Copyright © 2012 Tech Science Press CMES, vol.86, no.3, pp.199-223, 2012

On the Modeling of Surface Tension and its Applications by the Generalized Interpolation Material Point Method

L. Chen[1], **J. H. Lee**[1] **and C.-f. Chen**[1]

Abstract: This paper presents a numerical procedure to model surface tension using the Generalized Interpolation Material Point (GIMP) method which employs a background mesh in solving the equations of motion. The force due to surface tension is formulated at the mesh grid points by using the continuum surface force (CSF) model and then added to the equations of motion at each grid point. In GIMP, we use the grid mass as the color function in CSF and apply a moving average smoothing scheme to the grid mass to improve the accuracy in calculating the surface interface. The algorithm, named as GIMP-CSF, is implemented using the software package Uintah and benchmarked by three numerical examples: static equilibrium of a 2D liquid drop, dynamic evolution of a square drop in 2D and 3D, and the capillary rise. The benchmark results, when compared to analytical solutions and those obtained by other approaches, demonstrated the accuracy and effectiveness of the GIMP-CSF algorithm.

Keywords: MPM, GIMP, meshfree, particle, surface tension, CSF, smoothing, capillary rise.

1 Introduction

Surface tension exists at the interface of two immiscible fluids or at the junctions of solid, fluid, and gas. As a traction force that is proportional to the magnitude of curvature at the interface, surface tension exists wherever the interface is not flat. In particular, at small scales, surface tension becomes comparable in magnitude to other forces that govern fluid motion due to the large surface-to-volume ratio at these scales. An effective numerical method capable of modeling surface tension in its interaction with changing boundaries can be very useful to solve surface tension related problems.

Surface tension has been modeled by the volume of fluid (VOF) method [Bussmann, Mostaghimi, and Chandra (1999); Gueyffier, Li, Nadim, Scardovell, and

[1] Department of Mechanical Engineering, University of Alaska Fairbanks, Fairbanks,AK , USA.

Zaleski (1999); Sussman (2003)], level set method [Sussman, Smereka, and Osher (1994); Chang, Hou, Merriman, and Osher (1996); Liu, Fedkiw, and Kang (2000); Sussman (2003)], and the smoothed particle hydrodynamics (SPH) method [Nugent and Posch (2000); Morris (2000); Hu and Adams (2009); Zhang (2010)]. These methods model surface tension approximately as a "volume force" based on the continuum surface force (CSF) approach [Brackbill, Kothe, and Zemach (1992)].

Meshless or meshfree methods, such as SPH, have been gaining popularity due to their potential in dealing with complicated problems that involve failure, fragmentation and interactions of bodies with contact [Li and Liu (2002)]. The material point method (MPM) [Sulsky, Chen, and Schreyer (1994)] and Generalized Interpolation Material Point (GIMP) method [Bardenhagen and Kober (2004)] are promising meshfree methods in solving solid mechanics problems with large deformations to address mesh tangling and other issues. In particular, the GIMP method was developed to improve the instability issue in the original material point method. MPM has been applied to many engineering problems including contact [Bardenhagen, Guilkey, Roessig, Brackbill, Witzel, and Foster (2001)], crack and failure analysis [Nairn (2003)], explosion [Guilkey, Harman, and Banerjee (2007)], and brittle failure of disc particles under impact [Li, Pan, and Sinka (2011)].

MPM [Sulsky, Chen, and Schreyer (1994)] is an extension to solid mechanics from the fluid-implicit particle (FLIP) and particle in cell (PIC) methods in modeling highly distorted flow problems [Brackbill and Ruppel (1986)]. Consequently, MPM might be a suitable computational method to model fluid, structure and their interactions. However, to this date, the application of MPM for solving fluid-related problems has been limted [York, Sulsky, and Schreyer (1999); Lee and Guilkey (2008); Gan, Chen, and Montgomery-Smith (2011)]. In particular, the feasibility of modeling surface tension, as an important building block for the much more complicated problem of fluid-structure interactions at small scales, using MPM has not been demonstrated.

The objective of this paper is then to expand the modeling capability of the GIMP method to include surface tension. The paper is organized as follows. In Section 2 we review the GIMP method. The formulation of surface tension force by the CSF method for GIMP, and the numerical implementation of surface tension into an existing software package Uintah [Parker, Guilkey, and Harman (2006)] are given in Section 3. The newly developed algorithm is verified in Section 4 by a few numerical examples and compared to analytical solutions. Finally, discussions and major conclusion are given in Section 5.

2 Review of the GIMP Method

The numerical procedure of GIMP and other MPM-related methods consists of three major steps. First, the objects of interest are discretized into particles (i.e., material points) which carry all the material properties and history-dependent state variables. Second, a computational grid is constructed to cover the anticipated domain of simulation. Particle information is then mapped to the grid points where the equations of motion are solved to update the acceleration, velocity, and displacement at the grid points. Third, the status of each material point (such as displacement, velocity, acceleration, and state variables) is updated by mapping the information from the grid points. The procedure is repeated for each time step; details follow.

Using a circular disk (Fig. 1) for illustration, we review the GIMP method with a focus on the inclusion of the surface tension force in the equations of motion.

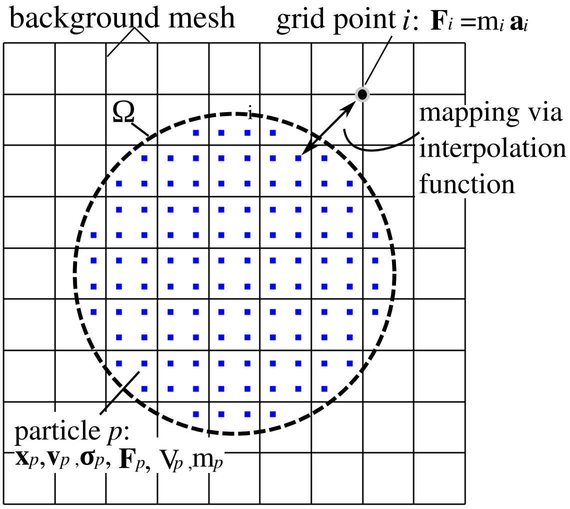
background mesh
grid point i: $\mathbf{F}_i = m_i \mathbf{a}_i$
Ω
mapping via interpolation function
particle p:
$\mathbf{x}_p, \mathbf{v}_p, \boldsymbol{\sigma}_p, \mathbf{F}_p, V_p, m_p$

Figure 1: Illustration of the procedure of GIMP. At the beginning of computation, the disk Ω is discretized into a collection of particles (i.e., material points) which are distributed over a background mesh. Each particle carries the local material properties and state variables of the disk: stress $\boldsymbol{\sigma}_p$, mass m_p, volume V_p, deformation gradient $\mathbf{F}_p$, position $\mathbf{x}_p$, and velocity $\mathbf{v}_p$. An interpolation function is then selected to map, in each time increment, the information from the particles to the grid points such that acceleration at each grid point can be solved via the equations of motion $\mathbf{F}_i = m_i\mathbf{a}_i$. Information at the grid points is then mapped back to the particles to complete the calculation of one time increment.

At the beginning of computation, the disk which occupies the domain Ω is discretized into N_p particles. A background mesh, also called the grid, is then constructed to cover the domain of simulation. The mesh consists of cells and grid points (nodes). The subscript p denotes a particle (i.e., a material point) and the subscript i indicates a node of the background mesh. In MPM, particles carry information including the material properties, mass (m_p), volume (V_p), all the state variables such as the stress tensor ($\boldsymbol{\sigma}_p$) and the deformation gradient ($\mathbf{F}_p$), and kinematic variables such as the velocity ($\mathbf{v}_p$). The particles may also carry external forces such as the body force $\mathbf{b}_p$ and surface traction $\boldsymbol{\tau}_p$.

At the beginning of each time step, the mass of the particle is mapped (extrapolated) to node i:

$$m_i = \sum_p S_{ip}(\mathbf{x}_p) m_p \tag{1}$$

where m_i is the mass at the node, $\sum\limits_p$ denotes summation over particles, and S_{ip} is the interpolation function. In this paper, we used the GIMP interpolation function [Bardenhagen and Kober (2004)] for S_{ip} which is computationally more robust than those used in the conventional MPM. Additional discussions on the implications of the interpolation function S_{ip} can be found in [Steffen, Wallstedt, Guilkey, Kirby, and Berzins (2008)].

