BACK AND FORTH ERROR COMPENSATION AND CORRECTION METHODS FOR SEMI-LAGRANGIAN SCHEMES WITH APPLICATION TO INTERFACE COMPUTATION USING LEVEL SET METHOD

TODD F. DUPONT* AND YINGJIE LIU†

Abstract. Semi-Lagranging schemes have been explored by several authors recently for transport problems in particular for moving interfaces using level set method. We incorporate the backward error compensation method developed in [2] into the semi-Lagranging schemes with almost the same simplicity and three times the complexity of a first order semi-Lagranging scheme but improve the order of accuracy. When applying this simple semi-Lagranging scheme to the level set method in interface computation, we observe good improvement comparable to results computed with other more complicated methods.

Key words. CIR scheme, front tracking, level set method, MacCormack scheme, Semi-Lagrangian scheme.

AMS subject classifications. 65M60, 65M12

1. Introduction. Semi-Lagranging schemes, e.g., the Courant-Isaacson-Rees (CIR) scheme [1], have a distinguished feature that there is no CFL restriction for time step size, thus local spatial refinement without changing the time step becomes possible even for explicit temporal discretization. Recently there are many researches on the application of semi-Lagranging schemes for transport equations, in particular for computing level set equation (Osher and Sethian [11]) describing interface movement. Strain [17, 18, 19] has developed a series of fast semi-Lagranging schemes for computing level set equation which incorporates techniques including essentially non-oscillatory (ENO [8]) spatial interpolation and predictor-corrector temporal approximation, velocity smoothing which removes artifacts and enables large time step even for mean curvature flow and quadtree meshes with fast algorithms. Enright *et. al.* [4] apply the CIR scheme to the hybrid particle level set method [3] to simplify the method with almost no loss of resolution.

For a linear transportation equation $u_t + \mathbf{v} \cdot u_x = 0$, the CIR scheme calculates the numerical solution defined on a space-time rectangular grid as $U(x_i, t_{n+1}) = U(\hat{x_i}, t_n)$, where $\hat{x_i} = \Gamma_i(t_n)$, and $\Gamma_i(t)$ is the approximate characteristics curve passing through (x_i, t_{n+1}). Different approximations of $\hat{x_i}$ and $U(\hat{x_i}, t_n)$ (since U is only defined at the grid points (x_i, t_n)) will generate schemes with different approximation properties. For example, if one choose $\hat{x_i} = x_i - \mathbf{v}(x_i, t_n)(t_{n+1} - t_n)$ and linearly interpolate $U(\hat{x_i}, t_n)$ by the U values at two nearest grid points x_j, x_{j+1} s.t. $\hat{x_i} \in [x_j, x_{j+1}]$, one obtains a first order CIR scheme which will not increase the L_∞ norm of U as time increases. If $t_{n+1} - t_n$ is small enough so that $\hat{x_i} \in [x_{i-1}, x_{i+1}]$, which is known as the CFL condition, then the CIR scheme is actually the first order upwind scheme. Therefore the CIR scheme removes the CFL restriction by interpolating $U(\hat{x_i}, t_n)$ through the nearby U values at, say x_j, x_{j+1} s.t. $\hat{x_i} \in [x_j, x_{j+1}]$, instead of extrapolating from the

*Department of Computer Science, University of Chicago, Chicago, Illinois 60637 (dupont@cs.uchicago.edu). The work of this author was supported by the ASCI Flash Center at the University of Chicago under DOE contract B532820, and by the MRSEC Program of the National Science Foundation under award DMR-0213745.

†School of Mathematics, Georgia Institute of Technology, Atlanta, GA 30332 (yingjie@math.gatech.edu).

U values at $[x_{i-1}, x_{i+1}]$ when $\hat{x_i}$ is not in $[x_{i-1}, x_{i+1}]$. In order to achieve higher order of accuracy, higher degree (2nd or higher) polynomial interpolation is necessary in order to approximate $U(\hat{x_i}, t_n)$ from U values at nearby grid points and corresponding order of Runge-Kutta type temporal numerical integration is also necessary for approximating the characteristics Falcone and Ferretti [5] analyze the stability and convergence of a general class of semi-Lagrangian schemes.

In higher spatial dimension the first order CIR scheme only requires e.g. bilinear (in 2D) or trilinear (in 3D) spatial interpolation, and Euler temporal integration and thus is most convenient to use. One may ask if there is a convenient way of manipulating the first order CIR scheme to achieve higher order of accuracy simultaneously in both space and time without explicit construction of higher order spatial polynomials and temporal integration. MacCormack scheme [10] uses an upwind scheme followed by a downwind scheme to obtain simutaneous improved order of accuracy in both space and time for hyperbolic equations. In the setting of semi-Lagrangian schemes, the integration is along the approximate characteristics and the upwind discretization is not clearly defined at the "root" of the characteristics. We are interested in whether the backward error compensation algorithm introduced in [2] can be successfully applied to the CIR scheme. The backward error compensation algorithm is based on a simple observation that if one solves a hyperbolic system forward in time for one time step with a scheme (e.g. a first order scheme) and then backward in time for one time step with the same scheme, one obtains another copy of the solution at the initial time. The two copies of the solution should have been equal if there were no numerical errors (away from singularities). Therefore comparing the two copies of the solutions may give us information of the error we can take advantage of to improve the accuracy. In Shu and Osher [16], total variation diminishing (TVD) time discretizations are introduced which incorporate a "downwind" spatial discretization in order to achieve the TVD property in higher order approximation. It is obtained by discretizing a time reversed conservation laws in certain middle time steps.

The difficulties involved in the numerical computation of the level set method are on how to reduce the diffusion and artifacts near the singular points of the interface. Typically high order ENO or WENO schemes designed for Hamilton-Jacobi equation are used for computing the level set equation and redistancing. In Sussman and Puckett [20], level set method and volume-of-fluid method are combined so that one can have the smoothness of the level set representation of the interface for extracting informations like mean curvature etc and also have the property of local volume conservation from volume-of-fluid method. Enright *et al* [3] proposed the hybrid particle level set method which takes advantage of the high resolution of Lagrangian type tracking schemes near interface singularities (see e.g. Rider and Kothe [13], Glimm *et al* [7, 6], Tryggvason*et al* [23]), and also has the convenience of the level set method being able to automatically resolve the interface topological changes. Strain [17, 18, 19] addresses these difficulties by using the semi-Lagrangian schemes to compute the level set equation so that local spatial refinement can be done near the singularity points of the interface without having to reduce the time step. Here we propose an alternative semi-Lagrangian scheme which incorporates the backward error compensation [2] and could result in an efficient and much simpler implementation of the level set method.

2. Backward Error Compensation for Semi-Lagranging Schemes. The level set method proposed by Osher and Sethian [11] uses a continuous function $\Phi(x,t) \in R$ to represent an evolving interface as the zero contour set $\{(x,t) : \Phi(x,t) = 0\}$. Here $x \in R^d$ is the spatial variable and $t \in R$ represents time variable. For a given velocity field $\mathbf{v}(x,t) \in R^d$, the level set function Φ satisfies

$$\frac{\partial \phi}{\partial t} + \mathbf{v} \cdot \bigtriangledown \phi = 0. \tag{2.1}$$

Assume a uniform rectangular grid in R^d with spatial mesh size

$$\Delta x = (\Delta x_1, \Delta x_2, \cdots, \Delta x_d)$$

and time step size $\Delta t_n = t_{n+1} - t_n$. Given $\Phi(\cdot, t_n)$ values at the grid points $\{x_i : i \in Z^d\}$, the first order CIR scheme can be formulated as follows,

$$\Phi(x_i, t_{n+1}) = \Phi(\hat{x_i}, t_n), \tag{2.2}$$

where $\hat{x_i} = x_i - \mathbf{v}(x_i, t_n)\Delta t_n$. In one space dimension ($d = 1$), $\Phi(\hat{x_i}, t_n)$ is the linear interpolation of $\Phi(x_j, t_n)$ and $\Phi(x_{j+1}, t_n)$ where $\hat{x_i} \in [x_j, x_{j+1}]$. In two space dimensions $\Phi(\hat{x_i}, t_n)$ can be approximated by the bilinear interpolation of the $\Phi(\cdot, t_n)$ values at the vertices (grid points) of a cell containing $\hat{x_i}$, similarly for general space dimensions one can use the tensor product of one dimensional linear polynomials to interpolate. Denote $\Phi_i^n = \Phi(x_i, t_n)$.

