A BACKWARD POINT SHIFTED LEVELSET METHOD FOR HIGHLY ACCURATE INTERFACE COMPUTATION

YINGJIE LIU*

Abstract. We propose a technique that could significantly improve the accuracy of the levelset method and has the potential for fully conservative treatment. Level set method uses the levelset function, usually an approximate signed distance function Φ to indirectly represent the interface by the zero set of Φ. When Φ is advanced to the next time level by an advection equation, it is no longer a signed distance function any more, therefore the uneven numerical dissipation associated with the discretization of the advection equation could distort the interface particularly in places where the radius of curvature of the interface changes dramatically or two segments of the interface are getting close. Also an auxiliary equation is usually solved at each time level to restore Φ into a signed distance function, which could further shift the interface position. We address the second problem by using an analytic point shifted algorithm to locally perturb the mesh without geometric reconstruction of the interface so that the zero set of Φ is located at grid nodes and therefore solving the auxiliary equation will not move the interface. Our strategy for solving the first problem is that when applying the advection equation for Φ, it does not initiate from a signed distance function, but ends up with one, which can be achieved by solving the advection equation backward in time. These two techniques are combined with an iterative procedure.

Key words. front tracking, levelset method, point shifted algorithm.

AMS subject classifications. 65M60, 65M12

1. Introduction. Interface computation is a challenging and rewarding area for scientific research in recent years. There are many different approaches on how to track or capture the interface and how to do it more accurately, more robustly during topological changes of the interface and how to do it conservatively across the interface. Front tracking (e.g. [6, 8, 7, 5]) uses markers to track the interface and typically solves Riemann problems to update the positions of these markers at the next time level. Conservative front tracking methods (e.g. [13, 20, 2, 10, 9, 11]) are developed to insure the conservation across the interface. Levelset method was proposed by Osher and Sethian [17] to compute the interface motion indirectly by use of the zero set of the levelset function. Recent developments on improving the levelset method can be found for example on [15, 19, 1, 21, 4, 12, 18, 3, 16].

Using level set method for interface tracking has a great benefit of more analytically solving the geometric problem. Following the pioneering work of [17], given a velocity field u, the level set function Φ satisfies

$$\frac{\partial \Phi}{\partial t} + v \cdot \bigtriangledown \Phi = 0, \tag{1.1}$$

where $v = (u \cdot \bigtriangledown \Phi / | \bigtriangledown \Phi|^2) \bigtriangledown \Phi$, the normal (to the contour of Φ) component of the velocity field. Since the normal velocity field could be compressible, there is also an auxiliary equation to solve at each time step to reshape the Φ back into a signed distance function without changing its zero level set [19],

$$\frac{\partial \Phi}{\partial t} + \mathrm{sgn}(\Phi)(| \bigtriangledown \Phi| - 1) = 0. \tag{1.2}$$

*School of Mathematics, Georgia Institute of Technology, Atlanta, GA 30332 (yingjie@math.gatech.edu).

Even for incompressible velocity field it is more accurate to use a signed distance function as the levelset function because it will not have the large gradients brought by the advection, see [12].

There is a subtle difficulty here. The signed distance function Φ has zero second derivative if the interface is a straight line. In this case the numerical error out of solving the above two differential equation is small enough. But the second derivatives of Φ gets larger and larger when the radius of curvature of the interface gets smaller and smaller to close to the mesh size, or if two segments of the interface are getting closer and closer to close to the mesh size. In this situation, the numerical dissipation from discretizing the above two equations can cause serious artifact to the interface structure. This is particularly damaging for the equation (1.2) which tries to restore Φ back into a signed distance function without changing the position of the zero set of Φ . Because this zero set does not lie on the grid nodes and has to be interpreted from the Φ values at the nearby grid nodes. When the dissipation of discretizing the equation (1.2) becomes large enough, it will cause significant amount of position shift of the zero set of Φ. This problem has been addressed in many papers e.g.[18, 3]. Here we propose another systematical way of solving this problem which incorporates the point shifted algorithm initially developed by O. McBryan[14]. This method has the benefit of locally perturbing the mesh without changing its topology so that the interface can pass through the grid nodes instead of the cell edges. It is also being adopted in the conservative front tracking method in two space dimension [9].

2. Outline of the Algorithm. Let Φ be defined numerically at each grid node and the states of the underlying partial differential equation are defined at each cell, with the cell corners being the grid nodes. In order to retain the information about the interface without actually reconstruct the interface (which causes loss of high order information depending on the method of reconstruction, which also involves substantial difficult in geometry for implementation), we can anchor the interface position by using the point shifted algorithm right after updating the interface by solving the equation (1.1) numerically but before solving the equation (1.2). The point shifted algorithm moves the nearby grid nodes to the interface and then the zero set of Φ will be represented at these shifted grid nodes. Therefore solving the equation (1.2) numerically on the locally irregular mesh (without mesh topological changes) will not move the zero set of Φ but still reshape Φ into a signed local distance function defined on the shifted new mesh. The point shifted level set method has another benefit that the conservation of each phase now has a clear definition simply because the interface now passes through the cell edges or diagonals leaving each side a whole cell (topologically) or a partial cell (a triangle in 2D).

