A Fast and Accurate Semi-Lagrangian Particle Level Set Method [*]

Douglas Enright [†]　　　Frank Losasso [‡]　　　Ronald Fedkiw [‡]

October 8, 2003

Abstract

In this paper, we present a fast semi-Lagrangian based particle level set method for the accurate capturing of interfaces. This method retains the robust topological properties of the level set method without the adverse effects of numerical dissipation. Both the level set method and the particle level set method typically use high order accurate numerical discretizations in time and space, e.g. TVD Runge-Kutta and HJ-WENO schemes. We demonstrate that these computationally expensive schemes are not required, rather fast, low order accurate numerical schemes suffice. That is, the addition of particles to the level set method not only removes the difficulties associated with numerical dissipation, but also alleviates the need for computationally expensive high order accurate schemes. We use a fast, first order accurate semi-Lagrangian advection scheme coupled with a first order accurate fast marching method to evolve the level set function. To accurately track the underlying flow characteristics, the particles are evolved with a second order accurate method. Since we avoid complex high order accurate numerical methods, extending the algorthim to arbitrary data strutures becomes more feasible, and we show preliminary results obtained with an octree-based adaptive mesh that captures the small scale features of the interface.

[†]Mathematics Department, UCLA, Los Angeles, CA 90095.
[‡]Computer Science Department, Stanford University, Stanford, CA 94305.

1 Introduction

Standard Eulerian advection algorithms, e.g. HJ-(W)ENO methods [17, 8] combined with TVD-based high order accurate Runge-Kutta schemes [17], require a strict bound on the maximum possible time step due to a stability-based CFL criterion. On the other hand, particles are not restricted by this stability criterion, rather the size of the time step used can be based solely on the degree of numerical accuracy desired. Grid based semi-Lagrangian advection methods are likewise not limited by a stability-based CFL condition, since each grid point is treated in a "particle-like" manner. However, semi-Lagrangian schemes can suffer from large amounts of numerical dissipation, making their use problematic. HJ-(W)ENO methods experience far less numerical dissipation due to their high order accurate adaptive nature. In order to take advantage of the stability afforded by particle based methods, the spatial and temporal coherency of Eulerian methods, and the opportunity for selective adaptive mesh refinement near the interface in order to resolve small scale features as discussed below, we propose to couple together a semi-Lagrangian advection scheme with a characteristic-based particle method to track a passively advected interface.

A flexible and easy-to-implement interface tracking technique is the level set method of Osher and Sethian [11]. By storing the distance to the interface at each point on a fixed computational grid, handling gross changes to interface topology, e.g. pinching and merging, becomes trivial as compared to standard Lagrangian techniques [24] which force the user to explicitly evolve and fix a moving interface mesh each time step. By avoiding these difficulties and utilizing well established numerical algorithms for the solution of nonlinear hyperbolic conservation laws, level set methods have been applied to a wide variety of problems including fluid mechanics, computer vision, material science, and computer graphics. One difficulty with the use of the level set method is the need to control the amount of numerical diffusion (or mass loss) present in the method, especially in areas of high curvature and long, thin filamentary regions. Various authors [23, 21, 22] have attempted to correct this problem by reinitializing the level set function, ϕ, to be signed distance to the interface after each time step. High order accurate TVD Runge-Kutta and HJ-(W)ENO techniques can be used to perform this reinitialization. While producing reasonable results, these methods suffer from the same Eulerian-based advection issues mentioned above such as small time step restrictions.

An alternative characteristic-based diffusion correction technique, the particle level set method [5], has been recently proposed. In this method,

2

two sets of marker particles are placed near the interface, one set associated with the interior ($\phi \leq 0$) region, and the other with the exterior ($\phi > 0$) region. Errors due to numerical dissipation can then be identified when interior particles appear in the exterior region or exterior particles appear in the interior region. Since the particles are able to more accurately track the underlying flow characteristics, these "escaped" particles can be used to correct the level set representation of the interface. The particle level set method has been shown to possess excellent volume conservation properties and a high degree of geometrical accuracy in tracking contact discontinuities, comparable to other interface methods including Volume-Of-Fluid (VOF) and explicit front tracking. At the same time the method maintains the highly desirable topological properties and ease-of-implementation of the original level set method. As an example of the flexibility of the particle level set method, its use in modeling complex three dimensional water surfaces can be seen in [6, 7].

In [5], HJ-WENO numerical methods were used to evolve and reinitialize ϕ. Reinitialization of ϕ was used in order to assist the particles in obtaining an accurate distance to the interface. Due to the HJ-WENO advection scheme used, a stability based CFL condition was imposed on the size of the time step. Use of this scheme in an adaptive mesh setting would impose a severe time step restriction on a simulation, discouraging the use of selective mesh adaptation to resolve small scale features. Due to the minimal amount of diffusion exhibited by the particle level set method, we can avoid the common, but computationally costly, approach of adaptively resolving the mesh just for the sake of minimizing numerical diffusion at the interface. Moreover, since the particles dictate a sharp and geometrically accurate interface, we propose to use a first order accurate semi-Lagrangian advection method [4, 14, 18]. Despite being unconditionally stable, use of a first order accurate sem-Lagrangian scheme has typically been shunned due to the large amount of numerical diffusion inherently incurred. Although a low order accurate semi-Lagrangian scheme was used by [19] for level set advection, computationally expensive higher order accurate semi-Lagrangian methods are usually preferred, see [20]. Due to the observed excellent diffusion limiting properties of the particles, we illustrate that the fast first order accurate semi-Lagrangian scheme is sufficient. Also, we replace the HJ-WENO reinitialization scheme with a fast, i.e. $O(N \log N)$, first order accurate marching technique first proposed by Tsitsiklis [25] and later popularized by Sethian and co-workers, see e.g. [16].

None of the methods we propose are bound by a grid based CFL stability condition, rather only numerical accuracy needs to be taken into account.

