BACK AND FORTH ERROR COMPENSATION AND CORRECTION METHODS FOR REMOVING ERRORS INDUCED BY UNEVEN GRADIENTS OF THE LEVEL SET FUNCTION

TODD F. DUPONT* AND YINGJIE LIU†

Abstract. We propose a method that significantly improves the accuracy of the level set method and could be of fundamental importance for numerical solutions of differential equations in general. Level set method uses the level set function, usually an approximate signed distance function Φ to represent the interface as the zero set of Φ. When Φ is advanced to the next time level by an advection equation, its new zero level set will represent the new interface position. But the non zero curvature of the interface will result in uneven gradients of the level set function which induces extra numerical error. In stead of attempting to reduce this error directly, we update the level set function Φ forward in time and then backward to get another copy of the level set function, say Φ_1. Φ_1 and Φ should have be equal if there were no numerical error. Therefore $\Phi - \Phi_1$ provides us the information of error induced by uneven gradients and this information can be used to compensate Φ before updating Φ forward again in time.

Key words. flux corrected transport, front tracking, level set method.

AMS subject classifications. 65M60, 65M12

1. Introduction.

Interface computation is a challenging and rewarding area for scientific research in recent years. There are many different approaches on how to track or capture the interface and how to do it more accurately, more robustly during topological changes of the interface.

Level set method was proposed by Osher and Sethian [8] to compute the interface motion indirectly by use of the zero set of the level set function. Recent developments on improving the level set method can be found for example on [6, 10, 1, 12, 4, 5, 9, 3, 7].

Using level set method for interface tracking enjoys great simplicity by solving a geometric problem with PDE based method. Following the pioneering work of [8], given a velocity field $\mathbf{u}$, the level set function Φ satisfies

$$\frac{\partial \Phi}{\partial t} + \mathbf{u} \cdot \bigtriangledown \Phi = 0. \tag{1.1}$$

Since the velocity field could create large variation in Φ, there is usually an auxiliary equation to solve till the steady state at each time step (Sussman *et. al.* [10]),

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

This procedure is supposed to transform the Φ into a signed distance function without changing its zero level set.

In Gomes and Faugeras [5], a modified advection equation is introduced in order to keep Φ a signed distance function while still advance the zero level set along the velocity field.

Even if the level set function is a perfect signed distance function, the curvature of the interface is still going to create variations in the gradient of the level set function,

*Department of Computer Science, University of Chicago, Chicago, Illinois 60637 (dupont@cs.uchicago.edu).
†School of Mathematics, Georgia Institute of Technology, Atlanta, GA 30332 (yingjie@math.gatech.edu).

which will cause uneven dissipative error. Also since the zero set of Φ is not located at the grid nodes, the discretization of equation (1.2) adds even more error to the interface position. In Enright *et. al.* [3], the excessive dissipation of discretizing (1.1) and (1.2) is addressed by using particles which move along characteristics of (1.1) to help regulate the interface. In Sussman and Fatemi [9], a volume conserving discretization method is introduced for more accurately approximating the solution of (1.2). Besides these techniques, very high order (3rd-5th order) non-oscillatory spatial and the corresponding high order Range-Kutta temporal differencing are used for discretization of (1.1) and (1.2).

Here we propose another simple systematical way of solving this problem. Though we use level set function and the level set advection equation (1.1) to describe the method we propose. It may find applications in much broader area in the numerical solutions of ordinary or partial differential equations.

