

Robust Treatment of Interfaces for Fluid Flows and Computer Graphics

Doug Enright¹ and Ron Fedkiw²

¹ Mathematics Department, University of California, Los Angeles,
enright@math.ucla.edu

² Computer Science Department, Stanford University, *fedkiw@cs.stanford.edu*

1 Introduction

Researchers have used numerical techniques to solve partial differential equations describing physical phenomena for many years. One challenging area, the numerical treatment of interfaces, motivated the creation of a topologically robust interface capturing algorithm, the level set method of Osher and Sethian [29]. The level set method has been used to track interfaces in a wide variety of applications. Utilizing geometrical information about the interface, which is naturally obtained from the level set function, an accurate treatment of material discontinuities across the interface can be obtained via the Ghost Fluid Method [15]. Discontinuities are implicitly enforced with the ghost fluid method, avoiding any numerical smoothing of discontinuous quantities across the interface. The ghost fluid method and related techniques have been used to model discontinuities in compressible and incompressible flows [15, 24, 22, 3], flames and detonations [27, 16], solid fluid coupling [14] and Stefan problems [19, 18, 4]. A newly proposed, fully conservative ghost fluid method has been used to track contact discontinuities, inert shocks and detonation waves [25]. Accurate modeling of the motion of a contact discontinuity itself for incompressible flows has been a challenge for level set methods. Recently a new method, the “particle level set method” [10], has been proposed to accurately track contact discontinuities for incompressible flows. The particle level set method conserves mass to an accuracy comparable to explicit front tracking and volume of fluid methods. Due to the robustness and ease of programming of these interface methods in three spatial dimensions combined with the ever increasing speed and memory of desktop computers, physics-based animation algorithms to model fire and water [26, 17, 11] have taken advantage of these methods in order to produce realistic looking behavior on the coarse computational grids commonly used in a production animation environment. In this article we give a brief overview of the level set method, the use of the ghost fluid method for modeling discontinuities across the interface, the “particle level set method” for tracking contact discontinuities, and illustrate the use of these methods in the context of computer animation. Additional details concerning these methods can be found in the recently published book, “Level Set Methods and Dynamic Implicit Surfaces” [28].

2 Level Set Method

In order to robustly deal with topological changes to a dynamically evolving interface, a simple and versatile method to treat this important problem is obtained by embedding the interface of an open region Ω as the level set of a smooth (at least Lipschitz continuous) higher dimensional function $\phi(\mathbf{x}, t)$. The level set function ϕ has the properties:

$$\begin{aligned}\phi(\mathbf{x}, t) &< 0 \text{ for } \mathbf{x} \in \Omega \\ \phi(\mathbf{x}, t) &> 0 \text{ for } \mathbf{x} \notin \bar{\Omega} \\ \phi(\mathbf{x}, t) &= 0 \text{ for } \mathbf{x} \in \partial\Omega = \Gamma(t).\end{aligned}$$

The description of the interface in this manner allows for the natural merging or separation of the interface without any additional involvement by the user.

The interface Γ is evolved in time by a velocity field $\mathbf{V}(\mathbf{x}, t)$ according to the simple advection equation

$$\frac{\partial\phi}{\partial t} + \mathbf{V} \cdot \nabla\phi = 0. \quad (1)$$

High order accurate WENO methods [21] methods can be used to discretize the spatial derivatives in equation 1 combined with an explicit TVD Runge-Kutta method utilizing convex combinations of simple forward Euler updates [32, 22] in order to integrate equation 1 forward in time. Local level set methods [1, 30] can substantially reduce the spatial complexity of equation 1 by reducing the calculation to a banded region about the interface.

Geometrical quantities can be easily calculated from the level set function. Unit normals are given by

$$\mathbf{N} = \frac{\nabla\phi}{|\nabla\phi|} \quad (2)$$

and the curvature by

$$\kappa = \nabla \cdot \left(\frac{\nabla\phi}{|\nabla\phi|} \right). \quad (3)$$

Two commonly performed operations using level set functions include the reinitialization of ϕ to be the signed distance to the interface Γ and the extrapolation of quantities across the interface from one side of the domain to the other. Reinitialization of ϕ can be achieved by solving to steady state (as fictitious time $\tau \rightarrow \infty$) the equation

$$\phi_\tau + S(\phi_0)(|\nabla\phi| - 1) = 0, \quad (4)$$

where $S(\phi_0) = \phi_0 / \sqrt{\phi_0^2 + (\Delta x)^2}$ [36]. Extrapolation of a variable I across the interface is obtained by again solving to steady state

$$I_\tau \pm \mathbf{N} \cdot \nabla I = 0 \quad (5)$$

[15, 5]. By making clever use of the way the information in equations 4 and 5 propagates, fast heap-based methods [37, 31, 2] may be used to solve these equations in $O(N \log N)$ time, where N is the number of grid points.

