A Remark on Computing Distance Functions

Giovanni Russo
Department of Mathematics, University of L'Aquila
L'Aquila, Italy

and

Peter Smereka
Department of Mathematics, University of Michigan
Ann Arbor, MI 48109

ABSTRACT

We propose a new method for the reconstruction of the signed distance function in the context of level set methods. The new method is a modification of the algorithm which makes use of the PDE equation for the distance function (introduced by Sussman, Smereka and Osher). It is based mainly on the use of a truly upwind discretization near the interface. Comparison with the previous algorithm shows a definite improvement. When used with a first order upwind scheme, the method provides first order accuracy for the signed distance function in the whole computational domain, and second order accuracy in the location of the interface.

1 Introduction

Level set methods have proven to be a useful tool for computing interface evolution. In this approach the interface, Σ, is represented as the zero level set of a continuous level set function, ϕ, defined in a domain $\Omega \subset I\!\!R^d$, i.e.

$$\Sigma = \{\boldsymbol{x} \in I\!\!R^d : \phi(\boldsymbol{x}) = 0\}.$$

The function ϕ is defined everywhere in the domain Ω. The interface Σ is updated by solving a transport equation for ϕ

$$\frac{\partial \phi}{\partial t} + \boldsymbol{v} \cdot \nabla \phi = 0, \tag{1}$$

where $\boldsymbol{v}$ is an extension of the interface velocity to Ω.

In many applications the level set function obtained by the solution of Eq. (1) may become distorted, which means that its gradient may become very large or very small around the interface. It is therefore useful to replace the level set function with a better behaved function which has the same zero level set. This process is called *reinitialization* [1].

The simplest and most useful choice is to replace the level set function by the signed distance function. A signed distance function associated to a level set function $\phi(\boldsymbol{x})$ is defined by

$$D(\boldsymbol{x}) = \min_{\boldsymbol{y}\in\Sigma} |\boldsymbol{x} - \boldsymbol{y}| \operatorname{sgn}(\phi(\boldsymbol{x})). \tag{2}$$

Reinitialization with the signed distance function has been used in a number of different circumstances; for example Chopp [1] (minimal surfaces), Sussman *et al.* [2],[3] (free boundary problems in two-phase flow), Chen *et al.* [4] (crystal growth) Merriman *et al.* [5] (motion of multiple junctions). Reinitialization with distance functions has also been used in the development of fast level set methods by Peng *et al.* [7]. A different fast method (the fast marching method) has been used by Adalsteinsson & Sethian [6]. They obtain the distance function as a byproduct of the method.

There are several methods to reinitialize the level set function to the signed distance function. One possibility would be to compute the distance function using a discretized version of Eq. (2). This approach can be used for very accurate calculations if the location of the interface is accurately known. An example of its use is shown in §3. Such technique, however, is not very practical in level set, because it requires the accurate evaluation of the position of several interface points, and because the straightforward algorithm based on the discretization of Eq. (2) would be too expensive, the number of operation required being proportional to $N_g \times N_\Sigma$, where N_g is the number of grid points, and N_Σ denotes the number of points which discretize the interface Σ. An efficient implementation of this algorithm for computing the distance function in the framework of a *narrow band* level set method has been presented by Adalsteinsson & Sethian [6]. Strain has developed fast methods for computing distance functions using tree methods see [10], and [11].

A different approach, introduced in [2], is based on solving the following PDE

$$\begin{aligned} \frac{\partial \phi}{\partial t} &= \operatorname{sgn}(\phi^0)(1 - |\nabla\phi|), \\ \phi(x,0) &= \phi^0(x). \end{aligned} \tag{3}$$

The zero level set of ϕ^0 represents the location of the interface. When this equation is solved up to time T, then $\phi(x,T)$ is the signed distance function for all the points within distance T from the interface. As pointed out in [2], this is a Hamilton-Jacobi equation, and one could use upwind methods to compute its unique viscosity solution.

It is possible to rewrite this equation in the following form

$$\frac{\partial \phi}{\partial t} + \operatorname{sgn}(\phi^0) n \cdot \nabla\phi = \operatorname{sgn}(\phi^0) \tag{4}$$

where $n \equiv \nabla\phi/|\nabla\phi|$ is the unit normal to the level sets. In this form the equation appears as a scalar convection equation. The direction of propagation of the signal

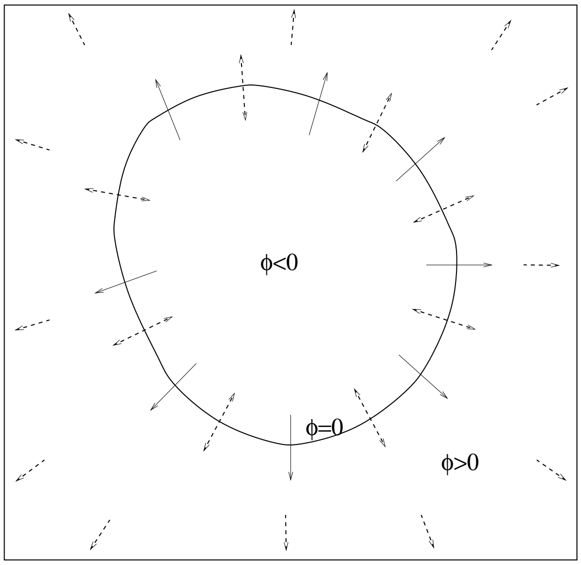

Figure 1: Propagation of the signal off the zero level set for equation (4). The arrows represent the unit normal to the level set. The dashed arrows represent the direction of propagation of the signal

is schematically illustrated in Figure 1. The continuous arrows represent the normal to the level set, n, while the dashed arrows represent the direction of propagation of the signal. It is clear from the figure that no boundary condition has to be assigned at the boarder of the computational domain, since the signal is propagating outward. The sign function sgn is defined as

$$\operatorname{sgn}(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x = 0 \\ -1 & \text{if } x < 0 \end{cases}$$

Note that on the zero level set of ϕ^0, the function ϕ is initialized to zero and it has to remain zero. This property is consistent with the above definition of the sign function. In numerical computation, smoothed versions $S(x)$ of the sign function will be used. They should maintain the property that $S(0) = 0$.

A similar approach, introduced by Sethian (see [6] and the references therein) is based on the idea of crossing times. One solves the equation

$$\frac{\partial \phi}{\partial t} + |\nabla \phi| = 0$$

both forwards in time and backward in time and calculates the time when ϕ changes sign at a particular node. This time is then the signed distance function.