The internal force at grid point i , $\mathbf{f}_i^{int}$, is calculated from the particle stress $\boldsymbol{\sigma}_p$ as:

$$\mathbf{f}_i^{int} = V_p \sum_p \mathbf{G}_{ip}(\mathbf{x}_p) \cdot \boldsymbol{\sigma}_p \tag{2}$$

where $\mathbf{G}_{ip} = \nabla S_{ip}$ is the gradient of the interpolation function, and V_p is the discretized volume at particle p.

The external force at grid point i, ($\mathbf{f}_i^{ext}$), includes contributions from the particle body force ($\mathbf{b}_p$), and particle traction force ($\boldsymbol{\tau}_p$):

$$\mathbf{f}_i^{ext} = \sum_p S_{ip}(\mathbf{x}_p)\mathbf{b}_p + \sum_p S_{ip}(\mathbf{x}_p)\boldsymbol{\tau}_p \tag{3}$$

At the grid, the equations of motion are:

$$\mathbf{m}\,\mathbf{a} = \mathbf{f}^{ext} - \mathbf{f}^{int} + \mathbf{f}^{sur} \tag{4}$$

where $\mathbf{m}$ is the mass matrix, $\mathbf{a}$ is the acceleration vector, $\mathbf{f}^{ext}$ is the external force vector, $\mathbf{f}^{int}$ is the internal force vector, and $\mathbf{f}^{sur}$ represents the force contributed by surface tension when present.

Here, except for the surface tension force $\mathbf{f}^{sur}$, all the terms in Eq. 4 are already explicitly formulated. The acceleration at grid point i can be calculated by:

$$\mathbf{a}_i = (\mathbf{f}_i^{ext} - \mathbf{f}_i^{int} + \mathbf{f}_i^{sur})/m_i \tag{5}$$

The formulation of the surface tension force $\mathbf{f}_i^{sur}$ will be presented shortly in Section 3.2. Once the grid acceleration is solved, the grid velocity $\mathbf{v}_i$ can be calculated via conservation of momentum:

$$\mathbf{v}_i = \frac{\sum_p S_{ip}(\mathbf{x}_p)\mathbf{v}_p m_p}{m_i} \tag{6}$$

After the grid velocity is calculated, one needs to update the state of each particle using $\mathbf{a}_i$ and $\mathbf{v}_i$. For a given time step Δt, the increment of the particle velocity, $\Delta \mathbf{v}_p$, is calculated from:

$$\Delta \mathbf{v}_p = \Delta t \sum_i S_{ip}(\mathbf{x}_i)\mathbf{a}_i \tag{7}$$

where $\sum\limits_i$ denotes summation over the grid points. It should be noted that the same interpolation function is used for the mapping from grid to particles as well as from particles to grid. The increment in the particle position, $\Delta \mathbf{x}_p$, is updated from $\mathbf{v}_i$ and $\mathbf{a}_i$:

$$\mathbf{v}_i^L = \mathbf{v}_i + \mathbf{a}_i \Delta t \tag{8}$$

$$\Delta \mathbf{x}_p = \sum_i S_{ip} \mathbf{v}_i^L \tag{9}$$

where $\mathbf{v}_i^L$ is the updated grid velocity. Particle position and velocity are then updated using Eq. 7 and Eq. 9.

The increment in the stress at particles, $\Delta \boldsymbol{\sigma}_p$, is updated using the update-stress-last (USL) algorithm [Wallstedt and Guilkey (2008)]:

$$\Delta \mathbf{F}_p = \mathbf{I} + \Delta t \sum_i \mathbf{G}_{ip} \mathbf{v}_i^L \tag{10}$$

$$\Delta \boldsymbol{\sigma} = f(\Delta \mathbf{F}_p) \tag{11}$$

where $\Delta \mathbf{F}_p$ is the increment of the deformation tensor, $\mathbf{I}$ is the identity tensor, and $f(\Delta \mathbf{F}_p)$ is a function associated with the constitutive law of the material.

For nearly-incompressible fluids of interest in this paper, we adopt the constitutive law given in [Monaghan, Cas, Kos, and Hallworth (1999); Cueto-Felgueroso, I. Colominas, Navarrina, and Casteleiro (2004)] to calculate the stress:

$$\boldsymbol{\sigma} = -p\mathbf{I} + 2\mu \mathbf{d}' \tag{12}$$

where μ is the dynamic viscosity, $\mathbf{d}'$ is the rate of deformation tensor based on $\Delta\mathbf{F}_p$, and p is the hydrostatic pressure determined by an equation of state:

$$p = K\left[\left(\frac{\rho}{\rho_0}\right)^{\lambda} - 1\right] \tag{13}$$

where K is the bulk modulus, ρ the fluid density, ρ_0 the initial density, and λ a constant. This completes the computation of a typical time step.

3 Modeling of Surface Tension

Expression of the surface tension force $\mathbf{f}_i^{sur}$ in Eq. 5 is elaborated in this section. We adopt the CSF model [Brackbill, Kothe, and Zemach (1992)] for the surface tension force since it is capable of handling complicated immiscible interfaces in the presence of contact angle. In the following, the CSF method is first reviewed, an approach is then proposed to include surface tension in GIMP.

3.1 Review of the CSF method

Fig. 2 illustrates how the CSF method models the surface tension between two immiscible fluids. The fluids are represented by different color functions (c_1 and c_2) (e.g., 0 and 1). Although color functions should be discontinuous at the interface in reality, a transition zone is used in the CSF method to bridge the discontinuity beween c_1 and c_2 using the color function $c(\mathbf{x})$ over the transition zone (i.e., the

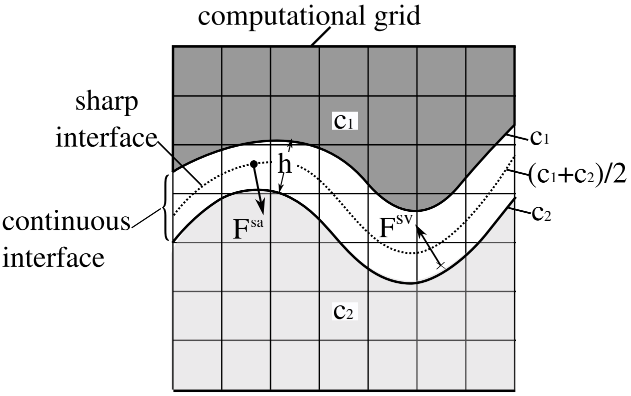

Figure 2: The CSF method for fluids with different colors (c_1 and c_2). Over the computational grid, contours of the continuous color function $c(\mathbf{x})$ are illustrated showing also a transition zone of width h. Surface tension force, $\mathbf{F}^{sa}$, is reformulated as a volume force $\mathbf{F}^{sv}$ within the transition zone.

continuous interface labeled in Fig. 2). $c(\mathbf{x})$ will converge to a step function as the width of the transition zone, h, approaches zero.

In the CSF method, the surface tension is modeled approximately as a body force or volume force, as opposed to its physical meaning by which the surface tension is force per length. The volume force, denoted by $\mathbf{F}^{sv}$, acts within the transition zone, shown in Fig. 2, in a continuous manner [Brackbill, Kothe, and Zemach (1992)] :

$$\mathbf{F}^{sv}(\mathbf{x}) = \gamma\kappa(\mathbf{x})\frac{\nabla c(\mathbf{x})}{[c]} \tag{14}$$

where γ is the coefficient of surface tension, $\nabla c(\mathbf{x})$ is the gradient of the color function, and $[c]$ is the jump of color function across the fluid interface ($[c] = |c_1 - c_2|$). κ represents the curvature evaluated at $\mathbf{x}$ at the interface; in computation, however, this curvature term is approximated by the gradient of the unit vector $\mathbf{n}$ at $\mathbf{x}$:

$$\kappa(\mathbf{x}) = \nabla\cdot\mathbf{n}(\mathbf{x}) = \nabla\cdot\left(\frac{\nabla c(\mathbf{x})}{|\nabla c(\mathbf{x})|}\right) \tag{15}$$

It should be noted that, in order to better approximate the curvature at the transition zone, the color function $c(\mathbf{x})$ can be further improved using a smoothed color function $\tilde{c}(\mathbf{x})$ [Brackbill, Kothe, and Zemach (1992); Sussman (2003)]:

$$\tilde{c}(\mathbf{x}) = \boldsymbol{K} * c(\mathbf{x}) = \int_{\Omega_k} c(\mathbf{x}')\boldsymbol{K}(\mathbf{x}' - \mathbf{x})d\mathbf{x}' \tag{16}$$

where Ω_k is the smoothing range of the smoothing kernel $\boldsymbol{K}$. Various smoothing kernels have been developed such as the Nordmark kernel [Nordmark (1991); Williams, Kothe, and Puckett (1999)], the B-spline kernel [Boor (1967); Brackbill, Kothe, and Zemach (1992)], and the quintic spline kernel used in SPH [Morris, Fox, and Zhu (1997); Hu and Adams (2009)]. In our study, a moving average smoothing kernel is used to be discussed in Section 3.4.