Theoretical justification of the level set method for geometrically-based motion came through the theory of viscosity solutions for scalar time-dependent partial differential equations [6, 12]. The notion of having a vanishing viscosity solution guarantees the existence of a unique solution which is consistent with equation 1 [9]. While this behavior is certainly comforting to a computational user, the vanishing viscosity solution approaches the true solution in the limit as $\Delta x \rightarrow 0$, a case never truly obtained in practice. The implications of calculating a vanishing viscosity solution to equation 1 on coarse computational grids, or in under-resolved regions of the flow (such as a sharp corner in a geometry driven flow) is discussed further in section 4.

3 Ghost Fluid Method

Spurious oscillations in material fields resulting from discontinuities due to shocks or contact discontinuities have been a source of difficulty in the numerical solution of hyperbolic conservation laws. In order to obtain correct shock speeds and strengths, the Lax-Wendroff theorem [23] states that a numerical method used should be fully conservative. An explicit way to deal with this requirement is by solving multidimensional Riemann problems at the location of the interface. This approach has been used by Glimm et al. [20] in conjunction with an explicit representation of the interface. However, complicated interfacial geometry along with changes in topology and a lack of a proper entropy condition built into the interface representation, place an onerous burden on any explicit method to capture these details in a robust manner. The Ghost Fluid Method on the other hand implicitly captures the Rankine-Hugoniot jump conditions at the interface in a manner similar to the implicit capturing of the location of an interface by the level set method. The result is an accurate, easy-to-implement, and topologically robust numerical algorithm.

Conservation of mass, momentum, and energy fluxes (F_ρ , $\mathbf{F}_{\rho\mathbf{V}}$, and F_E) across the interface results in the Rankine-Hugoniot jump conditions for the relevant physical variables. For an interface moving at speed D in the normal direction to the interface, F_ρ , $\mathbf{F}_{\rho\mathbf{V}}$, and F_E describing an inviscid compressible fluid are given by

$$F_\rho = \rho(V_N - D) \quad (6)$$

$$\mathbf{F}_{\rho\mathbf{V}} = \rho(\mathbf{V}^T - D\mathbf{N}^T)(V_N - D) + p\mathbf{N}^T \quad (7)$$

$$F_E = \left(\rho e + \frac{\rho|\mathbf{V} - D\mathbf{N}|^2}{2} + p \right) (V_N - D), \quad (8)$$

where ρ is the density, \mathbf{V} is the fluid velocity, \mathbf{N} is the normal to the interface, $V_N = \mathbf{V} \cdot \mathbf{N}$, e is the internal energy per unit mass, and p is the pressure.

Equations 6, 7, and 8 allow for chemical reactions at the interface, resulting in an interface velocity different from the underlying fluid velocity. For contact discontinuities with $\mathbf{V}_N = D$, two of these fluxes are zero across the interface, and the interface separates two gases (or materials) with possibly different equations of state.

The key to the ghost fluid method is that by defining a set of ghost cells on each side of the interface, the ghost cells can implicitly capture the physically correct boundary conditions as defined by equations 6, 7, and 8 in such a manner as to avoid any finite differencing across a discontinuity. The method also avoids the common approach of numerically smoothing a discontinuity with the aim of preventing the creation of nonphysical oscillations. The location of the ghost cells is the same as the real grid cell locations. Since both fluids are defined in a neighborhood about the front, one can solve for each fluid *independently* using standard schemes, regardless of the geometry of the front. After updating each fluid, the choice as to which of the two values, the “ghost fluid” or “real fluid”, to take near the interface is determined by the updated level set function describing the new location of the interface.