The backward error compensation algorithm [2] can be applied to the CIR scheme as follows.

Step 1. Solve equation (2.1) forward in time to obtain $\tilde{\Phi}^{n+1}$ using the CIR scheme (2.2), with Φ^n as the initial value at time t_n..

Step 2. Solve equation (2.1) using the same method backward in time to obtain $\breve{\Phi}^n$. This is equivalent to solve the time reversed equation $\frac{\partial \Phi}{\partial t} - \mathbf{v} \cdot \bigtriangledown \Phi = 0$ forward in time using (2.2), with $\tilde{\Phi}^{n+1}$ as the initial value and time interval size Δt_n.

Step 3. Let $\underline{\Phi}_i^n = \Phi_i^n + \frac{1}{2}(\Phi_i^n - \breve{\Phi}_i^n)$ for all i.

Step 4. Solve equation (2.1) forward in time to obtain Φ^{n+1} using (2.2), with $\underline{\Phi}^n$ as the initial value at time t_n.

It should be noticed that the velocity field $\mathbf{v}$ is only evaluated at the grid points at time level t_n and t_{n+1} in the above algorithm and the CIR scheme (2.2) involves only local linear interpolation of $\Phi(\hat{x_i}, \cdot)$. Therefore the implementation of the above algorithm is trivial even for three space dimensions.

The dual of the above algorithm, say the forward error correction algorithm, can be applied to the CIR scheme as follows:

Step 1. Solve equation (2.1) forward in time to obtain $\tilde{\Phi}^{n+1}$ using the CIR scheme (2.2), with Φ^n as the initial value at time t_n..

Step 2. Solve equation (2.1) using the same method backward in time to obtain $\breve{\Phi}^n$. This is equivalent to solve the time reversed equation $\frac{\partial \Phi}{\partial t} - \mathbf{v} \cdot \bigtriangledown \Phi = 0$ forward in time using (2.2), with $\tilde{\Phi}^{n+1}$ as the initial value and time interval size Δt_n.

Step 3. Solve equation (2.1) forward in time to obtain $\underline{\Phi}^{n+1}$ using (2.2), with $\breve{\Phi}^n$ as the initial value at time t_n.

Step 4. Let $\Phi_i^{n+1} = \tilde{\Phi}^{n+1} + \frac{1}{2}(\tilde{\Phi}^{n+1} - \underline{\Phi}_i^{n+1})$ for all i.

If the velocity field $\mathbf{v}$ depends only on x, t, the above two algorithms are equivalent in the sense they will result in the same Φ^{n+1}. If the velocity field $\mathbf{v}$ depends on Φ, i.e., $\mathbf{v} = \mathbf{v}(\Phi(x,t), x, t)$, in the backward error compensation algorithm we may use the same velocity field in Step 4 as in Step 1 , thus the velocity field needs only be computed twice at Step 1 and Step 2; In the forward error correction algorithm, we may use the same velocity field in Step 3 as in Step 1 so that the velocity field needs only be computed twice. Under this velocity linearization process, we can easily see that the above two algorithms applying to the CIR scheme are equivalent.

3. Stability. In [2], we have proved the l_2 stability of the backward error compensation algorithm applying to the first order upwind scheme for 1D equation $u_t + u_x = 0$. We are going to prove some more general results for higher space dimension. Throughout the section, we assume $\mathbf{v}$ is a constant vector in equation (2.1) unless specified otherwise. Let $L:\ \ U^{n+1} = L(U^n)$ such that $U_i^{n+1} = \sum_{j=i-j_1}^{i+j_2} \alpha_j U_j^n$ be a linear scheme for equation (2.1), where $\alpha_j' s$ are constants depending on $\Delta t_n/\Delta x_l, l = 1, \cdots, d$. As in [9], we assume j_1, j_2 are two finite integers (corresponding to an explicit scheme) or if not (e.g., an implicit scheme), c_j decays exponentially fast to zero as $|j|$ goes to infinity. Denote the Fourier symbol of L to be $\rho_L(\xi_h) = \sum_{j=i-j_1}^{i+j_2} c_j e^{(j-i)\xi_h}$ which can be obtained by applying a Fourier transform. $\xi_h = (\xi_1\Delta x_1, \xi_2\Delta x_2, \cdots, \xi_d\Delta x_d)$ where $\xi = (\xi_1, \cdots, \xi_d)$ is the dual variable. Since the Fourier symbol is 2π periodic in ξ_h we consider $\xi_h \in [-\pi, \pi]^d$. $max_{\xi_h\in[-\pi,\pi]^d}|\rho_L(\xi_h)|$ is called the amplification factor of the scheme L. let $L^*:\ \ W^n = L^*(W^{n+1})$ be the corresponding linear scheme to solve equation (2.1) backward in time using scheme L . Applying the backward error compensation algorithm to scheme L we obtain a linear scheme for equation (2.1),

$$F:\ \ V^{n+1} = F(V^n) = L(I + \frac{1}{2}(I - L^*L))(V^n), \tag{3.1}$$

where I is the identity operator. Let $\rho_{L^*}(\xi_h)$ and $\rho_F(\xi_h)$ be the Fourier symbols of the schemes L^* and F respectively, We have the following theorem.

THEOREM 1. *Supose $\rho_{L^*}(\xi_h) = \overline{\rho_L(\xi_h)}$ for any $\xi_h \in [-\pi, \pi]^d$. Then $|\rho_F(\xi_h)| \leq 1$ for any $\xi_h \in [-\pi, \pi]^d$ if and only if $|\rho_L(\xi_h)| \leq 2$ for any $\xi_h \in [-\pi, \pi]^d$.*

Proof. Applying the Fourier transform to both sides of (3.1) we can easily see that

$$\rho_F(\xi_h) = \rho_L(\xi_h)(1 + \frac{1}{2}(1 - \overline{\rho_L(\xi_h)}\rho_L(\xi_h)))$$

Let $\eta = |\rho_L(\xi_h)|$, $G(\eta) = |\rho_F(\xi_h)|$, then the theorem is proved by inspecting the function $G(\eta) = \eta|\frac{3}{2} - \frac{1}{2}\eta^2|$ for $\eta \in [0, \infty)$. □

The above Theorem 1 not only implies that the backward error compensation algorithm applying to a stable (in l^2) scheme is stable (in l^2), but it can also turn some unstable schemes into stable (in l^2) ones. Throughout the paper, we say a scheme is stable if it is stable in the l^2 sense, unless specified otherwise. It is easy to verify that the condition in Theorem 1 is satisfied when applying the the backward error compensation algorithm to some classical schemes as described in the following three examples.