Now we propose the outline of this method in general space dimensions. Suppose at time level t_n we have a point shifted mesh so that the zero set of Φ^n (the levelset function at time level t_n) are at the shifted grid nodes. The states are represented as cell averages (the corner points of the cell are the grid nodes, where the level set function Φ^n is defined).

We first solve the equation (1.1) from time level t_n to t_{n+1} in order to update the value of Φ^n defined on a point shifted mesh on time level t_n to Φ^{n+1} defined on the un-shifted uniform mesh on time level t_{n+1}. This updating from a locally irregular mesh to a uniform mesh needs a carefully designed difference scheme. But because the meshes have the same topology and each shifted grid nodes at time level t_n is within $\frac{1}{2}\Delta x$ distance away from its original position by property of the point shifted algorithm (to be described below), also because the normal velocity field is continuous

across the interface, the right difference scheme should not be too much difficult to find out.

On time level t_{n+1} we perform the point shifted algorithm again. The point shifted algorithm can be defined as a mapping Y which maps each grid node coordinate to a new grid nod coordinate without changing the mesh topology.

$$Y(x_i) = \begin{cases} x_i, & \text{if } d(x_i) \geq \frac{1}{2}\Delta x, \\ x_i - (d(x_i)/| \bigtriangledown \Phi|) \bigtriangledown \Phi, & \text{otherwise,} \end{cases}$$

where $d(x)$ is the function returns the approximate shortest signed distance of the point x to the interface in the $\bigtriangledown\Phi$ direction, with the same sign as $\Phi(x)$. $d(x)$ is supposed to be determined by the values of Φ at neighboring grid nodes and is a key subject we need to research on. For example, if Φ still represents the signed distance from the interface then $d(x_i) = \Phi(x_i)$. This could save us a lot of work from finding the crossings between the interface and the grid cell edges and geometrically shifting the nodes.

After this point shifted algorithm the mesh on time level t_{n+1} has been locally perturbed and we assign the zero set of Φ^{n+1} on the shifted grid nodes, we can now solve the equation (1.2) to reshape Φ^{n+1} as a signed distance function on the perturbed new mesh on time level t_{n+1}. We can also connect the corresponding (shifted) nodes on the time levels t_n and t_{n+1} to construct the hexahedra for finite volume computation of the cell averages of at time level t_{n+1}. It seems that the finite volume version of the ghost fluid method [4] can also be adapted to use in this case. When we don't enforce strict conservation across the interface, it is no longer a problem for the extreme situation that there are not enough grid nodes to shift in order to remove the crossing between the interface and cell edges. In fact, the levelset function based point shifted algorithm will simply shift the selected grid node to the nearest interface segment. We will discuss more about the different variations after we finish the outline of this algorithm.

Now how to address the dissipation associated with the discretization of the equation (1.1) and define the distance function $d(x)$? Actually these two problems are the same. Because after solving equation (1.1) to update Φ^n, the zero set of Φ^{n+1} is not at the grid nodes, also Φ^{n+1} is not a signed distance function any more even though it starts from one. Therefore the high order information regarding the interface position could easily be damaged (solving equation (1.2) could make things worse if not treated with extreme care, see[18] for a detailed discussion) and we may not be able to approximate $d(x)$ to more than first order. However, if we have a method so that numerically solving equation (1.1) happens to update Φ^n to a signed distance function again, we end up greatly improving the accuracy of the levelset method. The conclusion is that we may not obtain a signed distance function Φ^{n+1} by numerically solving equation (1.1) from an initial signed distance function but we may if equation (1.1) does not start from a signed distance function. Therefore we propose an iterative procedure as follows:

Let Φ^n be the signed distance function defined on a point shifted mesh on time level t_n, $\{x_i^n\}$ be the grid nodes of the point shifted mesh on time level t_n. Let $\Phi_1^n = \Phi^n$ and for $k = 1, 2, \cdots$,

Step 1 Update Φ_k^n to Φ_k^{n+1} which is defined on the original uniform mesh by solving the equation (1.1) from t_n to t_{n+1}.

Step 2 Perform the point shifted algorithm described above for the uniform mesh on time t_{n+1} with $d(x) = \Phi_k^{n+1}(x)$ and assign the zero set of Φ_k^{n+1} on the shifted grid nodes. Now Φ_k^{n+1} is defined on the point shifted mesh on t_{n+1}.