At each time step ghost cells are populated in a node-by-node fashion in order to preserve the continuity of the mass, momentum, and energy fluxes across the interface. This is achieved by solving the system of equations: $F_\rho^G = F_\rho^R$, $\mathbf{F}_{\rho\mathbf{V}}^G = \mathbf{F}_{\rho\mathbf{V}}^R$, and $F_E^G = F_E^R$ at each grid point with “R” representing the known real fluid values and “G” for the ghost values. The solution of this system of equations is simple compared to applying the Rankine-Hugoniot conditions at the interface. In the case of a contact discontinuity, we identify the continuous and discontinuous quantities and set them appropriately in order to ensure that the jump conditions are satisfied as follows. In this case, $[V] = 0$ and $[p] = 0$, where $[\cdot]$ is the jump in value across the interface, while the entropy, S , is discontinuous across the interface front. Since the pressure and velocity are continuous across the interface, we can take $V^G = V^R$ and $p^G = p^R$, however we need to extrapolate the value of the entropy across the interface to avoid differencing across the discontinuity and to ensure that the ghost fluid is indeed representing the real fluid it is standing in for. This is comparable to the ghost fluid taking on the correct equation of state. In multiple spatial dimensions, extrapolation of the discontinuous variables can be implemented according to equation 5.

Besides capturing discontinuous boundary conditions at an irregular interface for compressible flows, discontinuities in physical variables, e.g. pressure and density, can exist in incompressible flows. These discontinuities at the interface need to be accounted for when solving the resulting Poisson equation for the pressure. As shown in [24, 22, 27], a ghost fluid method approach for capturing these discontinuities across the interface is possible as well. The resulting numerical method for a variable coefficient Poisson equation in the presence of interfaces where the coefficients and the solution itself may be discontinuous is robust and easy to implement in multiple spatial dimen-

sions. In addition, the resulting coefficient matrix of the associated linear system is symmetric, allowing for the use of fast “black-box” solvers such as a preconditioned conjugate gradient method.

4 Particle Level Set Method

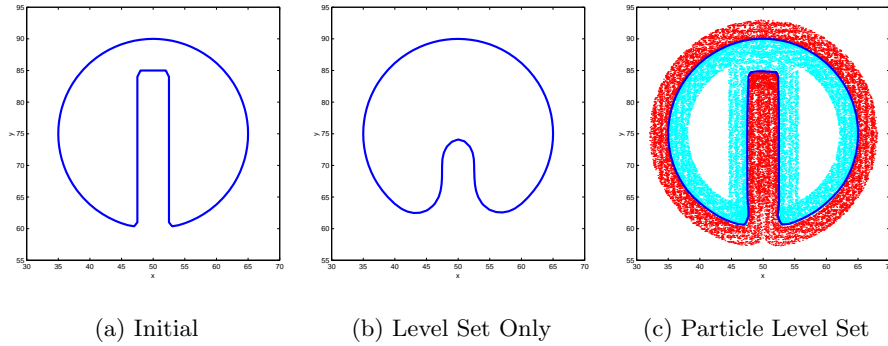


Fig. 1. Rigid Body Rotation of a Notched Disk

The robustness of the level set method is due to the regularization by curvature property that a numerical implementation the level set method intrinsically possesses. This regularization property allows for effortless changes in interface topology, e.g. the pinching off or merging together of the interface, a key aspect of the level set method. An upwind discretization of equation 1 results in a numerical truncation error of the form $\epsilon \Delta \phi$, with $\epsilon \approx O((\Delta x)^r)$, r being the order of accuracy of the discretization used. So instead of equation 1 being solved exactly, we actually are obtaining a solution to

$$\phi_t + \mathbf{V} \cdot \nabla \phi = \epsilon \Delta \phi. \quad (9)$$

The ϕ obtained from the above equation is actually the vanishing viscosity solution to equation 1. The viscosity term on the right hand side of equation 9 is proportional to the curvature of the interface and goes to zero as $\Delta x \rightarrow 0$. The effect of this unmodeled, but always present viscosity term can be seen in figure 1. Here a notched disk (with a notch width of 5 grid cells), undergoes a rigid body rotation. After one revolution, a level set only representation of the interface is seen in figure 1(b). The thin notch region along with the high curvature convex corners at the bottom of the disk have experienced large amounts of numerical diffusion. One solution to this problem is to increase the grid resolution, thereby decreasing the effect of numerical viscosity. However, this solution can dramatically increase the computational time needed,

especially when the level set method is used to track a contact discontinuity, e.g. an air-water interface, in a computational fluid dynamics simulation.

