

Second-Order Accurate Computation of Curvatures in a Level Set Framework Using Novel High-Order Reinitialization Schemes

Antoine du Chéné *

Chohong Min †

Frédéric Gibou ‡

26th November 2007

Abstract

We present a high-order accurate scheme for the reinitialization equation of Sussman *et al.* [16] that guarantees accurate computation of the interface's curvatures in the context of level set methods. This scheme is an extension of the work of Russo and Smereka [11]. We present numerical results in two and three spatial dimensions to demonstrate fourth-order accuracy for the reinitialized level set function, third-order accuracy for the normals and second-order accuracy for the interface's mean curvature in the L^1 - and L^∞ - norms. We also exploit the work of Min and Gibou [17] to show second-order accurate scheme for the computation of the mean curvature on non-graded adaptive grids.

1 Introduction

The level set method was introduced by Osher and Sethian [10] as a powerful technique to analyze and compute the motion of interfaces. The level set method is used in a myriad of applications such as in physical simulations, in computer graphics and computer vision to cite a few (see e.g. [8, 9, 12, 2, 7, 3] and the references therein). Level set methods are particularly well-suited for tracking interfaces that may undergo complex changes in topology, such as the merging or the pinching of two fronts. However, as it is the case for many interface tracking methods, a limitation is that the computation of the curvatures is often noisy. This is due to the fact that, in order to compute the second-order accurate derivatives used in the definition of the curvatures, the level set function must be guaranteed to have some smoothness. Unfortunately, the level set function evolution through an arbitrary velocity field does not guarantee that the smoothness of the original level set function is preserved. For this reason, a procedure to transform the level set function into a smooth function without altering the location of the zero-level set is desired.

In order to lower the amount of loss of area/volume characteristics of level set methods, Sussman *et al.* [16] introduced the so-called reinitialization equation. This equation transforms any level set function into a signed distance function, therefore producing a function for which the numerical discretization of its derivatives, necessary in the level set evolution, can be computed robustly. The signed distance function is of course a good candidate for guaranteeing the smoothness necessary in the computation of interface's curvatures. Traditionally, this equation is solved using high-order accurate Hamilton-Jacobi solvers, e.g. the HJ-WENO scheme of Jiang and Peng [4]. However, the reinitialization procedure in [4] is at best second-order accurate as noted in Gibou *et al.* [1]. In turns, this produces oscillatory results in the computations of the interface curvatures.

In [11], Russo and Smereka pointed out that the HJ-WENO scheme of [4] for the reinitialization equation violates the direction of propagation of the information along the characteristic curves and leads to a loss of mass. They proposed a simple solution to this problem. In particular, they developed a second-order accurate reinitialization procedure that produces much improved results in the computation of the interface location over the HJ-WENO of [4]. They also outlined a guideline to extend this methodology to higher order schemes. In this paper, we further exploit the idea of [11] in order to develop a higher-order scheme for the reinitialization equation that produces second-order accurate computations of the curvatures. We then turn to a variant of [11], presented in [17] and show that this too produces second-order accurate curvature. Finally we show that the latter method is better suited for adaptive mesh refinements and preserves second-order accuracy on quadtree and octree data structures.

*Mechanical Engineering Department, University of California, Santa Barbara, CA 93106 & École Polytechnique, Palaiseau, France.

†Mathematics Department, KyungHee University, Seoul, Korea

‡Mechanical Engineering Department & Computer Science Department, University of California, Santa Barbara, CA 93106.

2 The Level Set Method

The level set method, introduced by Osher and Sethian [10], represents an interface, noted Γ (a curve in two dimensions or a surface in three dimensions) as the zero level set of a higher dimensional Lipschitz continuous function ϕ , called the level set function. Traditionally, the level set function is negative in the region enclosed by the interface and positive outside. The interface deforms under a velocity field \mathbf{V} , according to the following partial differential equation, called the level set equation :

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

In order to reduce the ‘loss of area/volume’ inherent to the level set method, Sussman *et al.* introduced the reinitialization equation

$$\phi_\tau + \text{Sign}(\phi^0) (|\nabla \phi| - 1) = 0, \quad (2)$$

where τ represents a fictitious time. This equation transforms an arbitrary level set function ϕ^0 into a signed distance function ϕ . Equation (2) therefore guarantees that $|\nabla \phi| \approx 1$, which in turn produces more robust and accurate discretizations of (1). Traditionally, these Hamilton-Jacobi equations are discretized with high-order accurate schemes such as the HJ-WENO of Jiang and Peng [4]. The normal to the interface \mathbf{n} and the interface mean curvature κ are defined as:

$$\mathbf{n} = \frac{\nabla \phi}{|\nabla \phi|} \quad \text{and} \quad \kappa = \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right).$$

The mean curvature can be written in general as:

$$\kappa = \frac{\phi_x^2 \phi_{yy} - 2\phi_x \phi_y \phi_{xy} + \phi_y^2 \phi_{xx} + \phi_x^2 \phi_{zz} - 2\phi_x \phi_z \phi_{xz} + \phi_z^2 \phi_{xx} + \phi_y^2 \phi_{zz} - 2\phi_y \phi_z \phi_{yz} + \phi_z^2 \phi_{yy}}{|\nabla \phi|^3},$$

and reduces to the Laplace operator in the case where ϕ is a signed distance function. Standard central difference formulas are used in the discretizations of the normals and the mean curvature.

3 Standard Discretization of the Reinitialization Equation and its Limitations

Standard discretizations for the reinitialization equation combine Godunov spatial discretizations with high-order TVD Runge-Kutta schemes in time as described next.

3.1 Spatial Discretization - Godunov Scheme

The reinitialization equation is of type Hamilton-Jacobi and can be written as:

$$\phi_\tau + \text{Sign}(\phi^0) H(\phi, \nabla \phi) = 0,$$

where H is the corresponding Hamiltonian. This equation can be written in semi-discrete form as:

$$\phi_\tau + S(\phi^0) [H_G(D_x^+ \phi, D_x^- \phi, D_y^+ \phi, D_y^- \phi)] = 0, \quad (3)$$

where $D_x^+ \phi$, $D_x^- \phi$, $D_y^+ \phi$ and $D_y^- \phi$ are the one-sided derivatives of ϕ and where H_G is defined as:

$$H_G(a, b, c, d) = \begin{cases} \sqrt{\max(|a^+|^2, |b^-|^2) + \max(|c^+|^2, |d^-|^2)} - 1 & \text{if } \text{sgn}(\phi^0) \leq 0 \\ \sqrt{\max(|a^-|^2, |b^+|^2) + \max(|c^-|^2, |d^+|^2)} - 1 & \text{if } \text{sgn}(\phi^0) > 0 \end{cases},$$

with $f^+ = \max(f, 0)$ and $f^- = \min(f, 0)$.

