

## A NEW LEVEL SET METHOD FOR MOTION IN NORMAL DIRECTION BASED ON A SEMI-IMPLICIT FORWARD-BACKWARD DIFFUSION APPROACH\*

KAROL MIKULA<sup>†</sup> AND MARIO OHLBERGER<sup>‡</sup>

**Abstract.** We introduce a new level set method for motion in normal direction. It is based on a formulation in the form of a second order forward-backward diffusion equation. The equation is discretized by the finite volume method. We propose a semi-implicit time discretization taking into account the forward diffusion part of the solution in an implicit way, while the backward diffusion part is treated explicitly. When forward diffusion dominates, a straightforward reconstruction of the solution is used, while larger (smoothing) stencils are used when backward diffusion dominates. The method is precise on coarse grids and is second order accurate for smooth solutions. Numerical experiments show an optimal coupling of time and space steps with  $\tau = h$ , and no stronger CFL condition is required. Numerical tests with the scheme are discussed on representative examples.

**Key words.** level set method, finite volume method, forward-backward diffusion

**AMS subject classifications.** 35K20, 76M12, 76R50, 68U10

**DOI.** 10.1137/09075946X

**1. Introduction.** In this article we propose a new level set method for motion in normal direction. The standard level set techniques are based on finite difference discretization of the first order Hamilton–Jacobi equation,

$$(1.1) \quad u_t + F|\nabla u| = 0,$$

where  $F$  is a speed function. We refer to [15, 16, 14] and references therein for more details on this classical approach. For simplicity and clarity of the presentation, we will assume throughout the paper that  $F$  is a constant.

Recently, fully explicit finite volume schemes were suggested in [9, 10] for an advective formulation of (1.1)

$$(1.2) \quad u_t + F \frac{\nabla u}{|\nabla u|} \cdot \nabla u = 0$$

which was written in the equivalent form

$$(1.3) \quad u_t + \nabla \cdot (\bar{v}u) - u \nabla \cdot \bar{v} = 0$$

with  $\bar{v} = F \frac{\nabla u}{|\nabla u|}$  representing a velocity by which the level sets of the solution are driven. In [9], a first order upwind scheme with recursive flux redistribution was introduced, while in [10] a high resolution version has been developed. The first order scheme has theoretically no restriction on time step (cf. [8, 9]). In the high resolution scheme the first order polynomial reconstruction of the solution inside every finite

---

\*Received by the editors May 18, 2009; accepted for publication (in revised form) March 24, 2010; published electronically May 28, 2010. This work was partially supported by grants APVV-RPEU-0004-06, APVV-0351-07, APVV-LPP-0020-07, and VEGA 1/0269/09.

<http://www.siam.org/journals/sisc/32-3/75946.html>

<sup>†</sup>Department of Mathematics, Faculty of Civil Engineering, Slovak University of Technology, Radlinského 11, 81368 Bratislava, Slovakia (karol.mikula@stuba.sk).

<sup>‡</sup>Institut für Numerische und Angewandte Mathematik, Universität Münster, Einsteinstr. 62, D-48149 Münster, Germany (mario.ohlberger@uni-muenster.de).

volume brought the second order accuracy for smooth solutions. The high resolution method is computationally more complex than the first order upwind approximation and has a natural CFL restriction on the time step related to the local Courant numbers (cf. [9, 10]).

The new level set method developed in this paper is based on the observation that if we plug  $\bar{v} = F \frac{\nabla u}{|\nabla u|}$  directly into (1.3), we get a *second order partial differential equation* for the unknown level set function  $u$ , namely,

$$(1.4) \quad u_t + \nabla \cdot \left( Fu \frac{\nabla u}{|\nabla u|} \right) - u \nabla \cdot \left( F \frac{\nabla u}{|\nabla u|} \right) = 0.$$

In the first spatial differential term, the diffusion coefficient depends on  $u$  and thus results in a nonlinearly weighted mean curvature flow term [3, 2]. The second spatial differential term is in nondivergence form, where the solution  $u$  is multiplied by the (weighted) mean curvature of its level sets.