An alternative solution, is to use an error correction mechanism along portions of the interface which are susceptible to large amounts of diffusion. The error correcting mechanism we propose to use are diffusionless particles placed near the $\phi = 0$ isocontour. The results of this new “particle level set method” can be seen in figure 1(c) where the thin notch along with the sharp corners have maintained their original shape with little to no diffusion. The particles move according to $d\mathbf{x}_p/dt = \mathbf{V}(\mathbf{x}_p)$, and each particle possesses a radius r_p and a sign s_p . Since the level set is tracking a contact discontinuity, particles which correspond to the $\phi > 0$ region should always remain in the $\phi > 0$ region and vice versa, however excessive amounts of numerical diffusion can cause positive particles, i.e. particles with $s_p = +1$, to end up in a $\phi < 0$ region according to the level set function. These particles are said to have “escaped” from their respective side of the interface and indicate that a first order error in the location of the interface has occurred. This first order accurate error in ϕ can be corrected for by the particles since the radius of each particle defines a local level set function, ϕ_p , which we can compare against ϕ at the corners of the grid cell containing the particle. We take the value closest to zero as the new more accurate value of ϕ . After iterating through all the escaped particles and determining corrected ϕ values, the particles then resample their distance to the interface and adjust their radii accordingly. This error reduction technique can also be used to correct errors made when ϕ is reinitialized to be a signed distance function as well. In this case the particle velocity is assumed to be zero since the interface should not move. A complete description of this error reduction technique can be found in [10].

The robustness of the level set method is maintained by the particle level set method since the marker particles do not *explicitly* delineate the location of the interface. Rather, they locally capture the location of the interface through the ϕ_p function, and the level set function itself is used to automatically treat connectivity (merging and pinching of fronts). The ease-of-implementation of the level set method is maintained since the particles are disconnected and communicate with the level set function only during the error reduction stage described above. Since particles are placed within a band about the $\phi = 0$ isocontour, the interface is resolved on multiple scales by the particles. This multi-resolution approach is quite successful in preserving the volume of the level set when the interface undergoes large amounts of stretching induced by an incompressible flow field as seen in figure 2. A level set only approach as seen in figure 2(a) can not maintain regions of high curvature and the thin (approximately one grid cell thick) pancake region formed during the deformation process. On the other hand, the particle level set method can resolve these regions on the 100^3 grid used. Also, while tearing of the interface is seen in the thin pancake region with the particle level set method, particles

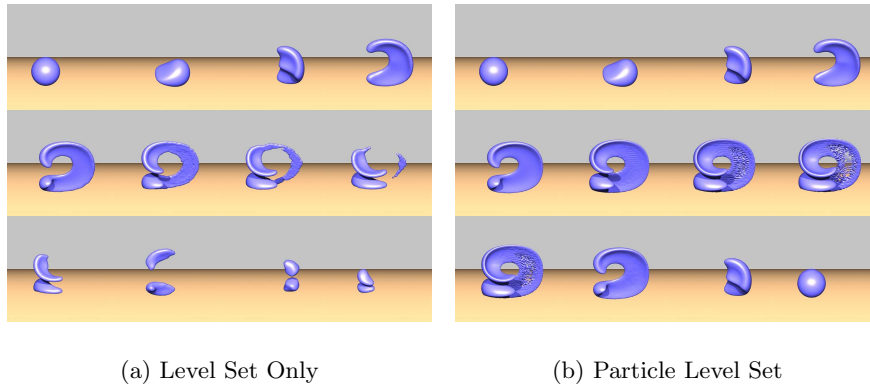


Fig. 2. 3D Deformation Test

which remain escaped are not deleted and can contribute to the rebuilding of the interface as seen in the last row of frames in figure 2(b). The sphere which loses over 80% of its volume by the end of the deformation process when represented with a level set only method, loses only 2% of its volume with the particle level set method. The additional cost of placing particles near the interface is offset by the ability to use much coarser volumetric grids when calculating the pressure during flow calculations without sacrificing a faithful representation of the interface.

5 Computer Graphics

The modeling of natural phenomena such as water and fire for computer graphics applications remains a major challenge. The complexity of the motion exhibited by these phenomena defies the ability of animators to realistically animate by hand. The ever increasing use of computer animation in feature films to create photorealistic effects in their own right and to supplement practical elements previously filmed have motivated researchers in computer graphics (CG) to examine the extensive computational fluid dynamics (CFD) literature for algorithms which can be adapted for use in an animation environment. An important criteria for the use of such algorithms is the ability to robustly model fully three dimensional effects on the coarse computational grids commonly used in a CG environment. Recent research [13, 17, 11, 26] has shown promise that when appropriate CFD algorithms are coupled with the level set related methods, the long sought after goal of the CG community of photorealistic fire and water behavior can be attained.