Example 1. For one space dimension $d = 1$, the first order upwind scheme for equation (2.1) has an amplification factor $|\rho| \leq 2$ if the CFL factor $|\mathbf{v}|\Delta t/\Delta x$ is no more than 1.5 (which is unstable for CFL factor greater than 1). Therefore applying the backward error compensation algorithm to it creates a stable scheme for CFL factor less than or equal to 1.5, with second order accuracy (See [2]).

Example 2. Using center spatial difference and forward Euler time difference for equation 2.1 will create an unstable scheme. When $d = 1$, the scheme has amplification factor $|\rho| \leq 2$ if the CFL factor is no more than $\sqrt{3}$, thus applying the backward error compensation algorithm to it creates a stable scheme for CFL factor less than or equal to $\sqrt{3}$, with second order accuracy (See Section 4).

Example 3. For one space dimension $d = 1$, the Lax-Friedreich scheme has an amplification factor no more than 2 if the CFL factor is less than or equal to 2 (it is stable only if the CFL factor is less than or equal to 1). Therefore applying the backward error compensation algorithm to it creates a stable scheme for CFL factor less than or equal to 2, with second order accuracy (See Section 4).

Now back to the CIR scheme for equation (2.1), we are going to verify that the condition of Theorem 1 is satisfied. Given Φ^n, Φ^{n+1} computed by the CIR scheme can be written as

$$\Phi_j^{n+1} = \sum_k \Phi_k^n \Psi_k(x_j - \mathbf{v}\Delta t_n) \tag{3.2}$$

where Ψ_k is the Lagrangian basis function which in each cell is a linear ($d = 1$) polynomial or a bilinear ($d = 2$) polynomial etc and satifies $\Psi_k \geq 0$, $\Psi_k(x_j) = \delta_{kj}$. Since the grid is uniform, it also has the symmetry $\Psi_k(x) = \Psi_k(2x_k - x)$ and $\Psi_k(x) = \Psi_j(x_j + x - x_k)$ for any $x \in R^d$ and any mesh indices $k, j \in Z^d$. Assume $x_j - \mathbf{v}\Delta t_n$ lies in a cell with vertices x_{j-s_l}, for some $s_l \in Z^d$, $l = 1, 2, \ldots, 2^d$, then $x_j + \mathbf{v}\Delta t_n$ will lie in a cell with vertices x_{j+s_l}, $s_l \in Z^d$, $l = 1, 2, \ldots, 2^d$. Let $c_l = \Psi_{j-s_l}(x_j - \mathbf{v}\Delta t_n)$, $l = 1, 2, \ldots, 2^d$. Due to the symmetry of the basis functions it can be verified that

$$\Psi_{j+s_l}(x_j + \mathbf{v}\Delta t_n) = c_l, \;\; l = 1, 2, \ldots, 2^d. \tag{3.3}$$

Since the basis functions have compact supports, (3.2) can be written as

$$\Phi_j^{n+1} = \sum_{l=1}^{2^d} \Phi_{j-s_l}^n \Psi_{j-s_l}(x_j - \mathbf{v}\Delta t_n). \tag{3.4}$$

We first prove the following stability result as a corolary of Theorem 1.

Corollary 2. *The CIR scheme with backward error compensation algorithm for equation (2.1) with constant coefficients has an amplification factor less than or equal to* 1 *for any mesh size Δx and time step size Δt_n.*

Proof. It has been shown in [5] that the CIR scheme has amplification factor less than or equal to 1. We only need to verify that the Fourier symbol of of the CIR scheme applying to the time reversed equation is the complex conjugate of that of the CIR scheme applying to 2.1, and the rest is implied by Theorem 1.

Applying the Fourier transform to (3.4) we find the Fourier symbol

$$\rho_{CIR} = \sum_{l=1}^{2^d} c_l e^{-is_l \cdot \xi_h}. \tag{3.5}$$

Similarly, given Φ^{n+1}, applying the CIR scheme for the time reversed equation yields

$$\Phi_j^n = \sum_{l=1}^{2^d} \Phi_{j+s_l}^{n+1} \Psi_{j+s_l}(x_j + \mathbf{v}\Delta t_n). \tag{3.6}$$

Applying the Fourier transform to (3.6) and use the symmetry (3.3) we obtain the Fourier symbol

$$\rho_{CIR^*} = \sum_{l=1}^{2^d} c_l e^{is_l \cdot \xi_h}. \tag{3.7}$$

Therefore we can see that the two symbols ρ_{CIR} and ρ_{CIR^*} are complex conjugates. □

Next we consider the equation (2.1) with variable coefficients $\mathbf{v}(x,t)$. For simplicity consider the spatial domain $\Omega = [0,1]^d$ with periodic boundary condition (periodic along each coordinate direction) and the time domain $[0,T]$. Let Φ_i^n be the numerical solution at (x_i, t_n) computed by the CIR scheme with backward error compensation. We obtain the following l^∞ stability result relative to Δt_n (but not to Δx) following [5].

Theorem 3. *Suppose $\mathbf{v}(x,t)$ is defined in $\Omega \times [0,T]$ and satisfies*

$$\sup_{(x,t)\in\Omega\times[0,T]} |\mathbf{v}(x,t)| \le c_0$$

for some positive constant c_0. Then there is a constant c depending on c_0 and Δx such that

$$\|\Phi^n\|_{l^\infty} \le c\|\Phi^0\|_{l^\infty}$$

for any time step size $\Delta t = T/N$, $n = 1, 2, \cdots, N$.

Proof. We first consider a uniform global mesh for R^d so that the uniform mesh in Ω coincides with the mesh in Ω. Consider the Lagrangian basis functions Ψ_j for every mesh point x_j in the global mesh. Given the numerical solution at time level t_n, Φ_j^n for every $x_j \in \Omega$, first extend $\{\Phi_j^n\}$ to the global mesh periodically, and still use the notation Φ_j^n without causing any ambiguity. According to (3.2) the CIR scheme compute the next time level solution as

$$\tilde{\Phi}_j^{n+1} = \sum_k \Phi_k^n \Psi_k(x_j - \mathbf{v}(x_j, t_n)\Delta t) \tag{3.8}$$

for every $x_j \in \Omega$. This can be written in a matrix form $\tilde{\Phi}^{n+1} = A\Phi^n$ where $\tilde{\Phi}^{n+1} = (\tilde{\Phi}_j^{n+1})$, $\Phi^n = (\Phi_j^n)$ are column vectors, $A = (a_{kj})$ is a matrix with $a_{kj} = \delta_{kj} + r_{kj}$, where k, j are such that $x_k, x_j \in \Omega$ because of the periodicity of Φ_k^n. There is a constant $C_1(\Delta x, c_0)$ such that $|r_{kj}| \le C_1\Delta t$ since the basis functions are of compact support, continuous and piece-wise linear ($d = 1$), bilinear ($d = 2$) etc. Therefore $A = I + R_1$ with R_1 satisfying $\|R_1\|_{l^\infty} \le C_2(\Delta x, c_0)\Delta t$. Similarly, corresponding to the Step 2 of the backward error compensation algorithm, we have

$$\breve{\Phi}_j^n = \sum_k \tilde{\Phi}_k^{n+1} \Psi_k(x_j + \mathbf{v}(x_j, t_{n+1})\Delta t), \tag{3.9}$$

which is equivalent to $\breve{\Phi}^n = B\tilde{\Phi}^{n+1}$ where $B = (b_{kj})$, $b_{kj} = \delta_{kj} + s_{kj}$, $|s_{kj}| \le C_3(\Delta x, c_0)\Delta t$. Thus we can also write $B = I + R_2$ with R_2 satisfying $\|R_2\|_{l^\infty} \le C_4(\Delta x, c_0)\Delta t$. Therefore the backward error compensation algorithm eventually yields a matrix form

$$\Phi^{n+1} = A(\frac{3}{2} - \frac{1}{2}BA)\Phi^n = (I + R_3)\Phi^n,$$

where R_3 satisfies $\|R_3\|_{l^\infty} \le C_5(\Delta x, c_0)\Delta t$. Finally we have that

$$\|\Phi^n\|_{l^\infty} \le (1 + C_5\Delta t)^n \|\Phi^0\|_{l^\infty} \le e^{C_5 T}\|\Phi^0\|_{l^\infty}.$$