The resulting numerical method is a computationally fast and geometrically accurate interface tracking technique that efficiently provides for the adaptive resolution of small scale features. We demonstrate this last claim by showing some preliminary computations on an octree data structure.

2 Numerical Method

2.1 Level Set Method

The underlying idea behind level set methods is to embed an interface Γ which bounds a region $\Omega \subset R^3$ as the zero level set of a higher dimensional function $\phi(\vec{x}, t)$. The level set function has the following properties,

$$\phi(\vec{x}, t) > 0 \qquad \text{for } \vec{x} \notin \Omega$$
$$\phi(\vec{x}, t) \leq 0 \qquad \text{for } \vec{x} \in \Omega,$$

where we include $\phi = 0$ with the negative ϕ values. Then the interface lies in between $\phi > 0$ and $\phi = 0$, but can of course be identified as $\phi = 0$. Note that ϕ is a scalar function in R^3 which greatly reduces the complexity of describing the interface, especially when undergoing topological changes such as pinching and merging.

The motion of the interface is determined by a velocity field, $\vec{u}$, which can depend on a variety of things including position, time, geometry of the interface, or be given externally for instance as the material velocity in a fluid flow simulation. In most of the examples below, the velocity field is externally given, and the evolution equation for the level set function is given by

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

In order to allow for a computationally efficient implementation, we solve this equation locally near the interface in a manner similar to [1, 12]. We solve equation 1 in a region of $\pm 5 \max(\Delta x, \Delta y)$ of the interface. The size of this region is chosen such that the $\phi = 0$ level set is not advected outside this region in one semi-Lagrangian time step as discussed below.

It is convenient to initialize ϕ to be a signed distance function with $|\nabla \phi| = 1$. This ensures that the level set is a smoothly varying function well suited for accurate numerical computations. Unfortunately, as noted in [23], the level set function can quickly cease to be a signed distance function especially for flows undergoing extreme topological changes. Reinitialization algorithms maintain the signed distance property by solving to steady state (as fictitious time $\tau \to \infty$) the equation

$$\phi_\tau + \text{sgn}(\phi_0)(|\nabla \phi| - 1) = 0 \tag{2}$$

where $\text{sgn}(\phi_0)$ is a one-dimensional smeared out signum function approximated numerically in [23] as

$$\text{sgn}(\phi_0) = \frac{\phi_0}{\sqrt{\phi_0^2 + (\Delta x)^2}}.$$

Efficient ways to solve equation 2 to steady state via fast marching methods are discussed in [15, 16]. Again, equation 2 only needs to be solved locally near the interface. We compute the solution to equation 2 in the same region near the interface as we solve for equation 1 in.

Geometric quantities are easily calculated using the level set function, including the unit normal,

$$\vec{N} = \frac{\nabla \phi}{|\nabla \phi|}, \tag{3}$$

and the curvature,

$$\kappa = \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right). \tag{4}$$

The spatial derivatives in equations 3 and 4 can be calculated using standard second order accurate central differencing when the denominators are non-zero. Otherwise, one sided differencing is used.

2.2 Particle Level Set Method

The particle level set method [5] is a thickened front tracking approach which uses particles to assist the level set method in accurately tracking flow characteristics in under-resolved regions (and consequently preserve volume). This is achieved through the placement of massless marker particles near the interface as a diffusion 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 $.1 \min(\Delta x, \Delta y)$ and maximum radius of $.5 \min(\Delta x, \Delta y)$ 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:

$$r_p = \begin{cases} r_{max} & \text{if } s_p\phi(\vec{x}_p) > r_{max} \\ s_p\phi(\vec{x}_p) & \text{if } r_{min} \leq s_p\phi(\vec{x}_p) \leq r_{max} \\ r_{min} & \text{if } s_p\phi(\vec{x}_p) < r_{min}, \end{cases} \tag{5}$$

6

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 interface.

2.2.1 Semi-Lagrangian Advection

In the original particle level set method, high order accurate numerical schemes, i.e. 3rd order accurate TVD Runge-Kutta [17] in time and 5th order accurate Hamilton-Jacobi WENO [8] for the advection term, were used to evolve level set function. Also, the particles were integrated forward in time with a 3rd order accurate TVD Runge-Kutta method with bilinear interpolation used to calculate the velocity of a given particle on the computational grid. We replace the high order advection and time integration schemes with a fast first order accurate semi-Lagrangian method, e.g. given a computational grid, $\vec{x}_{i,j} = (i\Delta x, j\Delta y)$, and temporal discretization, $t^n = n\Delta t$,

$$\phi_{i,j}^{n+1} = \alpha\beta\phi_{r+1,s+1}^n + (1-\alpha)\beta\phi_{r,s+1}^n + \alpha(1-\beta)\phi_{r+1,s}^n + (1-\alpha)(1-\beta)\phi_{r,s}^n, \quad (6)$$

where

$$r = i - \left\lceil u_{i,j}\frac{\Delta t}{\Delta x} \right\rceil, \qquad \alpha = \left(\frac{(i-r)\Delta x - u_{i,j}\Delta t}{\Delta x} \right), \tag{7}$$

$$s = j - \left\lceil v_{i,j}\frac{\Delta t}{\Delta y} \right\rceil, \qquad \beta = \left(\frac{(j-s)\Delta y - v_{i,j}\Delta t}{\Delta y} \right), \tag{8}$$

and $\vec{u}(\vec{x}_{i,j}) = (u_{i,j}, v_{i,j})$. This method is unconditionally stable due to the linear interpolation of ϕ in equation 6 and is convergent to the correct solution according to the Lax-Richtmyer theorem. Further discussion of this scheme can be found in [19]. Due to the unconditional stability of this scheme, the size of the time step is not governed by a stability-based CFL condition, and we use a CFL number of 4.9 (i.e. $\Delta t(|u|_{max}/\Delta x + |v|_{max}/\Delta y) = 4.9$) in the examples section.