For the purposes of CG, the motion of water and fire (low speed deflagrations) can be modeled using the inviscid, incompressible Euler equations,

$$\nabla \cdot \mathbf{V} = 0 \quad (10)$$

$$\mathbf{V}_t + (\mathbf{V} \cdot \nabla) \mathbf{V} + \frac{\nabla p}{\rho} = \mathbf{f}, \quad (11)$$

where \mathbf{f} can be a variety of body forces including gravity and buoyancy as appropriate. Additional transport equations for the reaction coordinate, temperature, and the density of soot resulting from the chemical reaction at the flame front need to be modeled in order to obtain the necessary information for the visualization of fire. These equations are discussed in detail in [26]. Due to the 1000 to 1 density ratio between water and air, the dynamics of the air on the water can be safely neglected, requiring only the solution of equations 10 and 11 on the water side of the interface. Fire on the other hand requires the solution of the Euler equations on both sides of the interface and more importantly the accurate capturing of the jump conditions resulting from equations 6 and 7.

A projection method [7] is used to update equation 11, where an intermediate velocity field \mathbf{V}^* is first obtained by neglecting the pressure term,

$$\frac{\mathbf{V}^* - \mathbf{V}^n}{\Delta t} + (\mathbf{V} \cdot \nabla) \mathbf{V} = \mathbf{0}. \quad (12)$$

Unconditional stability of a numerical scheme is important for its use in an animation environment, leading the CG community to adopt a semi-Lagrangian method [8, 34, 33] to discretize the convective term in equation 11. Use of a semi-Lagrangian method may introduce large amounts of numerical dissipation, especially on the coarse computational grids used. The method of choice to reduce this dissipation is the “vorticity confinement” method [35] (discussed below). To enforce mass conservation, the pressure is determined by the Poisson equation,

$$\nabla \cdot \left(\frac{1}{\rho} \nabla p \right) = \frac{\nabla \cdot \mathbf{V}^*}{\Delta t}. \quad (13)$$

The gradient of the pressure is then used to advance the velocity field to the $n + 1$ time level according to

$$\frac{\mathbf{V}^{n+1} - \mathbf{V}^*}{\Delta t} + \frac{\nabla p}{\rho} = \mathbf{0}. \quad (14)$$

The movement of the contact discontinuity describing the air-water interface in the pouring of a glass of water shown in figure 3(a) can be described by equation 1, where \mathbf{V} is the underlying liquid velocity at the interface. Being able to obtain realistic looking merging and pinching off of the water interface while at the same time limiting the amount of numerical diffusion resulting from the $55 \times 55 \times 120$ grid used for this simulation is a necessary requirement of any interface method used. The particle level set method is able to represent the complex liquid surface shown and maintain the “liveliness”



(a) CG Water

(b) CG Fire

Fig. 3. Physics Based Animation of Water and Fire

of the motion of the interface by limiting the amount of interface dissipation present. In addition, an extrapolation of the liquid velocity field into the unmodeled air using equation 5 can be used to provide a velocity field for the “air” particles associated with particle level set method and plausible velocity boundary conditions which satisfy equation 10, all of which result in a smoothly moving and visually pleasing liquid surface.

An important part of the visual appearance of fire is the expansion of the gas as it undergoes a transformation from unburnt fuel into hot gaseous products. The Rankine-Hugoniot jump conditions at the interface naturally capture this outward expansion of the gas that is next to impossible to achieve through the use of low level hacks and random numbers usually resorted to by the computer graphics animation community. The jump conditions at the interface are

$$[\mathbf{V}] = -\rho_{fuel} S \begin{bmatrix} 1 \\ \rho \end{bmatrix} \mathbf{N} \quad (15)$$

$$[p] = -\rho_{fuel}^2 S^2 \begin{bmatrix} 1 \\ \rho \end{bmatrix}, \quad (16)$$

where $[\rho] = \rho_{prod} - \rho_{fuel}$ and S is the flame speed. The flame front is not a contact discontinuity, rather the interface moves over the unreacted gas with a speed $S = S_o + \sigma\kappa$, where κ is the curvature of the interface. The overall speed of the interface in a reference frame at rest with respect to the moving fuel is $V_N + S$, where V_N is the normal velocity of the underlying