3.2 Time Discretization

The Godunov scheme is used in conjunction with high-order TVD Runge-Kutta schemes [14, 15]. For example, a third-order TVD Runge-Kutta scheme is written as a linear combination of Euler steps as follows: First, define $\tilde{\phi}^{n+1}$, $\tilde{\phi}^{n+2}$ and $\tilde{\phi}^{n+\frac{3}{2}}$ with Euler steps:

$$\begin{aligned} \frac{\tilde{\phi}^{n+1} - \phi^n}{\Delta\tau} + \mathbf{S}(\phi^0) [H_G (D_x^+ \phi^n, D_x^- \phi^n, D_y^+ \phi^n, D_y^- \phi^n)] &= 0, \\ \frac{\tilde{\phi}^{n+2} - \tilde{\phi}^{n+1}}{\Delta\tau} + \mathbf{S}(\phi^0) [H_G (D_x^+ \tilde{\phi}^{n+1}, D_x^- \tilde{\phi}^{n+1}, D_y^+ \tilde{\phi}^{n+1}, D_y^- \tilde{\phi}^{n+1})] &= 0. \end{aligned}$$

Second, define an intermediate value $\tilde{\phi}^{n+\frac{1}{2}}$ by simple averaging

$$\tilde{\phi}^{n+\frac{1}{2}} = \frac{3}{4}\phi^n + \frac{1}{4}\tilde{\phi}^{n+2},$$

before defining $\tilde{\phi}^{n+\frac{3}{2}}$ from $\tilde{\phi}^{n+\frac{1}{2}}$ with another Euler step:

$$\frac{\tilde{\phi}^{n+\frac{3}{2}} - \tilde{\phi}^{n+\frac{1}{2}}}{\Delta\tau} + \mathbf{S}(\phi^0) [H_G (D_x^+ \tilde{\phi}^{n+\frac{1}{2}}, D_x^- \tilde{\phi}^{n+\frac{1}{2}}, D_y^+ \tilde{\phi}^{n+\frac{1}{2}}, D_y^- \tilde{\phi}^{n+\frac{1}{2}})] = 0.$$

Finally, ϕ^{n+1} is defined by a linear combination of ϕ^n and $\tilde{\phi}^{n+\frac{3}{2}}$:

$$\phi^{n+1} = \frac{1}{3}\phi^n + \frac{2}{3}\tilde{\phi}^{n+\frac{3}{2}}.$$

3.3 Standard Computation Of One-Sided Derivatives: HJ-WENO Algorithm

Traditionally, the one-sided derivatives D_x^\pm , D_y^\pm of equation (3) are computed at each grid node in a dimension by dimension framework using the HJ-WENO scheme of Jiang and Peng [4], which is extended from the HJ-ENO algorithm and the fourth-order accurate WENO scheme of Liu *et al.* for conservation laws [5]. The philosophy behind the HJ-ENO is to choose the discretization that avoids interpolating across discontinuities. There are three possible divided differences for each one-sided derivative. The idea behind HJ-WENO is to use a convex combination of the three possible HJ-ENO approximations in such a way as to obtain fifth order accuracy in smooth regions, while preserving the HJ-ENO methodology near the discontinuities. There are two limitations to this algorithm applied to the reinitialization equation:

- It does not conserve the location of the interface when used in the reinitialization equation, which translates into a loss of area/volume.
- The computation of higher derivatives of the level set function are often noisy. This is particularly the case for the computations of the interface curvatures, which involve second-order derivatives.

These limitations stem from the fact that the HJ-WENO scheme of [4] only produces second-order accurate solutions when used in the reinitialization equation, as pointed out in [1]. Attempts to design numerical treatments that produce smoother curvatures can be found in Shin [13], which proposes a hybrid method using particles (see also Losasso *et al.* [6]). However, the reason behind low accuracy of the reinitialization equation has been described in [11]: the scheme in [4] violates the propagation of information (along the characteristics) near the interface. Russo and Smereka then proposed a simple procedure that guarantees that the information propagates along the characteristics. In their original work, they propose first and second-order accurate schemes for the reinitialization equations and a methodology to build higher order accurate schemes. In this article, we further develop the methodology of [11] to obtain a fourth-order accurate scheme for the reinitialization equation, which in turn lead to second-order accurate computations of interface curvatures. In [17], Min and Gibou proposed a variant of [11] that utilizes a smaller stencil. We show that this approach produces second-order accurate computation of the curvature similar to the results on uniform grids and is a better fit for adaptive mesh refinements. We first recall the scheme of [11] and then describe its extension.

Table 1: Errors in the reinitialized level set corresponding to Figure 2 after 150 iterations of equation (2).

	$\ \phi - \phi_h\ _1$	$\ \phi - \phi_h\ _\infty$
HJ-WENO Scheme	2.006×10^{-2}	5.903×10^{-2}
Russo & Smereka's Scheme	8.805×10^{-4}	3.580×10^{-3}

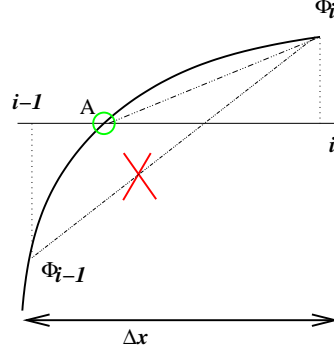


Figure 1: In [11], the interface location (A) where $\phi = 0$ is explicitly used in the approximation of the one-sided derivatives in (3), hence taking into account the fact that the information propagates outward from the interface

4 The ‘Subcell Fix’ of Russo and Smereka

The main idea in [11] is to discretely impose the fact that the information in the reinitialization equation propagates away from the interface in its normal direction. Therefore, while calculating the derivatives on one side of the interface, the value of ϕ at the interface should be used. Their algorithm therefore computes the first-order derivatives in (3) using the information that ϕ is zero on the interface instead of using the ϕ value at a node across the interface as depicted in figure 1. This ‘subcell fix’ is applicable in a dimension by dimension framework. For example, the first-order accurate scheme in the x -direction is as follows. Let d_i be the signed distance from the interface:

$$d_i = \Delta x \frac{\phi_i^o}{\Delta \phi_i^o},$$

where

$$\Delta \phi_i^o = \max \{ |\phi_{i+1}^o - \phi_{i-1}^o|/2, |\phi_{i+1}^o - \phi_i^o|, |\phi_i^o - \phi_{i-1}^o|, \epsilon \},$$

and ϵ is a small number compared to Δx . The smoothed sign function S becomes

$$S = \begin{cases} d_i/\Delta x, & \text{if } \phi_i^o \phi_{i+1}^o \leq 0 \text{ or } \phi_i^o \phi_{i-1}^o \leq 0 \\ \text{Sign}(\phi_i^o), & \text{otherwise.} \end{cases}$$