2 The Problem

In this paper we shall examine the approach suggested in Ref. [2]. Eq. (3) can be discretized by using upwind methods. The first order 1D version used in [2] is given by

$$\phi_i^{n+1} = \phi_i^n - \Delta t\, S(\phi_i^0) G(\phi)_i, \tag{5}$$

where

$$G(\phi)_i = \begin{cases} \max(|a_+|, |b_-|) - 1 & \text{if } \phi_i^0 > 0 \\ \max(|a_-|, |b_+|) - 1 & \text{if } \phi_i^0 < 0 \end{cases} \tag{6}$$

with

$$\begin{aligned} a &\equiv D_x^- \phi_i = (\phi_i - \phi_{i-1})/\Delta x, & (7) \\ b &\equiv D_x^+ \phi_i = (\phi_{i+1} - \phi_i)/\Delta x, & (8) \end{aligned}$$

and, for any real number h, it is $h_+ = \max(h, 0)$, $h_- = \min(h, 0)$. The smoothed sign function S is given by

$$S(\phi) = \frac{\phi}{\sqrt{\phi^2 + \Delta x^2}}.$$

This scheme has been successfully used in several contexts; nevertheless it suffers from drawbacks. In 1D, under certain conditions, the zero of the level set function will tend to approach to the closest grid node, after several iterations. To illustrate this, we consider the initial condition

$$\phi^0(x) = (x - 0.4\Delta x)(x + 6)/2 + 1. \tag{9}$$

The results of the implementation of the above algorithm are illustrated in Figs. (2) and (3). This effect was pointed out to the authors by A. Sarti [8].

The explanation of this effect, and a simple procedure to overcome this drawback, are illustrated in the next section.

3 The sub-cell fix

We begin by remarking that Eq. (3) is a first order hyperbolic equation, which can be written as

$$\frac{\partial \phi}{\partial t} + w \frac{\partial \phi}{\partial x} = \operatorname{sgn} \phi_0, \tag{10}$$

where

$$w = \operatorname{sgn}(\phi_0) \operatorname{sgn}\left(\frac{\partial \phi}{\partial x}\right).$$

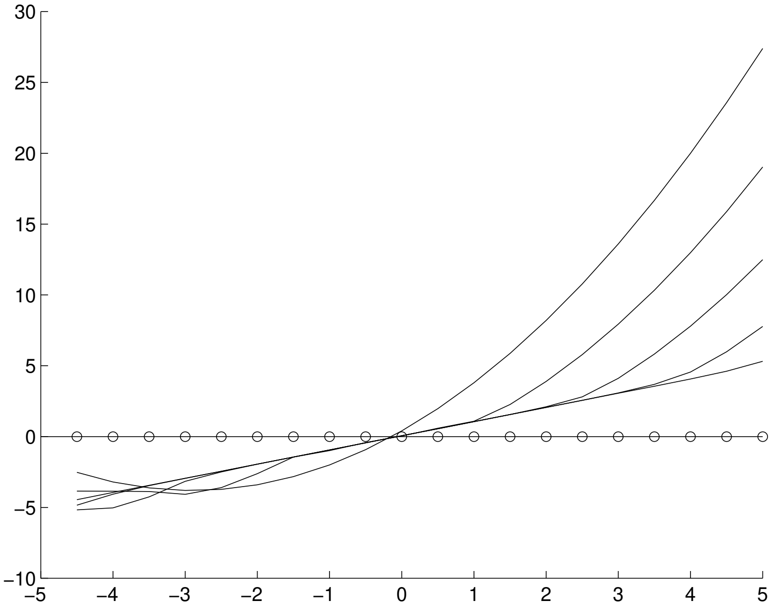

Figure 2: Construction of the distance function $\phi(x,t)$ using the original method given by Eq. (5) with the initial condition given by (9). Number of iterations $N_I = 0, 3, 6, 9, 12$. The domain is $\Omega = [-5, 5]$ and we take $\Delta t = 0.9\Delta x, \Delta x = 0.5$

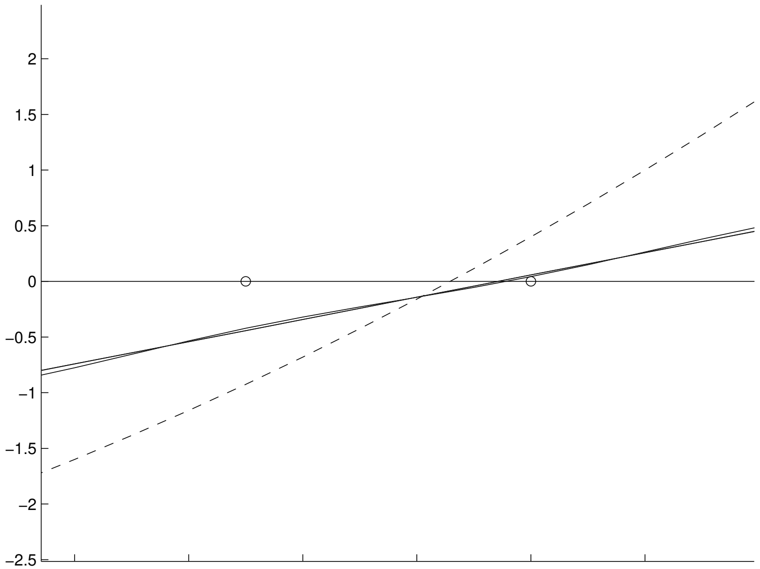

Figure 3: A close up of Figure 2 showing how the zero level set has moved. Cubic spline interpolation has been used to reconstruct $\phi(x,t)$ from its grid values for plotting purposes only. Dashed line: initial level set function $\phi^0(x)$.

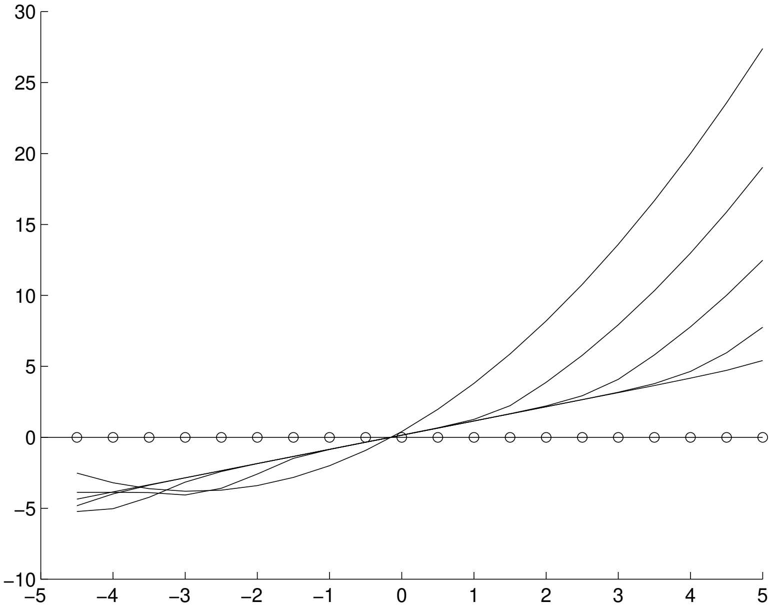

Figure 4: Construction of the distance function $\phi(x,t)$ using the new method given by Eq. (15) with the initial condition given by (9). Number of iterations $N_I = 0, 3, 6, 9, 12$. The domain is $\Omega = [-5, 5]$ and we take $\Delta t = 0.9\Delta x, \Delta x = 0.5$.

