Abstract

This paper introduces a fully-Eulerian interface tracking framework that preserves the fine details of liquids. Unlike existing Eulerian methods, the proposed framework shows good mass conservation even though it does not employ conventional Lagrangian elements. In addition, it handles complex merging and splitting of interfaces robustly due to the implicit representation. To model the interface more accurately, a high order polynomial reconstruction of the signed distance function is utilized based on a number of sub-grid quadrature points. By combining this accurate polynomial representation with a high-order re-initialization method, the proposed framework preserves the detailed structures of the interface. Moreover, the method is simple to implement, unconditionally stable, and is suitable for parallel computing environments.

CR Categories: I.3.7 [Computing Methodologies]: Computer Graphics—Three Dimensional Graphics and Realism;

Keywords: interface tracking, level set method, fully Eulerian, fluid simulation

1 Introduction

Interface tracking has long been an important issue in a wide variety of areas, such as computer graphics, image processing, and computational fluid dynamics. Interfaces usually experience complex movements such as merging or pinching. As such, tracking interfaces can be a challenging task. In general, there are two important aspects that are closely related to the quality of interface tracking algorithms: topology change and volume conservation. Interface tracking methods are generally categorized as either Eulerian or Lagrangian. Eulerian methods are typically good for handling complex topological changes. However, they tend to suffer from excessive numerical diffusion, which leads to disappearance of detailed structures. In contrast, Lagrangian methods show good mass conservation due to the preservation of characteristic information. However, they usually require complex re-meshing procedures to treat topological changes.

To get over the above weaknesses associated with the Eulerian and Lagrangian methods, several hybrid methods have been developed [Enright et al. 2002a; Mihalef et al. 2007]. Through the use of advected particles, those methods could produce impressive results in terms of mass conservation. However, the methods could experience some aliasing artifacts [Bargteil et al. 2006] and physical inconsistency problems. For example, when the characteristics in a given region are merged, some particles need to be deleted because they represent the situation differently from the Eulerian grids. Although several treatments have been developed [Enright et al. 2003; Rasmussen et al. 2004; Losasso et al. 2006], a completely satisfactory solution has not yet been proposed. Recently, Wang et al. [2009] also pointed out that accuracy of the particle level set method is sensitive to the particle reseeding strategy which varies for different problems.

In this paper, we introduce a fully-Eulerian interface tracking framework that can preserve fine details of the interface. In contrast to existing Eulerian frameworks, which suffer from severe mass loss, the proposed scheme exhibits good mass conservation, comparable to that of Lagrangian counterparts. To capture evolving surfaces more accurately, we employ a pseudo-spectral representation of the level set function [Desjardins and Pitsch 2009] that provides a precise sub-grid description of the interface. To prevent destruction of fine surface details during re-initialization, we propose a high-order PDE-based re-initialization method [Osher and Fedkiw 2002] that can be applied to a spectrally refined grid structure. By combining the above techniques, the signed distance property is preserved more accurately and hence small-scale surface details are well preserved without any Lagrangian computational element. We demonstrate our approach with several challenging liquid simulations that generate highly detailed, complex surfaces (see Figure 1).

2 Related Work

There are three popular frameworks for the interface tracking problem: level set, front tracking, and volume of fluid methods. The

Enright and colleagues [2002a] proposed a hybrid framework, called the particle level set method, that combines the Eulerian and Lagrangian approaches to achieve effective interface capturing. Recently, Wang et al. [2009] proposed several revisions to the particle level set method to remedy some limitations in the original scheme. As other hybrid approaches, several mesh-based interface tracking frameworks [Wojtan et al. 2009; Müller 2009] were proposed to overcome the difficulties associated with re-meshing in the front tracking method. Recently, Wojtan et al. [2010] proposed a new re-sampling method for connecting surface features during topological changes to enhance their original mesh-based surface tracking framework [Wojtan et al. 2009]. In addition, semi-Lagrangian contouring method [Bargeil et al. 2006; Chentanez et al. 2007], which utilizes both a Eulerian grid and triangle mesh structure, was developed for accurate computation of the signed distance, thus alleviating the numerical diffusion caused by linear interpolation. To enhance the surface tracking quality, adaptive methods have also been developed, including the octree [Losasso et al. 2004; Kim et al. 2009] and the hierarchical RLE grid [Houston et al. 2006].