Adequate resolution of the underlying flow field by the particles is necessary in order to maintain an accurate representation of the interface. We have found that a second order accurate Runge-Kutta midpoint rule is required for the time integration of the particles. *In fact, lowering the time integration of the particles from second order to first order does have an adverse effect on the numerical results, even though replacing the high order accurate integration of the level set equation with the first order accurate*

semi-Lagrangian and fast marching methods seems to have little effect. Bi-linear interpolation is again used to calculate the velocity of a particle on the computational grid. The time step used for the particles is the same as the one used for the level set advection.

2.2.2 Error Correction

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 interface 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, i.e. after the advection step and after the reinitialization step.

Identification of Error When particles appear on the wrong side of the interface by more than their radius, we indicate the presence of an error in the level set representation of the interface. These particles are said to have *escaped*. In smooth, well resolved regions of the flow where the level set method is accurate, the particles do not drift an appreciable amount across the interface, so we do not use the particle representation of the interface in this region. Only when the semi-Lagrangian advection of 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.

$$\phi_p(\vec{x}) = s_p(r_p - |\vec{x} - \vec{x}_p|). \tag{9}$$

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 the manitude of potential errors in the level set solution.

Error Correction We use the escaped positive particles to rebuild the $\phi > 0$ region and the escaped negative particles to rebuild the $\phi \leq 0$ region. For example, take the $\phi > 0$ region and an escaped positive particle. Using equation 9, the ϕ_p values of the eight grid points on the boundary of the cell containing the particle are calculated. Each ϕ_p is compared to the local value of ϕ and the maximum of these two values is taken as ϕ^+. This is done for all escaped positive particles, creating a reduced error representation of

the $\phi > 0$ region. That is, given a level set ϕ and a set of escaped positive particles E^+, we initialize ϕ^+ with ϕ and then calculate

$$\phi^+ = \max_{\forall p \in E^+} (\phi_p, \phi^+).$$ (10)

Similarly, to calculate a reduced error representation of the $\phi \leq 0$ region, we initialize ϕ^- with ϕ and then calculate

$$\phi^- = \min_{\forall p \in E^-} (\phi_p, \phi^-).$$ (11)

ϕ^+ and ϕ^- will not agree due to the errors in both the particle and level set methods as well as interpolation errors, etc. We merge ϕ^+ and ϕ^- back into a single level set by setting ϕ equal to the value of ϕ^+ or ϕ^- which is least in magnitude at each grid point,

$$\phi = \begin{cases} \phi^+ & \text{if } |\phi^+| \leq |\phi^-| \\ \phi^- & \text{if } |\phi^+| > |\phi^-|. \end{cases}$$ (12)

The minimum magnitude is used to reconstruct the interface (instead of, for example, taking an average), since it gives priority to values that are closer to the interface.

2.2.3 Reinitialization and Radii Adjustment

ϕ is maintained to be a signed distance function by solving equation 2 via a fast marching technique. Again, for the sake of efficiency, we only reinitialize ϕ within a band of the interface. Combining this narrow banding optimization with the fast marching method provides for a very fast reinitialization procedure. To ensure proper ϕ values for the semi-Lagrangian update, we reinitialize ϕ within a band of $\pm 6 \max(\Delta x, \Delta y)$ of the interface. This procedure is performed after each combined semi-Lagrangian update and error correction step. Unfortunately, reinitialization may cause the zero level set to move, which is not desirable, so we use the particles to correct these errors as well. Finally, the particles resample their position relative to the $\phi = 0$ level set and adjust their radii accordingly. Any particles which remain escaped have their radius set to the minimum particle radius value.

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 the fast marching method to reinitializae ϕ in a band near the interface, again correct errors in the level set function using particles, and finally adjust the particle radii.

Advection Method + Reinitialization Method	Area	% Area Loss	CPU Time (sec)
exact	582.2	-	-
HJ-WENO + HJ-WENO	604.4	-4.08%	86.36
HJ-WENO + FMM	181.3	68.78%	11.39
SL + HJ-WENO	328.7	43.40%	7.20
SL + FMM	298.8	48.55%	.45

Table 1: Zalesak's disk. Level set only method comparing different advection and reinitialization schemes on a 100x100 computational cell grid after one rotation.

3 Examples

3.1 Rigid Body Rotation of Zalesak's Disk

Consider the rigid body rotation of Zalesak's disk in a constant vorticity velocity field [26]. The initial data is a slotted circle centered at (50,75) with a radius of 15, a width of 5, and a slot length of 25. The constant vorticity velocity field is given by

$$
\begin{aligned}
u &= (\pi/314)(50 - y), \\
v &= (\pi/314)(x - 50),
\end{aligned}
$$

so that the disk completes one revolution every 628 time units.

To better understand the ability of the various coupled advection and reinitialization algorithms to accurately resprésent a passively advected interface, a comparison of a level set only method utilizing these algorithms has been performed. The use of the expensive, but accurate HJ-WENO schemes in a level set only method is clearly justified by the results seen in table 1 and figure 1. However, the ability of an HJ-WENO advection scheme to accurately advect the interface can be severely impaired when coupled with a low order accurate reinitialization method such as the fast marching method as seen in figure 1(b). The low order accruate errors introduced by the use of a fast marching reinitialization method are compounded due to the small time step used during the HJ-WENO advection phase, reducing the overall accuracy of the coupled method. Semi-Lagrangian based methods minimize this effect due to the larger time step allowed. That is, the larger time step means that the diffusion errors from the spatial mis-approximation of the interface are applied less often.