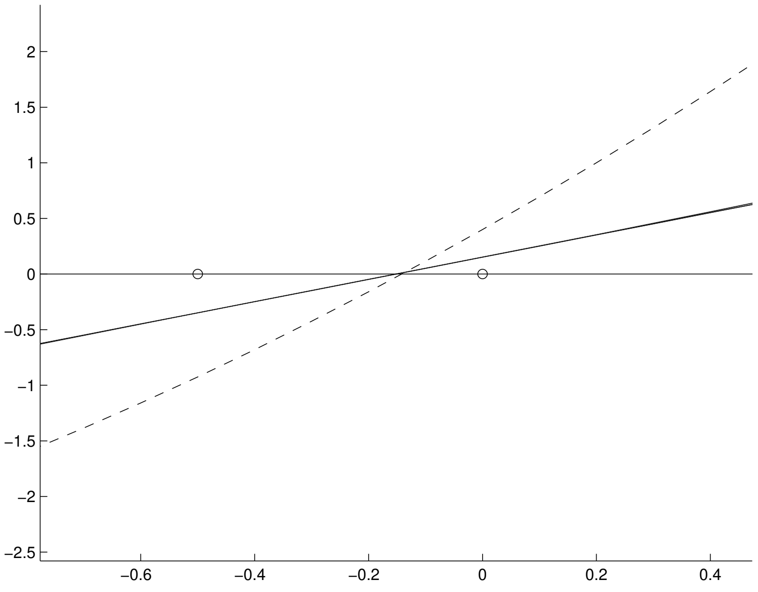

Figure 5: A close up of Figure 4 showing that the zero level set moves considerably less with the new method. Cubic spline interpolation has been used to reconstruct the function from its grid values. Dashed line: initial level set function $\phi^0(x)$.

In Eq. (10), the characteristics propagate outward from the interface, in the normal direction, with speed 1. The rate of change of the phase function along the characteristics is $+1$ (outside Ω), and -1 (inside Ω). Therefore after a time T, the value of the phase is the signed distance function from the interface.

Methods used to solve this equation are usually upwind methods, where the discrete derivatives are computed by upwind differencing according to the direction of the characteristics. In particular, this means that when differencing across the interface, this property will be violated. It is clear that the method presented in the introduction has differences across the interface.

The reasons for this is that discretization of the derivatives near the interface is not truly upwind, in the sense that part of the information is coming from the wrong side of the level set. This is illustrated in the following example (see Fig. (6)). Suppose we wish to update ϕ_i at $i = 4$. Then the application of the above algorithm would give

$$a = \frac{\phi_4 - \phi_3}{\Delta x}, \quad b = \frac{\phi_5 - \phi_4}{\Delta x},$$

In this case $a > 0$, $b > 0$, and $\phi_4^0 > 0$ and therefore we have from (5) that

$$\phi_4^{n+1} = \phi_4^n + \frac{|\phi_4^n - \phi_3^n|}{\Delta x}\Delta t.$$

Therefore, the value of the level set function ϕ_4^{n+1} depends on the value ϕ_3^n which is on the other side of the interface. This is inconsistent with upwinding since information should propagate outward from the interface.

As we shall see, modifying the numerical schemes to ensure that the schemes are truly upwind across the interface will dramatically reduce the movement of the interface. Moreover, we shall show that the motion of the interface is bounded in time by a constant that depends on the accuracy of the method. The new upwind scheme is obtained by a simple correction of the previous scheme. The new algorithm is obtained using Eqs. (5) and (6) with different expressions of $G(\phi)_i$ and of the sign function S. Near the interface, the function G is given by

$$G(\phi)_i = |D_x^{up}\phi_i| - 1, \tag{11}$$

where the upwind derivatives, D_x^{up}, of a function $\phi(x)$ near the interface are given by the geometrical consideration that the left derivative at point 4 (see Figure 6) is given by ϕ_4/D_4, where D_4 is the approximation of the distance function computed using the original level set function ϕ^0 (the length of the thick segment in Figure 6).

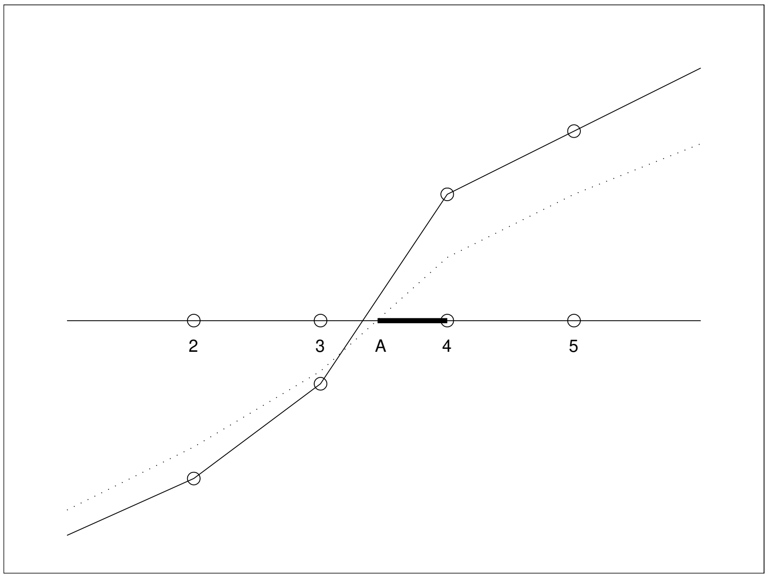

Figure 6: Example that shows why the original scheme (5–8) is not truly upwinding.The dashed line represents the piecewise linear reconstruction of the original level set function ϕ^0. Point A represents the intersection of the latter with the x axis, and the thick line is the approximation of the distance function at point 4.