Recently, a number of level set methods based on a finite element framework such as the spectral element method [Sussman and Hussaini 2003] or the discontinuous Galerkin method [Marchandise and Remacle 2006] are presented. In a similar context, Desjardins and Pitsch [2009] proposed a spectrally refined level set (SRL) method to provide a sub-cell representation of the interface through a pseudo-spectral description of the level set function. They demonstrated effectiveness of the proposed method by several experiments including the 2D Zalesak disk and the Enright test. Our work, which has been motivated by Desjardins and Pitsch [2009], suggests an accurate interface tracking framework by combining this pseudo-spectral representation with a high-order re-initialization method [Osher and Fedkiw 2002] which is designed for the SRL grid and further demonstrates the effectiveness of the approach with several challenging three-dimensional liquid simulation experiments that incorporate highly detailed, complex surface geometry (see Figure 8).

In this section, we present a fully-Eulerian interface tracking framework that preserves small details of the interface. Because it does not employ any Lagrangian computational element, it can robustly handle topological changes without any special treatment. Moreover, by utilizing accurate polynomial reconstruction and a high-order PDE-based re-initialization method, the mass conservation quality is comparable to that obtained with Lagrangian frameworks. Furthermore, the proposed approach is well suited to parallel computing environments.

3 Method

In this work, we employ the level set method, which tracks the phase interface in the form of an iso-contour in a higher dimensional space. The level set function $\phi$ is usually represented as a signed distance function such that $|\nabla\phi| = 1$, and the interface is defined as $\phi = 0$. From the signed distance property, various geometric quantities such as normal $\textbf{n} = \nabla\phi/|\nabla\phi|$ and curvature $\kappa = \nabla \cdot \textbf{n}$ can be easily computed.

The interface is evolved in time by a given velocity field $\textbf{u}$ according to the well-known advection equation

$$\phi_t + \textbf{u} \cdot \nabla \phi = 0. \tag{1}$$

The main difficulty associated with evolving the level set function is numerical diffusion, which tends to make small-scale droplets disappear and cause the overall liquid volume to decrease. These phenomena, especially the disappearance of small scale interfacial features, are visually undesirable.

To preserve the fine details of the liquid phase, we deploy a spectrally refined level set (SRL) method [Desjardins and Pitsch 2009] that provides an accurate description of the interface. Instead of using particles or meshes to provide sub-cell representations, SRL introduces a number of fixed grid points near the interface. Using these pre-assigned sub-cell computational elements, an accurate polynomial reconstruction of the signed distance function becomes possible. To avoid Runge phenomena [Press et al. 2007], SRL utilizes Gauss-Lobatto quadrature points (Figure 2).

After quadrature points have been properly defined ($r_0 = 0, r_1, \ldots, r_{N-1} = 1$), we can construct a high-order polynomial description of the level set function for each cell. For any given point, the interpolation weights are determined using the classical Lagrange polynomial. The multi-dimensional Lagrange interpolation can also be performed efficiently by the dimensional splitting.
In this paper, we present a high-order PDE-based re-distancing algorithm [Sethian 1995] and PDE-based methods [Peng et al. 1999; Peng et al. 2001; Kim et al. 2008], because of the lack of accuracy. In the SRL framework, traditional trilinear interpolation is replaced with high-order polynomial reconstruction [Desjardins and Pitsch 2009]. It reduces numerical diffusion significantly compared to the original semi-Lagrangian method (Figure 3).

In contrast to the original semi-Lagrangian method (which is monotonic), high-order interpolation sometimes introduces a non-monotonic solution. This new extremum may give rise to numerical instability [Selle et al. 2008]. To avoid such instabilities, we apply a simple limiting technique similar to that suggested by Selle et al. [2008]: when the interpolated result is larger/smaller than the maximum/minimum of the sub-grid values, we perform trilinear interpolation in the sub-grid. According to our experiments, such limiting is rarely required (below 1.5%), and hence has only a small impact on the overall accuracy.

