Proceedings of FEDSM'03
2003 4th ASME JSME Joint Fluids Engineering Conference
July 6-11, 2003, Honolulu, Hawaii USA

FEDSM2003-45144

USING THE PARTICLE LEVEL SET METHOD AND A SECOND ORDER ACCURATE PRESSURE BOUNDARY CONDITION FOR FREE SURFACE FLOWS

Doug Enright
Department of Mathematics
University of California, Los Angeles
enright@math.ucla.edu

Duc Nguyen, Frederic Gibou, Ron Fedkiw
Department of Computer Science
Stanford University
dqnguyen@stanford.edu
fgibou@math.stanford.edu
fedkiw@cs.stanford.edu

ABSTRACT

In this paper, we present an enhanced resolution capturing method for topologically complex two and three dimensional incompressible free surface flows. The method is based upon the level set method of Osher and Sethian to represent the interface combined with two recent advances in the treatment of the interface, a second order accurate discretization of the Dirichlet pressure boundary condition at the free surface (2002, *J. Comput. Phys.* **176**, 205) and the use of massless marker particles to enhance the resolution of the interface through the use of the particle level set method (2002, *J. Comput. Phys.*, **183**, 83). Use of these methods allow for the accurate movement of the interface while at the same time preserving the mass of the the liquid, even on coarse computational grids. Also, these methods complement the level set method in its ability to handle changes in interface topology in a robust manner. Surface tension effects can be easily included in our method. The method is presented in three spatial dimensions, with numerical examples in both two and three spatial dimensions.

1 Governing Equations

We solve for inviscid, incompressible flow. The governing equations for momentum and mass conservation are

$$\vec{u}_t + \vec{u}\cdot\nabla\vec{u} + \frac{\nabla p}{\rho} = \vec{g}, \tag{1}$$

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

where t is the time, $\vec{u} = (u,v,w)$ is the velocity field, p is the pressure, $\vec{g} = (0,g,0)$ is gravity, ρ is the constant density of the liquid, and $\nabla = (\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z})$. We use the free surface assumption so that the air has a constant pressure, p_{air}, in space and time.

The location of the liquid free surface is defined as the set of points where the level set function $\phi = 0$, and the region occupied by the liquid is given by $\phi < 0$. The level set equation [1],

$$\phi_t + \vec{u}\cdot\nabla\phi = 0, \tag{3}$$

determines the evolution of the free surface in space and time. Geometrical information about the interface, e.g. normals and curvature, can be easily obtained from ϕ which we take as the signed distance to the interface. The unit outward normal is $\vec{N} = \nabla\phi/|\nabla\phi|$ and the curvature is $\kappa = \nabla\cdot\vec{N}$. See [2] for more details. In addition to the level set function, massless marker particles are passively advected along with the flow using the particle level set method proposed in [3]. The particles move according to $d\vec{x}_p/dt = \vec{u}(\vec{x}_p)$, where $\vec{u}(\vec{x}_p)$ is the fluid velocity interpolated at the particle location $\vec{x}_p$.

2 Numerical Method

A standard MAC grid is used in the discretization with $p_{i,j,k}$ and $\phi_{i,j,k}$ defined at the cell centers and $u_{i\pm1/2,j,k}$, $v_{i,j\pm1/2,k}$, and $w_{i,j,k\pm1/2}$ defined at the appropriate cell faces, see [4,5] for more details. A pressure free projection method [6, 7] is used to calculate a divergence free velocity field during an Euler time step update using a two step process. First an intermediate velocity field, $\vec{u}^*$ is calculated,

$$\frac{\vec{u}^* - \vec{u}^n}{\Delta t} + \vec{u}^n \cdot \nabla \vec{u}^n = \vec{g}, \tag{4}$$

and then the incompressibility constraint is enforced by calculating a pressure field which makes the intermediate velocity field divergence free. The pressure field is scaled by the size of the time step and the liquid density, i.e. $\hat{p} = p\Delta t/\rho$ for robustness. A Poisson equation of the form