This geometrical construction leads to the following scheme

$$D_x^{up}\phi_i = \begin{cases} \dfrac{\phi_i}{|D_i|} & \text{if} \quad \phi_i^0\phi_{i-1}^0 < 0 \\ -\dfrac{\phi_i}{|D_i|} & \text{if} \quad \phi_i^0\phi_{i+1}^0 < 0 \end{cases} \tag{12}$$

where D_i is an approximation of the signed distance function from the interface to the i-th node. The derivation of this formula relies on the fact that ϕ is zero at the interface and the characteristics always point outward from the interface. A possible choice of D_i (see Figure 6) is given by

$$D_i = \Delta x \frac{2\phi_i^0}{|\phi_{i+1}^0 - \phi_{i-1}^0|}. \tag{13}$$

The smoothed sign function S is given by

$$S = \begin{cases} \dfrac{D_i}{\Delta x} & \text{if} \quad \phi_i^0\phi_{i-1}^0 \leq 0 \quad \text{or} \quad \phi_i^0\phi_{i+1}^0 \leq 0 \\ \operatorname{sgn}(\phi_i^0) & \text{otherwise} \end{cases} \tag{14}$$

Summarizing, we compute the derivatives as usual if we are not within one grid cell from the interface. If we are within one cell from the interface then we compute the derivatives using the information that ϕ is zero on the interface.

If we use (12) and (14) in (5) the following scheme is found

$$\phi_i^{n+1} = \begin{cases} \phi_i^n - \dfrac{\Delta t}{\Delta x}(\mathrm{sgn}(\phi_i^0)|\phi_i^n| - D_i) & \text{if } \phi_i^0\phi_{i+1}^0 \text{ or } \phi_i^0\phi_{i-1}^0 < 0 \\ \phi_i^n - \Delta t\ \mathrm{sgn}(\phi_i^0)G(\phi)_i & \text{otherwise} \end{cases} \tag{15}$$

where $G(\phi)_i$ is given by Eq. (6). In writing (15) we made use of the fact that $\mathrm{sgn}(D) = \mathrm{sgn}(\phi_0)$.

Remark Whenever there is a topology change it is conceivable that the denominator in Eq. (13) becomes very small. In order to overcome this difficulty, a more robust expression for the signed distance function would be the following

$$D_i = \Delta x \frac{\phi_i^0}{\Delta\phi_i^0}, \tag{16}$$

where

$$\Delta\phi_i^0 = \max\{|\phi_{i+1}^0 - \phi_{i-1}^0|/2, |\phi_{i+1}^0 - \phi_i^0|, |\phi_i^0 - \phi_{i-1}^0|, \epsilon\}. \tag{17}$$

Remark Note that the CFL stability condition for the above scheme is $\Delta t < \Delta x$. This uniform stability condition is obtained by using the smoothed sign function (14). This function has the property of being zero on the original level set, as required. Furthermore, a uniform CFL stability condition on the time step requires a smaller value of S near the interface, in order to compensate the effect of an effectively smaller local grid size (the space derivative is computed with a local grid size which is effectively equal to $|D_i|$). The fact that the smoothed sign function vanishes near the interface does not change the equilibrium solution for large time.

Remark A possible variant of this scheme is to assign the value of the signed distance function D_i to $\phi(x_i)$ and to use it as boundary condition for the upwind scheme. Such variant would provide essentially the same accuracy.

Here we use the new scheme on the same examples shown in the previous section. In Figs. (4-5) we show the evolution of the distance function in 1D. It is evident that after an initial transient, the distance function converges to the correct value up to second order in Δx, and no approach towards the closest node is observed for the zero-level point.

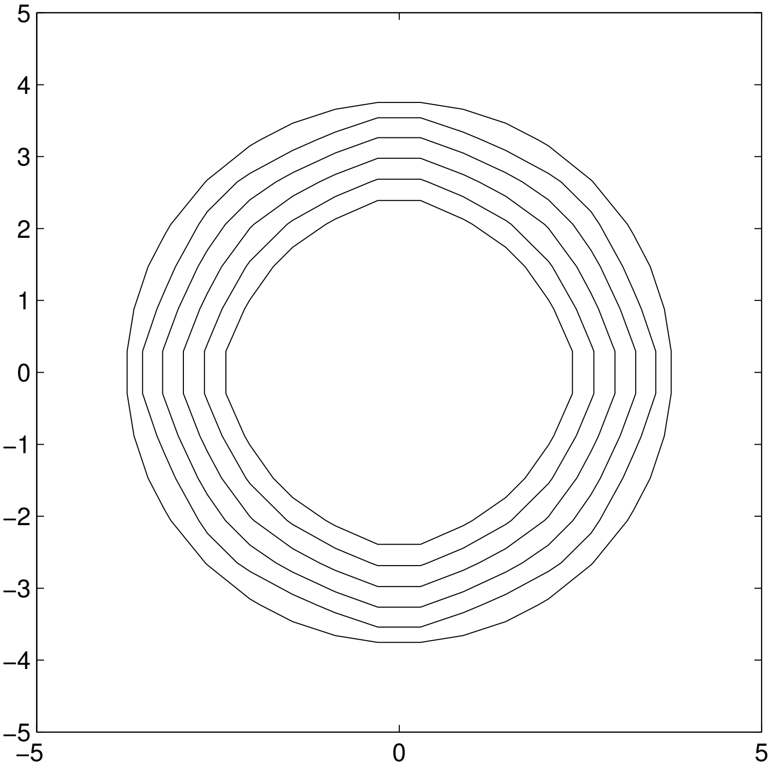

Figure 7: Shows the construction of the distance function in 2D using the original method as given by Eq. (19) with the initial condition given by (18). The key feature is that the interface moves considerably. In this figure we have plotted the zero level set of ϕ when the number of iterations is $N_I = 0, 160, 320, 480, 640, 800$. The domain is $\Omega = [-5, 5] \times [-5, 5]$ and we take $\Delta t = 0.5\Delta x, \Delta x = 10/16$

4 2D results

In 2 and 3 dimensions the problem is more severe, since repeated applications of the algorithm will cause the interface to loose area and shrink. We shall illustrate this with the following example. We consider

$$\phi^0(\boldsymbol{x}) = \sqrt{x^2 + y^2} - 4. \tag{18}$$

The zero level set is a circle with radius 4 and $\phi^0(\boldsymbol{x})$ is the signed distance function. If we apply the reinitialization algorithm to this function it should not move. A first order implementation of (3) in 2D is given by [2]

$$\phi_i^{n+1} = \phi_i^n - \Delta t \, S(\phi_i^0) G(\phi)_i \tag{19}$$

where

$$G(\phi)_i = \begin{cases} \sqrt{\max(a_+^2, b_-^2) + \max(c_+^2, d_-^2)} - 1 & \text{if } \phi_i^0 > 0 \\ \sqrt{\max(a_-^2, b_+^2) + \max(c_-^2, d_+^2)} - 1 & \text{if } \phi_i^0 < 0 \end{cases} \tag{20}$$