3.3 Accurate Re-distancing Algorithm

Re-distancing (i.e., re-initialization) is an essential step for accurate level set tracking. When interfaces are evolved via the advection process, it is almost impossible to maintain the signed distance property exactly. This inability to maintain the signed distance property usually degrades the overall accuracy of advection and the quality of geometric quantities such as normals and curvatures. In order to produce visually plausible fluid simulation, therefore, an accurate re-distancing algorithm is also required. Because of its importance, various re-distancing algorithms have been actively developed in CFD communities, including the fast marching method [Sethian 1995] and PDE-based methods [Peng et al. 1999; Osher and Fedkiw 2002].

In this paper, we present a high-order PDE-based re-distancing algorithm for a spectrally refined level set method. Compared to the mesh-based direct re-distancing algorithm [Desjardins and Pitsch 2009], PDE-based methods are more flexible to extend to high-order. The PDE-based re-distancing equation is given by

\[ \phi_t + S(\phi) (\nabla \phi - 1) = 0 \]  \hspace{1cm} (6)

where \( S(\phi) \) is a sign function which takes one in positive regions, minus one in negative regions, and zero at the interface [Osher and Fedkiw 2002]. For accurate integration of Equation 6, the spatial derivative \( \nabla \phi \) needs to be computed with some precision. In the adaptive or unstructured grid structure, however, an accurate high-order upwind method such as ENO or WENO is not efficiently performed compared to its performance in a regular grid [Wolf 2007; Cecil et al. 2008]. Because of the variable stencils, explicit formula for derivative computation does not exist in contrast to the regular grid structure [Osher and Fedkiw 2002]. Especially, the most popular adaptive data structure, the octree, has various types of variable stencils, making it difficult to provide explicit mathematical formulas for efficient computations.

In the SRL grid, however, the situation is slightly different. After the number of quadrature points is decided, because the stencil between two neighboring quadrature points is fixed, the computation of coefficients can be done as a pre-process. This reduces the run-time computation costs significantly, and makes it viable to employ ENO/WENO in the SRL grid as in the regular grid. During the re-initialization, high-order derivative computations are performed only in the narrow band that contains the sub-grid quadrature points (Figure 2 (b)). Outside of this band, simple 1st-order upwind method is used. Because the re-distancing process propagates information originating from the interface, limiting the high-order computation in the vicinity of the interface has little impact on the overall accuracy.

Compared to geometric algorithms such as the direct method based on an extracted mesh [Desjardins and Pitsch 2009] or the fast marching method [Sethian 1995], high-order PDE-based re-initialization has advantages in the aspect of preserving detailed interface structures (Figure 4). While the geometry based algorithms are hard to extend to high order, the PDE-based method is more flexible in this regard, requiring modifications only in the derivative computations. We provide a detailed derivation of 3rd-order WENO for variable stencils in Appendix A. A general framework for deriving WENO coefficients is presented by Chi-Shu [1997] and a detailed description about 5th-order variable WENO can be found in Wang et al. [2008]. In Appendix B, we make several revisions to the 5th-order variable WENO method of Wang et al. [2008]. (The revisions fix some errors in the original derivation.)
The suggested re-initialization framework is computationally more expensive than the coarse grid level re-initialization [Desjardins and Pitsch 2009]. However, the proposed framework can be easily parallelized due to its fully-Eulerian nature, and does not have to be performed as frequently as the advection process. In addition, because the coarse grid-level re-initialization can destroy sub-grid interfacial features, the more accurate re-initialization in the sub-grid structure achieved by the proposed method justifies the use of more computational resources.

Determining the optimal re-initialization frequency is an important issue for which, to date, no general solution has been proposed [Peng et al. 1999]. Too frequent re-initialization is known to be detrimental to the overall numerical accuracy. However, too infrequent re-initialization can incur some difficulties of maintaining the signed distance function. Actually, when we tested a 2D liquid simulation using the SRL framework, we observed that too infrequent re-initialization gave rise to some deterioration of the accuracy of the whole method. In the fluid scenes which have complex velocity fields, if the interface travels longer than 30L without re-initialization, the signed distance property was hard to preserve even though the SRL has good advection accuracy.

Intuitively, we expect that a rapidly evolving interface can easily deviate from the signed distance function. Hence, we set up a simple policy based on Eulerian movements of the interface. When the interface evolves by ∆t, the maximum Eulerian movements of the level set function with respect to the grid can be evaluated as ∆t(|ux|max/∆x + |uy|max/∆y + |uz|max/∆z) (i.e., CFL). If the accumulated Eulerian movement of the interface exceeds a user-defined threshold τ or the number of advections exceeds α, we perform re-initialization (We used 15 and 40 for the values of τ and α, respectively).