2. Backward Error Compensation and Forward Error Correction. Suppose we have a fixed computational mesh (with maximum cell size h) on a fixed domain $D \subset R^n$. let Φ^n be the level set function defined at time level t_n. $\mathbf{v}$ is a velocity field given in time intervals (t_n, t_{n+1}). Let $\mathcal{L}$ denote the mapping from an initial value of equation (1.1) at time t_n to the solution at time t_{n+1} with some given boundary value on the influx boundary. Let $\phi^{n+1} = \mathcal{L}(\Phi^n)$. We also solve the equation (1.1) numerically using a r-th order non-oscillatory scheme (e.g. upwind scheme, ENO, WENO, *etc*) to obtain a discretized level set function $\tilde{\Phi}^{n+1}$ at time level t_{n+1}, say $\tilde{\Phi}^{n+1} = \mathcal{L}_h \Phi^n$ where $\mathcal{L}_h$ represents the numerical operator. Let $\mathcal{L}_h^{-1}$ denotes the numerical operator of solving the equation (1.1) backward in time from t_{n+1} to t_n using the same scheme. Let $\Phi_1^n = \mathcal{L}_h^{-1}(\tilde{\Phi}^{n+1})$. If there were no numerical error, Φ^n and Φ_1^n would have been equal. Therefore $\Phi^n - \Phi_1^n$ provides information we may be able to take advantage of. Let x be a point at time level t_n which corresponds to the point $\hat{x}$ at time level t_{n+1} along the characteristics of equation (1.1). Let $e(\hat{x}) = \phi^{n+1}(\hat{x}) - \tilde{\Phi}^{n+1}(\hat{x})$ be the numerical error at $\hat{x}$ at time level t_{n+1}. Suppose we can write $e(\hat{x}) = e_g(\hat{x}) + e_o(\hat{x})$ where $e_g(\hat{x})$ is the portion of dissipative error contributed by the local distribution of the gradient of Φ^n near x. Also let $e(x) = \Phi^n(x) - \Phi_1^n(x) = e_g(x) + e_o(x)$ be the numerical error at x at time level t_n, where $e_g(x)$ is the portion of dissipative error contributed by the local distribution of the gradient of ϕ^{n+1} near $\hat{x}$. When the time step $dt = t_{n+1} - t_n$ is small enough we can assume the local distribution of the gradient of Φ^n near x is close enough to the local distribution of the gradient of ϕ^{n+1} near $\hat{x}$ and $e_g(\hat{x})$ is close enough to $\frac{1}{2}e_g(x)$. Therefore by assuming local linear dependence on small perturbation of the numerical scheme, we may add $\frac{1}{2}e_g(x)$ to Φ^n at x in order to to remove $e_g(\hat{x})$ from $e(\hat{x})$ at time level t_{n+1}! But it is almost impossible to separate $e_g(x)$ from $e(x)$. Since $e(x)$ is bounded by the local truncation error of $\mathcal{L}_h$, we can simply use $\frac{1}{2}e(x)$ to compensate Φ^n at x in order to achieve the same goal and the order of the local truncation error for the new scheme will be at least the same as that of $\mathcal{L}_h$ or better.

Therefore given Φ^n at time level t_n, we propose the following backward error compensation algorithm.

Step 1. Solve $\tilde{\Phi}^{n+1}$ using $\tilde{\Phi}^{n+1} = \mathcal{L}_h \Phi^n$.
Step 2. Solve Φ_1^n using $\Phi_1^n = \mathcal{L}_h^{-1}(\tilde{\Phi}^{n+1})$.
Step 3. Let $\Phi_2^n = \Phi^n + \frac{1}{2}(\Phi^n - \Phi_1^n)$.
Step 4. Solve Φ^{n+1} using $\Phi^{n+1} = \mathcal{L}_h \Phi_2^n$.

It is interesting to see what happen if we apply this method to a simple equation

$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = 0$$

on $[0, 1]$ with periodic boundary condition, and a first order upwind numerical scheme. Let $0 = x_0 < x_1 < \cdots < x_n = 1$ be a uniform partition and denote $u_i^n = u(x_i, t_n)$. Let t_n, t_{n+1} be the two adjacent time levels in time discretization and $\Delta t = t_{n+1} - t_n < \Delta x$, where $\Delta x = 1/n$. The 1st order upwind scheme can be written as

$$\tilde{u}_i^{n+1} = (1-\lambda)u_i^n + \lambda u_{i-1}^n, \tag{2.1}$$

where $\lambda = \Delta t/\Delta x < 1$. When solving the equation (2) backward in time with the same scheme, we have

$$\begin{aligned} b_i^n &= (1-\lambda)\tilde{u}_i^{n+1} + \lambda \tilde{u}_{i+1}^{n+1} \\ &= ((1-\lambda)^2 + \lambda^2)u_i^n + \lambda(1-\lambda)(u_{i-1}^n + u_{i+1}^n). \end{aligned} \tag{2.2}$$