$$\triangle \hat{p} = \nabla \cdot \vec{u}^*, \tag{5}$$

is solved for $\hat{p}$ through the use of a fast Poisson solver, e.g. a preconditioned conjugate gradient (PCG) method using a modified incomplete Cholesky preconditioner [8]. $\hat{p}$ is then used to make the intermediate velocity field divergence free,

$$\vec{u}^{n+1} - \vec{u}^* + \nabla \hat{p} = 0. \tag{6}$$

A third order TVD Runge-Kutta scheme [9] is used to update the velocity field in time. This method can be written as a convex combination of three simple Euler updates, therefore the above process is repeated three times to advance the velocity field from time level n to $n+1$. A third order accurate ENO discretization [10] of the convective derivatives in equation 4 is used, otherwise second order central differences is used in equations 5 and 6. An adaptive time stepping procedure is used to enforce an appropriate CFL condition incorporating convection, gravity, and surface tension forces as necessary. Our treatment follows that presented in [11].

2.1 Particle Level Set Method

The particle level set method is used to update the position of the interface in a mass preserving manner. The particle level set method is a thickened front tracking approach which utilizes particles to assist the level set method in accurately tracking flow characteristics in under-resolved regions and consequently preserve mass. This is achieved through the placement of massless marker particles near the interface as an error correction mechanism for the level set function.

Two sets of marker particles are randomly placed in a “thickened” surface region about the $\phi = 0$ level set. The thickness of the band used in the examples section is three grid cells on each side of the interface. *Positive* particles are located in the $\phi > 0$ region and *negative* particles in the $\phi < 0$ region. The number of particles placed in each cell can be adjusted according to the amount of surface resolution desired. For the examples presented, 16 particles per cell were used in 2D and 32 in 3D. Each particle possesses a radius, r_p, which is constrained to take on a value between a minimum and maximum value based upon the size of the underlying computational grid. A minimum radius of $.1\min(\Delta x, \Delta y, \Delta z)$ and maximum radius of $.5\min(\Delta x, \Delta y, \Delta z)$ is used. The radius of a particle changes dynamically throughout the simulation since the particle's relative location to the surface changes in time. This allows for a multiscale sampling of the interface by the particles. The radius of each particle is set according to:

$$r_p = \begin{cases} r_{max} & \text{if } s_p\phi(\vec{x}_p) > r_{max} \\ s_p\phi(\vec{x}_p) & \text{if } r_{min} \le s_p\phi(\vec{x}_p) \le r_{max} \\ r_{min} & \text{if } s_p\phi(\vec{x}_p) < r_{min}, \end{cases} \tag{7}$$

where s_p is the sign of the particle (+1 for positive particles and -1 for negative particles). This radius adjustment keeps the boundary of the particles tangent to the surface whenever possible. Also, particles are allowed to overlap in order allow for an enhanced reconstruction of the surface.

The particles are integrated forward in time using the 3rd order TVD Runge-Kutta scheme discussed previously. Simple bilinear or trilinear interpolation is used to calculate the velocity of a particle on the computational grid. The level set equation, 3, is discretized using a 5th order accurate Hamilton-Jacobi WENO method [12] on the advection terms and 3rd order accurate TVD Runge-Kutta for the time integration. After the level set function, ϕ, and both the positive and negative particles are integrated separately forward in time, the particles are used to correct any errors in the representation of the free surface according to the level set function. This particle correction mechanism is comprised of several steps discussed below. Note that we apply the error correction step after each modification of the level set.

Identification of Error Particles are able to detect errors in the level set representation of the free surface when they appear on the wrong side of the interface by more than their radius. These particles are said to have *escaped*. In smooth, well resolved regions of the flow where the level set method is highly accurate, the particles do not drift an appreciable amount across the interface, so we do not use the lower order accurate particle representation of the interface in this region. Only when the level set has clearly made an error do we resort to using the particle information to reconstruct the level set function.

Quantification of Error We associate a spherical level set function, ϕ_p, with each particle p whose size is determined by the particle radius, i.e.

$$\phi_p(\vec{x}) = s_p(r_p - |\vec{x} - \vec{x}_p|). \tag{8}$$

The particle defined level set function is computed locally on the eight corners of the cell containing the particle. The local values of ϕ_p are the particle predictions of the values of the overall level set function, ϕ, on the corners of the cell. Any variation of ϕ from ϕ_p indicates possible errors in the level set solution.