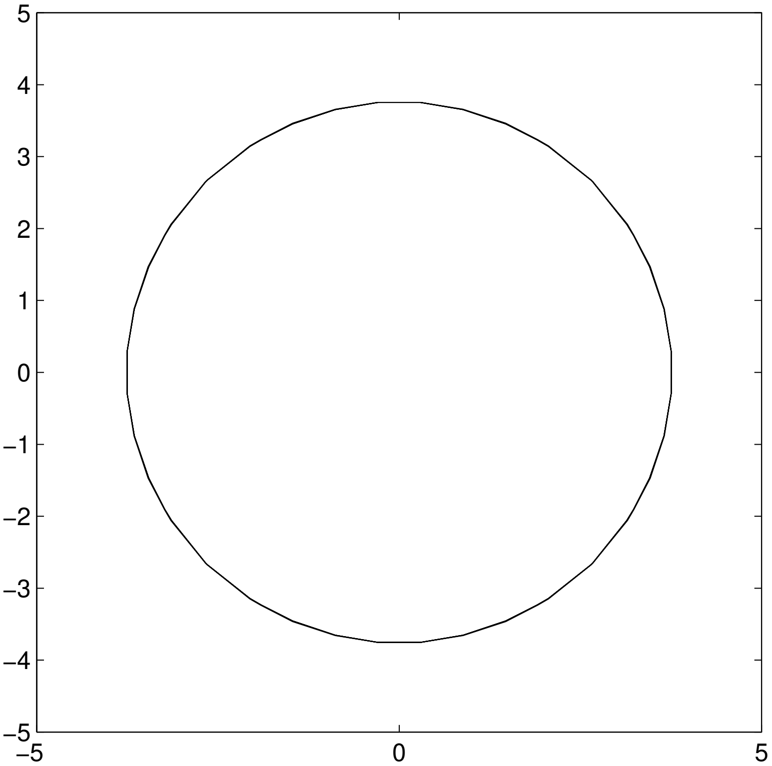

Figure 8: Shows the construction of the distance function in 2D using the new method as given by Eq. (19) with the initial condition given by (18). We see that with the new method the interface barely moves. In this figure we have plotted the zero level set of ϕ for the following number of iterations, $N_I = 0, 160, 320, 480, 640, 800$. The domain is $\Omega = [-5, 5] \times [-5, 5]$ and we take $\Delta t = 0.5\Delta x$, $\Delta x = 10/16$

with

$$\begin{cases} c & \equiv D_y^- \phi_i = (\phi_{i,j} - \phi_{i,j-1})/\Delta x \\ \\ d & \equiv D_y^+ \phi_i = (\phi_{i,j+1} - \phi_{i,j})/\Delta x \end{cases} \tag{21}$$

a and b are given by 2D versions of the expressions given by (5). The results are shown in shown in Fig.(7). We observe that not only does the circle shrink but there is considerable grid anisotropy. This effect is reduced if one uses higher order methods (Sussman, private communication). Thus we see that this algorithm produces an error that is proportional to the number of iterations. In most applications a small number of iterations of reinitialization procedure are applied each time step. Thus the total number iterations will be large; consequently, the error due to the reinitialization algorithm could in principle be rather large. Sussman and Fatemi [9] proposed to modify Eq. (5) by imposing the constraint that the total area must be preserved. More recent work [7] shows that there is an additional difficulty when the initial level set function is not close to a signed distance function. The authors propose to solve the problem by suitable modification of the mollified sign function, namely

$$S = \frac{\phi}{\sqrt{\phi^2 + |D\phi|^2 \Delta x^2}},$$

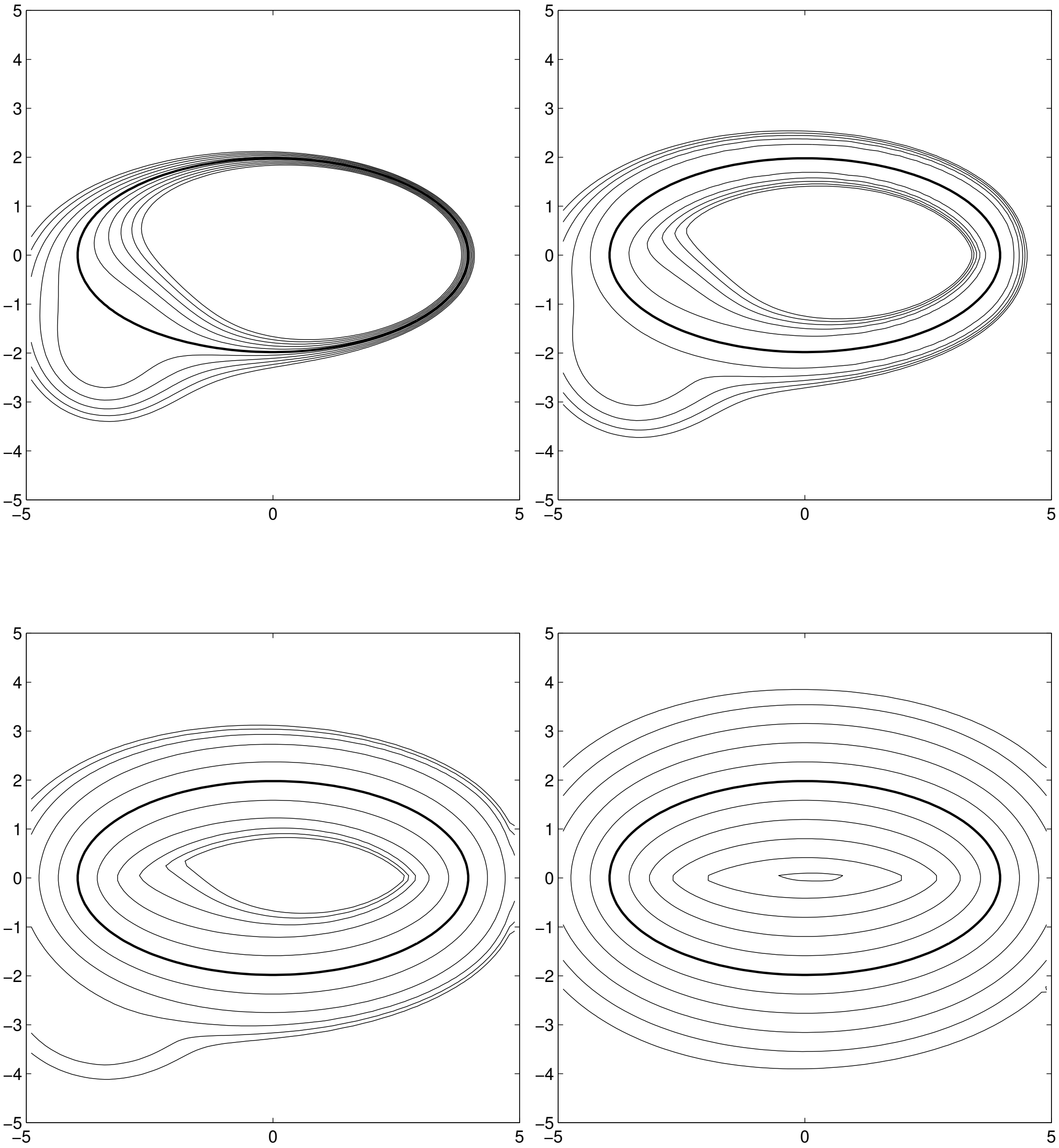

Figure 9: These figures show the result of applying the new method for the initial conditions given by (24). The domain is $\Omega = [-5, 5] \times [-5, 5]$ and we are using a 200×200 grid $dt = .5\Delta x$. The number of iterations is $0, 10, 25, 50$ starting from the top left and finishing in the lower right. The contours run from -1 to 1 and are spaced by 0.2