4 Results

In all experiments reported in this section, we used a Legendre polynomial to determine the Gauss-Lobatto quadrature points, and used five for the number (N) of nodes for one axis. We employ 2nd-order Runge-Kutta method for backtracking of the semi-Lagrangian method. In addition, we successfully achieved significant speedups by parallelization using OpenMP directives; Our implementation achieved about x3.7 and x7.3 speedups for 4 and 8 processors, respectively. The test was performed on Dell Precision T5500 which is an oct-core machine. Due to the fully-Eulerian nature, the proposed framework is readily parallelizable and shows good scalability.

![Figure 5: 3D Zalesak sphere test. The sphere completes one revolution in 628 time steps. (a) Using semi-Lagrangian contouring with effective resolution of 512^3. (b) Using the spectrally refined level set method with a 128^3 SRL grid. (c) Ground truth. (d) Using the spectrally refined level set method with a 100^3 SRL grid.](image)

![Figure 6: Comparison of re-initialization methods. The re-initialization is performed at every time step. The bunny experiences 950 advections and re-initializations in a 128^3 SRL grid. Total volume changes are 8.2%, 3.0%, and 1.7%, respectively. (a) Direct re-distancing based on an extracted mesh from the refined grid. (b) PDE-based re-distancing based on 3rd-order variable WENO. (c) PDE-based re-distancing based on 5th-order variable WENO.](image)
there are several interface tracking steps for every velocity update. In our demonstrations, we put different CFL restrictions for fluid simulation and all the other steps of the fluid solver. The reported total computational time taken for fluid simulation, surface advection, and re-initialization were about 20%, 45% and 35% of the total computational time over the course of simulations, respectively. The numbers of simulation time-steps per one animation frame (1/30 sec) were about 6.8, 6.0 and 12.6 for Figures 1, 8, and 9, respectively. In our demonstrations, we put different CFL restrictions for fluid simulation and interface tracking (about 3.5 and 1.5, respectively). So, there are several interface tracking steps for every velocity update.

We experimented the proposed interface tracking framework for the liquid simulations shown in Figures 1, 8, and 9, in which inviscid incompressible free-surface flow was used to model the liquid dynamics [Enright et al. 2002b]. We used 1st-order semi-Lagrangian method [Stam 1999] for velocity advection, and the ghost-fluid method [Enright et al. 2003] for projection and surface tensions. The velocity at the interface is extrapolated to the air region using the PDE-based extrapolation method [Peng et al. 1999]. A PDE-based re-distancing algorithm based on 5th-order variable WENO scheme is employed to reinitialize the level set.

Due to the fully-Eulerian nature, the proposed method captured complex liquid flows which experience lots of merging and splitting without any special treatment. Compared to conventional Eulerian methods, numerical diffusions are less noticeable. All demonstrations were performed on an Intel Core i7 940 2.93 GHz processor (which is a quad-core CPU) with 6 GB of memory. The time measurements reported in this section include the interface tracking and all the other steps of the fluid solver. The reported total computational time taken for fluid simulator, surface advection, and re-initialization were about 20%, 45% and 35% of the total computational time over the course of simulations, respectively. The numbers of simulation time-steps per one animation frame (1/30 sec) were about 6.8, 6.0 and 12.6 for Figures 1, 8, and 9, respectively. In our demonstrations, we put different CFL restrictions for fluid simulation and interface tracking (about 3.5 and 1.5, respectively). So, there are several interface tracking steps for every velocity update.

Overall volume errors for each re-initialization method are shown in Figure 7.