fuel. Appropriate ghost velocities determined by equation 15 are used in the update step given in equation 12. The jump in pressure is incorporated in a boundary condition capturing manner into the solution of equation 13. To combat numerical dissipation and the resulting loss of small scale rolling features characteristic of fire and smoke on the coarse computational grid used, a numerically consistent “vorticity confinement” [35, 13] body force term is used. This term introduces additional vorticity in regions of the flow which possess large gradients in vorticity and are thus sensitive to excessive amounts of artificial damping. As illustrated by the campfire in figure 3(b), a robust three dimensional interface method is required to attain the correct visual look since fire can physically wrap around objects like the top unlit log at the base of the campfire and it is a participating medium, acting as an unsteady volumetric light source. This aspect of fire can be detected by an observer due to the reflection and scattering of the emitted light off other objects in the scene such as the rocks surrounding the campfire. A complete description of the physics, numerical calculation, and visual appearance of fire can be found in [27] and [26].

6 Acknowledgments

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

References

1. Adalsteinsson, D. and Sethian, J., *A Fast Level Set Method for Propagating Interfaces*, J. Comp. Phys. 118, 269–277 (1995).
2. Adalsteinsson, D. and Sethian, J., *The Fast Construction of Extension Velocities in Level Set Methods*, J. Comp. Phys. 148, 2–22 (1999).
3. Caiden, R., Fedkiw, R. and Anderson, C., *A Numerical Method For Two-Phase Flow Consisting of Separate Compressible and Incompressible Regions*, J. Comp. Phys. 166, 1–27 (2001).
4. Chen, S., Merriman, B., Kang, M., Caflisch, R., Ratsch, C., Cheng, L.-T., Gyure, M., Fedkiw, R., Anderson, C. and Osher, S., *Level Set Method for Thin Film Epitaxial Growth*, J. Comp. Phys. 167, 475–500 (2001).
5. Chen, S., Merriman, B., Osher, S. and Smereka, P., *A Simple Level Set Method for Solving Stefan Problems*, J. Comp. Phys. 135, 8–29 (1997).
6. Chen, Y. G., Giga, Y. and Goto, S., *Uniqueness and existence of viscosity solutions of generalized mean curvature flow equations*, J. Differential Geom. 33, 749–786 (1991).
7. Chorin, A. J., *Numerical Solution of the Navier-Stokes Equations*, Math. Comp. 22, 745–762 (1968).