3.2 Formulation of surface tension in GIMP

We include the force attributed to surface tension as an additional external force to the equations of motion in Eq. 4. The modeling of surface tension is based on Eq. 14 together with a choice of the color function. Here we adopt the grid mass, m_i, as the color function $c(\mathbf{x})$. Therefore, the volume force due to surface tension at grid point i can be formulated as:

$$\mathbf{F}_i^{sv} = \gamma\kappa_i\frac{\nabla m_i}{[m_i]} \tag{17}$$

where each term is formulated as follows. First, ∇m_i, the gradient of grid mass, is calculated by:

$$\nabla m_i = \sum_{i'} \mathbf{G}_{ip}(\mathbf{x}_{i'}) m_{i'} \tag{18}$$

where $\sum_{i'}$ is a summation over the neighboring nodes at $\mathbf{x}_{i'}$.

Second, $[m_i]$ is defined as $(\rho_1 - \rho_2)V_c$, where ρ_1 and ρ_2 are the densities of the two immiscible fluids, respectively, and V_c is the volume of the cell of the background mesh.

Third, the curvature κ_i is calculated by:

$$\kappa_i = \nabla \cdot \mathbf{n}_i = \sum_{i'} \mathbf{G}_{ip}(\mathbf{x}_{i'}) \cdot (\mathbf{n}_{i'})^T \tag{19}$$

where T denotes the transpose of a vector. In turn, the unit normal vector used in the curvature calculation is based on a smoothed grid mass $\tilde{m}_i$:

$$\mathbf{n}_i = \left(\frac{\nabla \tilde{m}_i}{|\nabla \tilde{m}_i|} \right) \tag{20}$$

where $\nabla \tilde{m}_i$ is evaluated using Eq. 18. In Section 3.4 we discuss the application of moving average to smooth the mass at each grid point.

Once the volume force is calculated, the force contributed by surface tension, $\mathbf{f}^{sur}$, in the equations of motion in Eq. 4 can be calculated by:

$$\mathbf{f}^{sur} = V_c \mathbf{F}_i^{sv} \tag{21}$$

It should be noted that Eq. 17 can be further improved [Brackbill, Kothe, and Zemach (1992)] by:

$$\mathbf{F}_i^{sv} = \gamma \kappa_i \frac{\nabla m_i}{[m_i]} \frac{m_i}{< m_i >} \tag{22}$$

where $< m_i >= V_c(\rho_1 + \rho_2)/2$. Eq. 22 provides additional computational advantage in calculating the acceleration using GIMP:

$$\mathbf{a}_i^{sur} = \frac{V_c \mathbf{F}_i^{sv}}{m_i} = \gamma \kappa_i \frac{\nabla m_i}{(\rho_2 - \rho_1) < m >} \tag{23}$$

where $\mathbf{a}_i^{sur}$, the acceleration at grid point i due to surface tension, depends solely on the gradient of grid mass.

3.3 Treatment of contact angle as a boundary condition

In wetting problems, the balance in surface tension among solid, liquid, and gas results in the formation of the contact angle whose orientation is defined by the tagent of the surface profile at the point where the three phases meet. To expand GIMP's capability of handling the three-phase problem, we model the contact angle as a boundary condition illustrated in Fig. 3.

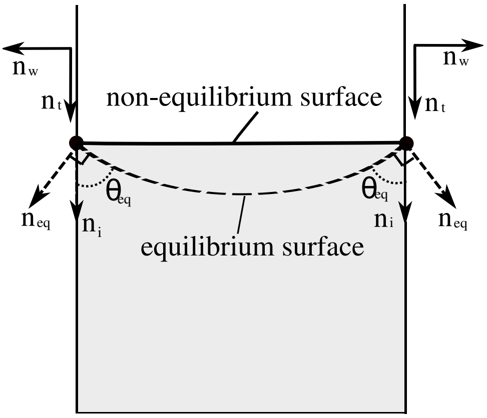

Figure 3: Contact angle θ_{eq} as a boundary condition. $\mathbf{n}_{eq}$ is the unit vector normal to the tagent along θ_{eq}; $\mathbf{n}_t$ is a unit vector tangent to the wall; $\mathbf{n}_w$ is a unit vector normal to the wall; and $\mathbf{n}_i$ is the unit normal vector along the liquid-wall contact line. During simulation, the profile of the liquid in equilibrium can be determined when $\mathbf{n}_i$ is aligned with $\mathbf{n}_{eq}$.

At the boundary where the wall interacts with two fluids (i.e., the liquid and gas), the unit normal calculated earlier in Eq. 20 is replaced by $\mathbf{n}_{eq}$:

$$\mathbf{n}_{eq} = \mathbf{n}_w \cos\theta_{eq} + \mathbf{n}_t \sin\theta_{eq} \tag{24}$$

where $\mathbf{n}_w$ and $\mathbf{n}_t$ are normal and tangent to the wall, respectively (Fig. 3). The replacement of $\mathbf{n}_i$ by $\mathbf{n}_{eq}$ in Eq. 20 for all the grid points at the boundary (where the meniscus is) will force the liquid profile to comply with a given contact angle. Consequently, the curvature of the liquid profile at the boundary will be updated incrementally during simulation, eventually reaching equilibrium when $\mathbf{n}_i$ aligns with $\mathbf{n}_{eq}$.

3.4 Implementation of GIMP-CSF

Fig. 4 outlines the main procedures discussed previously for our implementation

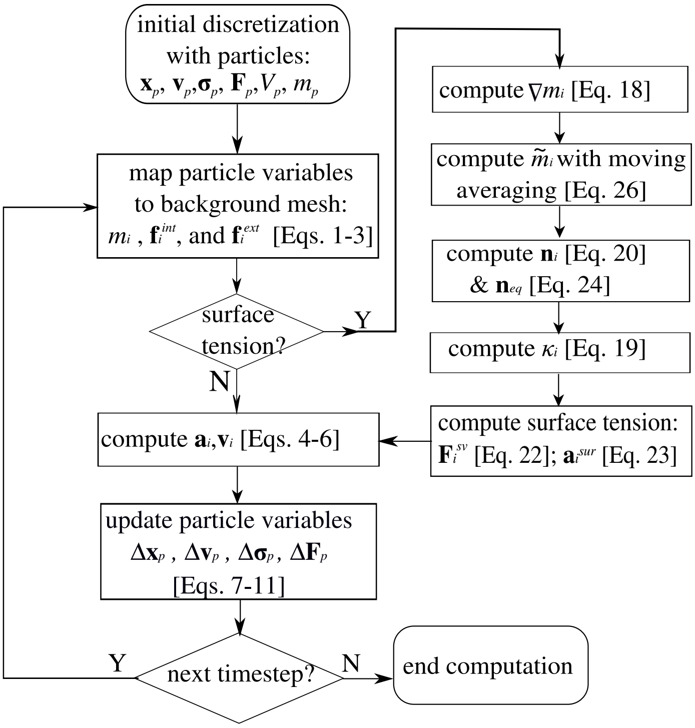
initial discretization with particles: $\mathbf{x}_p$, $\mathbf{v}_p$, $\boldsymbol{\sigma}_p$, $\mathbf{F}_p$, V_p, m_p
map particle variables to background mesh: m_i , $\mathbf{f}_i^{int}$, and $\mathbf{f}_i^{ext}$ [Eqs. 1-3]
surface tension?
Y
N
compute ∇m_i [Eq. 18]
compute $\tilde{m}_i$ with moving averaging [Eq. 26]
compute $\mathbf{n}_i$ [Eq. 20] & $\mathbf{n}_{eq}$ [Eq. 24]
compute κ_i [Eq. 19]
compute surface tension: $\mathbf{F}_i^{sv}$ [Eq. 22]; $\mathbf{a}_i^{sur}$ [Eq. 23]
compute $\mathbf{a}_i$,$\mathbf{v}_i$ [Eqs. 4-6]
update particle variables $\Delta\mathbf{x}_p$, $\Delta\mathbf{v}_p$, $\Delta\boldsymbol{\sigma}_p$, $\Delta\mathbf{F}_p$ [Eqs. 7-11]
next timestep?
Y
N
end computation

Figure 4: A flow chart for GIMP-CSF. ∇m_i–gradient of grid mass; $\tilde{m}_i$–smoothed grid mass; $\mathbf{n}_i$–grid surface normal; $\mathbf{n}_{eq}$–surface normal based on contact angle; κ_i–grid curvature; $\mathbf{F}_i^{sv}$–grid surface tension (volume) force; $\mathbf{a}_i^{sur}$–grid surface tension (acceleration) force.

of surface tension in MPM. The algorithm is named GIMP-CSF since it integrates the CSF model of surface tension with the GIMP method. GIMP-CSF is implemented in Uintah [Parker, Guilkey, and Harman (2006)], a comprehensive software package that includes the GIMP method. Our contribution to the improvement of Uintah is threefold: calculation of the gradient of grid mass, smoothing of the grid mass, and evaluation of the surface curvature; each aspect is presented below.