□

4. Accuracy. We study the accuracy improvement of the backward error compensation algorithm for a general linear scheme for equation (2.1) with constant coefficients. The result generalizes the accuracy improvement theorem in [2] for a linear ordinary differential equation and is based on comparison of the Fourier symbols of the differential equation (2.1) and its correponding numerical scheme, see Lax [9]. Let L, L^*, F be linear schemes defined as in Section 3 and $\rho_L(\xi_h), \rho_{L^*}(\xi_h), \rho_F(\xi_h)$ be their corresponding Fourier symbols respectively. Applying Fourier transform to equation (2.1) we have

$$\hat{\phi}_t = P(i\xi)\hat{\phi},$$

where P is a linear homogeneous polynomial with real coefficients. Therefore we can write

$$\hat{\phi}(\xi, t_n + \Delta t_n) = e^{\Delta t_n P(i\xi)}\hat{\phi}(\xi, t_n).$$

Assume $\Delta x_1 = \cdots = \Delta x_d = h$ and $\Delta t/h$ is fixed during mesh refinement. A scheme $L_1 : \quad \Phi^{n+1} = L_1(\Phi^n)$ is said to be accurate of order r if for any solution ϕ of equation (2.1) having continuous derivatives upto order $r+1$,

$$\phi(x_i, t_{n+1}) - L_1(\phi(\cdot, t_n))|_{x_i} = O(h^{r+1}).$$

We first state the theorem of Lax [9].

THEOREM 4. *Scheme L is accurate of order r if and only if*

$$\rho_L(\xi_h) = e^{\Delta t_n P(i\xi)} + O(|\xi_h|^{r+1}), \text{ as } h \to 0.$$

The "only if" part of the theorem is proved by Lax [9] for more general linear hyperbolic equations with variable coeffients. In constant coefficients' case, Lax's proof also implies the "if" part of this theorem. Using Lax's Theorem 4, we are able to prove the following theorem.

THEOREM 5. *Supose $\rho_{L^*}(\xi_h) = \overline{\rho_L(\xi_h)}$ for any $\xi_h \in [-\pi, \pi]^d$ and the scheme L is accurate of order r for equation (2.1) with constant coefficients and r is an odd positive integer, then scheme F is accurate of order $r+1$.*

Proof. The accuracy of the scheme L implies that

$$\rho_L(\xi_h) = e^{\Delta t_n P(i\xi)} + Q_{r+1}(i\xi_h) + O(|\xi_h|^{r+2}).$$

where Q_{r+1} is a homogeneous polynomial of order $r+1$ with real coefficients. Since $r+1$ is even, we have

$$\rho_{L^*}(\xi_h) = \overline{\rho_L(\xi_h)} = e^{-\Delta t_n P(i\xi)} + Q_{r+1}(i\xi_h) + O(|\xi_h|^{r+2}).$$

Therefore

$$\begin{aligned} \rho_F(\xi_h) &= \rho_L(\xi_h)\{1 + \tfrac{1}{2}[1 - \rho_{L^*}(\xi_h)\rho_L(\xi_h)]\} \\ &= \rho_L(\xi_h)\{1 - \tfrac{1}{2}[e^{-\Delta t_n P(i\xi)} + e^{\Delta t_n P(i\xi)}]Q_{r+1}(i\xi_h) + O(|\xi_h|^{r+2})\} \\ &= [e^{\Delta t_n P(i\xi)} + Q_{r+1}(i\xi_h) + O(|\xi_h|^{r+2})][1 - Q_{r+1}(i\xi_h) + O(|\xi_h|^{r+2})] \\ &= e^{\Delta t_n P(i\xi)} + O(|\xi_h|^{r+2})). \end{aligned} \tag{4.1}$$

The proof is complete. □

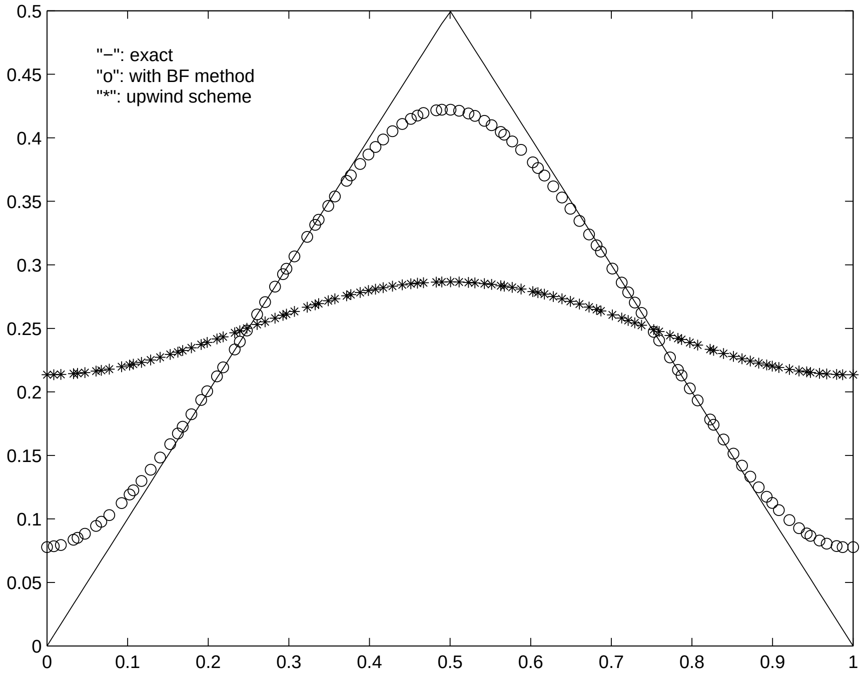

Fig. 4.1. *Translation of a pyramid over* 100 *irregular cells on domain* $[0, 1]$ *with the largest cell size* 4 *times the smallest cell size.* $CFL = 0.5, T = 10$*. Comparison of numerical results of the first order upwind scheme with and without backward error compensation (BF).*

Remark. In case of equation (2.1) with variable coefficients the Fourier-Stieltjes transform (see e.g.[9]) is used to replace the Fourier symbols and a formula similar to (4.1) can be derived for the Fourier-Stieltjes transform of the scheme F.

An interesting phenomenon is that the backward error compensation algorithm seems to improve the numerical result even for very irregular mesh. In the following example we use a first order upwind scheme with and without backward error compensation to compute the linear translation of a pyramid: $u_t + u_x = 0$, $x \in [0, 1]$ with periodic boundary condition. The grid points are distributed as

$$x_i = i * 0.01 + 0.003 * sin[(i - 0.2) * (i + 6.1789) * i], \quad i = 0, 1, \cdots, 99.$$

The solutions at final time $T = 10$ are shown in Fig. 4.1. More future experimental and theoretical efforts are necessary in order to draw a conclusion.