### 5 Conclusion

In this paper, we presented a new fully-Eulerian interface tracking framework for high-quality liquid simulation. In the absence of Lagrangian computational elements, the proposed framework could robustly handle topological changes without special treatments. To avoid numerical diffusion caused by 1st-order semi-Lagrangian method, we employed a pseudo-spectral representation of the level set function [Desjardins and Pitsch 2009] which provides a sub-grid description of the interface. To prevent destruction of the fine surface details during re-initialization, we proposed a high-order PDE-based re-initialization method [Osher and Fedkiw 2002] which is tailored to the spectrally refined grid structure. By reducing numerical diffusion in advection and re-initialization processes, detailed interface structures were preserved quite accurately. Despite of its fully-Eulerian nature, the suggested framework exhibited good mass conservation comparable to that achieved by existing Lagrangian-based surface tracking frameworks. Finally, the proposed method is suitable for parallel computing environments, which is becoming an attractive feature considering the ongoing development of more powerful parallel devices such as many-core processors [Seiler et al. 2008].

Our PDE-based re-distancing method is extremely accurate and is easily parallelized and extended to high-order. However, it is far more computationally expensive than other, less accurate techniques, such as the 1st-order fast marching method [Sethian 1995]. In our experiments, we have found the additional accuracy worth the cost. However, we have not done extensive experimentation with other techniques and, given that re-distancing isn’t performed as frequently as advection, it may be the case that a faster and less accurate method will prove sufficient in practice, especially in sequential computing environments. Further investigation of re-distancing methods in the context of the spectrally refined grid is an interesting area of future work.

Another possible avenue of future work would be to integrate the fully-Eulerian surface tracking framework with Lagrangian mass particles [Song et al. 2005]. Because of the inherent limitation of the grid, fluid or air features which are smaller than a grid size cannot be represented as a distance function. So, combining with Lagrangian mass particles can improve the visual quality of the animation. In addition, more detailed numerical properties of the spectrally refined grid are worth to investigate. The level set method usually exhibits 1st-order accuracy with topology changes because the interface simply merge regions together when they are smaller than a single grid spacing. Due to the variable-sized stencil, the
widest grid spacing in the spectrally refined grid is about 1.309 times larger than that of a regular grid. So, the numerical accuracy can be degraded when topology changes are occurred. In addition, the amount of deterioration of the numerical accuracy near the stiff regions such as thin fluid features is also worth to investigate.

Acknowledgements

This work was supported by Ministry of Culture, Sports and Tourism (MCST) and Korea Culture Content Agency (KOCCA) in the Culture Technology(CT) Research & Development Program 2010, Ministry of Science and Technology under National Research Laboratory (NRL) grant 20100008010, the Brain Korea 21 Project in 2010, Seoul Research and Business Development Program (10557), the IT R&D program of MKEMCSTIITA (ITAA1100090201260001000100100), Development of Digital Clothing Software Technology), and ASRI (Automation and Systems Research Institute at Seoul National University).

References


Appendix A : Derivation of 3rd-order variable WENO

When calculating \( \phi_{x}^{(1)} \), or \( \phi_{x}^{(1)}(i+1) \), the 2nd-order upwind method uses a subset of \( \{ \phi_{x}^{(0)}, \phi_{x}^{(1)}, \phi_{x}^{(i+1)}, \phi_{x}^{(i+2)} \} \) that depends on how the stencil is chosen. There are two 2nd-order upwind approximations of \( \phi_{x}^{(1)} \), and \( \phi_{x}^{(i+1)} \), respectively, \( \phi_{x} = x_{i+1} - x_{i+2} \)

\[
\begin{align*}
\phi_{x}^{(0)} &= (b_{01}h_{x} v_{2} + (b_{10}h_{x})v_{3} ) \\
\phi_{x}^{(1)} &= (b_{10}h_{x}v_{1} + (b_{11}h_{x})v_{2} ) \\
\phi_{x}^{(i+1)} &= (b_{01}h_{x} v_{2} + (b_{10}h_{x})v_{3} ) \\
\phi_{x}^{(i+2)} &= (b_{10}h_{x}v_{1} + (b_{11}h_{x})v_{2} )
\end{align*}
\]

\[
\begin{align*}
b_{01} &= b_{01} = \frac{b_{2} - b_{3}}{b_{2} + b_{3}} , \\
b_{10} &= b_{10} = \frac{b_{1} - b_{2}}{b_{1} + b_{2}} \\
b_{11} &= b_{11} = \frac{b_{3}}{b_{1} + b_{2}} \\
b_{20} &= b_{20} = \frac{b_{2} + b_{3}}{b_{2} + b_{3}} \\
b_{22} &= b_{22} = \frac{1}{b_{1} + b_{2}}
\end{align*}
\]