Following the above Step 3, we can define the compensated numerical solution at t_n as

$$\begin{aligned} c_i^n &= u_i^n + \tfrac{1}{2}(u_i^n - b_i^n) \\ &= u_i^n - \tfrac{1}{2}\lambda(1-\lambda)(u_{i+1}^n - 2u_i^n + u_{i-1}^n). \end{aligned} \tag{2.3}$$

If we write $c^n = \mathcal{S}u^n$, then $\mathcal{S}$ may be viewed as a sharpening operator. So this method can be related to the flux corrected transport method (Boris and Book [2]) which compares the difference of the numerical solutions from a low order scheme and a high order scheme to obtain the anti-diffusion term, and then "correct" the anti-diffusion term to avoid oscillation. Our strategy doesn't require two schemes to compute the solutions, and doesn't have to "correct" the error with a limiter. In stead, we compensate the solution before evolving a small time step so that the numerical diffusion during this time step will actually make the compensated numerical solution more accurate. Finally applying the upwind scheme forward again we have

$$\begin{aligned} u_i^{n+1} &= (1-\lambda)c_i^n + \lambda c_{i-1}^n \\ &= (-\tfrac{1}{2}\lambda^2 + \tfrac{1}{2}\lambda^3)u_{i-2}^n + (\tfrac{1}{2}\lambda + 2\lambda^2 - \tfrac{3}{2}\lambda^3)u_{i-1}^n + \\ &\quad (1 - \tfrac{5}{2}\lambda^2 + \tfrac{3}{2}\lambda^3)u_i^n + (-\tfrac{1}{2}\lambda + \lambda^2 - \tfrac{1}{2}\lambda^3)u_{i+1}^n. \end{aligned} \tag{2.4}$$

The local truncation error for the scheme (2.4) is $O(\Delta x^3)$. Therefore it improves the order of scheme (2.1) by one. We also conduct a numerical test to see if it can stand up to the numerical diffusion after a large number of iterations. We set up the initial data as a square: $u = 0$ for $x \in (0, 0.3) \cup (0.7, 1)$ and $u = 1$ for $x \in (0.3, 0.7)$. With $n = 100$, $\Delta x = 0.01$, $\Delta t = 0.005$, we plot the numerical solution against the exact one at time $t = 10$, *i.e.* after 2000 iterations. See Fig. 2.1.

The dual of the above backward error compensation method can be formulated as follows. Let us called it the forward error correction method. For given Φ^n at time level t_n,

Step 1. Solve $\tilde{\Phi}^{n+1}$ using $\tilde{\Phi}^{n+1} = \mathcal{L}_h \Phi^n$.
Step 2. Solve Φ_1^n using $\Phi_1^n = \mathcal{L}_h^{-1}(\tilde{\Phi}^{n+1})$.
Step 3. Solve $\tilde{\Phi}_1^{n+1}$ using $\tilde{\Phi}_1^{n+1} = \mathcal{L}_h \Phi_1^n$.
Step 4. Let $\Phi^{n+1} = \tilde{\Phi}^{n+1} + \frac{1}{2}(\tilde{\Phi}^{n+1} - \tilde{\Phi}_1^{n+1})$.