5. Application to Level Set Method. Since the velocity field could create too large gradient in Φ, there is usually an auxiliary equation to solve until the steady state at each time step [22],

$$\frac{\partial \Phi}{\partial \tau} + \text{sign}(\Phi)(| \bigtriangledown \Phi| - 1) = 0. \tag{5.1}$$

This procedure is supposed to transform the Φ into a signed distance function without changing its zero level set. This step also helps clean the error pollution coming from the "skeleton", *i.e.*, the non-smooth area of the level set function. As in [22], equation (5.1) can be written as

$$\tilde{\Phi}_\tau + W \cdot \bigtriangledown\tilde{\Phi} = S(\tilde{\Phi}^0), \tag{5.2}$$

where $W = S(\tilde{\Phi}^0) \bigtriangledown \tilde{\Phi}/| \bigtriangledown \tilde{\Phi}|$ and $S(\tilde{\Phi}^0)$ is the sign function of $\tilde{\Phi}^0$, $S(\tilde{\Phi}^0) = 1$, if $\tilde{\Phi}^0 > 0$; $S(\tilde{\Phi}^0) = -1$ if $\tilde{\Phi}^0 < 0$. $\tilde{\Phi}^0$ is the initial value for (5.2) and is the current level set function obtained by solving equation (2.1).

At each time step we will first compute the equation (2.1) using CIR scheme with backward error compensation to obtain the approximate level set function at time level t_n, say Φ^n. Then let $\tilde{\Phi}^0 = \Phi^n$ and solve equation (5.2) for a few time steps (e.g. m_1 steps). Then replace Φ^n with $\tilde{\Phi}^{m_1}$ and finish the redistancing at this time step. A very simple and effective procedure was proposed in [2] as follows.

At each time step of solving equation (5.2), given $\tilde{\Phi}^m$ at time τ_m, we compute equation (5.2) only at places, say x_i, to obtain $\tilde{\Phi}_i^{m+1}$ where $|\tilde{\Phi}_i^m| > \Delta x$. For other grid nodes, say x_k, simply let $\tilde{\Phi}_k^{m+1} = \tilde{\Phi}_k^m$. This allows us to use a simple low cost first order upwind scheme to discretize equation (5.2) without generating large diffusion and distortion, yet keep an upper bound for the norm of the gradient of $\tilde{\Phi}^{m_1}$ (in 1D it is easy to see that at equilibrium state $|\operatorname{grad}\tilde{\Phi}^{m_1}| \leq 2$ near the interface and $|\operatorname{grad}\tilde{\Phi}^{m_1}| = 1$ elsewhere away from the non-smooth area). For some problems, e.g. an expanding bubble, the level set function Φ^n could become flatter and flatter near the interface. To overcome this problem, we could simply update the $\tilde{\Phi}^m$ using (5.2) only at places, say x_i, to obtain $\tilde{\Phi}_i^{m+1}$ where $|\tilde{\Phi}^m(x_i)| > \Delta x$ or $\tilde{\Phi}^m(x_i)$ is of the same sign with $\tilde{\Phi}^m$ at all neighboring grid nodes x_j of x_i (*i.e.*, x_j and x_i share the same mesh cell). For other grid nodes, say x_k, simply let $\tilde{\Phi}_k^{m+1} = \tilde{\Phi}_k^m$.

In order to discretize equation (5.2), W can be first discretized by the centered finite difference of $\tilde{\Phi}^m$. Now knowing the wind direction W at each grid node, equation (5.2) can then be discretized by the simple first order upwind scheme.

To give the credit where it is due, Russo and Smereka [14] seem to be the first to realize that not changing the values of the level set function at grid nodes adjacent to the interface produces good result in redistancing. In [14], they propose that upwind discretization of equation (5.2) shouldn't go across the interface. So the the value of the level set function at a grid node adjacent to the interface is recomputed instead by its value divided by the norm of the approximated gradient of the level set function at the grid node. In one remark of [14], the approximated gradient will result in the value of the level set function at the grid nodes adjacent to the interface unchanged during the redistancing.

We first conduct a convergence test with and without the redistancing. We compute the rotation of circle for one revolution in a domain $(0, 100) \times (0, 100)$. Initially there is a circle centered at $(50, 75)$ with radius 15. Set a rotational velocity field $(u, v) = (\frac{\pi}{314}(50 - y), \frac{\pi}{314}(x - 50))$. Every point of this circle is supposed to move along the local velocity field. One revolution will be at time $T = 628$. We set the initial level set function Φ to be a signed distance function which is negative inside the circle and positive outside. The max error between the computed and exact level set function Φ's at grid nodes near the interface is shown in Table 5.1. Clearly we have second order convergence for CIR scheme with backward error compensation without redistancing. And this simple redistancing causes the order of convergence to lie between 1 and 2.

In next example we replace the circle with a cutout circle. It is the so called Zalesak's Problem [24] which is one of the difficult test problems for interface tracking methods such as level set method or volume of fluid method, because of their Eulerian

Δx	error without redistancing	order	error with redistancing	order
2	0.623	-	0.454	-
1	0.110	2.50	0.154	1.56
0.5	0.0262	2.07	0.0536	1.52
0.25	0.00638	2.04	0.0208	1.37

Table 5.1

Rotating circle: max error between the computed and exact level set function at grid nodes near the interface, computed using CIR scheme with backward error compensation, CFL=3.

Δx	average distance	order
1	0.138	-
0.5	0.0497	1.47
0.25	0.0211	1.23

Table 5.2

Rotating slotted disk: average distance between the exact interface and the one computed computed using CIR scheme with backward error compensation and redistancing, CFL=3.

representation of the interface. Initially the cutout circle is centered at $(50, 75)$ with radius 15. The slot being cut out has width 5 and length 25. 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 times mesh cell size respectively). In the first test we compute this problem with $N = 100$, $\Delta x = 1$, CFL factor 3. The level set advection equation (2.1) is computed by CIR scheme with backward error compensation and redistancing. In all the following test examples, the redistancing was done for only two time steps with CFL factor 0.25 after each time step of computing equation 2.1. In Fig. 5.1 the computed disk (dash line) is drawn against the exact one (solid line) after one (left figure) and two revolutions (right figure). The result seems to match the resolution computed with the coupled level set and volume-of-fluid method in [21]. In Fig. 5.2, The same tests were done with $N = 200$, $\Delta x = 0.5$. Since the CIR scheme has no restriction for the CFL number, it is ideal for local spatial refinement while keeping the time step unchanged. In Fig. 5.3, we recompute the same problem with $N = 300$, $\Delta x = 1/3$, CFL factor 3. This is almost equivalent to applying a 3 times local spatial refinement near the interface on top of a coarse mesh with $\Delta x = 1$ without changing the time step size. The average distances (defined and computed as in [20]) between the exact and computed interfaces are shown in Table 5.2 for three meshes: 100×100, 200×200 and 400×400. The relative error of the computed disk area A is plotted against time for the three meshes. See Fig. 5.4.

5.1. Inferface Moving with Non-Smooth Velocity.

When the interface has corner points, its unit normal vector field is not continuous at the corner points. A simple way to overcome this problem is to set the backward compensation term to be zero wherever the nonsmoothness in the velocity field is detected. In the following examples we use the following detector. For a given velocity field (u, v) in 2D defined in a uniform grid, if at grid point (x_i, y_j)

$$\begin{aligned}&|u_{i+1,j} - 2u_{i,j} + u_{i-1,j}| \leq \min(|u_{i+1,j} - u_{i,j}|, |u_{i,j} - u_{i-1,j}|) \text{ and}\\&|v_{i,j+1} - 2v_{i,j} + v_{i,j-1}| \leq \min(|v_{i,j+1} - u_{i,j}|, |v_{i,j} - v_{i,j-1}|),\end{aligned} \tag{5.3}$$

then we use the backward error compensation; otherwise we set the backward compensation term to be zero.