The first-order scheme using the Euler discretization is then:

$$\phi_i^{n+1} = \begin{cases} \phi_i^n - \frac{\Delta t}{\Delta x} (\text{Sign}(\phi_i^o) |\phi_i^n| - d_i), & \text{if } \phi_i^o \phi_{i+1}^o < 0 \text{ or } \phi_i^o \phi_{i-1}^o < 0 \\ \phi_i^n - \Delta t \text{Sign}(\phi_i^o) H_G(D_x^+ \phi, D_x^- \phi, 0, 0), & \text{otherwise,} \end{cases}$$

where H_G is given by the Eq. (3.1) and the one-sided differences $D_x^+ \phi$ and $D_x^- \phi$ are:

$$(D_x^- \phi)_{ij} = \frac{\phi_{ij} - \phi_{i-1j}}{\Delta x}, \quad \text{and} \quad (D_x^+ \phi)_{ij} = \frac{\phi_{i+1j} - \phi_{ij}}{\Delta x}.$$

This discretization can be extended to two and three spatial dimensions and to a second-order accurate scheme (see [11] for more details).

Since the location of the interface is preserved, the loss of area/volume becomes independent of the number of iteration of the algorithm. A clear improvement in the preservation of the location of the interface is observed in figure 2 and the associated table 1, which compare the results obtained after 150 iterations of the reinitialization equation with the HJ-WENO algorithm of Jiang and Peng [4] and that of Russo and Smereka [11].

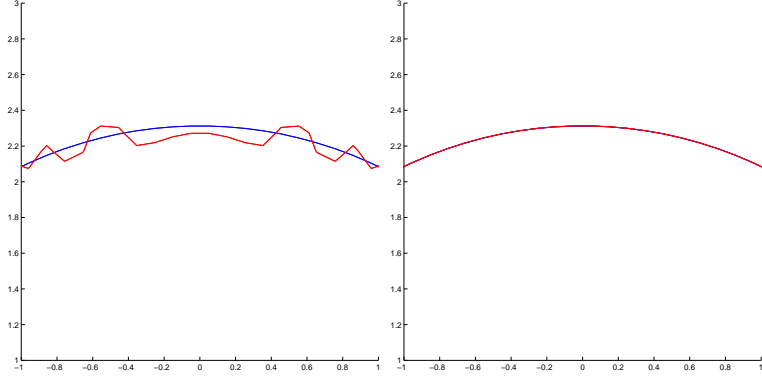


Figure 2: Zoom of the interface location after 150 iterations of the reinitialization equation on a 100×100 grid. The original level set function (blue) is defined as $\phi : (x, y) = x^2 + y^2 - 2.313^2$. The red curve depicts the reinitialized function. Left: using the HJ-WENO Scheme of [4]. Right: Using the first-order accurate scheme of Russo and Smereka.

4.1 High-Order Accurate Schemes

The subcell fix of [11] can be extended to general ENO reconstructions in order to develop higher order schemes. These schemes are constructed by successively adding nodes to build higher interpolants. Two constraints need to be satisfied when computing the one-sided derivatives for a node adjacent to the interface:

- The nodes across the interface should not be used in the computation of the first-order degree interpolant since the information propagates outward from the interface.
- The nodes to be added in the construction of higher degree interpolants are chosen in the direction of smoothness of ϕ .

For the nodes not immediately adjacent to the interface, we use the standard HJ-WENO algorithm. In this algorithm, we use the spatial Godunov approximation and the time discretization described in sections 3.1 and 3.2. We describe the fourth-order accurate scheme next.

4.1.1 Location Of The Interface

The interface location is calculated using a cubic interpolation of the initial level set function ϕ^o , using two points on each side of the interface.

4.1.2 Computation Of The One-Sided Derivatives For The Nodes Close To The Interface

We use the Newton form of the polynomial interpolant of degree three by first computing the divided difference tables. Because we want to include the location of the interface in the computation of the interpolant, the spatial steps can vary (see figure 3). Following the notations in [11], the divided differences for a stencil $((x(k), h(k)), k \in [-3, 3])$ are now written as follows:

$$\begin{aligned}
 D_{k+1/2}^1 &= \frac{h(k+1) - h(k)}{x(k+1) - x(k)}, & k \in [-3, 2] \\
 D_k^2 &= \frac{D_{k+1/2}^1 - D_{k-1/2}^1}{x(k+1) - x(k-1)}, & k \in [-2, 2] \\
 D_{k+1/2}^3 &= \frac{D_{k+1}^2 - D_k^2}{x(k+2) - x(k-1)}, & k \in [-2, 1]
 \end{aligned}$$

We define the MinAbs and MinMod functions as follows:

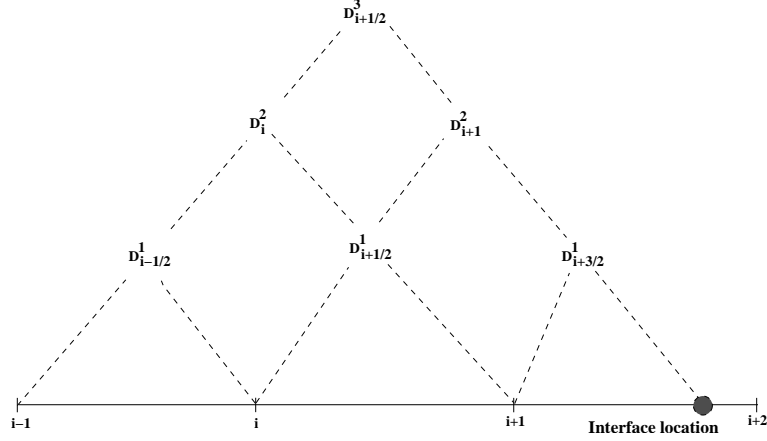


Figure 3: Computation of the table of divided differences with a stencil including the interface location for the stencil $[(X_{i-1}, \phi_{i-1}); (X_i, \phi_i); (X_{i+1}, \phi_{i+1}); (X_{interface}, 0)]$.

$$\text{MinAbs}(\alpha, \beta) = \begin{cases} \alpha & \text{if } |\alpha| < |\beta|, \\ \beta & \text{otherwise.} \end{cases} \quad \text{and} \quad \text{MinMod}(\alpha, \beta) = \begin{cases} \alpha & \text{if } |\alpha| \leq |\beta| \text{ and } \alpha\beta > 0, \\ \beta & \text{if } |\alpha| > |\beta| \text{ and } \alpha\beta > 0, \\ 0 & \text{if } \alpha\beta \leq 0. \end{cases}$$