Figures 2 and 3 compare the evolution of a high order accurate level set only method (3rd order TVD RK in time and 5th order HJ-WENO in space), the original high order accurate particle level set method, and the newly proposed fast semi-Lagrangian particle level set method (coupled to a fast marching method), after one and two revolutions, respectively. The exact solution is also plotted for the sake of comparison. As expected, the level set only method applies an excessive amount of regularization in the sharp corners which the particles correct. The ability of the particles to maintain sharp features, even when a highly diffusive first order accurate semi-Lagrangian advection scheme is combined with a first-order accurate fast marching reinitialization method, is quite remarkable. We note that to obtain reasonable results on the coarse computational grid shown, a second order Runge Kutta midpoint scheme is used for the particle advection. Use of a higher-order accurate time integration scheme for the particles or a HJ-WENO based advection and/or reinitialization scheme did not significantly add to the quality of the solution already obtained.

Tables 2 and 3 compare the area loss (or gain) of the original high-order accurate particle level set method to the newly proposed semi-Lagrangian based particle level set method on three different grids. A CFL number of 4.9 is used on all grids for the semi-Lagrangian calculation, while a CFL number of .5 is used in all HJ-WENO based calculations. The area is calculated using a second order accurate unbiased level set contouring algorithm [3]. In addition, we calculate the accuracy of the interface location using the first order accurate error measure introduced in [21],

$$\frac{1}{L} \int |H(\phi_{expected}) - H(\phi_{computed})| dx dy, \qquad (13)$$

where L is the length of the expected interface. This integral is numerically calculated as in [21]:

- partition the domain into many tiny pieces (1000×1000),

- interpolate $\phi_{computed}$ onto the newly partitioned domain and calculate $\phi_{expected}$ for the domain,

- numerically integrate equation 13, where $H(\phi)$ is the indicator function for $\phi \leq 0$, i.e. $H(\phi) = 1$ if $\phi \leq 0$ and $H(\phi) = 0$ otherwise.

We note that both schemes are comparable in the quality of interface reconstruction, while the fast semi-Lagrangian based method is far superior in CPU time used.

11

	Grid Cells	Area	% Area Loss	L_1 Error	Order	CPU Time (sec)
	exact	582.2	-	-	-	-
One Revolution	50	562.2	3.6 %	.302	N/A	25.076
	100	578.4	.41%	.073	2.04	134.043
	200	581.7	.08%	.031	1.22	840.038
Two Revolutions	50	552.2	5.32%	.357	N/A	50.052
	100	576.3	.78%	.092	1.95	267.214
	200	580.6	.20%	.037	1.33	1691.55

Table 2: Zalesak's disk. HJ-WENO based Particle Level Set method.

	Grid Cells	Area	% Area Loss	L_1 Error	Order	CPU Time (sec)
	exact	582.2	-	-	-	-
One Revolution	50	565.2	3.09%	.434	N/A	.891
	100	574.5	1.07%	.181	1.25	4.045
	200	580.5	.22%	.105	.79	18.215
Two Revolutions	50	567.7	2.66%	.610	N/A	1.752
	100	574.9	1.01%	.206	1.57	7.941
	200	580.5	.22%	.103	1.00	36.341

Table 3: Zalesak's disk. Semi-Lagrangian based Particle Level Set method.

3.2 Single Vortex

While Zalesak's disk is a good indicator of diffusion errors in an interface capturing method, it does not test the ability of an Eulerian scheme to accurately resolve thin filaments on the scale of the mesh which can occur in stretching and tearing flows. A flow which exhibits interface stretching is the "vortex-in-a-box" problem introduced by Bell, Colella and Glaz [2]. The velocity field is defined by the stream function

$$\Psi = \frac{1}{\pi} \sin^2(\pi x) \sin^2(\pi y).$$

A unit computational domain is used with a circle of radius .15 placed at (.5,.75). The resulting velocity field stretches out the circle into a very long, thin fluid element which progressively wraps itself towards the center of the box. In under-resolved regions, the particles will not be close enough together to accurately represent the interface and thin filament structures will break apart. However, the particles still track the interface motion with second order accuracy, and thus the resulting pieces are in accurate locations.

For the purposes of error analysis, the velocity field is time reversed by multiplying by $\cos(\pi t/T)$ where T is the time at which the flow returns to its initial state, see LeVeque [9]. The reversal period used in the error analysis of the vortex problem is $T = 8$ producing a maximal stretching as seen in figure 4 for both the HJ-WENO based particle level set method and the semi-Lagrangian based particle level set method. As can be seen from the error tables 4 and 5 as well as figures 5 and 6, the ability of the fast, first order accurate semi-Lagrangian particle level set method to model interfaces undergoing substantial stretching is comparable to the significantly slower HJ-WENO method. The L_1 errors reported here for both cases compare favorably with those reported by Rider and Kothe in [13] using a VOF PLIC method. Again, the semi-Lagrangian method is substantially faster due to the much larger characteristic-based CFL number, 4.9, used as compared to the "safe" stability-based CFL number of .5 for the HJ-WENO method.

3.3 Octree Example

LeVeque [9] proposed a three dimensional incompressible flow field which combines a deformation in the x-y plane with one in the x-z plane. The

Grid Cells	Area	% Area Loss	L_1 Error	Order	CPU Time (sec)
exact	.0707	-	-	-	-
64	.0694	1.68%	2.89e-3	N/A	152.448
128	.0701	.79%	1.40e-3	1.0	885.974
256	.0705	.32%	5.43e-4	1.4	6609.78

Table 4: One period of vortex flow. HJ-WENO Particle Level Set method.