We compute the Zalesak's slotted disk as in Fig. 5.1 shrinking under the velocity field $v = -0.2 \bigtriangledown \Phi/| \bigtriangledown \Phi|$. The graphs at different time levels are plotted in Fig. 5.5. When the interface normal velocity depends on its mean curvature, the numerical evaluation of the mean curvature if not treated properly could cause instability of the interface. We use a slope limiter similar to those used in the MUSCL scheme. When evaluating $div(\bigtriangledown\Phi/| \bigtriangledown \Phi|)$, we use center difference to approximate $\mathbf{W} = \bigtriangledown\Phi$ since Φ is Lipschitz continuous. The evaluation of derivatives of $(u, v) = \mathbf{W}/|\mathbf{W}|$ uses the following limiter. Let $D_x u_{i,j}$ be the approximation of $\frac{\partial u}{\partial x}$ at (x_i, y_j) and be defined as follows.

$$D_x u_{i,j} = \begin{cases} \frac{u_{i+1,j}-u_{i-1,j}}{2\Delta x}, & \text{if } (u_{i+1,j} - u_{i,j})(u_{i,j} - u_{i-1,j}) > 0 \\ & \text{and } |u_{i+1,j} - 2u_{i,j} + u_{i-1,j}| \leq \\ & \min(|u_{i+1,j} - u_{i,j}|, |u_{i,j} - u_{i-1,j}|), \\ \text{minmod}(\frac{u_{i+1,j}-u_{i,j}}{\Delta x}, \frac{u_{i,j}-u_{i-1,j}}{\Delta x}), & \text{otherwise,} \end{cases} \tag{5.4}$$

where

$$\text{minmod}(a, b) = \begin{cases} \text{sign}(a)\text{min}(|a|, |b|), & \text{if } \ ab > 0, \\ 0, & \text{otherwise.} \end{cases} \tag{5.5}$$

Our two test examples are volume-preserving mean curvature flows with the velocity field given by $-(k - \bar{k}) \bigtriangledown \Phi/| \bigtriangledown \Phi|$. where k is the mean curvature and $\bar{k}$ is the average mean curvature along the interface approximated as in [12]:

$$\bar{k} = \frac{\int_\Omega k\delta(\Phi)| \bigtriangledown \Phi|dx}{\int_\Omega \delta(\Phi)| \bigtriangledown \Phi|dx}, \tag{5.6}$$

where

$$a(\Phi) = \begin{cases} \frac{1}{2\epsilon}(1 + cos(\frac{\pi\Phi}{\epsilon})), & \text{if } |\Phi| < \epsilon, \\ 0, & otherwise. \end{cases}$$

The mean curvature in (5.6) is approximated by center difference.

The first test example is an unwinding spiral under the volume-preserving mean curvature flow. The computational domain is $\Omega = [0, 100] \times [0, 100]$ and the mesh is 128×128 with $\epsilon = 0.5\Delta x$, $\Delta t = 0.4\Delta x$. The unwinding spiral at different time level are shown in (a), (b), (c) and (d) of Fig. 5.6 the relative volume loss as a function of time is drawn in Fig. 5.7. Through out the computational time interval, the relative area loss is within 0.6%.

The second test example is bubbles merging under the volume-preserving mean curvature flow. Initially in the domain $\Omega = [0, 100] \times [0, 100]$ there are 100 bubbles of radii between 2 and 3. The computational mesh is 128×128 with $\epsilon = 2\Delta x$, $\Delta t = 0.4\Delta x$. The bubbles at different time level are shown in (a), (b) and (c) of Fig. 5.8 and the relative volume loss as a function of time is drawn in (d). Through out the computational time interval, the relative area loss is within 3%.

REFERENCES

[1] R. Courant, E. Isaacson, and M. Rees, *On the solution of nonlinear hyperbolic differential equations by finite differences*, Comm. Pure Appl. Math., 5 (1952), pp. 243–255.

[2] T. F. Dupont and Y. Liu, *Back and forth error compensation and correction methods for removing errors induced by uneven gradients of the level set function*, J. Comput. Phys., 190 (2003), pp. 311–324.

[3] D. Enright, R. Fedkiw, J. Ferziger, and I. Mitchell, *A hybrid particle level set method for improved interface capturing*, J. Comput. Phys., 183 (2002), pp. 83–116.

[4] D. Enright, F. Losasso, and R. Fedkiw, *A fast and accurate semi-lagrangian particle level set method*, submitted, (2003).

[5] M. Falcone and R. Ferretti, *Convergence analysis for a class of high-order semi-lagrangian advection schemes*, SIAM J. Numer. Anal., 35 (1998), pp. 909–940.

[6] J. Glimm, J. Grove, X.-L. Li, K.-M. Shyue, Y. Zeng, and Q. Zhang.

[7] J. Glimm, D. Marchesin, and O. McBryan, *Subgrid resolution of fluid discontinuities. ii.*, J. Comput. Phys., 37 (1980), pp. 336–354.

[8] A. Harten, S. Osher, B. Engquist, and S. Chakravarthy, *Uniformly high order accurate essentially non-oscillatory schemes, III*, J. Comput. Phys., 71 (1987), pp. 231–303.

[9] P. D. Lax, *On the stability of difference approximations to solutions of hyperbolic equations with variable coefficients*, Comm. Pure Appl. Math., 14 (1961), pp. 497–520.

[10] R. W. MacCormack, AIAA Paper 69-354, (1969).

[11] S. Osher and J. Sethian, *Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi equations*, J. Comput. Phys, 79 (1988), pp. 12–49.

[12] D. Peng, B. Merriman, S. Osher, H.-K. Zhao, and M. Kang, *A pde-based fast local level set method*, J. Comput. Phys, 155 (1999), pp. 410–438.

[13] W. Rider and D. Kothe, *A marker particle method for interface tracking*, Proceedings of the Sixth International Symposium on Computational Fluid Dynamics, 163 (1995), p. 976.

[14] G. Russo and P. Smereka, *A remark on computing distance functions*, J. Comput. Phys, 163 (2000), pp. 51–67.

[15] C.-W. Shu, *Total-variation-diminishing time discretizations*, SIAM J. Sci. Stat. Comput., 9 (1988), pp. 1073–1084.

[16] C.-W. Shu and S. Osher, *Efficient implementation of essentially non-oscillatory shock-capturing schemes*, J. Comput. Phys, 77 (1988), pp. 439–471.

[17] J. Strain, *Semi-lagrangian methods for level set equations*, J. Comput. Phys, 151 (1999), pp. 498–533.

[18] ———, *A fast modular semi-lagrangian method for moving interfaces*, J. Comput. Phys, 161 (2000), pp. 512–536.

[19] ———, *A fast semi-lagrangian contouring method for moving interfaces*, J. Comput. Phys, 170 (2001), pp. 373–394.

[20] 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 (1999), pp. 1165–1191.

[21] M. Sussman and E. G. Puckett, *A coupled level set and volume-of-fluid method for computing 3d and axisymmetric incompressible two-phase flows.*, J. Comput. Phys, 162 (2000), pp. 301–337.

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

[23] G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, , N. Al-Rawahi, W. Tauber, J. Han, S. Nas, and Y.-J. Jan, *A front-tracking method for the computations of multiphase flow*, J. Comput. Phys, 169 (2001), pp. 708–759.

[24] S. T. Zalesak, *Fully multidimensional flux-corrected transport*, J. Comput. Phys., 31 (1979), pp. 335–362.