In order to discuss (1.4), let us consider the product  $Fu$ , of a constant  $F$  and the solution  $u$ . If it is negative, the first spatial differential term represents a forward diffusion and the second one a backward diffusion. On the other hand, if it is positive, the situation is vice versa; the first term represents a backward diffusion and the second one a forward diffusion. Thus, if we are interested in motion of the zero level set representing an interface (closed curve), starting from, e.g., a signed distance function, there is always this forward-backward diffusion (FBD) coupling included in (1.4), and the difference of these two terms gives an overall forward or backward diffusion character of (1.4) in a point. A semi-implicit finite volume discretization of this FBD formulation is the basis of our new numerical method.

The rest of the paper is organized as follows. In section 2 we motivate our new level set method and give a basic version of the scheme in Definition 2.1. In section 3 an alternative variant of the method is proposed that takes into account a more evolved reconstruction technique. In section 4 several numerical experiments are given that analyze the convergence and efficiency of the new method. Thereby we also compare our semi-implicit approach with the fully explicit finite volume flux-based level set method (FBLSM) from [10]. In section 5, we finally address that the presented approach for motion in normal direction can be embedded into a more general framework of inflow-implicit/outflow-explicit finite volume methods for variable velocity advection equations.

**2. The new level set FBD scheme.** Let us consider (1.4) in a bounded polygonal domain  $\Omega \subset \mathbb{R}^d$ ,  $d = 2, 3$ , and time interval  $[0, T]$ . Let  $\mathcal{Q}_h$  denote a primal polygonal partition of  $\Omega$ . Let  $p$  be a finite volume of a corresponding dual Voronoi tessellation  $\mathcal{T}_h$  with measure  $m_p$ , and let  $e_{pq}$  be an edge between  $p$  and  $q$ ,  $q \in N(p)$ , where  $N(p)$  is a set of neighboring finite volumes (i.e.,  $\bar{p} \cap \bar{q}$  has nonzero  $(d-1)$ -dimensional measure). Let  $c_{pq}$  be the length of  $e_{pq}$  and  $n_{pq}$  be the unit outer normal vector to  $e_{pq}$  with respect to  $p$ . We shall consider  $\mathcal{T}_h$  to be an admissible mesh in the sense of [6], i.e., there exists a representative point  $x_p$  in the interior of every finite volume  $p$  such that the joining line between  $x_p$  and  $x_q$ ,  $q \in N(p)$ , is orthogonal to  $e_{pq}$ . We denote by  $x_{pq}$  the intersection of this line segment with the edge  $e_{pq}$ . The length of this line segment is denoted by  $d_{pq}$ , i.e.,  $d_{pq} := |x_q - x_p|$ . As we have built  $\mathcal{T}_h$  based on the primal mesh  $\mathcal{Q}_h$ , we assume that the points  $x_p$  coincide with the vertices of  $\mathcal{Q}_h$ . Let us denote by  $u_p$  a (constant) value of the solution in a finite volume  $p$  computed by the scheme. We shall use also  $\bar{u}_p$ , a reconstructed (but also constant) value of the solution in  $p$  (e.g., given as an average of neighboring finite volume val-

ues), and  $\bar{u}_{pq}$ , a reconstructed (but constant) value of the solution assigned to the edge  $e_{pq}$ .

To derive and motivate our scheme, let us approximate  $u$  in the second spatial term of (1.4) by an appropriate constant function  $\bar{u}_p$  on  $p$ , integrate the spatial differential terms of (1.4) over  $p$ , and use the divergence theorem. We get

$$\begin{aligned} & \int_p \nabla \cdot \left( F u \frac{\nabla u}{|\nabla u|} \right) dx - \int_p u \nabla \cdot \left( F \frac{\nabla u}{|\nabla u|} \right) dx \\ &= \int_{\partial p} F \frac{u}{|\nabla u|} \nabla u \cdot n ds - \bar{u}_p \int_{\partial p} F \frac{1}{|\nabla u|} \nabla u \cdot n ds \\ &= \sum_{q \in N(p)} \int_{e_{pq}} F \frac{u}{|\nabla u|} \frac{\partial u}{\partial n_{pq}} ds - \bar{u}_p \sum_{q \in N(p)} \int_{e_{pq}} F \frac{1}{|\nabla u|} \frac{\partial u}{\partial n_{pq}} ds. \end{aligned}$$