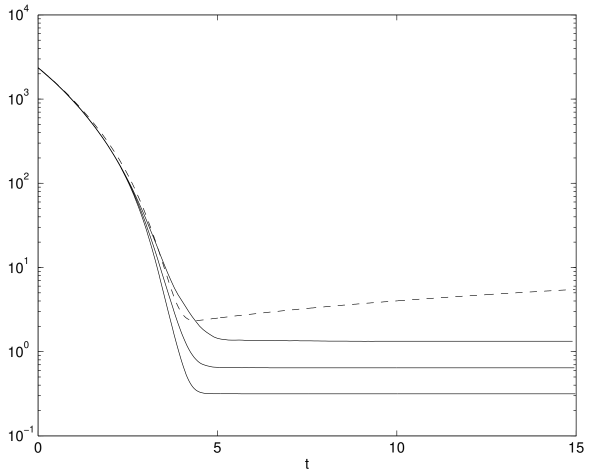

Figure 10: The L_1 error between the numerically computed distance function and the exact distance function (Eq. 25) is plotted as a function of time for the new method for $N = 50, 100, 200$ (solid lines) and for the old method with $N = 200$ (dotted line). These errors are for the same initial condition as used in Fig. 9

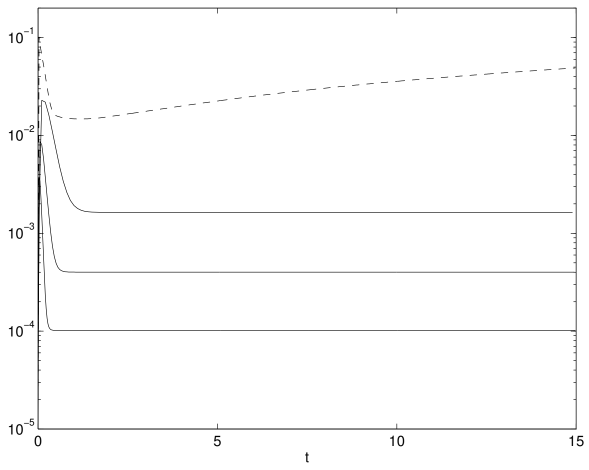

Figure 11: The L_1 error between the zero level set of ϕ^0 and $\phi(x,t)$ (defined by (27)) is plotted as a function of time for the new method for $N = 50, 100, 200$ (solid lines) and for the old method with $N = 200$ (dotted line). These errors are for the same initial condition as used in Fig. 9

where $D\phi$ is a discretization of $\nabla\phi$. In this paper we show that a simple modification of scheme (15) virtually removes both of these difficulties. We shall see that our improvement of the algorithm has an error bound that is *independent of the number of iterations.*

The scheme presented in the previous section can be straightforwardly extended in two dimensions. The resulting scheme is the following

$$\phi_{i,j}^{n+1} = \begin{cases} \phi_{i,j}^n - \dfrac{\Delta t}{\Delta x}(\mathrm{sgn}(\phi_{i,j}^0)|\phi_{i,j}^n| - D_{i,j}) & \text{if } (i,j) \in \Sigma_{\Delta x} \\ \phi_{i,j}^n - \Delta t \ \mathrm{sgn}(\phi_{i,j}^0) G(\phi)_{i,j} & \text{otherwise} \end{cases} \tag{22}$$

where the set $\Sigma_{\Delta x}$ defines the points which are within one grid point from the level set. More specifically, we say that $(i,j) \in \Sigma_{\Delta x}$ if

$$\phi_{i,j}^0\phi_{i-1,j}^0 < 0 \text{ or } \phi_{i,j}^0\phi_{i+1,j}^0 < 0 \text{ or } \phi_{i,j}^0\phi_{i,j-1}^0 < 0 \text{ or } \phi_{i,j}^0\phi_{i,j+1}^0 < 0 \ .$$

The quantity $D_{i,j}$ represents the distance of node (i,j) from the interface, and can be computed, for example, by

$$D_{i,j} = \frac{2\Delta x\, \phi_{i,j}^0}{[(\phi_{i+1,j}^0 - \phi_{i-1,j}^0)^2 + (\phi_{i,j+1}^0 - \phi_{i,j-1}^0)^2]^{1/2}}, \tag{23}$$

or by a more robust formula, analogue to the one used for the one dimensional scheme. The quantity G is computed according to formula (20).

In Fig.(8) we show the evolution of the zero level set of a phase function, with the same initial condition of the example shown in Fig. (7), but satisfying the new evolution equations.

Next, we compare the old and new scheme for the computation of the signed distance function from an ellipse.

We start with

$$\phi(x,y,0) = f(x,y)\left(\sqrt{\left(\frac{x^2}{a^2} + \frac{y^2}{b^2}\right)} - 1\right), \tag{24}$$

where

$$f(x,y) = \varepsilon + (x - x_0)^2 + (y - y_0)^2,$$

and the parameters are given by $a = 4$, $b = 2$, $\varepsilon = .1$, $x_0 = 3.5$, and $y_0 = 2$. This choice of $\phi(x,y,0)$ means that our initial condition has both small and large gradients near its zero level set. In order to check the validity of our new scheme, we compute the L^1 norm of the difference between the level set function and the distance function. More precisely, we compute

$$||\phi^n - D||_1 = \sum_{i,j} |\phi_{i,j}^n - D(\boldsymbol{x}_{i,j})|\Delta x^2, \tag{25}$$

where $D(\boldsymbol{x}_{i,j})$ is a very accurate approximation of the exact signed distance function which is computed as follows

$$D(\boldsymbol{x}_{i,j}) = \min_{1\leq p\leq N_\Sigma} |\boldsymbol{x}_{i,j} - \boldsymbol{x}_p| \operatorname{sgn}(\phi^0(\boldsymbol{x}_{i,j})),$$

where $\boldsymbol{x}_p$ is point that is exactly on the interface. There are N_Σ such points. For the ellipse we use $\boldsymbol{x}_p = (x_p, y_p)$ where

$$x_p = a\cos(2\pi p/N_\Sigma) \quad \text{and} \quad y_p = b\sin(2\pi p/N_\Sigma).$$

To estimate how much the interface moves we compute the following integral

$$E_\Sigma = \int_\Sigma |\phi(\boldsymbol{x}, t)| ds. \tag{26}$$

The discrete form is

$$E_\Sigma = \frac{1}{2}\sum_{p=1}^{N_\sigma} \left(|\widetilde{\phi}(\boldsymbol{x}_p, t)| + |\widetilde{\phi}(\boldsymbol{x}_{p+1}, t)|\right) |\boldsymbol{x}_{p+1} - \boldsymbol{x}_p|, \tag{27}$$

where $\boldsymbol{x}_p$ is given above and $\widetilde{\phi}(\boldsymbol{x}_p, t)$ is a third order interpolation of ϕ at $\boldsymbol{x}_p$.

