolution in two dimensions using volume of fluids, Technical Report LBNL-40693, Lawrence Berkeley National Laboratory, 1997.
[11] D. Jamet, D. Torres, J.U. Brackbill, The second gradient method for the direct numerical simulation of liquid–vapor flows with phase change, J. Comput. Phys. 182 (2002) 262–276.
[12] M. Kang, R.P. Fedkiw, X.-D. Liu, A boundary condition capturing method for multiphase incompressible flow, J. Sci. Comput. 15 (2000) 323–360.
[13] D.B. Kothe, W.J. Rider, S.J. Mosso, J.S. Brock, J.I. Hochstein, Volume tracking of interfaces having surface tension in two and three dimensions, Technical Report AIAA 96-0859, in: Proceedings of the 34th Aerospace Sciences Meeting and Exhibit, January 15–18, Reno, NV, 1996.
[14] B. Lafaurie, C. Nardone, R. Scardovelli, S. Zaleski, G. Zanetti, Modelling merging and fragmentation in multiphase flows with SURFER, J. Comput. Phys. 113 (1994) 134–147.
[15] X.-D. Liu, R.P. Fedkiw, M. Kang, A boundary condition capturing method for Poisson's equation on irregular domains, J. Comput. Phys. 160 (2000) 151–178.
[16] D. Lorstad, M.M. Francois, W. Shyy, L. Fuchs, Assessment of volume of fluid and immersed boundary methods for droplet computations, Int. J. Numer. Meth. Fluids 46 (2004) 109–125.
[17] M. Meier, G. Yadigaroglu, B. Smith, A novel technique for including surface tension in PLIC-VOF methods, Eur. J. B 21 (2002) 61–73.
[18] S. Popinet, S. Zaleski, A front-tracking algorithm for accurate representation of surface tension, Int. J. Numer. Meth. Fluids 30 (1999) 775–793.
[19] Y. Renardy, M. Renardy, PROST: a parabolic reconstruction of surface tension for the volume-of-fluid method, J. Comput. Phys. 183 (2002) 400–421.
[20] C.M. Rhie, W.L. Chow, A numerical study of turbulent flow past an isolated airfoil with trailing edge separation, AIAA J. 21 (1983) 1525–1532.
[21] W.J. Rider, D.B. Kothe, Reconstructing volume tracking, J. Comput. Phys. 141 (1998) 112–152.
[22] M. Rudman, Volume tracking methods for interfacial flow calculations, Int. J. Numer. Meth. Fluids 24 (1998) 671–691.
[23] E. Shirani, N. Ashgriz, J. Mostaghimi, Interface pressure calculation based on conservation of momentum for front capturing methods, J. Comput. Phys. 203 (2005) 154–175.
[24] M. Sussman, A second order coupled level set and volume-of-fluid method for computing growth and collapse of vapor bubbles, J. Comput. Phys. 187 (2003) 110–136.
[25] The Telluride Team, Truchas: physics and algorithms, Technical Report LA-UR-03-0166, Los Alamos National Laboratory, 2003.
[26] D.J. Torres, J.U. Brackbill, The point-set method: front-tracking without connectivity, J. Comput. Phys. 165 (2000) 620–644.
[27] M. Torrey, L. Cloutman, R. Mjolsness, C. Hirt, NASA-VOF2D: a computer program for incompressible flows with free surfaces, Technical Report LA-10612-MS, Los Alamos National Laboratory, 1985.
[28] G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, S. Nas, Y.-J. Jan, A front-tracking method for the computations of multiphase flow, J. Comput. Phys. 169 (2001) 708–759.
[29] M.W. Williams, Numerical methods for tracking interfaces with surface tension in 3-D mold filling processes, Ph.D. Dissertation, University of California, Davis, also Technical Report LA-13776-T, Los Alamos National Laboratory, 2000.
[30] M.W. Williams, D.B. Kothe, E.G. Puckett, Convergence and accuracy of continuum surface tension models, in: W. Shyy, R. Narayanan (Eds.), Fluid Dynamics at Interface, Cambridge University Press, Cambridge, 1999, pp. 294–305.