Let us denote the absolute value of the reconstructed gradient on an edge  $e_{pq}$  (given, e.g., by the so-called diamond-cell strategy specified later) by  $|\nabla u_{pq}|$ , and let the normal derivative on this edge be approximated by  $(u_q - u_p)/d_{pq}$ . Then as an approximation of the spatial differential terms, we get the following expression:

$$(2.1) \quad \sum_{q \in N(p)} \frac{c_{pq}}{d_{pq}} \frac{F(\bar{u}_p - \bar{u}_{pq})}{|\nabla u_{pq}|} (u_p - u_q).$$

Now let us think about a matrix representation of a numerical scheme that is based on such a space discretization. If the term  $F(\bar{u}_p - \bar{u}_{pq})$  is positive, it gives a “forward diffusion contribution” to the system matrix, which means that the coefficient on the diagonal is positive while off-diagonal coefficients are negative. Thus, in this situation, we get the M-matrix property which guarantees the solvability and  $L_\infty$ -stability of a linear system represented by the matrix. On the other hand, if  $F(\bar{u}_p - \bar{u}_{pq})$  is negative, it gives opposite signs to the matrix coefficients. In our understanding, it represents a “backward diffusion contribution” which destroys the favorable matrix solvability and stability properties. Taking into account these observations, we define the diffusion coefficient on the edge  $e_{pq}$  as

$$(2.2) \quad a_{pq} = \frac{F(\bar{u}_p - \bar{u}_{pq})}{|\nabla u_{pq}|}.$$

The coefficient  $a_{pq}$  contains forward and backward diffusion contributions if  $\bar{u}_p$  and  $\bar{u}_{pq}$  have equal signs. Forward diffusion is dominant if  $a_{pq}$  is positive, while backward diffusion is dominant if  $a_{pq}$  is negative. Hence, we define dominant forward and backward diffusion parts, respectively, as

$$(2.3) \quad a_{pq}^f = \max(a_{pq}, 0), \quad a_{pq}^b = \min(a_{pq}, 0).$$

In the next step we approximate  $u_t$  in (1.4) by the time difference  $\frac{u_p^n - u_p^{n-1}}{\tau}$ , where  $\tau$  is a uniform time step, and integrate the whole (1.4) in every finite volume  $p$  according to the space discretization given in (2.1). Then, due to the above-mentioned contributions to the matrix properties, we take the forward diffusion contribution implicitly and the backward diffusion contribution explicitly. This results in the following linear system at the  $n$ th discrete time step:

$$(2.4) \quad m_p u_p^n + \tau \sum_{q \in N(p)} \frac{c_{pq}}{d_{pq}} a_{pq}^f (u_p^n - u_q^n) = m_p u_p^{n-1} + \tau \sum_{q \in N(p)} \frac{c_{pq}}{d_{pq}} a_{pq}^b (u_q^{n-1} - u_p^{n-1}).$$

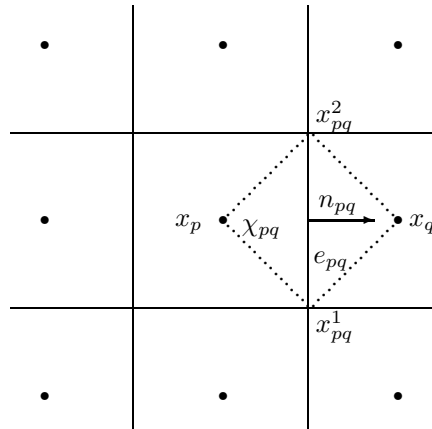


FIG. 1. The diamond-cell covolume  $\chi_{pq}$  associated to the edge  $e_{pq}$ .