To compute the one-sided differences at a point X_i next to the interface, we take a stencil of three points (including the interface) on each sides of X_i : $((x(k), h(k)), k \in [-3, 3])$ (where $x(0) = x_i$ and $h(0) = \phi_i$). We evaluate the divided differences as explained above. Then, the formulation of the one-sided derivatives is given by:

$$a = \begin{cases} (x(0) - x(-1)) * (x(0) - x(-2)) * \text{MinAbs}(D_{-1/2}^3, D_{-3/2}^3), & \text{if } |D_{-1}^2| < |D_0^2| \\ (x(0) - x(-1)) * (x(0) - x(1)) * \text{MinAbs}(D_{-1/2}^3, D_{1/2}^3), & \text{otherwise} \end{cases}$$

$$b = \begin{cases} (x(0) - x(-1)) * (x(0) - x(1)) * \text{MinAbs}(D_{-1/2}^3, D_{1/2}^3), & \text{if } |D_0^2| < |D_1^2| \\ (x(0) - x(1)) * (x(0) - x(2)) * \text{MinAbs}(D_{1/2}^3, D_{3/2}^3), & \text{otherwise} \end{cases}$$

$$D_x^- \phi(x(0)) = D_{-1/2}^1 + \text{MinMod}(D_{-1}^2, D_0^2) * (x(0) - x(-1)) + a$$

$$D_x^+ \phi(x(0)) = D_{1/2}^1 + \text{MinMod}(D_0^2, D_1^2) * (x(0) - x(1)) + b$$

Similarly, the same procedure is employed in the y - and z - directions.

5 Numerical Results

In this section, we present numerical results to demonstrate the accuracy of the algorithms described in Section 4. In all the simulations, the initial implicit functions are chosen so that the interface does not necessarily lie on grid nodes.

5.1 Interface Location

Consider a domain $\Omega = [-1, 1]^2$ and a level set function ϕ defining a square as:

$$\phi(x, y) = \begin{cases} 1, & \text{if } |x| > 0.5 + \Delta x/3 \quad \text{or} \quad |y| > 0.5 + \Delta x/3, \\ -1, & \text{otherwise,} \end{cases}$$

where Δx is the distance between two adjacent grid nodes in the x - or y - directions. We compare the results obtained after 150 iterations of the reinitialization equation using the Hamilton-Jacobi WENO solver of [4], the second-order

accurate scheme of Russo and Smereka and the high-order accurate scheme of section 4.1. As pointed out in [11], the HJ-WENO algorithm leads to a loss in area whereas the other two conserve the original location of the interface as illustrated in figure 4. We then test the high-order accurate scheme in the case where the initial square is rotated by $\pi/4$ so that the edges of the square are not parallel to the Cartesian directions. Figure 5 illustrates that the initial location of the level set function is unchanged. Not surprisingly, the nonlinear features of the reinitialization equation along the diagonals of the square are treated properly by the Godunov scheme.

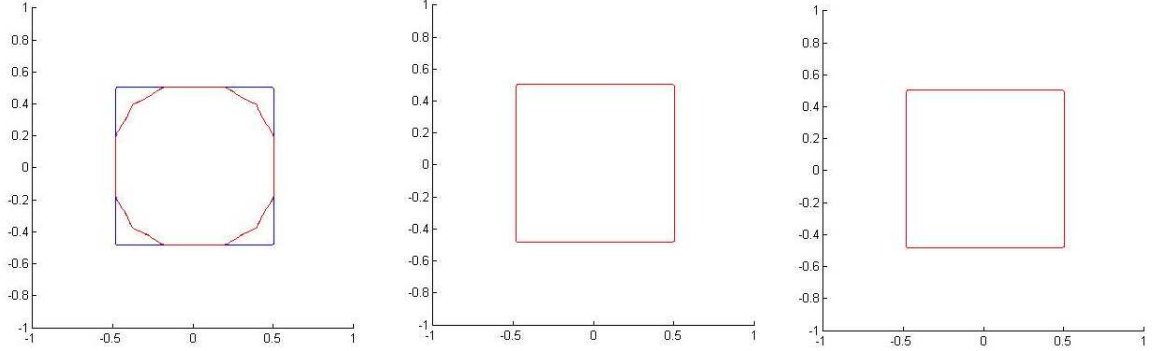


Figure 4: Initial (blue) and reinitialized (red) zero level set after 150 iterations of equation (2) on a 100×100 grid using the HJ-WENO scheme of [4] (left), the second-order accurate scheme of [11] (center) and the high-order scheme of section 4.1 (right).

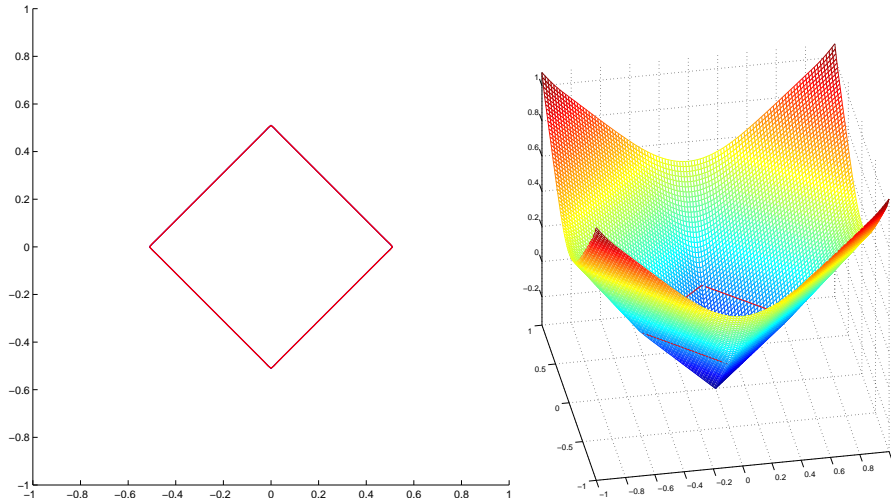


Figure 5: Left: Initial (blue) and reinitialized (red) zero level set after 100 iterations of equation (2) on a 100×100 grid using the high-order scheme of section 4.1. Right: The reinitialized level set function.

5.2 Computation of the Interface's Curvature in 2D

The main improvement of the high-order algorithm of Section 4.1 is clearly the ability to compute quantities that involve first or second-order derivatives of ϕ , such as the normals to the interface and the curvatures. In particular the smoothness of the reinitialized level set avoids the spurious numerical noise when computing the curvatures.