Step 3 Solve equation (1.2) to update Φ_k^{n+1} on the point shifted mesh of time level t_{n+1} (so that it becomes a signed distance function).

Step 4 Solve equation (1.1) backward in time from t_{n+1} to t_n to get $\tilde{\Phi}_k^n$ defined on the point shifted mesh on time level t_n.

Step 5 Define $\Phi_{k+1}^n(x_i^n) = \begin{cases} \tilde{\Phi}_k^n(x_i^n), & \text{if } \Phi_k^n(x_i^n) \cdot \tilde{\Phi}_k^n(x_i^n) > 0, \\ \Phi_k^n(x_i^n), & \text{otherwise.} \end{cases}$

This iteration only needs to perform a few times until the change of Φ_k^{n+1} during the third step is within the given tolerance. In this case, Φ_k^{n+1} becomes the required signed distance function on time level t_{n+1} so that we can denote $\Phi^{n+1} = \Phi_k^{n+1}$.

Notice that all the above virtual geometric operation introduced by the point shifted algorithm are performed analytically. If we don't want to enforce the strict conservation across the interface we can associate the levelset function Φ with another independent mesh for performing the analytic point shift algorithm while the states are being computed in the uniform fixed mesh, therefore the above algorithm should be implemented without too much difficulty.

3. A Uniform Grid Version. Most of computational difficulties of the previous algorithm come from the updating of Φ_k^{n+1} to the signed distance function using equation (1.2) on a point shifted mesh because it is locally irregular. Therefore we come up with another version of the algorithm with uniform fixed mesh on time level t_{n+1} but point shifted mesh on time level t_n. In this situation equation (1.2) will be solved on a uniform mesh and the point shifted algorithm only need to perform once on time level t_n. This also simplify the work of solving equation (1.1) backward in time since the velocity field is supposed to be defined on the uniform fixed mesh.

Let Φ^n be the signed distance function defined on a point shifted mesh on time level t_n, $\{x_i^n\}$ be the grid nodes of the point shifted mesh on time level t_n. Let $\Phi_1^n = \Phi^n$ and for $k = 1, 2, \cdots$,

Step 1 Update Φ_k^n to Φ_k^{n+1} which is defined on the uniform fixed mesh by solving the equation (1.1) from t_n to t_{n+1}.

Step 2 Solve equation (1.2) to update Φ_k^{n+1} on the uniform fixed mesh on time level t_{n+1} (so that it becomes a signed distance function).

Step 3 Solve equation (1.1) backward in time from t_{n+1} to t_n to get $\tilde{\Phi}_k^n$ defined on the point shifted mesh on time level t_n.

Step 4 Define $\Phi_{k+1}^n(x_i^n) = \begin{cases} \tilde{\Phi}_k^n(x_i^n), & \text{if } \Phi_k^n(x_i^n) \cdot \tilde{\Phi}_k^n(x_i^n) > 0, \\ \Phi_k^n(x_i^n), & \text{otherwise.} \end{cases}$

This iteration only needs to perform a few times until the change of Φ_k^{n+1} during the second step is within the given tolerance. In this case, Φ_k^{n+1} becomes the required signed distance function on time level t_{n+1}. Then perform the point shifted algorithm described above for the uniform mesh on time t_{n+1} with $d(x) = \Phi_k^{n+1}(x)$ and assign the zero set of Φ_k^{n+1} on the shifted grid nodes. Now Φ_k^{n+1} is defined on the point shifted mesh on t_{n+1}. Finally let $\Phi^{n+1} = \Phi_k^{n+1}$.

4. Remarks on Interface Topological Changes. Levelset representation on a uniform fixed mesh accelerates interface topological changes in certain situations. It is easy to see this phenomenon for two phase fluid in 2D case. Suppose there is a cell with its grid nodes A, B, C, D oriented clockwisely. An interface segement located in the cell could make $\Phi(A) > 0$, $\Phi(B) > 0$, $\Phi(C) < 0$, $\Phi(D) < 0$. Suppose there is another interface segment which moves into the cell but has not interacted with the first interface segment yet. It could still turn $\Phi(C)$ and $\Phi(D)$ positive, which is interpreted as a topological change from the levelset function Φ. On the other hand,

suppose Q is a grid node surrounded by four grid nodes E, F, G, H clockwisely where $\Phi(E) > 0$, $\Phi(F) < 0$, $\Phi(G) > 0$, $\Phi(H) < 0$ and $\Phi(Q) < 0$. This could be interpreted as that two interface segments are getting very close but are still separated by the grid nodes Q, F and H. In fact they could get as close as possible without topological changes while still having $\Phi(Q) < 0$, $\Phi(F) < 0$ and $\Phi(H) < 0$. Note that if we apply the above analytical point shifted algorithm for the later case, premature interface topological changes could happen. Fortunately this could be easily fixed if we add a few more rules to the point shifted algorithm so that grid nodes like Q's will not be shifted and $\Phi(Q)$'s keep their original values. These extra rules can be set by comparing the sign of Φ at a grid node to the signs of Φ at its neighbors. This modification to the point shift algorithm will not change the algorithms described in the previous sections at all simply because we don't need to reconstruct the interface and what we have done is to move some grid nodes slightly and redefine the levelset function Φ on the new mesh.