For a uniform squared grid in two dimensions (2D) with a finite volume side width  $h$ , the dual Voronoi mesh  $\mathcal{T}_h$  is again a uniform squared grid with side width  $h$ , shifted by  $(h/2, h/2)^\top$ . For such grids we obtain  $m_p = h^2$ ,  $\frac{c_{pq}}{a_{pq}} = 1$ , and in three dimensions (3D) on uniform Cartesian meshes analogously  $m_p = h^3$ ,  $\frac{c_{pq}}{a_{pq}} = h$ . Thus, in these cases we end up with the simple system

$$(2.5) \quad u_p^n + \frac{\tau}{h^2} \sum_{q \in N(p)} a_{pq}^f (u_p^n - u_q^n) = u_p^{n-1} + \frac{\tau}{h^2} \sum_{q \in N(p)} a_{pq}^b (u_q^{n-1} - u_p^{n-1}).$$

This system should be accompanied by suitable initial and boundary conditions in order to get unique solvability. In our convergence test examples (see section 4), we use the distance function for the level set representing an initial curve as initial data. At the boundaries we may, for example, prescribe exact Dirichlet data: for  $u_q$  on the left-hand side of (2.4) at the new time level and for  $u_q$  on the right-hand side at the old time level. It is clear that at any edge  $e_{pq}$ , only one of the terms  $a_{pq}^f$  or  $a_{pq}^b$  appears. Thus, the boundary values are needed either on the left- or right-hand side of (2.4). In experiments where the exact solution on the boundary is not known, we use boundary conditions which extrapolate the solution at the boundary of  $\Omega$ , using the slope of an approximate normal derivative from the interior cells.

The system (2.4) can be considered as a general formulation of our semi-implicit level set FBD method on any admissible dual Voronoi finite volume grid. The next important question is how the values  $\bar{u}_p$ ,  $\bar{u}_{pq}$ , and  $|\nabla u_{pq}|$  in the *diffusion* coefficients  $a_{pq}$  are computed. First of all, the term  $|\nabla u_{pq}|$  is given by using the so-called diamond-cell strategy. For simplicity, let us consider a uniform grid in two space dimensions as depicted in Figure 1. If we denote by  $u_{pq}^1$  and  $u_{pq}^2$  approximate values of the solution in the end points  $x_{pq}^1$  and  $x_{pq}^2$  of the edge  $e_{pq}$ , we may define

$$(2.6) \quad |\nabla u_{pq}^{n-1}| := \left( \left( \frac{u_q^{n-1} - u_p^{n-1}}{h} \right)^2 + \left( \frac{u_{pq}^{2,n-1} - u_{pq}^{1,n-1}}{h} \right)^2 \right)^{\frac{1}{2}}.$$

In this case,  $u_{pq}^{1,n-1}$  and  $u_{pq}^{2,n-1}$  are given by the average of the values at time  $t^{n-1}$  in the four adjacent finite volumes for which the points  $x_{pq}^1$ , respectively,  $x_{pq}^2$ , are common (cf. [4]). For more general meshes and the 3D case, we refer to [1, 7, 12, 5].

An important point in our method is the fact that the further reconstructed values  $\bar{u}_p$  and  $\bar{u}_{pq}$  in the numerator of (2.2) are computed differently, depending on the forward-backward character of the diffusion in a finite volume  $p$ . The strategy is as follows. In a first step we simply take

$$(2.7) \quad \bar{u}_p^{n-1} = u_p^{n-1} \quad \text{and} \quad \bar{u}_{pq}^{n-1} = \frac{1}{2}(u_p^{n-1} + u_q^{n-1})$$

for all  $p$  and  $q \in N(p)$ , and we compute our first estimate of  $a_{pq}^f$  and  $a_{pq}^b$  by (2.2). Then, for every finite volume  $p$ , we compute the “strength” of forward and backward diffusion by summing all corresponding contributions and define

$$(2.8) \quad D_p^f = \sum_{q \in N(p)} a_{pq}^f \geq 0 \quad \text{and} \quad D_p^b = \sum_{q \in N(p)} a_{pq}^b \leq 0.$$

If the forward diffusion dominates in the finite volume  $p$ , i.e.,  $D_p^f \geq -D_p^b$ , then we use the already computed reconstructions (2.7). On the other hand, if the backward diffusion dominates, i.e.,  $D_p^f < -D_p^b$ , we replace  $\bar{u}_p$  and  $\bar{u}_{pq}$  by a larger spatial average. For example, on Cartesian meshes in 2D, we define

$$(2.9) \quad \text{and} \quad \begin{aligned} \bar{u}_{pq}^{n-1} &:= \frac{1}{4} \sum_{q \in N(p)} (u_p^{n-1} + u_q^{n-1} + u_{pq}^{1,n-1} + u_{pq}^{2,n-1}) \\ \bar{u}_p^{n-1} &:= \frac{1}{4} \sum_{q \in N(p)} \bar{u}_{pq}^{n-1}. \end{aligned}$$