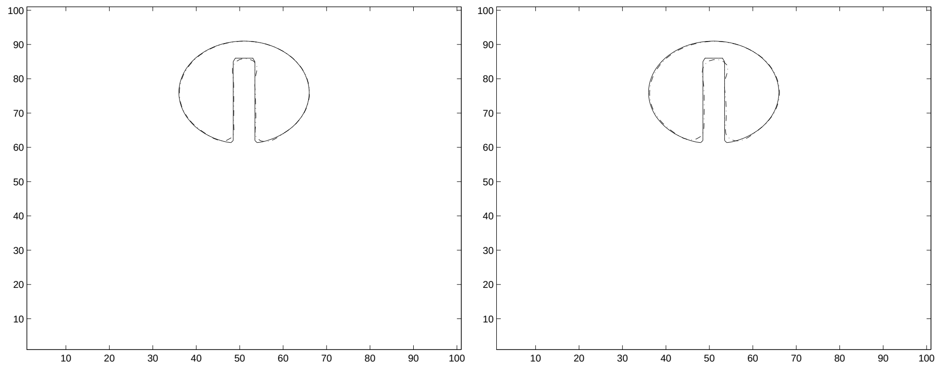

Fig. 5.1. *Zalesak's problem. Comparison of a notched disk that has been rotated one (left) and two revolutions (right). Level set equation is computed using CIR scheme with backward error compensation and redistancing, CFL=3,* 100×100 *(*$\Delta x = 1$*).*

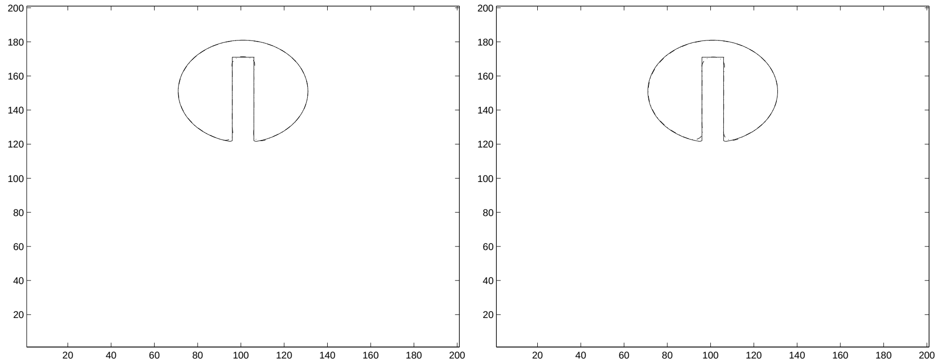

Fig. 5.2. *Zalesak's problem. Comparison of a notched disk that has been rotated one (left) and two revolutions (right). Level set equation is computed using CIR scheme with backward error compensation and redistancing,* $CFL = 3$*,* 200×200 *(*$\Delta x = 0.5$*).*

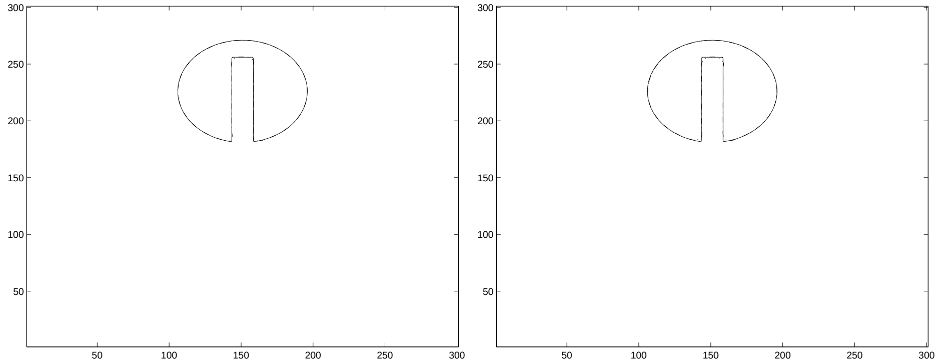

Fig. 5.3. *Zalesak's problem. Comparison of a notched disk that has been rotated one (left) and two revolutions (right).Level set equation is computed using CIR scheme with backward error compensation and redistancing,* $CFL = 3$*,* 300×300 *(*$\Delta x = 1/3$*).*

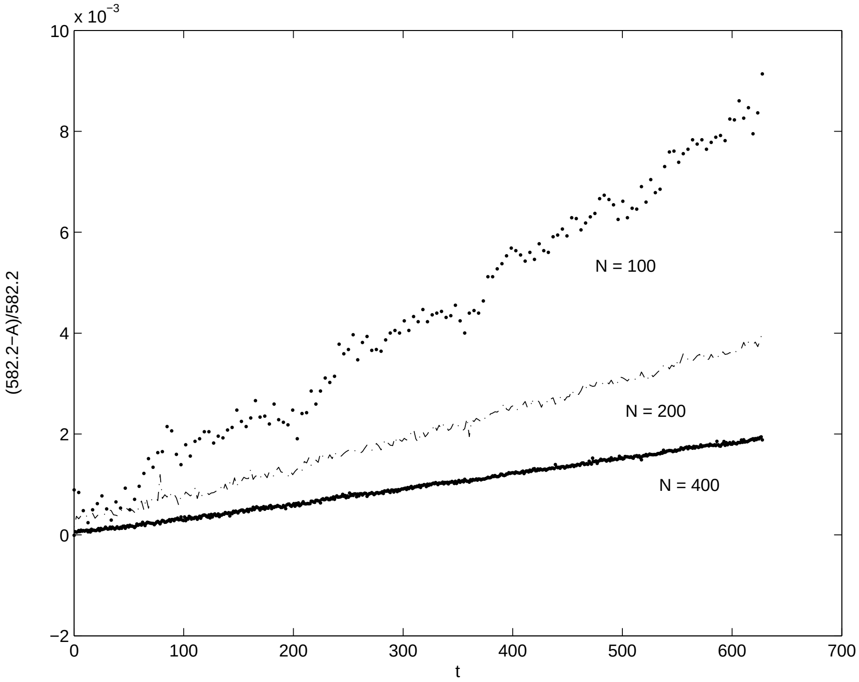

Fig. 5.4. *Zalesak's problem. Relative area loss of the notched disk as a function of time. Level set equation is computed using CIR scheme with backward error compensation and redistancing,* $CFL = 3$.

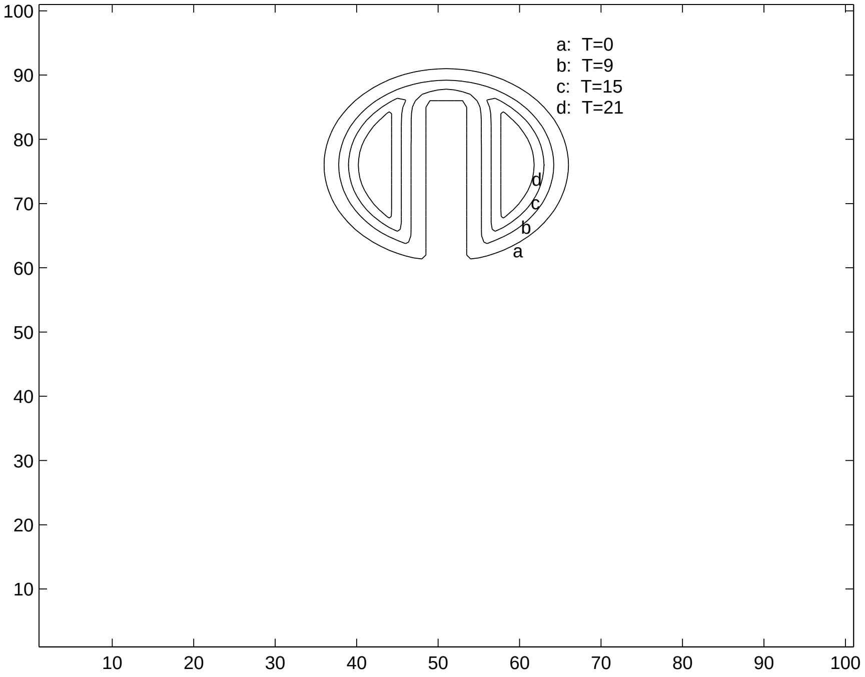

Fig. 5.5. *Shrinking slotted disk,* 100×100, $\Delta t = 0.4\Delta x$.