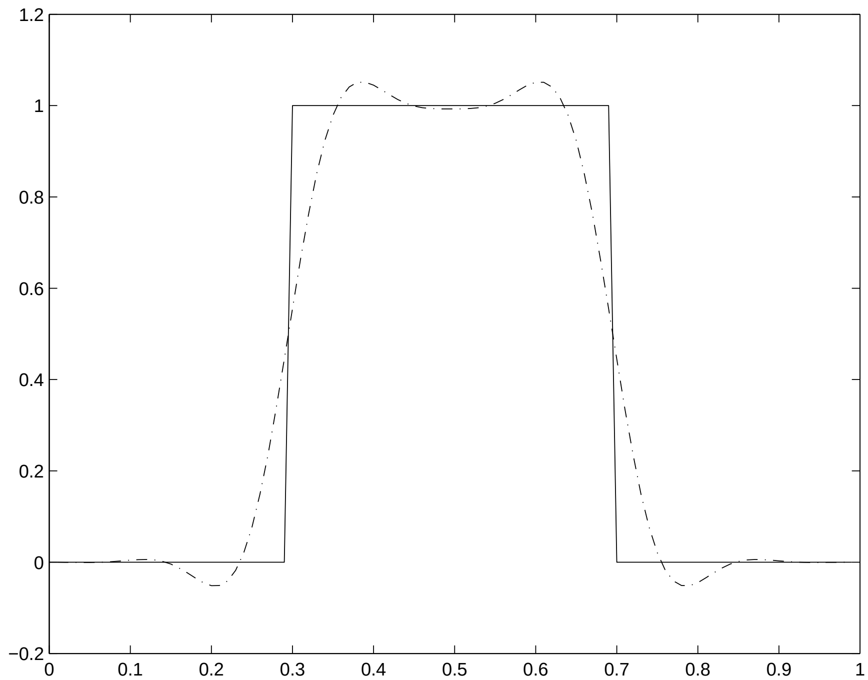

FIG. 2.1. *Square wave after* 2000 *time steps of computation (*$T = 10$*). The linear equation is computed using a 1st order upwind scheme with backward error compensation,* $n = 100$*.*

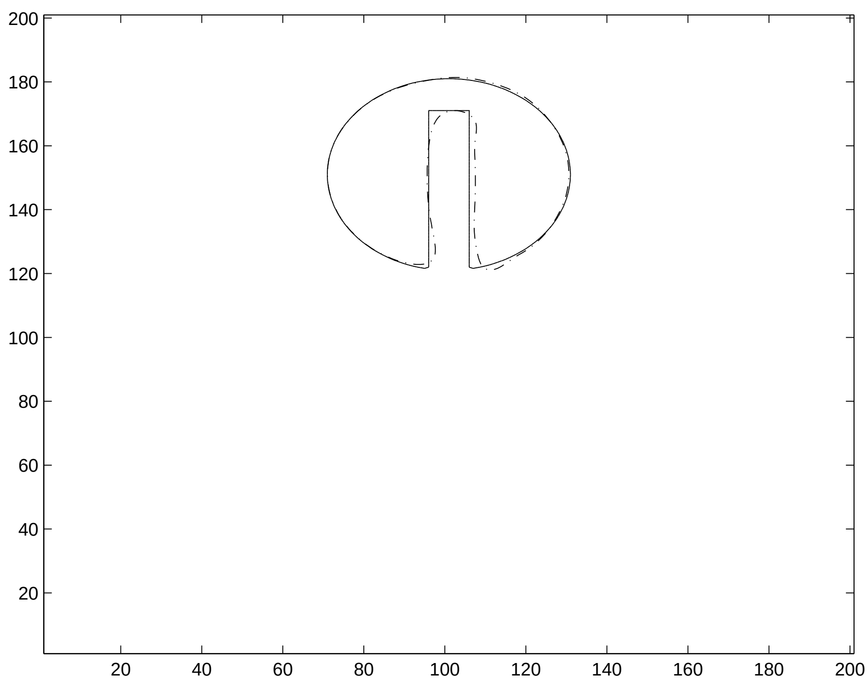

FIG. 2.2. *Zalesak's problem. Comparison of a notched disk that has been rotated one revolution. Level set equation is computed using 1st order upwind scheme with forward error correction,* $n = 200$*,* $\Delta x = 0.5$*.*