Consider a domain $\Omega = [-5, 5]^2$ and an initial level set function $\phi(x, y) = e^{\sqrt{x^2+y^2}-2.313} - 1$. Figure 6 compares the results obtained with the HJ-WENO scheme of [4], the scheme of [11] and the high-order accurate scheme of Section 4.1. In particular, it illustrates the presence of numerical noise in the case where the reinitialization scheme

does not guarantee enough smoothness for ϕ , i.e. the case of [4]. In the case of the second-order accurate scheme of [11], table 4 demonstrates second-order accuracy for the reinitialized level set function, whereas table 5 demonstrates that the interface's curvature is only first order, due to the lack of smoothness of the reinitialized level set. In the case of the high-order scheme of Section 4.1, table 6 demonstrates fourth-order accuracy for the reinitialized ϕ . This high degree of smoothness guarantees second-order accuracy for the computation of the interface curvature and third-order accuracy for the computation of the normals as shown in tables 7 and 8, respectively. The interface's curvature is computed using standard second-order accurate central difference formulas, whereas the normals are computed with standard fourth-order accurate central difference formulas.

Figures 7 and 8 further demonstrate the smoothness of the isocontours of the mean curvature in the case of an ellipse and in the case of two nearby circles, respectively.

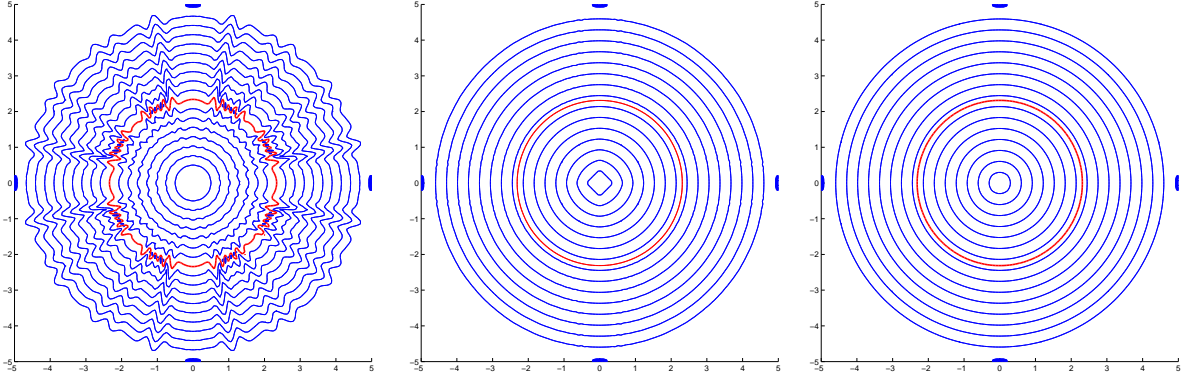


Figure 6: Isocontours of the curvature of ϕ after solving equation (2) for 150 iterations on a 128×128 grid. The red curve depicts the curvature at the interface. Left: Results obtained with the HJ-WENO scheme of [4]. Middle: Results obtained with second-order accurate scheme of [11]. Right: Results obtained with the high-order accurate scheme of section 4.1. Tables 3, 5 and 7 give the accuracy results for these three cases, respectively.

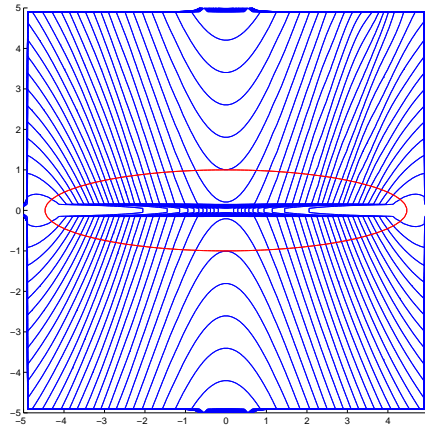


Figure 7: Isocontours of the curvature on a 100×100 grid in the case where 100 iterations of the high-order algorithm of section 4.1 is used. The red curve corresponds to the curvature at the interface. The initial implicit function is given by $\phi(x, y) = \min \{ (x - 1.5)^2 + y^2 - 1.313^2, (x + 1.5)^2 + y^2 - 1.313^2 \}$

5.3 Computation of the Interface's Mean Curvature in 3D

Consider a domain $\Omega = [-1, 1]^3$ and an initial level set function $\phi(x, y) = x^2 + y^2 + z^2 - (.2222)^2$. Table 11 demonstrates fourth-order accuracy for the reinitialized ϕ in the case of the high-order scheme of Section 4.1. This

Table 2: Accuracy results for the reinitialized level set function ϕ for example 5.2. In this case the HJ-WENO scheme produces second-order accurate results.

	$\ \phi - \phi_h\ _1$	Order	$\ \phi - \phi_h\ _\infty$	Order
16×16	1.512×10^{-2}	–	2.306×10^{-2}	–
32×32	4.241×10^{-3}	1.83	5.773×10^{-3}	2.00
64×64	8.414×10^{-4}	2.33	1.656×10^{-3}	1.80
128×128	2.470×10^{-4}	1.77	3.441×10^{-4}	2.27

Table 3: Accuracy results for the mean curvature κ for example 5.2. In this case the HJ-WENO scheme is used and the resulting lack of smoothness in ϕ does not guarantee smooth computations of the curvatures.

	$\ \kappa - \kappa_h\ _1$	Order	$\ \kappa - \kappa_h\ _\infty$	Order
16×16	0.0272	–	0.0596	–
32×32	0.0126	1.11	0.0290	1.04
64×64	0.0305	-1.27	0.0979	-1.76
128×128	0.0207	0.55	0.0732	0.42

Table 4: Accuracy results for the reinitialized level set function ϕ for example 5.2. In this case the second-order accurate algorithm of [11] produces more than second-order accurate results.

	$\ \phi - \phi_h\ _1$	Order	$\ \phi - \phi_h\ _\infty$	Order
16×16	3.236×10^{-3}	–	1.716×10^{-2}	–
32×32	5.661×10^{-4}	2.52	2.710×10^{-3}	2.66
64×64	4.829×10^{-5}	3.55	2.020×10^{-4}	3.75
128×128	6.787×10^{-6}	2.83	3.790×10^{-5}	2.41

Table 5: Accuracy results for the mean curvature κ for example 5.2. In this case the second-order accurate scheme of [11] is used.

	$\ \kappa - \kappa_h\ _1$	Order	$\ \kappa - \kappa_h\ _\infty$	Order
16×16	1.08×10^{-2}	–	1.68×10^{-2}	–
32×32	2.04×10^{-3}	2.40	5.86×10^{-3}	1.52
64×64	8.00×10^{-4}	1.35	2.56×10^{-3}	1.20
128×128	3.35×10^{-4}	1.26	1.28×10^{-3}	0.99

Table 6: Accuracy results for the reinitialized level set function ϕ for example 5.2. In this case the high-order accurate algorithm of Section 4.1 produces fourth-order accurate results.