For illustration, we consider a 3D domain in Cartesian space meshed uniformly into $L \times M \times N$ cells in the x, y, and z direction, respectively. Each cell has volume $V_c = \Delta x \Delta y \Delta z$, where Δx, Δy, and Δz are the cell sizes. The position of a grid node is represented by $\mathbf{x}(i,j,k) = (i\Delta x, j\Delta y, k\Delta z)$, where $1 \leq i \leq L$, $1 \leq j \leq M$, and $1 \leq k \leq N$.

The gradient of the grid mass for a grid node at $\mathbf{x}_{i,j,k}$ can be calculated by:

$$\nabla m_{i,j,k} = \sum_{i',j',k'} \mathbf{G}_{ip}(\mathbf{x}_{i',j',k'})\, m_{i',j',k'} \tag{25}$$

where $m_{i',j',k'}$ is the grid mass at the neighboring node $\mathbf{x}_{i',j',k'}$. With the GIMP interpolation function, there are 27 neighboring nodes where the gradient of interpolation function $\mathbf{G}_{ip}$ is evaluated at each node.

Smoothing of the grid mass is conducted by using moving average as the smoothing kernel (cf. Eq. 16):

$$\tilde{m}_{i,j,k} = \frac{1}{n^3} \sum m_{i',j',k'} \tag{26}$$

where n is the number of neighboring nodes in each coordinate direction; the summation is over neighboring nodes; $\tilde{m}^x_{i,j,k}$, and $\tilde{m}^y_{i,j,k}$, and $\tilde{m}^z_{i,j,k}$ are the smoothed grid mass along each direction. We did not apply the smoothing average kernel at or near the boundaries since there are not sufficient neighboring nodes for Eq. 26. Further smoothing can be done by conducting more iterations of the smoothing process by substituting the left-hand side of Eq. 26 to its right-hand side.

With $\tilde{m}_{i,j,k}$ found, the unit normals are obtained by:

$$\mathbf{n}_{i,j,k} = \frac{\nabla \tilde{m}_{i,j,k}}{|\nabla \tilde{m}_{i,j,k}|} \tag{27}$$

where $\nabla \tilde{m}$ is calculated similar to ∇m in Eq. 25. It should be noted that if contact angle is specified, the boundary conditions described in Section 3.3 will be applied to modify $\mathbf{n}_{i,j,k}$.

The curvature is evaluated via:

$$\begin{aligned} \kappa_{i,j,k} &= \nabla \cdot \mathbf{n}_{i,j,k} \\ &= \sum_{i',j',k'} G_{ip} \cdot \mathbf{n}^T_{i',j',k'} \end{aligned} \tag{28}$$

4 Numerical Examples

Three numerical examples are given in this section to assess the efficacy of the GIMP-CSF algorithm implemented in Uintah. These examples consider only single-phase fluids with surface tension, applicable to situations when the effect of one phase on the other is negligible [Blanchette and Bigioni (2009); Zhang (2010)]. Although the algorithm is also capable of simulating surface tension problems in multi-phase fluids, they are beyond the scope of this paper.

We used the fluid properties listed in Tab. 1 and GIMP interpolation function given in Bardenhagen and Kober (2004) for all the examples below.

Table 1: Fluid properties used in the numerical examples

Surface tension (dynes/cm)	Density (g/cm^3)	Bulk modulus (dynes/cm^2)	Dynamic viscosity (dynes $\cdot$ s/cm^2)	λ
2.4	1	1.5 $\times 10^5$	0.5	7.0

4.1 2D equilibrium droplet

A droplet in equilibrium experiences a pressure drop, p_{drop}, across the interface due to surface tension. The theoretical solutions for p_{drop} are

$$p_{drop} = \gamma\kappa = \gamma/R, \text{ for 2D} \tag{29}$$

$$p_{drop} = 2\gamma\kappa = 2\gamma/R, \text{ for 3D} \tag{30}$$

where γ is the surface tension coefficient, and R the radius of the droplet. In this example, we simulate the 2D case with GIMP-CSF and compare the results with those obtained by a different approach [Brackbill, Kothe, and Zemach (1992)].

A 1-cm-radius droplet in the x–y plane is simulated with a typical set-up shown in Fig. 5. The droplet has an initial non-circular shape and is discretized into 1264 particles; the initial shape was chosen to be close to a circle to save computational time. The problem domain is overlaid with a 4×4 cm grid with 40×40 cells; there

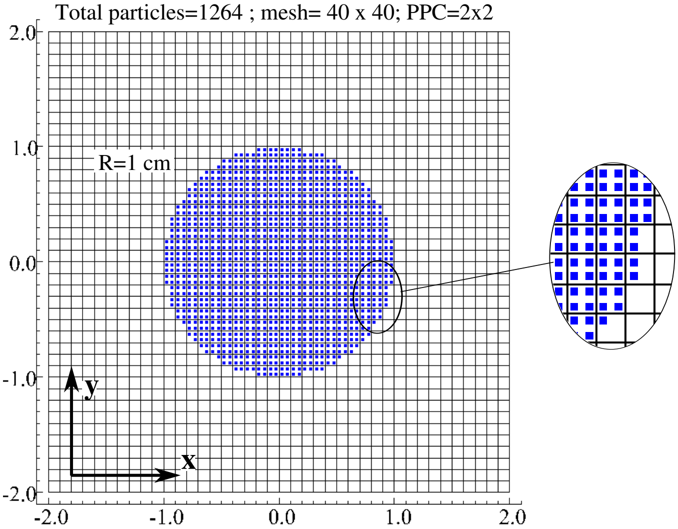

Figure 5: Example set-up of 2D equilibrium drop with GIMP using Uintah.

are 2 particles per cell in the x and y directions, respectively (PPC=2 × 2). In the z direction, there is only one layer of particles and cells for this 2D case. Besides the set-up in Fig. 5, various numbers of PPC and cells were also investigated.

As an example, Fig. 6 shows the contours of grid mass and its gradient (∇m) for

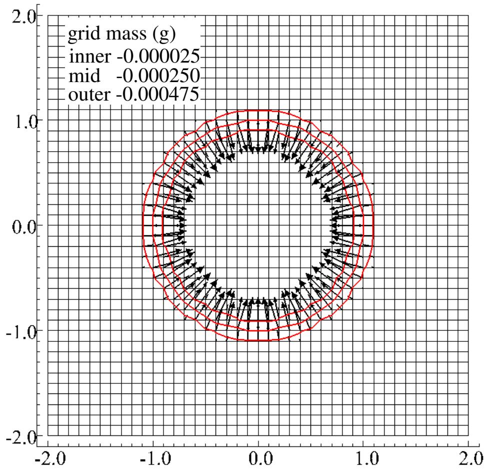

Figure 6: Contours of the grid mass and the gradients at t=0 s.

the situation in Fig. 5 at t=0 s. The grid mass is mapped from the particle mass by Eq. 1, and its gradient calculated by Eq. 18. The contours of the grid mass define a transition zone which is about two cells wide. Only within the transition zone are the gradients nonzero, so is the surface tension force. Thus there is no need for the CSF model to track the evolution of the boundary. Since the surface tension force is zero when ∇m is zero, it can save computational time by skipping the calculation of the curvature and surface tension when the gradient ∇m is smaller than a threshold (e.g., 10 % of $[m]/\Delta x$). Fig. 7 shows the curvatures evaluated at the nodes when the gradient ∇m is greater than the predefined threshold. It shows the influence of different smoothing schemes on the curvature. Without smoothing, Fig. 7a shows that the curvature has a large error compared to the theoretical value of 1 cm^{-1} at the boundary of the droplet. With 1 iteration of moving average smoothing, the curvature (Fig. 7b) is improved. More iterations of moving average smoothing can greatly improve the accuracy of the curvature, as shown in Fig. 7c and Fig. 7d. Similar results were also found in [Brackbill, Kothe, and Zemach (1992)].