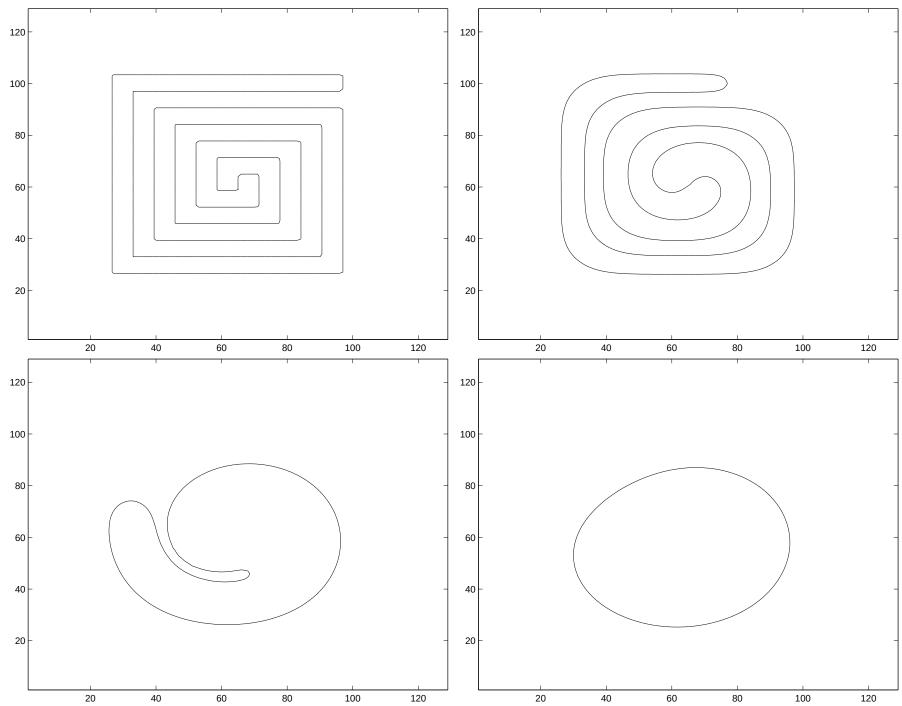

Fig. 5.6. *Area conserving interface movement,* 128×128*,* $\Delta t = 0.4\Delta x$*. (a)*$T = 0$*, upper left; (b)*$T = 30$*, upper right; (c)*$T = 160$*, lower left; (d)*$T = 300$*, lower right.*

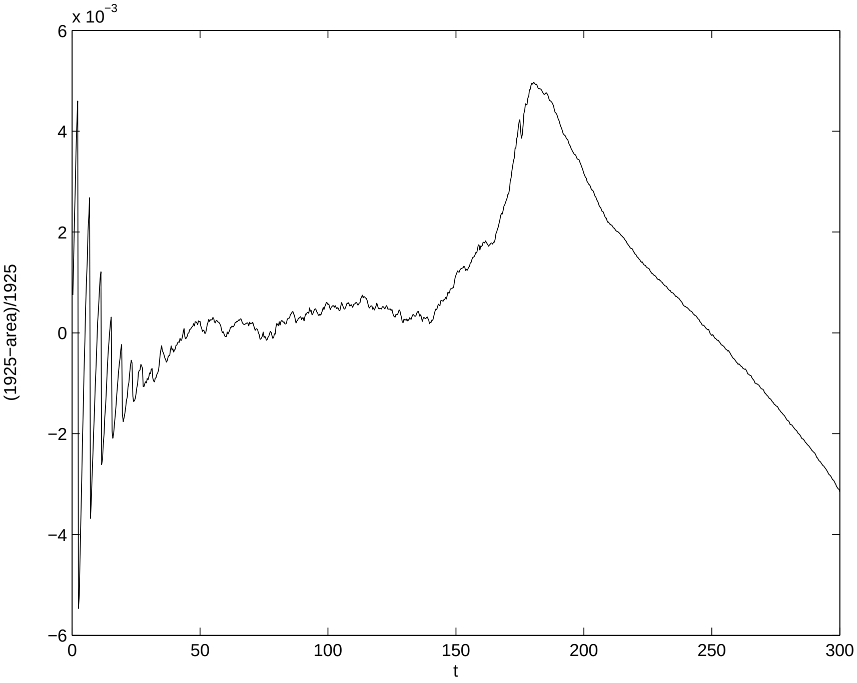

Fig. 5.7. *Unwinding spiral: relative area loss as a function of time.*

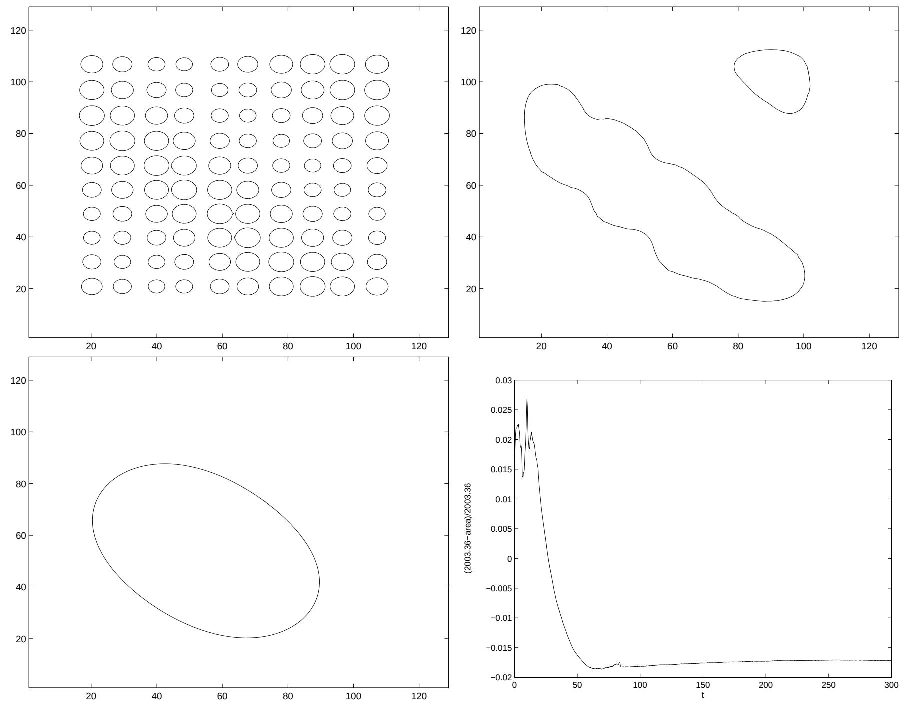

Fig. 5.8. *Area conserving interface movement,* 128×128, $\Delta t = 0.4\Delta x$. *(a)*$T = 0$, *upper left; (b)*$T = 20$, *upper right; (c)*$T = 300$, *lower left; (d)relative area loss as a function of time, lower right.*