	$\ \phi - \phi_h\ _1$	Order	$\ \phi - \phi_h\ _\infty$	Order
16×16	1.566×10^{-3}	–	3.115×10^{-3}	–
32×32	1.669×10^{-4}	3.23	3.198×10^{-4}	3.28
64×64	5.848×10^{-6}	4.84	1.516×10^{-5}	4.40
128×128	4.822×10^{-7}	3.60	8.868×10^{-7}	4.10

Table 7: Accuracy results in computing the interface's mean curvature in the case of example 5.2. The high-order accurate algorithm of Section 4.1 produces second-order accurate results.

	$\ \kappa - \kappa_h\ _1$	Order	$\ \kappa - \kappa_h\ _\infty$	Order
16×16	8.87×10^{-3}	–	1.92×10^{-2}	–
32×32	1.42×10^{-3}	2.65	4.59×10^{-3}	2.06
64×64	4.24×10^{-4}	1.74	1.05×10^{-3}	2.13
128×128	1.02×10^{-4}	2.05	2.08×10^{-4}	2.34

Table 8: Accuracy results in computing the normals to the interface \vec{n} in the case of example 5.2. The high-order accurate algorithm of Section 4.1 produces third-order accurate results.

	$\ \vec{n} - \vec{n}_h\ _1$	Order	$\ \vec{n} - \vec{n}_h\ _\infty$	Order
16×16	1.68×10^{-3}	–	6.38×10^{-3}	–
32×32	1.20×10^{-4}	3.80	6.81×10^{-4}	3.23
64×64	1.71×10^{-5}	2.81	6.02×10^{-5}	3.50
128×128	1.51×10^{-6}	3.51	7.28×10^{-6}	3.05

Table 9: Accuracy results in computing the interface's mean curvature using the Laplacian in the case of example 5.2. The high-order accurate algorithm of Section 4.1 produces second-order accurate results.

	$\ \kappa - \kappa_h\ _1$	Order	$\ \kappa - \kappa_h\ _\infty$	Order
16×16	1.12×10^{-2}	–	2.59×10^{-2}	–
32×32	2.16×10^{-3}	2.38	6.53×10^{-3}	1.99
64×64	5.86×10^{-4}	1.88	1.85×10^{-3}	1.82
128×128	1.44×10^{-4}	2.03	3.20×10^{-4}	2.53

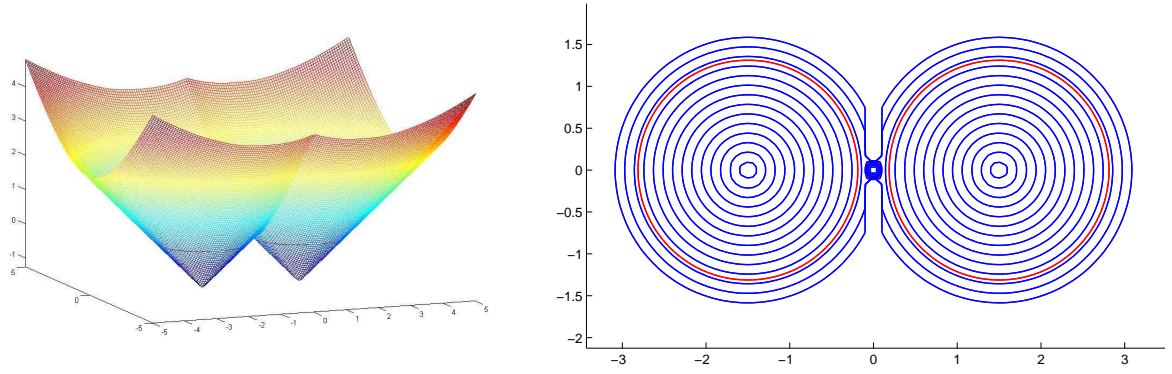


Figure 8: Isocontours of the curvature on a 150×150 grid in the case where 100 iterations of the high-order algorithm of section 4.1 is used. The red curve corresponds to the curvature at the interface. The initial implicit function is given by $\phi(x, y) = 0.1(\sqrt{x^2/20 + y^2} - 1)(1 + (x - 3.5)^2 + (y - 2)^2)$.

Table 10: Accuracy results for the computation of the interface mean curvature using the Laplacian in the case of example 5.3. The high-order accurate algorithm of section 4.1 produces second-order accurate results.

	$\ \kappa - \kappa_h\ _1$	Order	$\ \kappa - \kappa_h\ _\infty$	Order
19^3	2.73×10^{-2}	–	1.52×10^{-1}	–
38^3	7.04×10^{-3}	1.95	2.02×10^{-2}	2.91
76^3	2.07×10^{-3}	1.77	6.01×10^{-3}	1.75

high degree of smoothness guarantees second-order accuracy for the computation of the interface curvature as shown in table 12. Figure 9 illustrates the smoothness of the curvature computations.

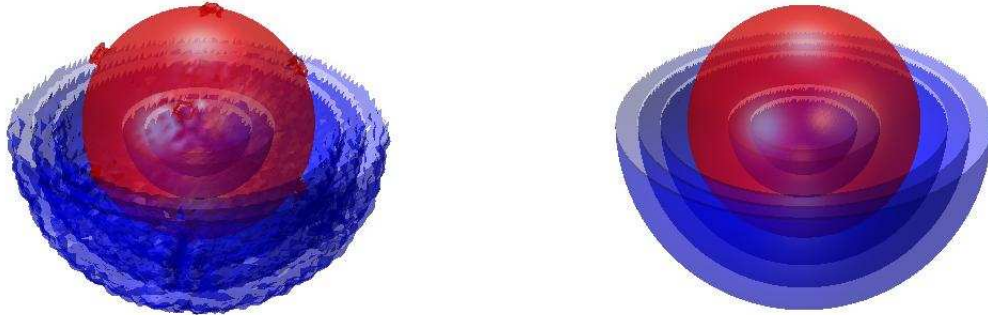


Figure 9: Isosurface of the mean curvature after solving equation (2) for 80 iterations on a 76^3 grid in the case of example 5.3. Left: The HJ-WENO scheme of [4]. Right: The high-order accurate scheme of section 4.1. The red isosurface corresponds to the mean curvature at the zero level set.

Remark: As mentioned in section 2, the formula for the curvature reduces to that of the Laplacian in the case where the level set function is a signed distance function. Tables 9 and 10 show that second-order accuracy is obtained as well when we use the Laplacian to compute the curvature. We don't find any benefits in terms of accuracy. This procedure is slightly faster and easier to implement.