Next we plot the surface tension in terms of acceleration $\mathbf{a}^{sur}$ (cf. Eq. 23) in Fig. 8, comparing the result of 1 iteration of the moving average smoothing with that of 4. The directions of surface tension at the grid points in Fig. 8a and Fig. 8b are

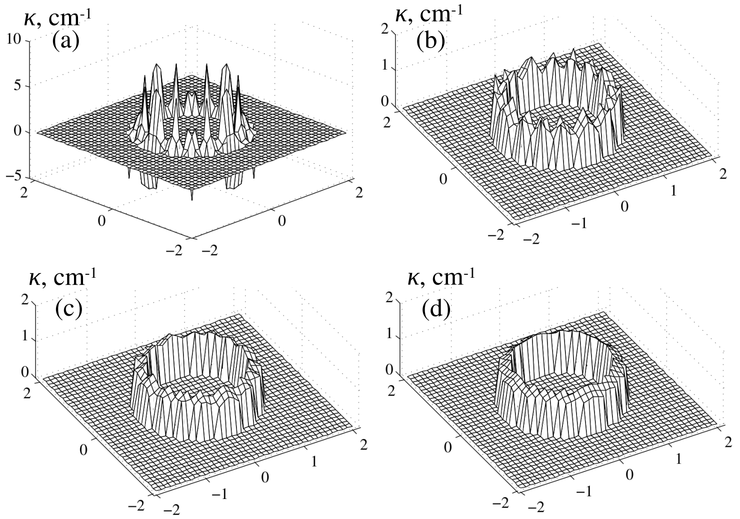

Figure 7: Curvatures (t=0) with different smoothed grid mass: (a) zero iteration of moving averaging; (b) 1 iteration of moving averaging ; (c) 2 iterations of moving averaging; (d) 4 iterations of moving averaging. Theoretical curvature is 1 cm^{-1}.

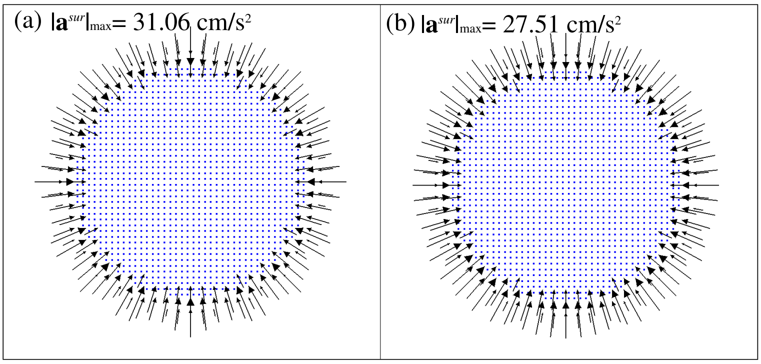

Figure 8: Surface tension, $\mathbf{a}^{sur}$, at t=0 with different curvatures from (a) one iteration of moving averaging; (b) 4 iterations of moving averaging.

the same since they are based on the gradient of unsmoothed grid mass (Fig. 6). However, the magnitude of the surface tension for each case is different since it is determined by the curvature which, in turn, is affected by the smoothing scheme used.

In a dynamic simulation, the droplet oscillates and will reach an equilibrium state. Given the initial configuration, we found that 0.25 second is sufficient for the droplet to stabilize. The grid pressure is then obtained by mapping the particle stress to the background mesh. Fig. 9 shows the distribution of pressure for different numbers of smoothing iterations. Accuracy of the calculated pressure can be assessed by the following measures [Brackbill, Kothe, and Zemach (1992)]:

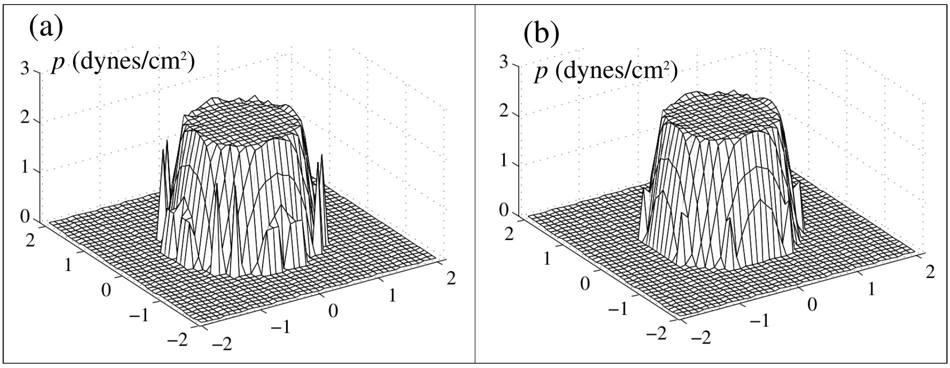

Figure 9: Pressure at t=2.5 s from different smoothing schemes: (a) one iteration of moving averaging; (b) 4 iterations of moving averaging. Theoretical pressure drop is 2.4 dynes/cm^2.

$$< p >= \frac{1}{N_d} \sum_{i,j=1}^{N_d} p_{i,j} \tag{31}$$

$$L_2 = \left[\frac{\sum_{i,j=1}^{N_d} (p_{i,j} - p_{drop})^2}{N_d p_{drop}^2} \right] \tag{32}$$

where <p> is the average pressure, N_d is the number of grid nodes within the droplet radius, i and j are node indices, and L_2 is the root-mean-square (rms) error relative to the theoretical pressure drop p_{drop} which is 2.4 dyne/cm^2 for our case (Eq. 29). With Eq. 31 and Eq. 32, we have, for Fig. 9a, <p>/p_{drop} = 98.36% and L_2= 0.043; for Fig. 9b, we have <p>/p_{drop} = 98.16% and $L_2 = 0.043$. It shows that the results in pressure in Fig. 9 are very close such that the number of smoothing iterations has little effect on the distribution of pressure.

Tab. 2 shows the effects of grid resolution, number of particles per cell, and smoothing schemes. Generally, the GIMP-CSF method with smoothing has the same order of accuracy (less 5% error in <p>) as those in Brackbill, Kothe, and Zemach (1992). Smoothing is shown to be very important; when not used, the error could be as high as 125.7%. However, smoothing once is adequate for this example as smoothing more times does not change the results significantly.

Table 2: Comparison of results using GIMP-CSF to those using the arbitrary Lagrangian-Eulerian (ALE) method for simulating a 2D equilibrium droplet. (Δh is the mesh size; PPC–particles per cell.)

<table>
<tr><th colspan="5">ALE [Brackbill, Kothe, and Zemach (1992)]</th></tr>
<tr><th>$R/\Delta h$</th><th># of particles (PPC)</th><th>smoothing</th><th>$< p > /p_{drop}$</th><th>rms error (Eq. 32)</th></tr>
<tr><td>10</td><td>/</td><td>B-spline</td><td>1.034</td><td>0.056</td></tr>
<tr><td>20</td><td>/</td><td>B-spline</td><td>1.016</td><td>0.028</td></tr>
<tr><td colspan="5">GIMP-CSF</td></tr>
<tr><td>10</td><td>1264 (2×2)</td><td>none</td><td>1.641</td><td>0.655</td></tr>
<tr><td>10</td><td>1264 (2×2)</td><td>1 moving averaging</td><td>0.983</td><td>0.043</td></tr>
<tr><td>10</td><td>1264 (2×2)</td><td>2 moving averaging</td><td>0.982</td><td>0.042</td></tr>
<tr><td>10</td><td>1264 (2×2)</td><td>4 moving averaging</td><td>0.984</td><td>0.043</td></tr>
<tr><td>10</td><td>5024 (4×4)</td><td>none</td><td>1.212</td><td>0.279</td></tr>
<tr><td>10</td><td>5024 (4×4)</td><td>1 moving averaging</td><td>0.984</td><td>0.046</td></tr>
<tr><td>10</td><td>5024 (4×4)</td><td>2 moving averaging</td><td>0.992</td><td>0.045</td></tr>
<tr><td>10</td><td>5024 (4×4)</td><td>4 moving averaging</td><td>0.991</td><td>0.045</td></tr>
<tr><td>20</td><td>5024 (2×2)</td><td>none</td><td>2.257</td><td>1.286</td></tr>
<tr><td>20</td><td>5024 (2×2)</td><td>1 moving averaging</td><td>0.967</td><td>0.040</td></tr>
<tr><td>20</td><td>5024 (2×2)</td><td>2 moving averaging</td><td>0.980</td><td>0.031</td></tr>
<tr><td>20</td><td>5024 (2×2)</td><td>4 moving averaging</td><td>0.978</td><td>0.033</td></tr>
<tr><td>20</td><td>20108 (4×4)</td><td>none</td><td>1.941</td><td>0.971</td></tr>
<tr><td>20</td><td>20108 (4×4)</td><td>1 moving averaging</td><td>0.973</td><td>0.035</td></tr>
<tr><td>20</td><td>20108 (4×4)</td><td>2 moving averaging</td><td>0.989</td><td>0.026</td></tr>
<tr><td>20</td><td>20108 (4×4)</td><td>4 moving averaging</td><td>0.987</td><td>0.027</td></tr>
</table>

In terms of the effect of grid resolution, results in Tab. 2 indicate that a cell size (Δh) 10% of the droplet radius ($R/\Delta h$ = 10) is adequate since the results are comparable to those with $R/\Delta h$ = 20. Regarding the effect of the number of particles per cell, PPC=4×4 gives slightly better results than PPC=2×2. For computational efficiency, we recommend a PPC= 2×2 (or 2×2×2 for 3D).