Grid Cells	Area	% Area Loss	L_1 Error	Order	CPU Time (sec)
exact	.0707	-	-	-	-
64	.0693	1.83%	3.34e-3	N/A	4.316
128	.0701	.73%	9.73e-4	1.8	23.653
256	.0704	.38%	5.58e-4	.8	149.995

Table 5: One period of vortex flow. Semi-Lagrangian Particle Level Set method.

velocity field is given by

$$
\begin{aligned}
u(x, y, z) &= 2\sin^2(\pi x)\sin(2\pi y)\sin(2\pi z), \\
v(x, y, z) &= -\sin(2\pi x)\sin^2(\pi y)\sin(2\pi z), \\
w(x, y, z) &= -\sin(2\pi x)\sin(2\pi y)\sin^2(\pi z)
\end{aligned}
$$

and the flow field is modulated in time with a period of $T = 3$. A sphere of radius .15 is placed within a unit computational domain at (.35, .35, .35). The authors in [5] used this 3D deformation field test to demonstrate the ability of the particles to help conserve the volume of the sphere as shown in figure 7. A uniform 100^3 grid cell domain was used in this test case. However, this test case also shows some surface aliasing when the resolution of the computational grid is insufficient to resolve the small scale features and thin filaments formed during the simulation. Increasing the resolution of the uniform grid used to store ϕ in order to accurately resolve the surface is prohibitively expensive in terms of computational time and memory usage. Since particles track the surface well even when the level set is underresolved, the original particle level set method is able to show excellent volume conservation properties on the coarse grid used.

Octrees allow for significantly higher effective grid resolutions and selective parts of the computational domain can be refined to accurately resolve the surface where needed. Figure 8 shows the same test using an octree grid refinement scheme. A maximum refinement depth of 10, resulting in

14

an effective resolution of 512^3 grid cells, was used. Use of octree grid refinement guarantees that the level set is never underresolved anytime during the simulation and that the aliasing problems that occur in the lower resolution uniform grid simulation are non-existent. The time and memory required for the levelset in this example is significantly less than running the same simulation with a highly refined uniform grid.

We stress that the octree based implementation and efficiency is facilitated since we only require a semi-Lagrangian advection scheme and a fast marching method, i.e. as opposed to TVD RK for time evolution and HJ-WENO for both the advection and the reinitialization equation. For a more complete description of the octree particle level set method see [10].
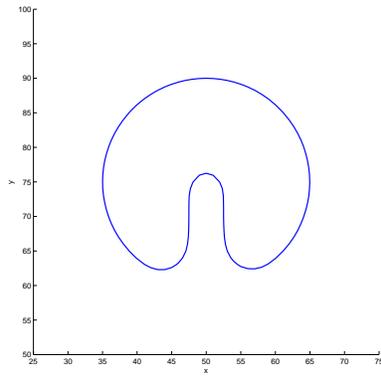
(a) HJ-WENO advection + HJ-WENO reinitializaiton
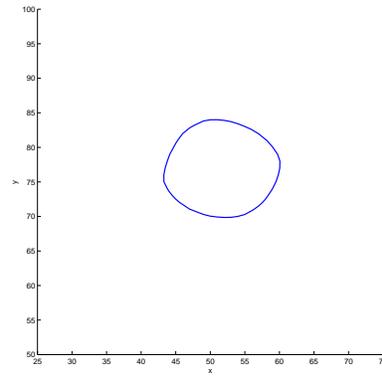
(b) HJ-WENO advection + FMM reinitialization
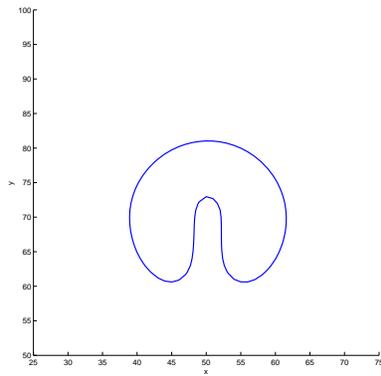
(c) Semi-Lagrangian advection + HJ-WENO reinitialization
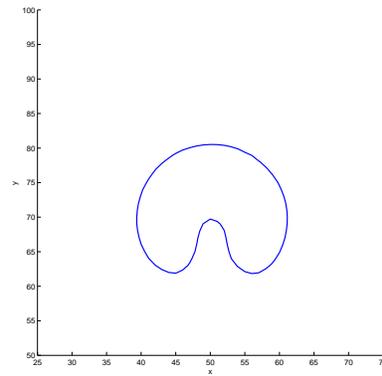
(d) Semi-Lagrangian advection + FMM reinitialization

Figure 1: Level set only method. Comparison of various advection and reinitialization algorithms after one revolution on 100×100 grid cell computational domain.
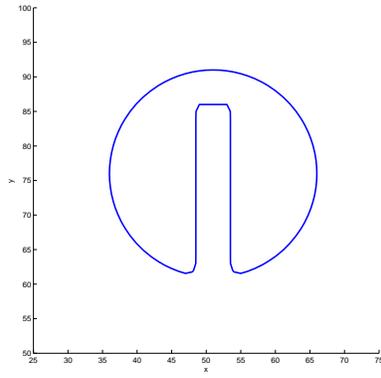
(a) Initial Notched Disk
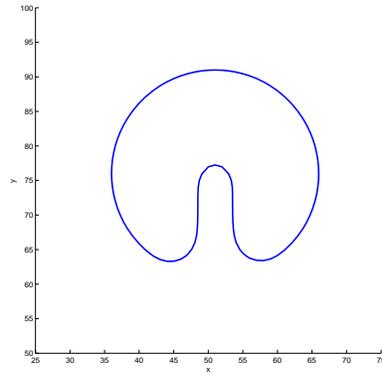
(b) Level Set Solution
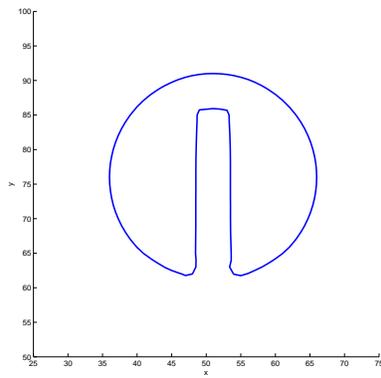
(c) HJ-WENO Particle Level Set
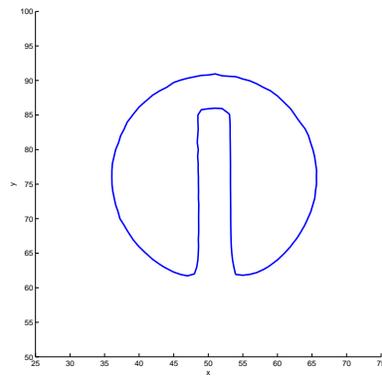
(d) Semi-Lagrangian Particle Level Set