Error Correction We use escaped positive particles to rebuild the $\phi > 0$ region and escaped negative particles to rebuild the $\phi \leq 0$ region. The reconstruction of the implicit surface occurs locally within the cell that each escaped particle currently occupies. Using equation 8, the ϕ_p values of escaped particles are calculated for the eight grid points on the boundary of the cell containing the particle. This value is compared to the current value of ϕ for each grid point and we take the smaller value (in magnitude) which is the value closest to the "final version" of the $\phi = 0$ contour defining the surface. This is done for each escaped positive and escaped negative particle resulting in an improved representation of the surface of the liquid.

ϕ is maintained to be a signed distance function through a reinitialization process using $\phi_\tau + \text{sgn}(\phi_0)(|\nabla\phi| - 1) = 0$, where $\text{sgn}(\phi_0)$ is a one-dimensional smeared sign function, see [13] for more details. This procedure is performed after each combined Runge-Kutta cycle and error correction step. Unfortunately, reinitialization may cause the zero level set to move, which is not desirable, so we use the particle level set method to correct these errors as well.

In summary, the order of operations is: evolve both the particles and the level set function forward in time, correct errors in the level set function using particles, apply reinitialization, again correct errors in the level set function using particles, and finally adjust the particle radii. In order to retain a smooth interface, escaped particles which are more than 1.5 times their radius removed from the appropriate side of the interface are deleted.

Particle reseeding strategies are an important aspect of the particle level set method in order to ensure that the interface is properly resolved. For the examples shown, we reseeded particles about the $\phi = 0$ contour after every 20 time steps. We have found that this rule-of-thumb works well, but other metrics can be used to indicate when the particles near the interface need to reseeded as discussed in [3].

2.2 Pressure Boundary Treatment

While the constant coefficient Poisson equation for the pressure, equation 5, has been extensively analyzed for a fixed, Cartesian computational domain, difficulties arise when solving this equation on a domain with a dynamically changing irregular boundary. For the free surface problem, a physically correct Dirichlet boundary condition needs to be applied at the sub-cell location of the free surface. Naive central differencing of the left-hand side of equations 6 and 5 near the free surface can lead to $O(\Delta x)$ errors in the pressure as noted by [14]. Later, [15] reduced the magnitude of these first order errors with the introduction of micro-cells.

We instead use the fully second order accurate discretization proposed in [16]. We use a dimension by dimension approach to discretize equation 5 near the free surface. Consider the solution of a one dimensional constant coefficient Poisson equation, $p_{xx} = u^*_x$, where the free surface interface is located at x_{fs} with $x_i < x_{fs} < x_{i+1}$, and a Dirichlet boundary condition of $p = p_{fs}$ is applied at that interface. The standard second order accurate discretization of the Poisson equation can no longer be applied at x_i, since the stencil cuts across the interface. This can be remedied by defining a ghost value of p^G_{i+1} at x_{i+1} and rewriting the standard discretization of equation 5 as

$$\frac{p^G_{i+1} - 2p_i + p_{i-1}}{(\triangle x)^2} = (u^*_x)_i \tag{9}$$

in order to solve for p_i. Possible candidates for p^G_{i+1} can be derived using extrapolation. The discretization obtained from using constant extrapolation is only first order accurate since it displaces the physically correct boundary by an $O(\Delta x)$ amount. Several authors (see for example [17, 18]) used schemes equivalent to quadratic extrapolation obtaining second order accurate, but nonsymmetric discretizations. In [16], the authors proposed using linear extrapolation of the form

$$p^G_{i+1} = \frac{p_{fs} + (\theta - 1)\,p_i}{\theta} \tag{10}$$

were $\theta \in [0,1]$ is defined by $\theta = (x_{fs} - x_i)/\triangle x$. This leads to a symmetric discretization of

$$\frac{\left(\frac{p_{fs}-p_i}{\theta\triangle x}\right) - \left(\frac{p_i-p_{i-1}}{\triangle x}\right)}{\triangle x} = (u^*_x)_i \tag{11}$$

which was shown to be second order accurate. This symmetric discretization allows for the use of a fast Poisson solver, e.g. we use a PCG method with a modified incomplete Cholesky preconditioner [8].