4.2 Evolution of non-equilibrium drops in 2D and 3D

In this example, we consider the evolution of a droplet from a square in 2D (or a cubic in 3D) to the circular (spherical) profile at equilibrium under surface tension. This example has been studied by other numerical methods [Brackbill, Kothe, and Zemach (1992); Zhang (2010)] and serves as another benchmark problem for our GIMP-CSF algorithm.

We first consider the 2D case where a square drop (1 × 1 cm) is simulated shown in Fig. 10. The drop is discretized with 400 particles and overlaid with a 2 × 2

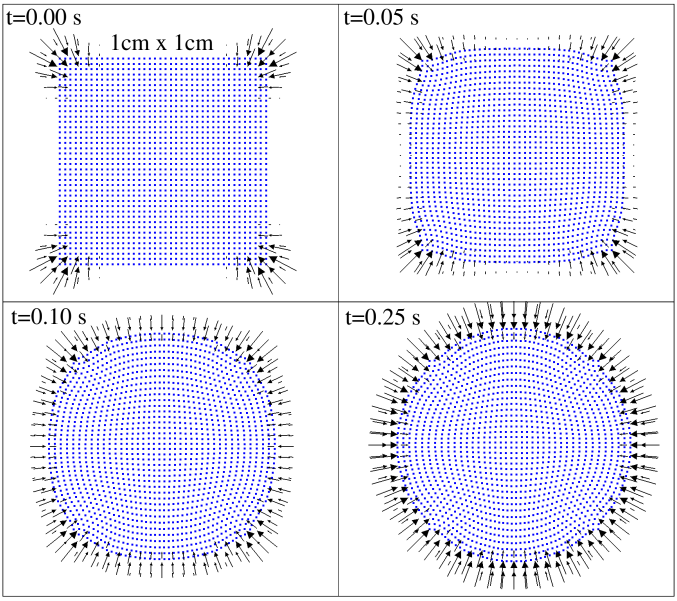

Figure 10: A 2D 1 × 1 cm square drop evolves into a circular shape under surface tension using two iterations of moving average smoothing. The arrows represent the surface tension force.

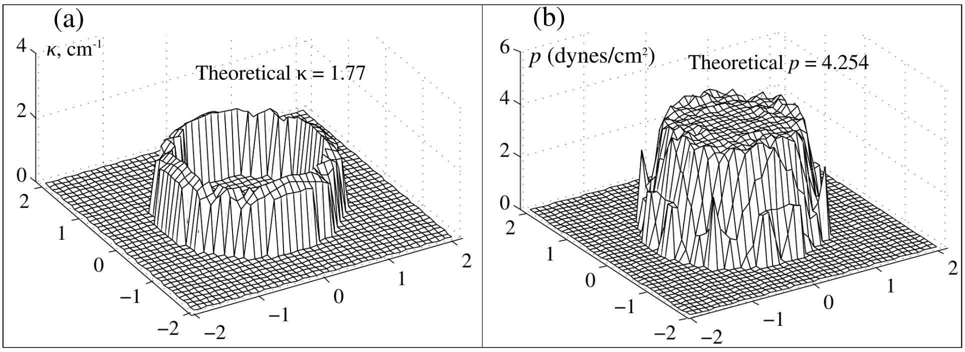

Figure 11: Distribution of the calculated curvature (a), and pressure (b) after a 2D non-equilibrium drop becomes stable at time = 0.25 s. The average pressure is 4.09 (or 96.1% of the theoretical value), with an rms error $L_2 = 0.044$.

cm background grid of 40 × 40 cells with a PPC of 2×2. Similar to the previous example, there is a single layer of particles and cells in the z direction. Initially, the surface tension forces, indicated by the arrows, are concentrated at the corners where the curvatures are very large; these forces cause the drop to evolve reaching an equilibrium state at t = 0.25 s. The theoretical equilibrium state is circular with an area equal to that of the initial square, 1 cm^2, which gives a theoretical radius of 0.56 cm, and a curvature of 1.77 cm^{-1}; the pressure drop across the interface is 4.25 dynes/cm^2. Fig. 11 shows the calculated curvature and pressure at t=0.25 s when equilibrium is reached; the average pressure <p> is 4.09 dynes/cm^2 (or 96.1% of the theoretical value) with an rms error L_2=0.044. A similar example was studied in Brackbill, Kothe, and Zemach (1992) but the equilibrium state was not shown.

The same problem of a non-equilibrium drop evolving to its equilibrium state is simulated in 3D shown in Fig. 12. In this simulation, the drop is initialized as a

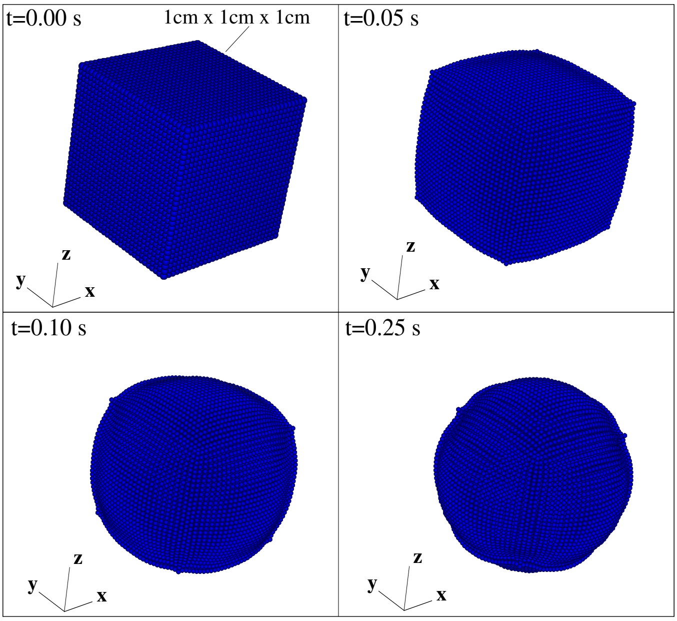

Figure 12: Evolution of a 3D cubic drop into a sphere under surface tension using two iterations of moving average smoothing.

1 × 1 × 1–cm cube placed over a 2 × 2 × 2–cm background mesh. The drop is discretized into a total of 27000 particles and the background mesh has 30 × 30 × 30 cells; the PPC is 2×2×2. The theoretical radius of the final sphere is 0.62 cm corresponding to a curvature of 1.61 cm^{-1}, and a pressure drop of 7.74

dynes/cm^2 (cf. Eq. 30). The calculated curvature and pressure on a plane through the origin are shown in Fig. 13; the average pressure is 8.05 dyn/cm^2 (or 104.1 % of the theoretical value) with an rms error of 0.0451. The results suggest that the GIMP-CSF algorithm is capable of handling 3D droplet simulation with good accuracy.

In summary, the proposed GIMP-CSF algorithm can produce comparable results as benchmarked by examples available in the literature or with known theoretical solutions for a drop evolving to its equilibrium state in 2D or 3D.

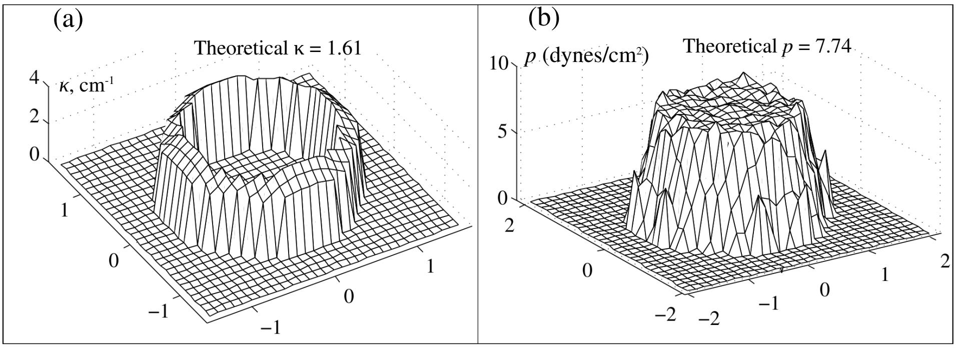

Figure 13: Distribution of the calculated curvature (a), and pressure (b) on a plane through the origin of a 3D drop. The average pressure is 8.05 (or 104.1% of the theoretical value) with an rms error $L_2 = 0.0451$.