The results are summarized in Fig. (10) and (11), where the log plot of the error is shown as function of time for the new algorithm (solid line) and for the old algorithm (dashed line). The computation has been performed using 50×50 , 100×100 and 200×200 grid points for the new algorithm, and 200×200 for the old one. In the approximation of $D(\boldsymbol{x}_{ij})$ we used $N_\Sigma = 2000$ points, and we checked that by using $N_\Sigma = 4000$ we obtain the same value of the error (within 0.2%). Notice that the new method is first order accurate (as expected), and that the error approaches very quickly a stationary value, while the classical algorithm produces a result that degrades with time.

Figure (11) shows that the error in the position of the zero level decreases by a factor four when the mesh grid size Δx is halved. This means that the position of the interface is preserved to second order accuracy. It is not surprising that a first order upwind gives second order accuracy, since the error within a fixed number of grid points from the interface is proportional to the local truncation error, which is second order in Δx. Furthermore, this property is essential for the construction of a consistent first order scheme that makes use of the reconstruction of the distance function at every time step as an intermediate stage of the computation, provided time step and grid size are of the same order of magnitude.

5 High order schemes

It is possible to construct high order versions of the present scheme. In this section we consider a second order scheme, which is based on second order formulas for the evaluation of the derivatives. The second order scheme in 1D is still given by Eqs.(5) and (6), but now Eqs. (7,8) are substituted by a second order approximation of space derivatives. Far from the interface, the one sided derivatives are obtained by the same scheme used in [9], appendix B. We report here the scheme for completeness. Given five points of the stencil around point x_i, $(x_p(k), f_p(k), k = -2, \ldots, 2)$, the left and right derivatives a and b are given as follows. First compute the table of divided differences

$$\begin{aligned}
\Phi[k,k+1] &= \frac{f_p(k+1) - f_p(k)}{x_p(k+1) - x_p(k)}, \quad k = -2, \ldots, 1 \\
\Phi[k,k+2] &= \frac{\Phi[k+1,k+2] - \Phi[k,k+1]}{x_p(k+2) - x_p(k)}, \quad k = -2, \ldots, 0
\end{aligned}$$

Then compute a and b as

$$c_- = \mathrm{MM}(\Phi[-2,0], \Phi[-1,1]), \quad c_+ = \mathrm{MM}(\Phi[-1,1], \Phi[0,2])$$

$$\begin{aligned}
a &\equiv D_x^- \phi_i = \Phi[-1,0] + c_-(x_p(0) - x_p(-1)), \\
b &\equiv D_x^+ \phi_i = \Phi[0,1] + c_+(x_p(0) - x_p(1)),
\end{aligned}$$

where MM is the minmod function defined as

$$\mathrm{MM}(\alpha,\beta) = \begin{cases} \alpha & \text{if } |\alpha| \le |\beta| \text{ and } \alpha\beta > 0 \\ \beta & \text{if } |\alpha| > |\beta| \text{ and } \alpha\beta > 0 \\ 0 & \text{if } \alpha\beta \le 0 \end{cases}$$

The points of the stencil are chose as follows. If point x_i is not close to the zero level set then

$$x_p(k) = x_{i+k}, \quad f_p(k) = \phi_{i+k}, \quad k = -2, \ldots, 2$$

If point x_i is within one grid size from the interface, then the stencil will include the intersection of the function with the axis (see Figure 12). In most cases, the point of intersection can be efficiently computed by fitting a third order polynomial $x = x(\phi)$ through the grid points near the zero (marked by a circle).

The extension of such second order scheme to two dimension is straightforward. Next we show the numerical results obtained with the second order scheme in two dimensions.

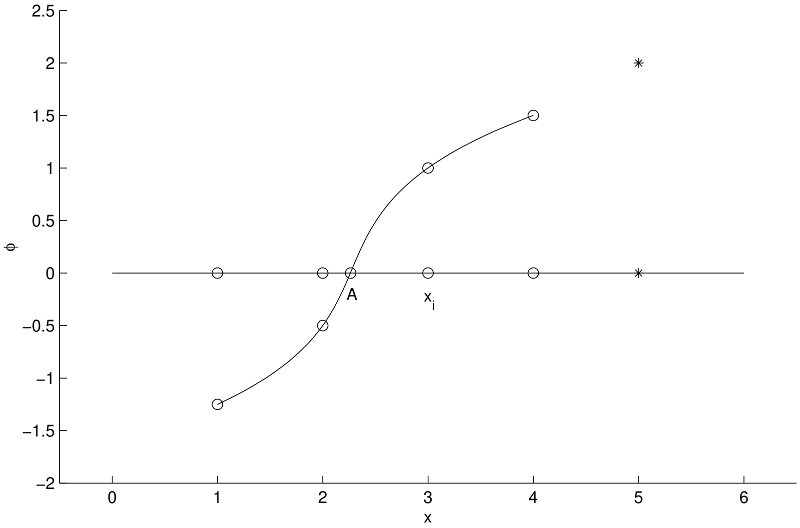

Figure 12: Use of the stencil for the approximation of left and right derivatives at point x_i. The vector x_p is given by $x_p = (x_{i-1}, x_A, x_i, x_{i+1}, x_{i+2})$. Point A is obtained constructing a third order polynomial $x = x(\phi)$ that fits the circled points.

In Figure 13 we reproduce the L^1 norm of the error obtained with the second order scheme. It is evident that the scheme provides a second order accurate evaluation of the distance function in the whole domain.

In Figure 14 we reproduce the error in the evaluation of the position of the level set. The scheme maintains the position of the level set with third order accuracy.

As a final remark observe that if one is interested in a second order accurate evaluation of the distance function, it is not necessary to reconstruct the space derivatives near the interface with the accurate procedure outlined above. It is sufficient to use the first order approximation of the derivatives near the interface, according to Eq. (12,13) in 1D and Eq. (**??**,23) in 2D, and the second order ENO scheme for the propagation of the distance function for all other points not adjacent to the interface. Figure 15 reproduces the L^1 norm of the error obtained with this intermediate scheme. It is evident that the distance is computed with second order accuracy.