Table 11: Accuracy results for the reinitialized level set function ϕ for example 5.3. In this case the high-order accurate algorithm of Section 4.1 produces fourth-order accurate results.

	$\ \phi - \phi_h\ _1$	Order	$\ \phi - \phi_h\ _\infty$	Order
19^3	2.439×10^{-5}	—	6.765×10^{-5}	—
38^3	1.791×10^{-6}	3.77	7.977×10^{-6}	3.08
76^3	1.212×10^{-7}	3.88	6.225×10^{-7}	3.68

Table 12: Accuracy results for the computation of the interface mean curvature κ in the case of example 5.3. The high-order accurate algorithm of Section 4.1 produces second-order accurate results.

	$\ \kappa - \kappa_h\ _1$	Order	$\ \kappa - \kappa_h\ _\infty$	Order
19^3	2.10×10^{-2}	—	1.52×10^{-1}	—
38^3	5.31×10^{-3}	1.98	2.03×10^{-2}	2.90
76^3	1.54×10^{-3}	1.79	4.96×10^{-3}	2.04

6 Extension to Adaptive Grids

In the case of adaptive grids, it is difficult to construct numerical schemes with large stencils, especially in the case of non-graded grids, i.e. grids for which the size ratio between adjacent cells is not constrained. In this case, the difficulty comes from the fact that infinitely many possible arrangements could exist for the next to adjacent cells, making a general procedure difficult to write. In [17], Min and Gibou extended the work of Russo and Smereka [11] to the case of non-graded Cartesian grids. We show here that this scheme leads to second-order accurate computations for the mean curvature. In the case of adaptive grids, [17] made two changes to the original work of Russo and Smereka: (1) different interpolation to locate the interface and (2) in computing the second divided differences at a grid point adjacent to the interface. We refer the reader to [17] for more details.

We next present numerical evidence in tables 13 and 14 that the reinitialization algorithm implemented on the non-graded adaptive Cartesian grid produces comparable results to the one obtained from the uniform grid with the high-order algorithm of section 4.1. In the case of the adaptive grids, table 15 demonstrates third-order accuracy for the reinitialized level set function and second-order accuracy for the computation of the mean curvature. These results for the computation of the mean curvature are similar to the ones obtained with the high-order algorithm of section 4.1 (see table 7). The adaptive grid thus allows to concentrate the computational effort in the region of interest, i.e. close to the interface. As a consequence one is able to obtain results with the adaptive framework that are not possible on uniform grids. Table 16 gives accuracy results for the mean curvature in three spatial dimensions in the case of adaptive grids. Figures 11 and 12 demonstrate a comparable smoothness of the isocontours of the mean curvature in the same cases as the ones using the high-order algorithm of section 4.1 (figures 7 and 8).

Table 13: Accuracy results for the reinitialized level set ϕ and the mean curvature κ from the exponential test using the adaptive framework with a uniform grid.

	$\ \phi - \phi_h\ _\infty$	Order	$\ \kappa - \kappa_h\ _\infty$	Order
(2^4)	1.30×10^{-2}	—	2.45×10^{-2}	—
(2^5)	2.48×10^{-3}	2.38	6.27×10^{-3}	1.96
(2^6)	3.95×10^{-4}	2.65	1.57×10^{-3}	1.99
(2^7)	5.19×10^{-5}	2.93	3.93×10^{-4}	2.00
(2^8)	6.56×10^{-6}	2.98	9.82×10^{-5}	2.00

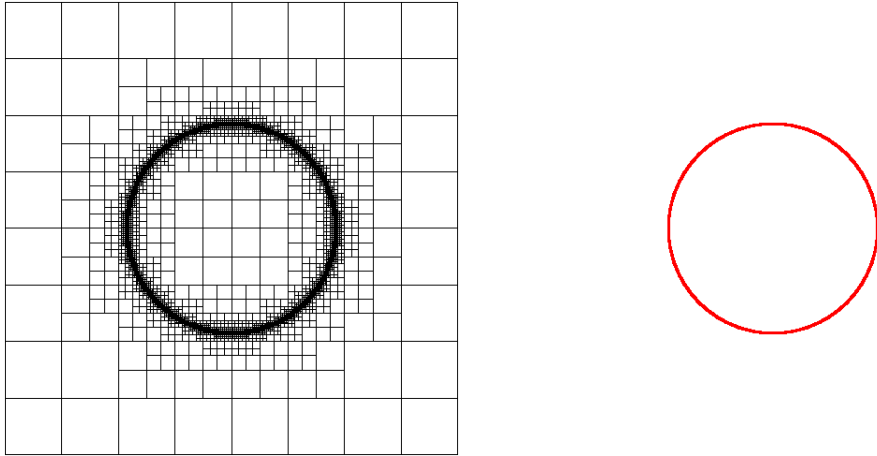


Figure 10: Left: Adaptive grid used. Right: Smooth computation of the mean curvature after 100 iterations of the reinitialization equation. The smallest grid size is $\Delta x_s = 1/1024$. - The initial implicit function is $\phi : (x, y) \mapsto \exp(\sqrt{x^2 + y^2} - 2.313) - 1$.

Table 14: Accuracy results for the reinitialized level set ϕ and the mean curvature κ from the exponential test in 3D using the adaptive framework with a uniform grid.

	$\ \phi - \phi_h\ _\infty$	Order	$\ \kappa - \kappa_h\ _\infty$	Order
$(2^4)^3$	2.082×10^{-3}	—	7.57870×10^{-1}	—
$(2^5)^3$	2.819×10^{-4}	2.88	1.835×10^{-1}	2.04
$(2^6)^3$	3.948×10^{-5}	2.83	4.520×10^{-2}	2.02
$(2^7)^3$	5.007×10^{-6}	2.97	1.645×10^{-2}	1.45

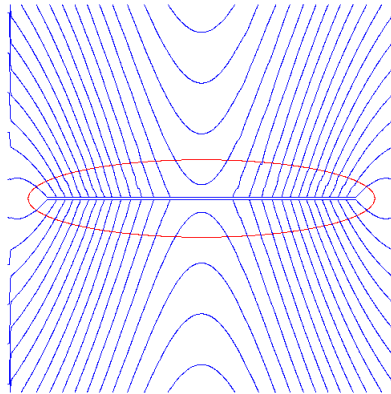


Figure 11: Isocontours of the curvature on a 256×256 grid in the case where 200 iterations of the algorithm detailed in [17] are used. The red curve corresponds to the curvature at the interface. The initial implicit function is given by $\phi : (x, y) \mapsto 0.1(\sqrt{x^2/20 + y^2} - 1)(1 + (x - 3.5)^2 + (y - 2)^2)$.