4.3 Wall adhesion and capillary rise

The last example demonstrates the capability of the GIMP-CSF algorithm in problems involving contact angles and the gravitational force. The contact angle problem was modeled in Brackbill, Kothe, and Zemach (1992) using the CSF method but without quantitative comparisons with theoretical results. In this example, we simulate the problem of half-plane capillary rise and compare the results with theoretical solutions.

The problem is shown schematically in Fig. 14 where a contact angle θ_{eq} is specified. The meniscus profile can be described by a closed-form solution derived from the work in Anderson, Bassom, and Fowkes (2006):

$$\xi = \cosh^{-1}\left(\frac{2}{\zeta}\right) - \sqrt{4-\zeta^2} + \sqrt{2(1-\sin\theta)} - \cosh^{-1}\left(\frac{\sqrt{2}}{\sqrt{1-\sin\theta}}\right) + \sqrt{2+2\sin\theta} \tag{33}$$

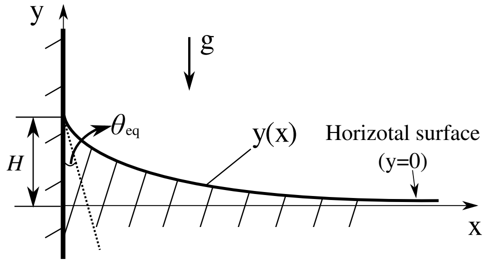

Figure 14: Schematic of half-plane liquid meniscus $y(x)$ under gravitational pull. The liquid is at rest in contact with a hydrophilic wall with a given contact angle θ_{eq}. H is the capillary rise to be simulated.

where the two dimensionless parameters are $\xi = x/\sqrt{\gamma/\rho g}$, $\zeta = y/\sqrt{\gamma/\rho g}$; x is distance measured from the wall, and g the gravitational constant (981 cm/s^2 in the negative-y direction). The capillary rise H is measured from the horizontal surface at $y = 0$ shown in Fig. 14.

The initial set-up of the capillary-rise simulation is shown in Fig. 15 where a 2D tank (0.5 cm by 0.3 cm) is partially filled with a 0.20-cm-deep fluid. A $75 \times 45 \times 1$ background mesh with 6750 particles are used (PPC=2×2). It should be noted that the normal vector to the initial meniscus, as shown in Fig. 15, points to the negative $y-$direction at all the grid nodes lying on the meniscus except at the boundaries which join the walls. At the two side boundaries, the vectors are set in the direction defined by the contact angle θ_{eq} as a boundary condition (cf. Section 3.3). Such a boundary condition results in a net surface tension force which drives the fluid upwards along the wall, while under the influence of gravitational pull, until an equilibrium state is established.

The evolution of the meniscus profile is shown in Fig. 16 with a contact angle gradually approaching the specified θ_{eq} of 30^o. The simulation is terminated when the contact angle is equal to the specified 30^o, shown in Fig. 16, at t=0.1 s. The profile of the meniscus is obtained using particle positions.

We compare the simulation results for two different θ_{eq} values, 30^o and 60 ^{o}, with the theoretical solutions (Eq. 33) in Fig. 17. The agreement with the theoretical solutions for various θ_{eq} is good suggesting that the proposed GIMP-CSF algorithm is capable of handling problems with wetting or capillary rise such as liquid extraction in oil recovery or capillary driven flows.

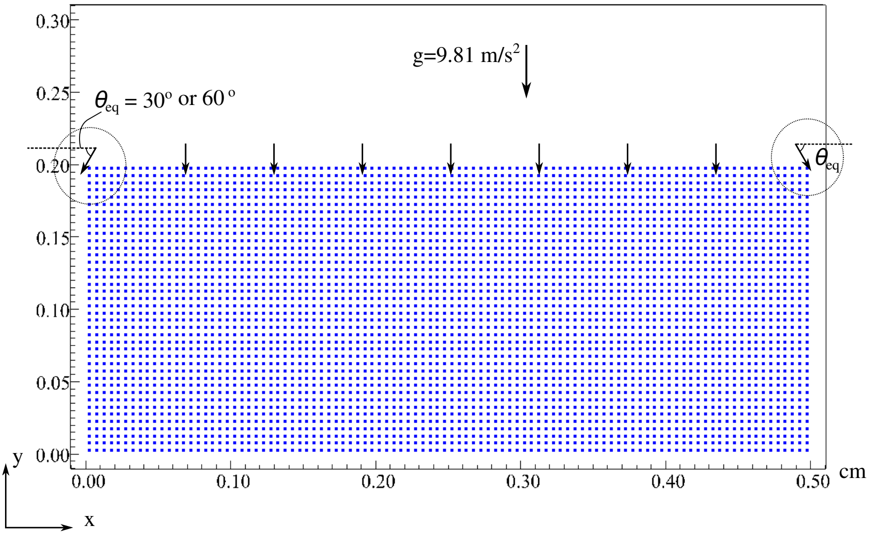

Figure 15: Set-up for the simulation of the capillary rise. The contact angle θ_{eq} is prescribed at the wall boundary; the surface profile will evolve toward its equilibrium state under surface tension.

vskip1em

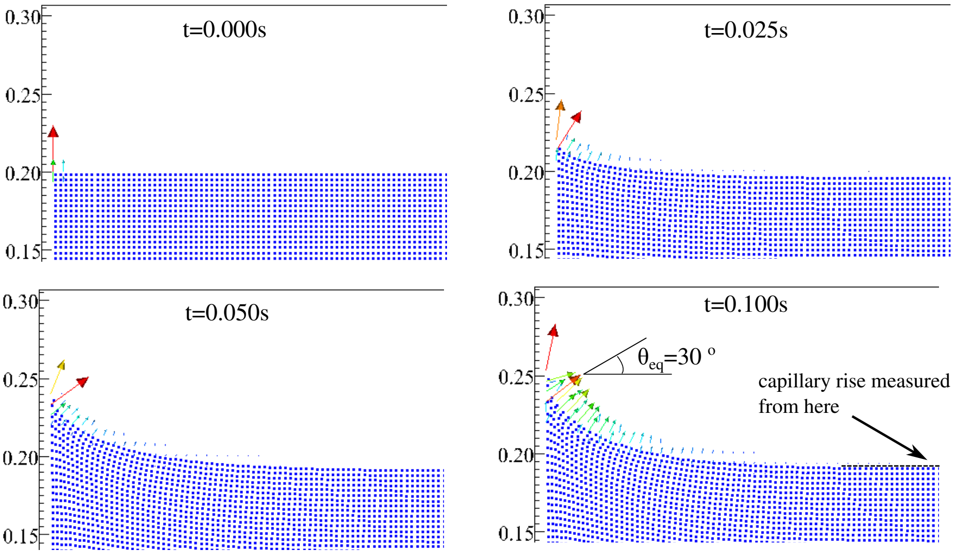

Figure 16: Evolution of the surface profile due to surface tension. Arrows indicate the surface tension force which are in equilibrium with gravitational pull at t=0.1 s ($\theta_{eq} = 30^o$).

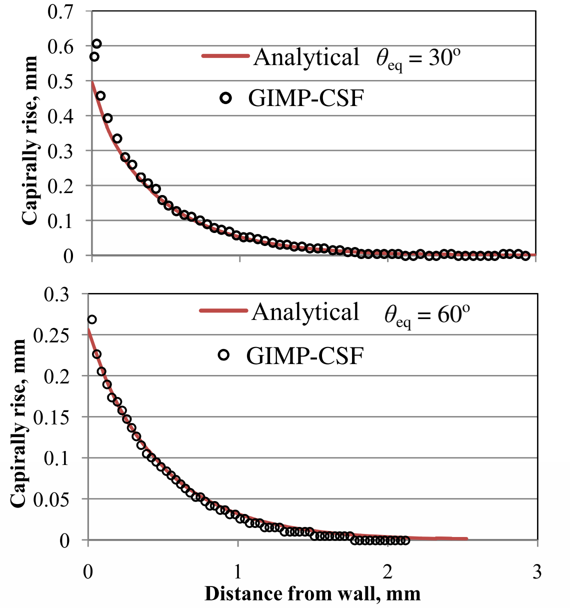

Figure 17: Comparison of the calculated capillary rise at t=0.1 s to theoretical solutions for different contact angles (30^o and 60^0).