One can impose $p_{fs} = p_{air}$ or $p_{fs} = p_{air} + \sigma\kappa_{fs}$, depending upon whether or not surface tension is included in the calculation. κ_{fs} is the curvature at the interface, and is computed using standard central finite differences. Finally, one should be careful to use the same discretization of the gradient operator in both equations 5 and 6 in order to maintain the conservative properties of the method. Also, since we wish to make u^{n+1} divergence-free, the location of the interface should be determined by ϕ^{n+1} rather than ϕ^n.

2.3 Velocity Extrapolation

While the Navier-Stokes equations can be used to obtain velocities inside the liquid volume, velocity boundary conditions

Grid Cells	$\max\|\vec{u}\|$
25	1.435e-3
50	1.26e-3
100	8.0e-4
200	1.51e-4

Table 1. Parasitic velocity on a 1x1 domain after 500 time steps.

are needed on the unmodeled, i.e. $\phi > 0$, side of the free surface. A commonly used condition [4,19] is to set the undetermined velocities so that $\nabla \cdot \vec{u} = 0$ is satisfied. Probably the best treatment of this sort is contained in [20]. Since we will use these velocity boundary conditions to not only update the velocity and pressure fields, but also to evolve both the particles and the level set, we need velocity boundary conditions a few grid cells deep into the $\phi > 0$ region. Thus, we take a different approach using constant extrapolation of the liquid velocities from inside the liquid in the direction normal to the interface. The extrapolated velocities, $\vec{u}_{ext}$ can be obtained by solving to steady state

$$\frac{\partial \vec{u}_{ext}}{\partial \tau} + \vec{N} \cdot \nabla \vec{u}_{ext} = 0 \tag{12}$$

where τ is a fictitious time. At steady state we note that $\nabla\phi \cdot \nabla\vec{u}_{ext} = 0$ so that a faster method can be devised by marching out from the interface, see [21,22]. We usually populate a 3 to 5 grid cell band in the $\phi > 0$ region with extension velocities. Note that as the grid resolution approaches zero, these extension velocities do indeed solve the $\nabla \cdot \vec{u} = 0$ condition.

3 Examples

3.1 Parasitic Currents

The motion of bubbles and drops of small radius can be affected by spurious velocities generated due to the non-zero curvature of the interface. In order to obtain accurate behavior of the interface over long time periods, these numerical inaccuracies resulting from parasitic currents need to be suppressed. These spurious velocities have been analyzed for VOF [23,24], front tracking [25–27], and level set [11] methods. Kang *et. al.* [11] noted that the ghost fluid treatment of the discontinuity at the interface reduces the effect of parasitic currents on a stationary bubble by a factor of 1000 over the typical delta function treatment as in [13]. Our method creates a negligible amount of parasitic currents under grid refinement for an inviscid, stationary water drop as seen in table 1.

3.2 Drop Oscillation

To further assess the ability of the second order accurate pressure solver, we calculated the period of oscillation of a perturbed, inviscid 2D droplet. Lamb [28] showed that the frequency of oscillation, ω, of a 2D droplet is given by $\omega_n^2 = (n^3 - n)[\sigma/(\rho a^3)]$, where the interface of the droplet in polar coordinates is determined by $r = a + \varepsilon \cos(n\theta)$. ρ is the liquid density, a is the unperturbed radius, σ is the surface tension coefficient and n is the mode of oscillation in the plane with amplitude ε. In order to obtain a theoretical period of oscillation of π for the $n = 2$ mode, a drop with $\rho = 27$, $a = 1/3$, $\sigma = 2/3$, and $\varepsilon = .05a$ was used. The period of oscillation calculated by the second order accurate pressure solver shows excellent agreement as compared to the theoretical result, unlike the period of oscillation obtained by a first order accurate method. Also, the amount of numerical viscosity present in the calculation is greatly reduced with the use of the second order accurate solver as seen in figure 1 and table 2.