5. Acknowledgments. The author is grateful to Dr. Hongkai Zhao in UC Irvine for the discussion on levelset method which helps motivate the current work.

REFERENCES

[1] D. Adalsteinsson and J. A. Sethian. A fast level set method for propagating interfaces. *J. Comp. Phys.*, 118:269–277, 1993.

[2] I-L. Chern and P. Colella. A conservative front tracking method for hyperbolic conservation laws. LLNL Rep. No. UCRL-97200, Lawrence Livermore National Laboratory, 1987.

[3] D. Enright, R. Fedkiw, J. Ferziger, and I. Mitchell. A hybrid particle level set method for improved interface capturing. *J. Comp. Phys.*, (in press).

[4] R. P. Fedkiw, T. Aslam, B. Merriman, and S. Osher. A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method). *J. Comp. Phys.*, 152:457–492, 1999.

[5] J. Glimm, J. W. Grove, X. L. Li, W. Oh, and D. H. Sharp. A critical analysis of Rayleigh-Taylor growth rates. *J. Comp. Phys.*, 169:652–677, 2001.

[6] J. Glimm, J. W. Grove, X.-L. Li, K.-M. Shyue, Q. Zhang, and Y. Zeng. Three dimensional front tracking. *SIAM J. Sci. Comp.*, 19:703–727, 1998.

[7] J. Glimm, J. W. Grove, X.-L. Li, and D. C. Tan. Robust computational algorithms for dynamic interface tracking in three dimensions. *SIAM J. Sci. Comp.*, 21:2240–2256, 2000.

[8] J. Glimm, J. W. Grove, X.-L. Li, and N. Zhao. Simple front tracking. In G.-Q. Chen and E. DiBebedetto, editors, *Contemporary Mathematics*, volume 238, pages 133–149. Amer. Math. Soc., Providence, RI, 1999.

[9] J. Glimm, X.-L. Li, and Y.-J. Liu. Conservative front tracking in higher space dimensions. *Transactions of Nanjing University of Aeronautics and Astronautics*, 18, Suppl.:1–15, 2001. Proceedings of International Workshop on Computational Methods for Continuum Physics and Their Applications (IWCCPA), Nanjing, China.

[10] J. Glimm, X.-L. Li, and Y.-J. Liu. Conservative front tracking in one space dimension. *Contemporary Mathematics*, 2002. Proceedings of the Joint Summer Research Conference on Fluid Flow and Transport in Porous Media: Mathematical and Numerical Treatment. In Press, Report SUNYSB-AMS-01-16.

[11] J. Glimm, X.-L. Li, Y.-J. Liu, and N Zhao. Conservative front tracking and level set algorithms. *Proc. National Academy of Sci.*, 98:14198–14201, 2001.

[12] J. Gomes and O. Faugeras. Advancing and keeping a signed distance function. Technical Report 3666, INRIA, 1999.

[13] A. Harten. High resolution scheme for hyperbolic conservation laws. *J. Comp. Phys.*, 49:357, 1983.

[14] O. McBryan. Elliptic and hyperbolic interface refinement in two phase flow. In J. Miller, editor, *Boundary and Interior Layers – Computational and Asymptotic Methods*. Boole Press, Dublin, 1980.

[15] B. Merriman, J. Bence, and S. Osher. Motion of multiple functions: a level set approach. *J. Comp. Phys.*, 112:334, 1994.

[16] D. Nguyen, F. Gibou, and R. Fedkiw. A fully conservative ghost fluid method & stiff detonation waves. *12th International Detonation Symposium, San Diego, CA*, 2002.

[17] S. Osher and J. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi equations. *Jour. Comp. Phys*, 79:12–49, 1988.

[18] M. Sussman and E. Fatemi. An efficient, interface preserving level set re-distancing algorithm and its application to interfacial incompressible fluid flow. *SIAM J. Sci. Comput.*, 20:1165–1191, 1999.

[19] M. Sussman, P. Smereka, and S. Osher. A level set method for computing solutions to imcompressible two-phase flow. *J. Comput. Phys*, 119:146–159, 1994.

[20] B. Swartz and B. Wendroff. Aztec: A front tracking code based on Godunov's method. *Appl. Num. Math.*, 2:385–397, 1986.

[21] H.-K. Zhao, T. Chan, B. Merriman, and S. Osher. A variational level set approach to multiphase motion. *J. Comput. Phys.*, 127:179–195, 1996.