3. Tests On Zalesak's Problem. The two space dimensional Zalesak's Problem [11] can be described as follows: Set a rotational velocity field $(u, v) = (\frac{\pi}{314}(50 - y), \frac{\pi}{314}(x - 50))$ in a domain $(0, 100) \times (0, 100)$. Initially there is a cutout circle centered at $(50, 75)$ with radius 15. The slot being cut out has width 5 and length 25. Every point of this cutout circle is supposed to move along the local velocity field. We set the initial level set function Φ to be a signed distance function which is negative inside the notched disk and positive outside. The computation is up to the final time $t = 628$, allowing a full revolution of the disk. The challenge for computation with level set method is that this disk has corner points, curves, straight lines and a very narrow slot (when the mesh size is 1 or 0.5, the slot width is 5 or 10 mesh cell sizes respectively). In the first test we compute this problem with $n = 100$, $\Delta x = 1$, CFL factor 0.5. The level set advection equation (1.1) is discretized by a first order upwind scheme and the backward error compensation method described in the previous section is applied. There is no re-distancing of the level set function at each time level. In Fig. 3.1 the computed disk is drawn against the exact one after one revolution. Though the error is large due to the large cell size and the 1st order scheme we are using, the essential characters like the sizes of the disk and the slot survive, which is particularly surprising because if this test were done without backward error compensation, the whole disk would have disappeared after one revolution.

In the second test we basically repeat the first test but with the cell size shrinked by one half, *i.e.*, $n = 200$, $\Delta x = 0.5$. The final solution is shown in Fig. 3.2. If this test were done without backward error compensation, the disk would have shrinked to about one half of the original size and the slot would have disappeared (not shown in the picture). We also did the second test using forward error correction described in the previous section, see Fig. 2.2. It is similar to the result in Fig. 3.2 using backward error compensation.

The comparison tests without backward error compensation or forward error correction are shown in Fig. 3.3 with $n = 300$, $\Delta x = 1/3$, and in Fig. 3.3 with $n = 600$, $\Delta x = 1/6$. The level set advection equation (1.1) is still discretized by the first order upwind scheme, and we also re-distance the level set function at each time level using equation (1.2) discretized by the same first order upwind scheme, which improves the numerical results.

4. Acknowledgments. The authors thank Dr. Hongkai Zhao for the helpful discussion on the curvature induced error for level set method, and thank Dr. Luca Dieci for helping us compare this method to some numerical methods for ODE.