Final remarks Because of its simplicity, accuracy, and efficiency, we believe that the present schemes can be effectively used as a tool for the computation of a signed distance function, either for problems where this function is required, or as an intermediate step in level set calculations. Because of accuracy and efficiency, the scheme can be used at each time step, without affecting the overall accuracy or efficiency of level set-based methods.

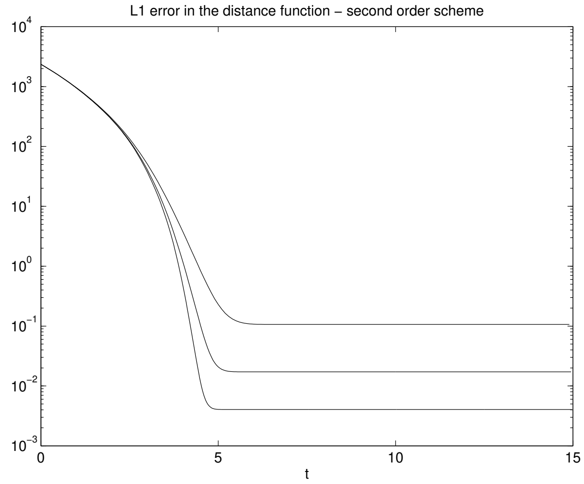

Figure 13: Second order method: the L_1 error between the distance numerically computed distance function and the exact distance function (Eq. 25) is plotted as a function of time for $N = 50, 100, 200$. These errors are for the same initial condition as used in Fig. 9.

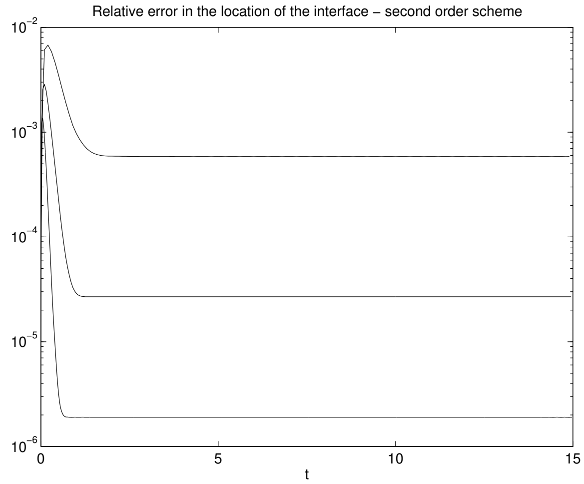

Figure 14: The L_1 error between the zero level set of ϕ^0 and $\phi(x,t)$ (defined by (27)) is plotted as a function of time for the second order method for $N = 50, 100, 200$. These errors are for the same initial condition as used in Fig. 9

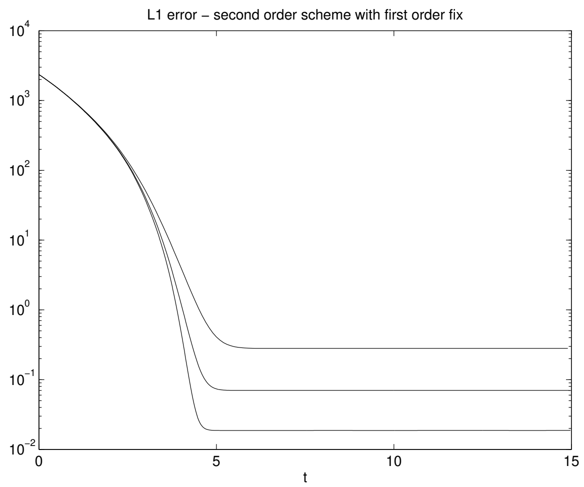

Figure 15: Second order method with linear fix near the zero level set. The L_1 error between the numerically computed distance function and the exact distance function (Eq. 25) is plotted as a function of time for $N = 50, 100, 200$. These errors are for the same initial condition as used in Fig. 9.

About the efficiency of the scheme, we observe that if the scheme is used as an intermediate step for the construction of the distance function near the zero level set in a *narrow band* level set method, then its complexity is only $O(N)$, where N is the total number of grid points. This is because the number of time steps for which the equation has to be solved is a fixed number, independent on the size of the problem, and therefore the number of operation is proportional to the number of the unknowns.

On the other hand, if one wants to use this method for the computation of the distance function at all points of the computational domain, then the complexity of the scheme (for the simple geometry illustrated in Figure 1) would be $O(N^{3/2})$ for two dimensional computation and $O(N^{4/3})$ for three dimensional computation. In this case the complexity of the algorithm would be higher than the complexity of the fast marching method. It is conceivable to imagine a more sophisticated variant of the method, in which only the values of the level set far from a front moving with speed one will be updated. In this way the scheme would be closer in spirit to the time marching method, and it would possibly be competitive with it. Such variant is, however, far beyond the scope of the present paper.

Acknowledgments

This work was supported in part by the National Science Foundation through a Mathematical Sciences Career grant (Grant No. DMS-9625190) and by DARPA and ONR through the Virtual Integrated Prototyping (VIP III) initiative.

References

[1] D. L. Chopp, *Computing Minimal Surfaces via Level Set Curvature Flow*, J. Comput. Phys., **106** 77-91 (1993).

[2] M. Sussman, P. Smereka, and S. Osher *A level set method for computing solutions to incompressible two phase flow* J. Comput. Phys. **119**, 146-159 (1994).

[3] M. Sussman and P. Smereka *Axisymmetric free boundary problems* J. Fluid Mech. **341**, 269-294 (1997).

[4] S. Chen, B. Merriman, S. Osher, and P. Smereka *A simple level set method for solving Stefan problems* J. Comput. Phys., **135** 8-29 (1997)

[5] B. Merriman, J. Bence, and S. Osher, *Motion of Multiple Junctions: a Level Set Approach*, J. Comput. Phys., **112** 334-363 (1994).

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

[7] D. Peng, B. Merriman,S. Osher,H. Zhao, and M. Kang, *A PDE based fast local level set method*, UCLA- CAM report 98-25 1998.

[8] A. Sarti, private communication.

[9] M. Sussman and E. Fatemi, *An efficient, interface-preserving level set redistancing algorithm and its application to interfacial incompressible fluid flow*, SIAM J. Sci. Comput. **20**, 1165-1191 (1999).

[10] J. Strain, *Fast tree-based redistancing for level set computations*, J. Comput. Phys. **152**, 664-686 (1999).

[11] J. Strain, *Tree methods for moving interfaces*, J. Comput. Phys. **151**, 616-648 (1999).

[12] J.A. Sethian, *Level Set Methods: evolving interfaces in geometry, fluid mechanics, computer vision, and materials science* Cambridge University Press (1996).