Figure 2: Comparison of particle level set methods after one revolution on 100×100 grid cell computational domain.
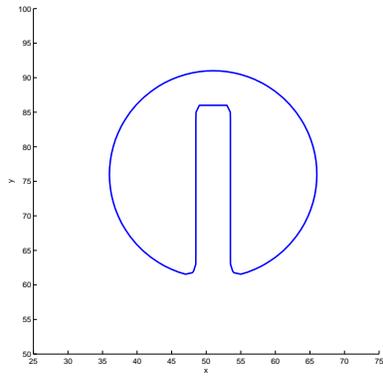
17

(a) Initial Notched Disk

(b) Level Set Solution
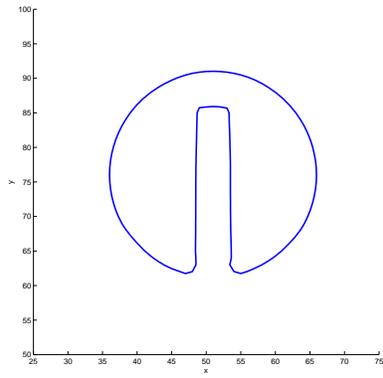
(c) HJ-WENO Particle Level Set
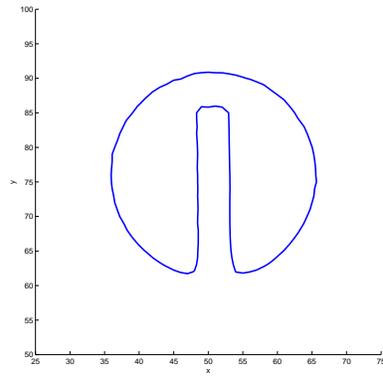
(d) Semi-Lagrangian Particle Level Set

Figure 3: Comparison after two revolutions on 100×100 grid cell computational domain.
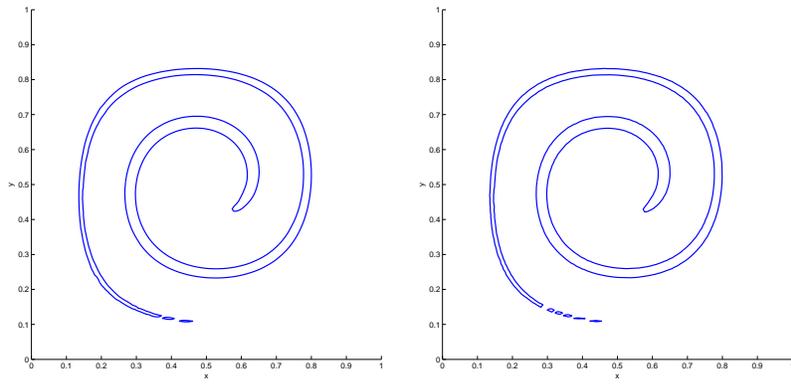
(a) HJ-WENO Particle Level
Set

(b) Semi-Lagrangian Particle
Level Set

Figure 4: Comparison of methods on 128×128 cell computational grid for
the vortex flow at $t = 4$.

19
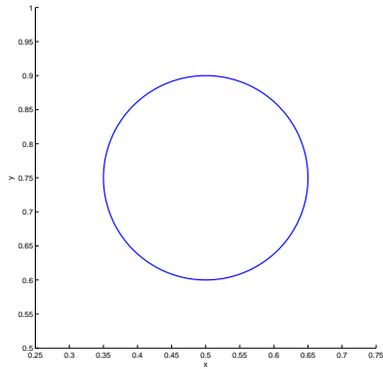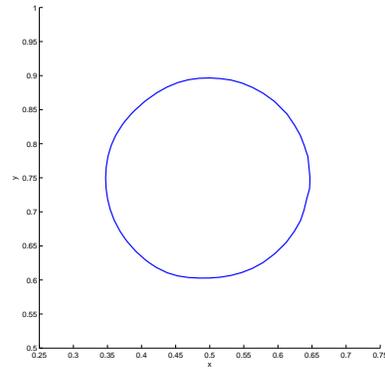
(a) Initial Circle
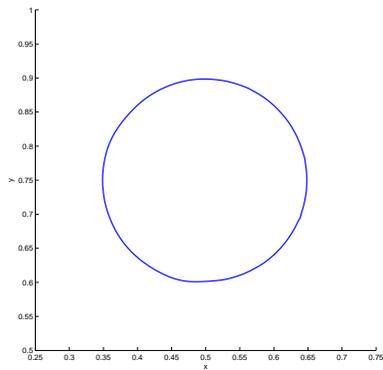
(b) 64×64 grid cells
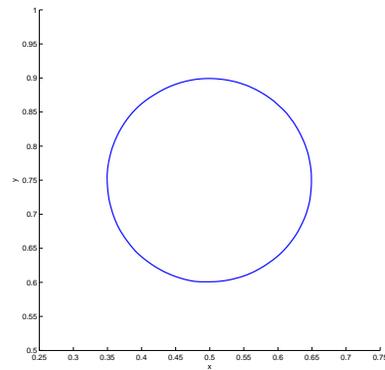
(c) 128×128 grid cells