This can be considered as a stencil for a spatial smoothing of the solution. By that approach we get a meaningful (stable) approximate solution. It is known that a backward diffusion process can be solved for a short time uniquely only starting from smooth initial data (cf. [11]). So our switch between (2.7) and (2.9) is also motivated by this fact. Finally, for such backward diffusion dominated finite volumes  $p$ , we recompute (2.2) and get the final coefficients of the linear system (2.4) or (2.5). The resulting linear systems can then be solved by an appropriate linear solver. In the numerical experiments in section 4, we use the Gauss–Seidel iterative method.

We summarize our new forward-backward diffusion level set scheme in the following definition.

**DEFINITION 2.1** (the general level set FBD scheme). *Let a sufficiently smooth ( $C^0$ ) interface curve  $\Gamma_l \subset \Omega$  be given, and let  $u_0 \in C^0(\Omega)$  denote an initial level set function with  $\{u_0 = 0\} = \Gamma_l$  (e.g., given by the signed distance function to  $\Gamma_l$  in  $\Omega$ ). In addition, let us assume that generalized boundary data  $u_B \in C^0(\partial\Omega \times [0, T] \times \mathbb{R})$  are given for the level set problem such that the following compatibility condition is satisfied:*

$$(2.10) \quad u_B(x, 0, u_0(x)) = u_0(x) \quad \text{for all } x \in \partial\Omega.$$

*In the case of standard Dirichlet boundary data  $u_D \in C^0(\partial\Omega \times [0, T])$ , we have  $u_B(x, t, v) = u_D(x, t)$  for any  $v \in \mathbb{R}$ .*

*Then the general level set FBD scheme is given as follows.*

**Initial data:** *For  $n = 0$ , define the piecewise constant approximation  $u_h^0$  through*

$$(2.11) \quad u_h^0|_p(x) := u_p^0 := \pi_p(u_0) \quad \text{for all } x \in p, p \in \mathcal{T}_h,$$

*where  $\pi_p : C^0(p) \rightarrow \mathbb{P}_0(p)$  is a suitable local projection to a constant.*

**Time step  $(n - 1) \rightarrow n$ :** For  $n > 0$ , we define  $u_h^n$  through  $u_p^n, p \in \mathcal{T}_h$  as follows:

- (a) **Implicit definition of boundary values at time  $t^n$ :** For all  $x_p \in \partial\Omega$ , we set

$$(2.12) \quad u_p^n := u_B(x_p, t^n, E_p^B(u_h^n)).$$

Here  $E_p^B$  denotes a discrete operator that defines an extrapolation of the interior discrete solution to the boundary point  $x_p$ . Note that this definition is explicit in the case of standard Dirichlet data.

- (b) **Definition of the interior values at time  $t^n$ :**

- (i) For all interior cells  $p \in \mathcal{T}_h, q \in N(p)$ , we compute

$$\bar{u}_p^{n-1} := R_p(u_h^{n-1}), \quad \bar{u}_{pq}^{n-1} := R_{pq}(u_h^{n-1}), \quad |\nabla u_{pq}^{n-1}| := |DR_{pq}(u_h^{n-1})|,$$

where  $R_p, R_{pq}$ , and  $DR_{pq}$  denote suitable constant local reconstructions of the solution or the gradient of the solution, respectively.