	Grid Cells	Period	Amplitude
1st Order Pressure Solver	50	3.542	.3400
	100	3.260	.3443
	200	3.219	.3470
2nd Order Pressure Solver	50	3.145	.3475
	100	3.168	.3487
	200	3.160	.3493

Table 2. Period of oscillation and amplitude after one period of perturbed 2D droplet (1x1 domain).

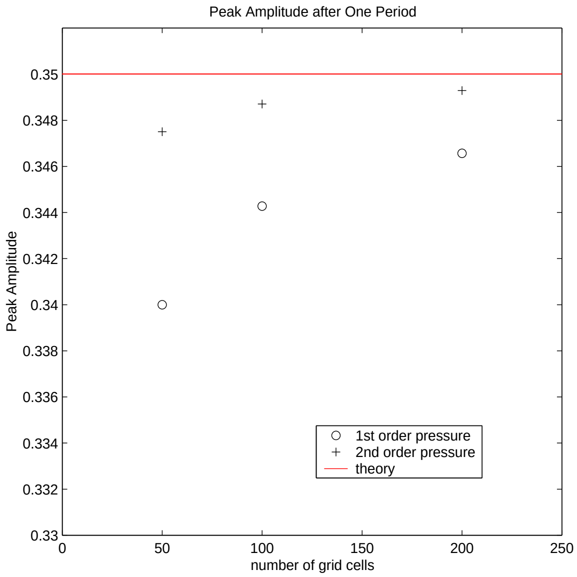

Figure 1. Peak amplitude after one period.

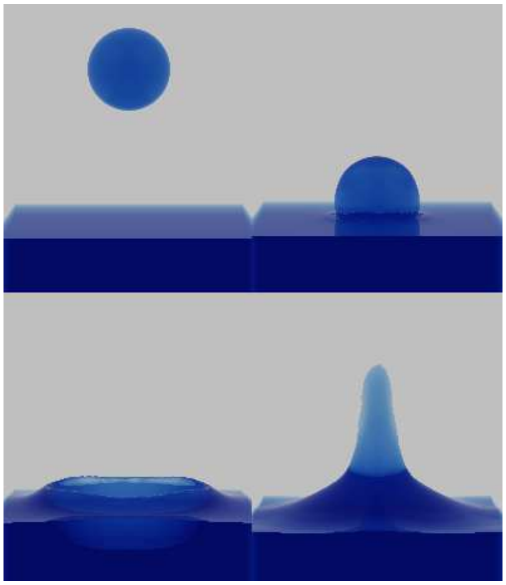

Figure 2. Drop impact onto liquid pool.

3.3 Liquid Mass Impact

Liquid drop impact on liquid and solid surfaces has important engineering and biological implications. Ink-jet printing and the dispersal of fungal spores are examples of two processes which are governed by drop impact dynamics. A variety of different behaviors can be observed depending upon the initial conditions used and whether or not the drop impinges upon a liquid or solid surface, including splashing, coalescence, and bouncing. We have performed a number of 3D inviscid drop calculations. For example, in figure 2 a drop of radius $1/3m$ with initial velocity $-5\text{m}/\text{s}$ impinges upon a pool of similar liquid $1m$ in depth. A surface tension coefficient of $\sigma = .0728\,\text{kg}/\text{s}^2$, corresponding to water, was used. The calculation was performed on a computational domain of size (-1m,1m)x(-1m,1m)x(-2m,1m) and was discretized using 60x60x90 grid cells. *We note that the particle level set method seems to perform as well as the level set method in the case of merging. Moreover, it usually performs better since there tends to be area loss or gain when using the level set only method for merging and pinching.*

3.4 Breaking Wave

We have also performed a 2D breaking wave calculation. Similar work can be found in [22] and [29]. A computational domain of size (-50m,175m)x(0m,35m) was discretized using 900x140 grid cells. The initial velocity and shape of the wave was initialized according to [29] with a wave height of 5m and water depth of 10m. No surface tension was used during this calculation. The wave breaks against a beach with a slope of 1 : 14. Our particle level set method is able to robustly handle the overturning and splashdown phase of the breaking wave and capture the formation of the secondary jet as seen in figure 3.