(d) 256×256 grid cells

Figure 5: HJ-WENO Particle Level Set Solutions after one period ($t = 8$) of vortex flow.
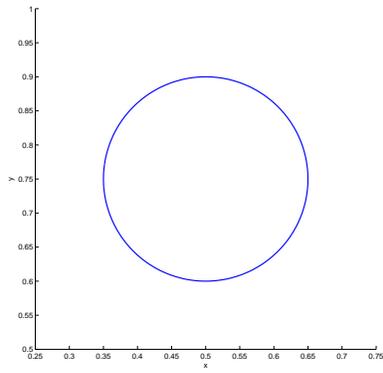
(a) Initial Circle
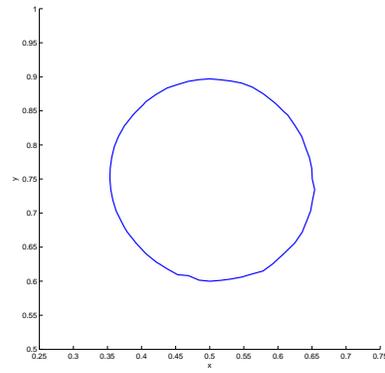
(b) 64×64 grid cells
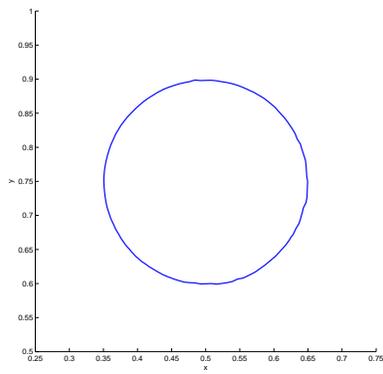
(c) 128×128 grid cells
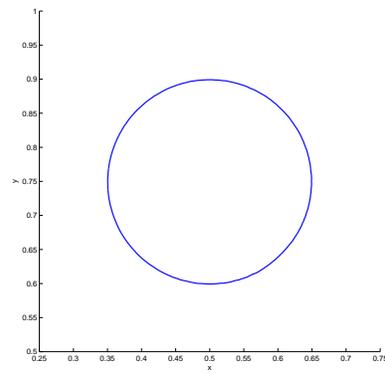
(d) 256×256 grid cells

Figure 6: Semi-Lagrangian Particle Level Set Solutions after one period ($t = 8$) of vortex flow.
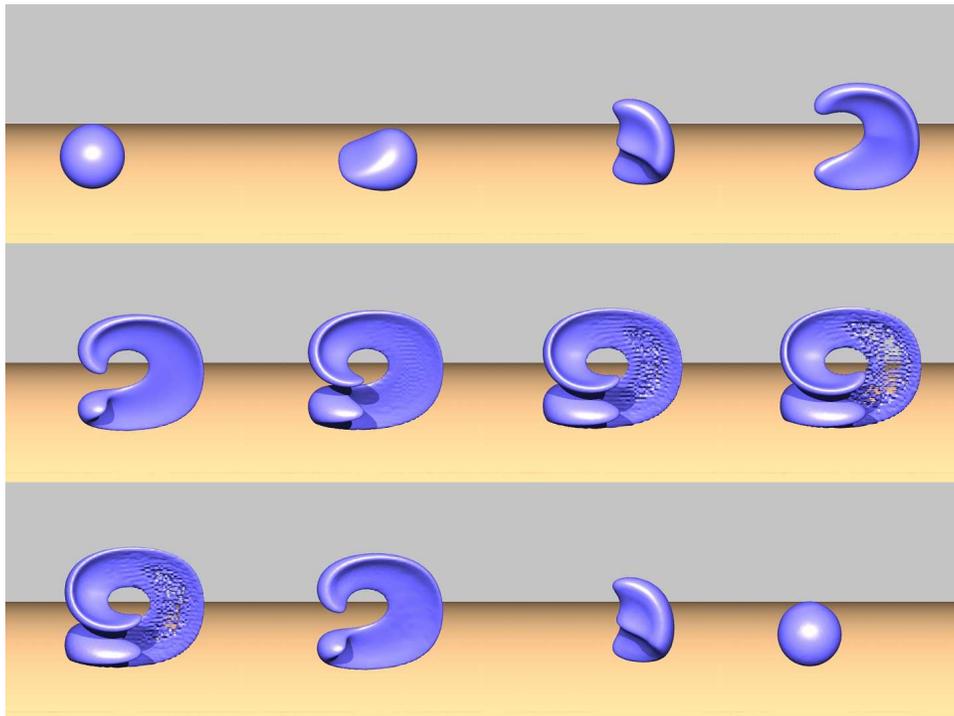
Figure 7: A Uniform Grid HJ-WENO Based Particle Level Set Three Dimensional Deformation Test. Reprinted from [5].
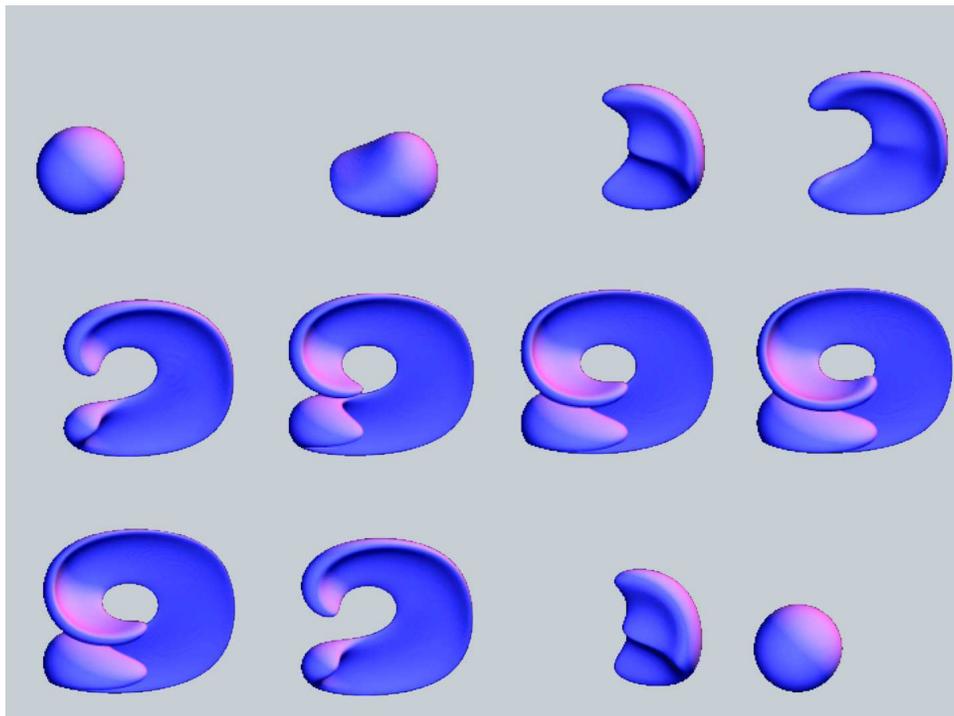
Figure 8: An Octree Based Semi-Lagrangian Particle Level Set Three Dimensional Deformation Test.