The 3rd-order WENO approximation of \( \phi_{x}^{(0)} \), or \( \phi_{x}^{(i+1)} \), is defined as:

\[
\begin{align*}
\phi_{x}^{(0)} &= \omega_{0} \phi_{x}^{(0)} + \omega_{1} \phi_{x}^{(1)} \\
\phi_{x}^{(i+1)} &= \omega_{0} \phi_{x}^{(i+1)} + \omega_{1} \phi_{x}^{(i+2)}
\end{align*}
\]

where \( 0 \leq \omega_{0}, \omega_{1} \leq 1 \) with \( \omega_{0} + \omega_{1} = 1 \).

In order to obtain high-order accuracy in smooth regions, we should estimate the smoothness of the stencils in Equation 7, 8, 9, 10. The smoothness measures \( S_{0}, S_{1} \) are provided by:

\[
\begin{align*}
S_{0} &= b_{2}^{10} ((b_{01}h_{x}v_{2} + (b_{10}h_{x})v_{3} )^{2} \\
S_{1} &= b_{2}^{10} ((b_{10}h_{x}v_{1} + (b_{11}h_{x})v_{2} )^{2} \\
B_{0}^{10} &= B_{1}^{10} - 2 / (b_{1} + b_{2}) \\
B_{1}^{10} &= 2 / ((b_{1} + b_{2})h_{x})
\end{align*}
\]

The convex combination weights that achieve the optimal 3rd-order accurate approximation of \( \phi_{x}^{(0)} \), and \( \phi_{x}^{(i+1)} \), are computed as:

\[
\begin{align*}
\omega_{0} &= \frac{d_{0}}{d_{1}} = \left( \frac{h_{x}}{b_{1} + b_{2}} \right) , \\
\omega_{1} &= \frac{d_{1}}{d_{0}} = \left( \frac{h_{x}}{b_{1} + b_{2}} \right)
\end{align*}
\]

From the smoothness measures and the optimal combination weights, we can define \( \omega_{0}, \omega_{1} \) as follows (\( \epsilon = 10^{-8} \)).

\[
\begin{align*}
\omega_{0} &= \alpha_{0} / (\alpha_{0} + \alpha_{2}) , \\
\omega_{1} &= \alpha_{2} / (\alpha_{0} + \alpha_{2})
\end{align*}
\]

As shown above, all coefficients for derivative computations can be pre-computed after the stencils (i.e., \( b_{1}, b_{2}, b_{3} \)) are determined.

Appendix B : Derivation of 5th-order variable WENO

In this section, we note that the 5th-order variable WENO method described in Wang et al. [2008] needs some corrections.

Equation (32) should be fixed as:

\[
\begin{align*}
b_{22} &= \frac{(b_{1} + b_{2})h_{x}}{(b_{1} + b_{2} + b_{3})(b_{1} + b_{2} + b_{3})} \\
b_{20} &= b_{20} + \frac{1}{b_{1} + b_{2} + b_{3}} - \frac{1}{b_{3}} \\
b_{20} &= \frac{(b_{1} + b_{2})h_{x}}{(b_{1} + b_{2} + b_{3})(b_{1} + b_{2} + b_{3})} \\
b_{11} &= b_{11} + \frac{1}{b_{2} + b_{3}} - \frac{1}{b_{2} + b_{3}}
\end{align*}
\]

Equation (44) and the equation between (44) and (45) should be fixed as:

\[
\int_{x_{i-1/2}}^{x_{i+1/2}} h_{3} (P_{x}^{(1)}(x))^{2} dx
\]

\[
= \frac{1}{6} h_{3} (P_{x}^{(1)}(x_{i-1/2}))^{2} + 4 P_{x}^{(1)}(x_{i-1/2})^{2} + P_{x}^{(1)}(x_{i+1/2})^{2}
\]

and

\[
P_{x}^{(1)}(x) = \sum_{j=0}^{2} a_{j} C_{j}(x) u_{i-r+j}
\]

Equation (51) should be fixed as:

\[
B_{i2}^{(1)}(x_{i+1/2}) = \frac{1}{4} (b_{3} + 2b_{4}) - (4b_{3} + 2b_{4}) (b_{3} + b_{4} + b_{5}) (b_{3} + b_{4} + b_{5})
\]