8. Courant, R., Issacson, E. and Rees, M., *On the Solution of Nonlinear Hyperbolic Differential Equations by Finite Differences*, Comm. Pure and Applied Math 5, 243–255 (1952).
9. Crandall, M. G., Ishii, H. and Lions, P.-L., *User's guide to viscosity solutions of second order partial differential equations*, Bull. Amer. Math. Soc. 27, 1–67 (1992).
10. Enright, D., Fedkiw, R., Ferziger, J. and Mitchell, I., *A Hybrid Particle Level Set Method for Improved Interface Capturing*, J. Comp. Phys. (2002), in press.
11. Enright, D., Marschner, S. and Fedkiw, R., *Animation and Rendering of Complex Water Surfaces*, ACM Trans. on Graphics (SIGGRAPH 2002 Proceedings) 21, 736–744 (2002).
12. Evans, Y. C. and Spruck, J., *Motion of level sets by mean curvature. I.*, J. Differential Geom. 33, 635–681 (1991).
13. Fedkiw, R., Stam, J. and Jensen, H. W., *Visual Simulation of Smoke*, in Fiume, E., ed., *Proceedings of SIGGRAPH 2001*, Computer Graphics Proceedings, Annual Conference Series, pp. 15–22, ACM, ACM Press / ACM SIGGRAPH, 2001.
14. Fedkiw, R. P., *Coupling an Eulerian Fluid Calculation to a Lagrangian Solid Calculation with the Ghost Fluid Method*, J. Comp. Phys. 175, 200–224 (2002).
15. Fedkiw, R. P., Aslam, T., Merriman, B. and Osher, S., *A Non-oscillatory Eulerian Approach to Interfaces in Multimaterial Flows (The Ghost Fluid Method)*, J. Comp. Phys. 152, 457–492 (1999).
16. Fedkiw, R. P., Aslam, T. and Xu, S., *The Ghost Fluid Method for Deflagration and Detonation Discontinuities*, J. Comp. Phys. 154, 393–427 (1999).
17. Foster, N. and Fedkiw, R., *Practical Animation of Liquids*, in Fiume, E., ed., *Proceedings of SIGGRAPH 2001*, Computer Graphics Proceedings, Annual Conference Series, pp. 23–30, ACM, ACM Press / ACM SIGGRAPH, 2001.
18. Gibou, F., Fedkiw, R., Caffisch, R. and Osher, S., *A Level Set Approach for the Numerical Simulation of Dendritic Growth*, J. Sci. Comput. (2002), in press.
19. Gibou, F., Fedkiw, R. P., Cheng, L.-T. and Kang, M., *A Second-Order-Accurate Symmetric Discretization of the Poisson Equation on Irregular Domains*, J. Comp. Phys. 176, 205–227 (2002).
20. Glimm, J., Grove, J. W., Li, X. and Zhao, N., *Simple Front Tracking*, Contemp. Math. 238, 133–149 (1999).
21. Jiang, G.-S. and Peng, D., *Weighted ENO Schemes for Hamilton-Jacobi Equations*, SIAM J. Sci. Comput. 21, 2126–2143 (2000).
22. Kang, M., Fedkiw, R. and Liu, X.-D., *A Boundary Condition Capturing Method for Multiphase Incompressible Flow*, J. Sci. Comput. 15, 323–360 (2000).
23. Lax, P. and Wendroff, B., *Systems of Conservation Laws*, Comm. Pure Appl. Math. 13, 217–237 (1960).
24. Liu, X.-D., Fedkiw, R. and Kang, M., *A Boundary Condition Capturing Method for Poisson's Equation on Irregular Domains*, J. Comp. Phys. 160, 151–178 (2000).
25. Nguyen, D., Gibou, F. and Fedkiw, R., *A Fully Conservative Ghost Fluid Method and Stiff Detonation Waves*, in *12th International Detonation Symposium*, ONR, 2002.
26. Nguyen, D. Q., Fedkiw, R. and Jensen, H. W., *Physically Based Modeling and Animation of Fire*, ACM Trans. on Graphics (SIGGRAPH 2002 Proceedings) 21, 721–728 (2002).

27. Nguyen, D. Q., Fedkiw, R. P. and Kang, M., *A Boundary Condition Capturing Method for Incompressible Flame Discontinuities*, J. Comp. Phys. 172, 71–98 (2001).
28. Osher, S. and Fedkiw, R., *Level Set Methods and Dynamic Implicit Surfaces*, Springer-Verlag, New York, 2002.
29. Osher, S. and Sethian, J., *Fronts Propagating with Curvature Dependent Speed: Algorithms Based On Hamilton-Jacobi Formulations*, J. Comp. Phys. 79, 12–49 (1988).
30. Peng, D., Merriman, B., Osher, S., Zhao, H.-K. and Kang, M., *A PDE-Based Fast Local Level Set Method*, J. Comp. Phys. 155, 410–438 (1999).
31. Sethian, J., *A Fast Marching Level Set Method for Monotonically Advancing Fronts*, Proc. Natl. Acad. Sci. 93, 1591–1595 (1996).
32. Shu, C. and Osher, S., *Efficient Implementation of Essentially Non-Oscillatory Shock Capturing Schemes*, J. Comp. Phys. 77, 439–471 (1988).
33. Stam, J., *Stable Fluids*, in *Proceedings of SIGGRAPH 99*, Computer Graphics Proceedings, Annual Conference Series, pp. 121–128, ACM, ACM SIGGRAPH / Addison Wesley Longman, 1999.
34. Staniforth, A. and Côté, J., *Semi-Lagrangian Integration Schemes for Atmospheric Models - A Review*, Monthly Weather Review 119, 2206–2223 (1991).
35. Steinhoff, J. and Underhill, D., *Modification of the Euler Equations for "vorticity confinement": Application to the computation of interacting vortex rings*, Phys. Fluids 6, 2738–2744 (1994).
36. Sussman, M., Smereka, P. and Osher, S., *A Level Set Approach for Computing Solutions to Incompressible Two-Phase Flow*, J. Comp. Phys. 114, 146–159 (1994).
37. Tsitsiklis, J., *Efficient Algorithms for Globally Optimal Trajectories*, IEEE Trans. on Automatic Control 40, 1528–1538 (1995).