4   Conclusions

We proposed a fast semi-Lagrangian based particle level set method for the accurate capturing of passively advected interfaces. By utilizing massless marker particles nearby the interface as a characteristic-based diffusion control mechanism, we were able to forgo the use of higher order accurate advection schemes. Use of these schemes add significantly to the time of computation due to the small time step required for stability of the scheme. Semi-Lagrangian methods do not suffer from this constraint and the excessive amount of numerical diffusion present in a first order accurate semi-Lagrangian method is successfully counteracted by the particles. Also, selective adaptive mesh refinement for small scale feature resolution is computationally feasible with the semi-Lagrangian based particle level set method, unlike other more complicated Eulerian based advection methods. Moreover, the computational overhead incurred by the use of an octree grid representation is minimized by the large semi-Lagrangian based time steps allowed. Additional optimization is also gained by confining the level set update to a narrow band about the interface and the use of a fast marching method to reinitialize ϕ. The method also maintains the topological robustness and ease of implementation of the original level set method.

References

[1] Adalsteinsson, D. and Sethian, J., *A Fast Level Set Method for Propagating Interfaces*, J. Comp. Phys. 118, 269–277 (1995).

[2] Bell, J., Colella, P. and Glaz, H., *A Second-Order Projection Method for the Incompressible Navier-Stokes Equations*, J. Comp. Phys. 85, 257–283 (1989).

[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] 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).

[5] Enright, D., Fedkiw, R., Ferziger, J. and Mitchell, I., *A Hybrid Particle Level Set Method for Improved Interface Capturing*, J. Comp. Phys. 183, 83–116 (2002).

[6] 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).

[7] Enright, D., Nguyen, D., Gibou, F. and Fedkiw, R., *Using the Particle Level Set Method and a Second Order Accurate Pressure Boundary Condition for Free Surface Flows*, in Kawahashi, M., Ogut, A. and Tsuji, Y., eds., *Proceedings of the 4th ASME-JSME Joint Fluids Engineering Conference*, FEDSM2003–45144, ASME, 2003.

[8] Jiang, G.-S. and Peng, D., *Weighted ENO Schemes for Hamilton-Jacobi Equations*, SIAM J. Sci. Comput. 21, 2126–2143 (2000).

[9] LeVeque, R., *High-Resolution Conservative Algorithms For Advection In Incompressible Flow*, SIAM J. Numer. Anal. 33, 627–665 (1996).

[10] Losasso, F. and Fedkiw, R., *Extending the Particle Level Set Method to Octree Data Structures* (October 2003, In Review).

[11] Osher, S. and Sethian, J., *Fronts Propagating with Curvature Dependent Speed: Algorithms Based On Hamiliton-Jacobi Formulations*, J. Comp. Phys. 79, 12–49 (1988).

[12] 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).

[13] Rider, W. and Kothe, D., *Reconstructing Volume Tracking*, J. Comp. Phys. 141, 112–152 (1998).

[14] Roberts, A., *A Stable Numerical Integration Scheme for the Primitive Meterological Equations*, Atmos. Ocean 19, 35–46 (1981).

[15] Sethian, J., *A Fast Marching Level Set Method for Monotonically Advancing Fronts*, Proc. Natl. Acad. Sci. 93, 1591–1595 (1996).

[16] Sethian, J., *Fast Marching Methods*, SIAM Rev. 41, 199–235 (1999).

[17] Shu, C. and Osher, S., *Efficient Implementation of Essentially Non-Oscillatory Shock Capturing Schemes*, J. Comp. Phys. 77, 439–471 (1988).

[18] Staniforth, A. and Côté, J., *Semi-Lagrangian Integration Schemes for Atmospheric Models - A Review*, Monthly Weather Review 119, 2206–2223 (1991).

[19] Strain, J., *Semi-Lagrangian Methods for Level Set Equations*, J. Comp. Phys. 151, 498–533 (1999).

[20] Strain, J., *A Fast Modular Semi-Lagrangian Method for Moving Interfaces*, J. Comp. Phys. 161, 512–536 (2000).

[21] Sussman, M. and Fatemi, E., *An Efficient, Interface-Preserving Level Set Redistancing Algorithm and Its Application to Interfacial Incompressible Fluid Flow*, SIAM J. Sci. Comput. 20, 1165–1191 (1999).

[22] Sussman, M., Fatemi, E., Smereka, P. and Osher, S., *An Improved Level Set Method for Incompressible Two-Phase Flows*, Computers and Fluids 27, 663–680 (1998).

[23] 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).

[24] Tryggvason, G., Bunner, B., Esmaeeli, A., Juric, D., Al-Rawahi, N., Tauber, W., Han, J., Nas, S. and Jan, Y.-J., *A Front-Tracking Method for the Computations of Multiphase Flow*, J. Comp. Phys. 169, 708–759 (2001).

[25] Tsitsiklis, J., *Efficient Algorithms for Globally Optimal Trajectories*, in *Proceedings of the 33rd Conference on Decision and Control*, pp. 1368–1373, 1994.

[26] Zalesak, S., *Fully Multidimensional Flux-Corrected Transport Algorithms for Fluids*, J. Comp. Phys. 31, 335–362 (1979).