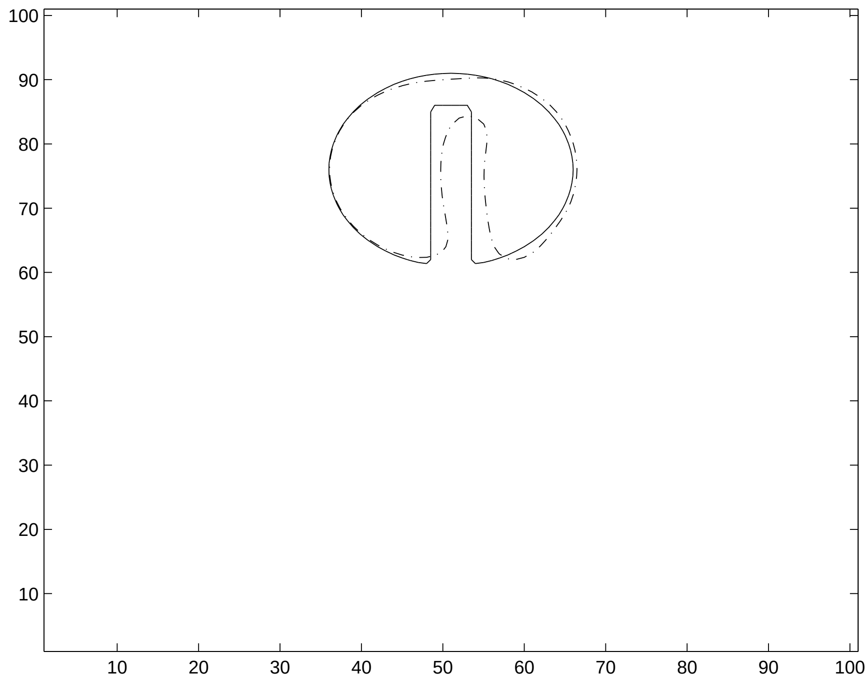

Fig. 3.1. *Zalesak's problem. Comparison of a notched disk that has been rotated one revolution. Level set equation is computed using 1st order upwind scheme with backward error compensation,* $n = 100$, $\Delta x = 1$.

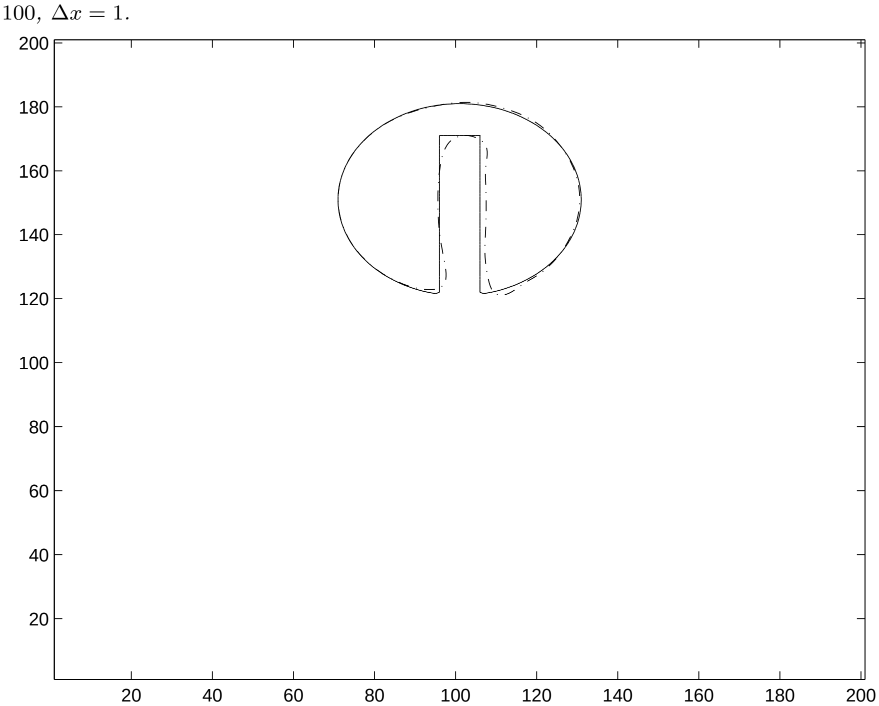

Fig. 3.2. *Zalesak's problem. Comparison of a notched disk that has been rotated one revolution. Level set equation is computed using 1st order upwind scheme with backward error compensation,* $n = 200$, $\Delta x = 0.5$.