5 Discussions and Conclusion

Surface tension exists at the interface between two immiscible fluids and becomes significant at small spatial scales. In this paper, we presented a new particle-based algorithm, GIMP-CSF, to model surface tension and capillary rise in the presence of contact angles. The contribution of this paper is in the adaptation of the CSF surface tension model for the GIMP method. One key aspect of the algorithm is about calculating and adding the surface tension force to the equations of motion at the nodes of the background mesh within the framework of the GIMP method. The algorithm used a moving average smoothing scheme for the color function to improve the accuracy of the curvature at the interface. The effects of smoothing, grid resolution, and number of particles per cell were also studied with practical suggestions regarding a balance between accuracy and computational efficiency.

The computational time for each example is summarized in Tab. 3 based upon the Uintah package installed on a Sun Ultra 40 computer with a 2.4 GHz CPU.

As demonstrated through the benchmarked examples, we have shown that the proposed GIMP-CSF algorithm, using a nearly-incompressible constitutive law for fluids, can be as effective and accurate as other methods in literature. This paves

Table 3: Summary of computational time. Δh is mesh size, t_p is physical time, and t_c is computational time.

Example	Mesh	Δh (cm)	# of particles	$t_p(s)$	$t_c(s)$
2D-equilibrium	40x40x1	0.1	1264	0.25	195
2D-equilibrium	40x40x1	0.1	5024	0.25	482
2D-equilibrium	80x80x1	0.05	5024	0.25	1304
2D-nonequilibrium	40x40x1	0.1	1600	0.25	571
3D-nonequilibrium	30x30x30	0.13	27000	0.25	4612
Capillary rise	75x45x1	0.0067	6750	0.1	4556

the way for applying GIMP toward more difficult problems such as fluid-structure interactions at small scales where surface tension plays an important role.

References

Anderson, M. L.; Bassom, A. P.; Fowkes, N. (2006): Exact solutions of the Laplace-Young equation. *Proceedings of the Royal Society A*, vol. 462, pp. 3645–3656.

Bardenhagen, S.; Guilkey, J.; Roessig, K.; Brackbill, J.; Witzel, W.; Foster, J. (2001): An improved contact algorithm for the material point method and application to stress propagation in granular material. *CMES: Computer Modeling in Engineering and Sciences*, vol. 2, pp. 509–522.

Bardenhagen, S.; Kober, E. (2004): The generalized interpolation material point method. *CMES: Computer Modeling in Engineering and Sciences*, vol. 5, pp. 477–495.

Blanchette, F.; Bigioni, T. P. (2009): Dynamics of drop coalescence at fluid interfaces. *J. Fluid Mech.*, vol. 620, pp. 333–352.

Boor, C. (1967): *A Practical Guide to Splines.* Springer-Verlag, New York.

Brackbill, J. U.; Kothe, D.; Zemach, C. (1992): A continuum method for modeling surface tension. *Journal of Computational Physics*, vol. 100, pp. 335–354.

Brackbill, J. U.; Ruppel, H. (1986): FLIP: a method for adatively zoned, particel-in-cell calculation of fluid flows in two dimensions. *Computer Physics Communications*, vol. 95, pp. 314–343.

Bussmann, M.; Mostaghimi, J.; Chandra, S. (1999): On a three-dimensional volume tracking model of droplet impact. *Physics of Fluids*, vol. 11, pp. 1406–1418.

Chang, Y. C.; Hou, T. Y.; Merriman, B.; Osher, S. (1996): A level set formulation of Eulerian interface capturing methods for incompressible fluid flows. *Journal of Computational Physics*, vol. 124, pp. 449–464.

Cueto-Felgueroso, L.; I. Colominas, G. M.; Navarrina, F.; Casteleiro, M. (2004): On the Galerkin formulation of the smoothed particle hydrodynamics method. *International Journal for Numerical Methods in Engineering*, vol. 60, pp. 1475–1512.

Gan, Y.; Chen, Z.; Montgomery-Smith, S. (2011): Improved material point method for simulating the zona failure response in piezo-assisted intracytoplasmic sperm injection. *CMES: Computer Modeling in Engineering and Sciences*, vol. 73, pp. 45–75.

Gueyffier, D.; Li, J.; Nadim, A.; Scardovell, R.; Zaleski, S. (1999): Volume-of-fluid interface tracking with smoothed surface stress methods for three-dimensional flows. *Journal of Computational Physics*, vol. 152, pp. 423–456.

Guilkey, J. E.; Harman, T. B.; Banerjee, B. (2007): An Eulerian-Lagrangian approach for simulating explosions of energetic devices. *Computers and Structures*, vol. 85, pp. 660–674.

Hu, X.; Adams, N. (2009): A constant-density approach for incompressible multi-phase SPH. *Journal of Computational Physics*, vol. 228, pp. 2082–2091.

Lee, J. H.; Guilkey, J. E. (2008): Grain-scale modelling of saturated soils Using the generalized interpolation material point method. In *Proceedings of the 16th International Conference of the International Society for Terrain-Vehicle Systems*.

Li, F.; Pan, J.; Sinka, C. (2011): Modelling brittle impact failure of disc particles using material point method. *International Journal of Impact Engineering*, vol. 38, pp. 653–660.

Li, S.; Liu, W. K. (2002): Meshfree and particle methods and their applications. *Applied Mechanics Review*, vol. 55, pp. 1–34.

Liu, X.-D.; Fedkiw, R.; Kang, M. (2000): A boundary condition capturing method for multiphase incompressible flow. *Journal of Scientific Computing*, vol. 160, pp. 151–178.

Monaghan, J. J.; Cas, R. A. F.; Kos, A. M.; Hallworth, M. (1999): Gravity currents descending a ramp in a stratified tank. *Journal of Fluid Mechanics*, vol. 379, pp. 39–70.

Morris, J. P. (2000): Simulating surface tension with smoothed particle hydrodynamics. *International Journal for Numerical Methods in Fluids*, vol. 33, pp. 333–353.

Morris, J. P.; Fox, P. J.; Zhu, Y. (1997): Modeling low Reynolds number incompressible flows using SPH. *Journal of Computational Physics*, vol. 136, pp. 214–226.

Nairn, J. A. (2003): Material point method calculations with explicit cracks. *CMES: Computer Modeling in Engineering and Sciences*, vol. 4, pp. 649–663.

Nordmark, H. O. (1991): Rezoning for higher order vortex methods. *Journal of Computational Physics*, vol. 97, pp. 366–397.

Nugent, S.; Posch, H. A. (2000): Liquid drops and surface tension with smoothed particle applied mechanics. *Physical Review E*, vol. 62, pp. 4968–4975.

Parker, S.; Guilkey, J.; Harman, T. (2006): A component-based parallel infrastructure for the simulation of fluid-structure interaction. *Engineering With Computers*, vol. 22, pp. 277–292.

Steffen, M.; Wallstedt, P.; Guilkey, J.; Kirby, R.; Berzins, M. (2008): Examination and analysis of implementation choices within the material point method (MPM). *CMES: Computer Modeling in Engineering and Sciences*, vol. 31, pp. 107–127.

Sulsky, D.; Chen, Z.; Schreyer, H. L. (1994): A particle method for history dependent materials. *Comput. Methods Appl. Mech. Engrg.*, vol. 118, pp. 179–196.

Sussman, M. (2003): A second order coupled level set and volume-of-fluid method for computing growth and collapse of vapor bubbles. *Journal of Computational Physics*, vol. 187, pp. 110–136.

Sussman, M.; Smereka, P.; Osher, S. (1994): A level set approach for computing solutions to incompressible two-phase flow. *Journal of Computational Physics*, vol. 114, pp. 146–159.

Wallstedt, P. C.; Guilkey, J. E. (2008): An evaluation of explicit time integration schemes for use with the generalized interpolation material point method. *Journal of Computational Physics*, vol. 227, pp. 9628–9642.

Williams, M.; Kothe, D.; Puckett, E. (1999): Convergence and accuracy of continuum surface tension models. *Fluid Dynamics at Interface*, pp. 294–305.

York, A. R.; Sulsky, D. L.; Schreyer, H. L. (1999): The material point method for simulation of thin membranes. *International Journal for Numerical Methods in Engineering*, vol. 44, pp. 1429–1456.

Zhang, M. Y. (2010): Simulation of surface tension in 2D and 3D with smoothed particle hydrodynamics method. *Journal of Computational Physics*, vol. 229, pp. 7238–7259.