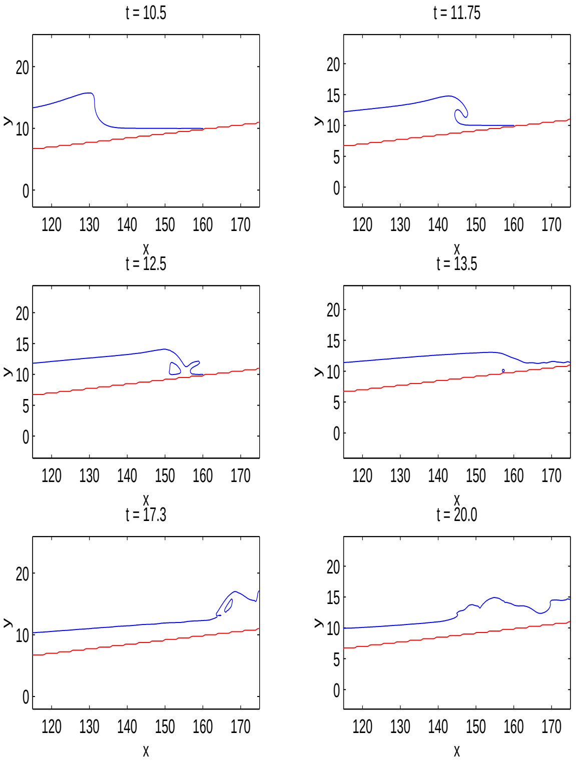

Figure 3. Breaking wave.

4 Conclusion

A mass conserving particle level set discretization was presented for topologically complex two and three dimensional incompressible free surface flows. This includes a second order accurate pressure discretization which allows for the straightforward inclusion of surface tension forces. A velocity extrapolation technique was used to obtain the necessary velocity boundary conditions at and near the free surface. Future work includes the incorporation of viscosity, and a detailed study of the free-surface turbulence generated by the motion of a ship through the ocean.

ACKNOWLEDGMENT

Research supported in part by an ONR YIP and PECASE award (ONR N00014-01-1-0620), a Packard Foundation Fellowship, a Sloan Research Fellowship, ONR N00014-03-1-0071, NSF DMS-0106694 and NSF ITR-0121288. In addition, D.E. and F.G. were each supported in part by an NSF postdoctoral fellowship (NSF #DMS-0202459 and #DMS-0102029).

REFERENCES

[1] Osher, S., and Sethian, J., 1988. "Fronts propagating with curvature dependent speed: Algorithms based on hamiliton-jacobi formulations". J. Comp. Phys., **79** (1), pp. 12–49.

[2] Osher, S., and Fedkiw, R., 2002. *The Level Set Method and Dynamic Implicit Surfaces*. Springer-Verlag, New York.

[3] Enright, D., Fedkiw, R., Ferziger, J., and Mitchell, I., 2002. "A hybrid particle level set method for improved interface capturing". J. Comp. Phys., **183** (1), pp. 83–116.

[4] Harlow, F., and Welch, J., 1965. "Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface". Phys. Fluids, **8** (1), pp. 2182–2189.

[5] Peyret, R., and Taylor, T. D., 1983. *Computational Methods for Fluid Flow*. Spring-Verlag, New York.

[6] Kim, J., and Moin, P., 1985. "Application of a fractional-step method to incompressible navier-stokes equations". *J. Comp. Phys.*, **59**, pp. 308–323.

[7] Brown, D. L., Cortez, R., and Minion, M. L., 2001. "Accurate projecion methods for the incompressible navier-stokes equations". J. Comp. Phys., **168** (2), pp. 464–499.

[8] Saad, Y., 1996. *Iterative methods for sparse linear systems*. PWS Publishing. New York, NY.

[9] Shu, C., and Osher, S., 1988. "Efficient implementation of essentially non-oscillatory shock capturing schemes". J. Comp. Phys., **77** (2), pp. 439–471.