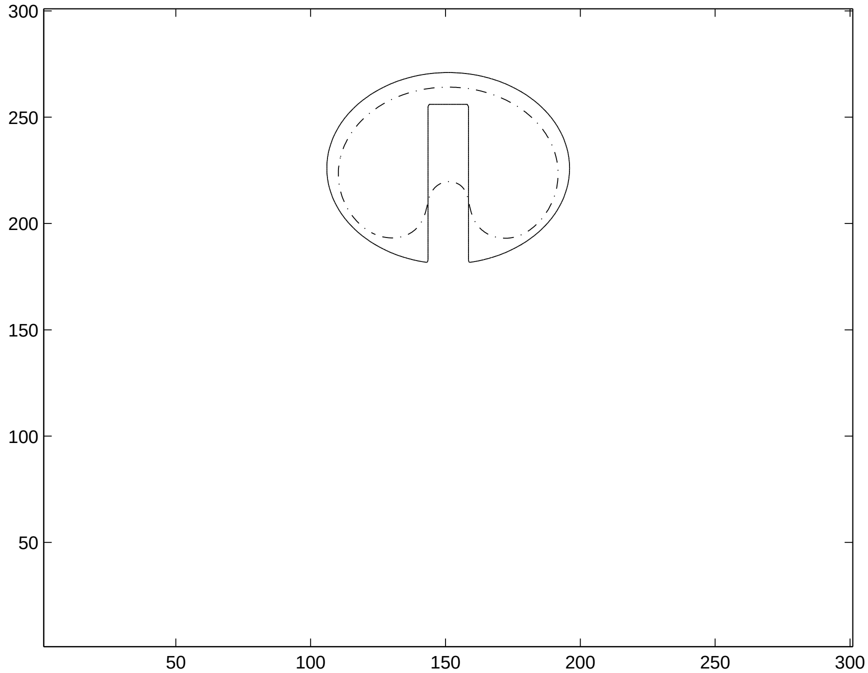

Fig. 3.3. *Zalesak's problem. Comparison of a notched disk that has been rotated one revolution. Level set equation is computed using 1st order upwind scheme without backward error compensation,* $n = 300$, $\Delta x = 1/3$.

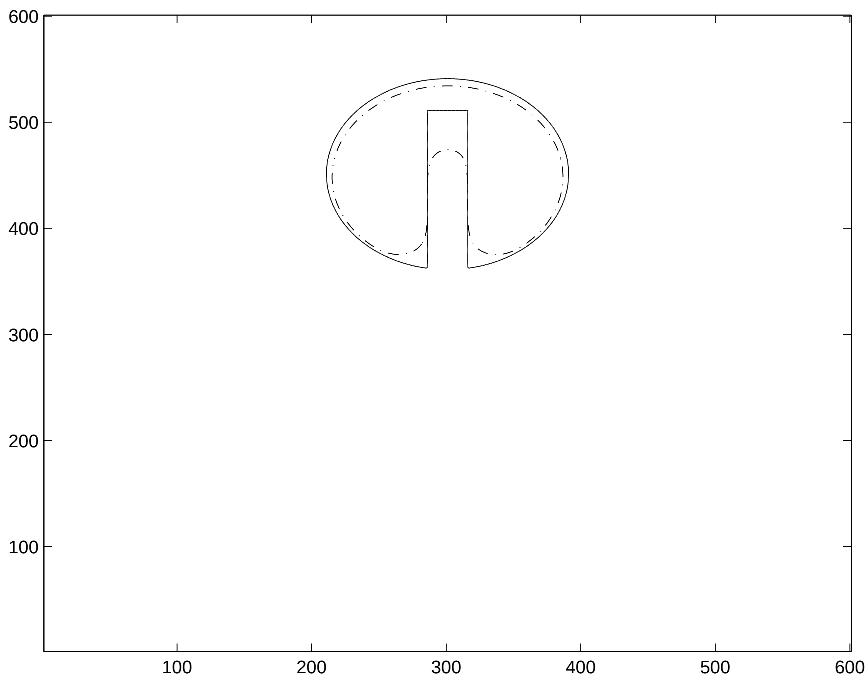

Fig. 3.4. *Zalesak's problem. Comparison of a notched disk that has been rotated one revolution. Level set equation is computed using 1st order upwind scheme without backward error compensation,* $n = 600$, $\Delta x = 1/6$.

REFERENCES

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

[2] J. P. Boris and D. L. Book. Flux-corrected transport i. shasta, a fluid transport algorithm that works. *J. Comp. Phys.*, 11:38–69, 1973.

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

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

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

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

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

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

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

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

[11] S. T. Zalesak. Fully multidimensional flux-corrected transport. *J. Comput. Phys.*, 31:335–362, 1979.

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