Table 15: Accuracy results for the reinitialized level set ϕ and the mean curvature κ from the example of Figure 10.

	$\ \phi - \phi_h\ _\infty$	Order	$\ \kappa - \kappa_h\ _\infty$	Order
(2^4)	1.32×10^{-2}	—	2.45×10^{-2}	—
(2^5)	2.48×10^{-3}	2.41	6.27×10^{-3}	1.96
(2^6)	3.99×10^{-4}	2.63	1.57×10^{-3}	1.99
(2^7)	5.08×10^{-5}	2.97	3.93×10^{-4}	2.00
(2^8)	6.47×10^{-6}	2.97	9.82×10^{-5}	2.00
(2^9)	8.39×10^{-7}	2.94	2.45×10^{-5}	1.99
(2^{10})	1.04×10^{-7}	3.01	6.14×10^{-6}	2.00
(2^{11})	1.29×10^{-8}	3.00	1.53×10^{-6}	2.00
(2^{12})	1.78×10^{-9}	2.85	3.83×10^{-7}	1.99

Table 16: Accuracy results for the reinitialized level set ϕ and the mean curvature κ from the exponential test in 3D using the adaptive framework with an adaptive grid.

	$\ \phi - \phi_h\ _\infty$	Order	$\ \kappa - \kappa_h\ _\infty$	Order
$(2^4)^3$	2.081×10^{-3}	—	7.578×10^{-1}	—
$(2^5)^3$	2.819×10^{-4}	2.88	1.835×10^{-1}	2.04
$(2^6)^3$	3.948×10^{-5}	2.83	4.520×10^{-2}	2.02
$(2^7)^3$	5.013×10^{-6}	2.97	1.554×10^{-2}	1.53

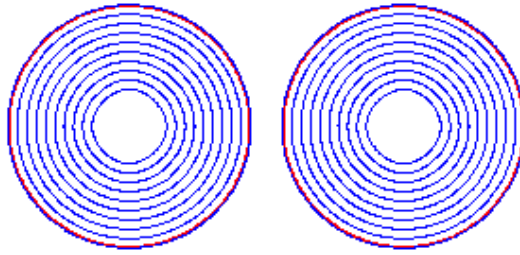


Figure 12: Isocontours of the curvature on a 128×128 grid in the case where 100 iterations of the algorithm detailed in [17] are used. The red curve corresponds to the curvature at the interface. The initial implicit function is given by $\phi : (x, y) \mapsto \min((x - 1.5)^2 + y^2 - 1.313^2, (x + 1.5)^2 + y^2 - 1.313^2)$.

7 Acknowledgments

The research of A. du Chene and F. Gibou was supported in part by a Sloan Research Fellowship in Mathematics and by NSF under grant agreement DMS 0713858. The research of C. Min was supported in part by the Kyung Hee University Research Fund (KHU-20070608) in 2007.

8 Conclusion

We have presented two novel high-order accurate schemes for the reinitialization equation of Sussman *et al.* [16] that guarantee accurate computation of the interface's curvature in the context of level set methods. These schemes build on the work of Russo and Smereka [11]. We have presented numerical results in two and three spatial dimensions

that demonstrate fourth-order accuracy for the reinitialized level set function, third-order accuracy for the normals and second-order accuracy for the interface's curvatures in the L^1 - and L^∞ - norms. We also exploited the reinitialization scheme introduced by Min and Gibou in [17] and demonstrated that one can obtain second-order accuracy for the computation of curvatures on highly non-graded adaptive Cartesian grids. A hallmark of this work is that state-of-the-art second-order accurate reinitialization procedures of [4] can now be replaced by fourth-order accurate reinitialization schemes, hence solving a long standing problem in the accurate computation of geometrical quantities in the level-set framework, such as normals and curvatures. In addition, unlike reinitialization schemes based on WENO technologies, our approach can be extended to nonuniform meshes producing results with a level of resolution unattainable on uniform grids.

References

- [1] F. Gibou and R. Fedkiw. A fourth order accurate discretization for the Laplace and heat equations on arbitrary domains, with applications to the Stefan problem. *J. Comput. Phys.*, 202:577–601, 2005.
- [2] F. Gibou, R. Fedkiw, R. Caflisch, and S. Osher. A level set approach for the numerical simulation of dendritic growth. *J. Sci. Comput.*, 19:183–199, 2003.
- [3] F. Gibou, D. Levy, C. Cardenas, P. Liu, and A. Boyer. Partial differential equation based segmentation for radiotherapy treatment planning. *Mathematical Biosciences and Engineering*, 2:209–226, 2005.
- [4] G.-S. Jiang and D. Peng. Weighted ENO schemes for Hamilton-Jacobi equations. *SIAM J. Sci. Comput.*, 21:2126–2143, 2000.
- [5] X.-D. Liu, S. Osher, and T. Chan. Weighted essentially non-oscillatory schemes. *J. Comput. Phys.*, 126:202–212, 1996.
- [6] F. Losasso, R. Fedkiw, and S. Osher. Spatially adaptive techniques for level set methods and incompressible flow. *Computers and Fluids (in press)*.
- [7] F. Losasso, F. Gibou, and R. Fedkiw. Simulating water and smoke with an octree data structure. *ACM Trans. Graph. (SIGGRAPH Proc.)*, pages 457–462, 2004.
- [8] S. Osher and R. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Springer-Verlag, 2002. New York, NY.
- [9] S. Osher and N. Paragios. *Geometric Level Set Methods in Imaging, Vision, and Graphics*. Springer-Verlag, 2003. New York, NY.
- [10] S. Osher and J. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *J. Comput. Phys.*, 79:12–49, 1988.
- [11] G. Russo and P. Smereka. A remark on computing distance functions. *J. Comput. Phys.*, 163:51–67, 2000.
- [12] J. A. Sethian. *Level set methods and fast marching methods*. Cambridge University Press, 1999. Cambridge.
- [13] S. Shin. Computation of the curvature field in numerical simulation of multiphase flow. *J. Comput. Phys.*, doi:10.1016/j.jcp.2006.08.009:32–54.
- [14] C.-W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock capturing schemes. *J. Comput. Phys.*, 77:439–471, 1988.
- [15] C.-W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock capturing schemes II (two). *J. Comput. Phys.*, 83:32–78, 1989.
- [16] M. Sussman, P. Smereka, and S. Osher. A level set approach for computing solutions to incompressible two-phase flow. *J. Comput. Phys.*, 114:146–159, 1994.
- [17] C. Min and F. Gibou. A Second Order Accurate Level Set Method on Non-Graded Adaptive Cartesian Grids. under review, available at UCLA CAM report (06-22) <http://www.math.ucla.edu/applied/cam/index.html>, 2006.