[10] Shu, C., and Osher, S., 1989. "Efficient implementation of essentially non-oscillatory shock capturing schemes ii (two)". *J. Comp. Phys.*, **83**, pp. 32–78.

[11] Kang, M., Fedkiw, R., and Liu, X.-D., 2000. "A boundary condition capturing method for multiphase incompressible flow". J. Sci. Comput., **15** (3), pp. 323–360.

[12] Jiang, G.-S., and Peng, D., 2000. "Weighted eno schemes for hamilton-jacobi equations". *SIAM J. Sci. Comput.*, **21**, pp. 2126–2143.

[13] Sussman, M., Smereka, P., and Osher, S., 1994. "A level set approach for computing solutions to incompressible two-phase flow". J. Comp. Phys., **114** (1), pp. 146–159.

[14] Chan, R. K.-C., and Street, R. L., 1970. "A computer study of finite-amplitude water waves". J. Comp. Phys., **6** (1), pp. 68–94.

[15] Raad, P., Chen, S., and Johnson, D., 1994. "The introduction of micro cells to treat pressure in free surface flow problems". In Proceedings of the Fluids Engineering Division, vol. 202, ASME International Mechanical Engineering Congress and Exposition, pp. 43–54.

[16] Gibou, F., Fedkiw, R. P., Cheng, L.-T., and Kang, M., 2002. "A second–order–accurate symmetric discretization of the poisson equation on irregular domains". J. Comp. Phys., **176** (1), pp. 205–227.

[17] Chen, S., Merriman, B., Osher, S., and Smereka, P., 1997. "A simple level set method for solving stefan problems". *J. Comp. Phys.*, **135**, pp. 8–29.

[18] Udaykumar, H., Mittal, R., and Shyy, W., 1999. "Computation of solid-liquid phase fronts in the sharp interface limit on fixed grids". *J. Comp. Phys.*, **153**, pp. 535–574.

[19] Amsden, A., and Harlow, F., 1970. The SMAC method: A numerical technique for calculating incompressible fluid flows. Tech. Rep. LA–4370, Los Alamos Scientific Laboratory.

[20] Chen, S., Johnson, D., and Raad, P., 1995. "Velocity boundary conditions for the simulation of free surface fluid flow". J. Comp. Phys., **116** (2), pp. 262–276.

[21] Adalsteinsson, D., and Sethian, J., 1999. "The fast construction of extension velocities in level set methods". *J. Comp. Phys.*, **148**, pp. 2–22.

[22] Enright, D., Marschner, S., and Fedkiw, R., 2002. "Animation and rendering of complex water surfaces". ACM Trans. on Graphics (SIGGRAPH 2002 Proceedings), **21** (3), pp. 736–744.

[23] Brackbill, J., Kothe, D., and Zemach, C., 1992. "A continuum method for modeling surface tension". J. Comp. Phys., **100** (2), pp. 335–354.

[24] Lafaurie, B., Nardone, C., Scardovelli, R., Zaleski, S., and Zanetti, G., 1994. "Modelling merging and fragmentation in multiphase flows with surfer". J. Comp. Phys., **113** (1), pp. 134–147.

[25] Torres, D., and Brackbill, J., 2000. "The point-set method: Front tracking without connectivity". J. Comp. Phys., **165** (2), pp. 620–644.

[26] Tryggvason, G., Bunner, B., Esmaeeli, A., Juric, D., Al-Rawahi, N., Tauber, W., Han, J., Nas, S., and Jan, Y.-J., 2001. "A front-tracking method for the computations of multiphase flow". J. Comp. Phys., **169** (2), pp. 708–759.

[27] Shin, S., and Juric, D., 2002. "Modeling three-dimensional multiphase flow using a level contour reconstruction method for front tracking without connectivity". J. Comp. Phys., **180** (2), pp. 427–470.

[28] Lamb, H., 1932. *Hydrodynamics*. Dover, New York.

[29] Radovitzky, R., and Ortiz, M., 1998. "Lagrangian finite element analysis of newtonian fluid flows". Int. J. Numer. Meth. Engng., **43** (4), pp. 607–619.