- (ii) Define the preliminary diffusion coefficients as

$$(2.13) \quad a_{pq}^{n-1} := \frac{F(\bar{u}_p^{n-1} - \bar{u}_{pq}^{n-1})}{|\nabla u_{pq}^{n-1}|},$$

$$(2.14) \quad a_{pq}^{f,n-1} = \max(a_{pq}^{n-1}, 0), \quad a_{pq}^{b,n-1} = \min(a_{pq}^{n-1}, 0).$$

- (iii) For all interior cells  $p$ , compute the forward and backward diffusion contributions as

$$D_p^{f,n-1} := \sum_{q \in N(p)} a_{pq}^{f,n-1} \geq 0 \quad \text{and} \quad D_p^{b,n-1} = \sum_{q \in N(p)} a_{pq}^{b,n-1} \leq 0.$$

If, for a cell  $p$ , backward diffusion dominates, i.e.,  $-D_p^{b,n-1} > D_p^{f,n-1}$ , recompute  $\bar{u}_p, \bar{u}_{pq}, q \in N(p)$  by

$$\bar{u}_p^{n-1} := R_p^b(u_h^{n-1}), \quad \bar{u}_{pq}^{n-1} := R_{pq}^b(u_h^{n-1}),$$

where  $R_p^b$  and  $R_{pq}^b$  are spatial reconstructions for the backward diffusion dominated case that use a larger spatial averaging than  $R_p$  and  $R_{pq}$ , respectively. In addition, for those cells recompute  $a_{pq}^{f,n-1}, a_{pq}^{b,n-1}$  according to (2.13) and (2.14).

- (iv) For all  $x_p \in \Omega \setminus \partial\Omega$ , we define  $u_p^n$  as the solution of the following linear system:

$$\begin{aligned} m_p u_p^n + \tau \sum_{q \in N(p)} \frac{c_{pq}}{d_{pq}} a_{pq}^f (u_p^n - u_q^n) \\ = m_p u_p^{n-1} + \tau \sum_{q \in N(p)} \frac{c_{pq}}{d_{pq}} a_{pq}^b (u_q^{n-1} - u_p^{n-1}). \end{aligned}$$

Note that in the case of generalized boundary conditions, the boundary reconstruction is taken into account explicitly on backward diffusion boundaries and implicitly on forward diffusion boundaries.

- (c) **Definition of  $u_h^n$ :**

We define the piecewise constant approximation  $u_h^n$  as

$$u_h^n|_p(x) := u_p^n \quad \text{for all } x \in p, p \in \mathcal{T}_h.$$

In the general definition of the new level set FBD scheme, we assumed that the suitable local reconstruction operators  $R_p, R_{pq}, DR_{pq}$ , and  $R_p^b, R_{pq}^b$  are given. In the following definition we summarize the definition of those reconstruction operators in the case of Cartesian meshes in two space dimensions.

DEFINITION 2.2 (a specific FBD scheme on Cartesian meshes in 2D). *For Cartesian meshes in 2D, the basic FBD scheme is given by Definition 2.1 with the following reconstruction operators (cf. the motivation above):*

$$\begin{aligned} R_p(u_h) &:= u_p, & R_{pq}(u_h) &:= \frac{1}{2}(u_p + u_q), \\ DR_{pq}(u_h) &:= \frac{u_q - u_p}{h}n_{pq} + \frac{u_{pq}^2 - u_{pq}^1}{h}t_{pq}, \\ R_{pq}^b(u_h) &:= \frac{1}{4} \sum_{q \in N(p)} (u_p + u_q + u_{pq}^1 + u_{pq}^2), \\ R_p^b(u_h) &:= \frac{1}{4} \sum_{q \in N(p)} R_{pq}^b(u_h). \end{aligned}$$

Here  $n_{pq}$  denotes the outer normal to  $e_{pq}$  with respect to  $p$ , and  $t_{pq} := (x_{pq}^2 - x_{pq}^1)/h$  is a tangential vector to  $e_{pq}$ .

Remark 2.3 (choice of reconstructions). The suitability of the specific reconstructions in Definition 2.2 is documented by the presented numerical experiments. In the next subsection, we give also a more advanced reconstruction for uniform Cartesian meshes in two space dimensions. We will compare the performance of both reconstruction schemes in section 4. However, we note that other reconstruction stencils and gradient approximations are possible, and we leave it to the reader and further studies to test and compare further approaches.

Remark 2.4 (time discretization). Our choice of the semi-implicit time discretization is motivated by the good properties of the resulting linear system matrix. As a result, the system can be solved efficiently using only a few Gauss–Seidel iterations, usually less than 15 for very fine grids and about 5 for coarse grids with stopping criterion  $\|R_k\|_2^2 \leq TOL\|R_0\|_2^2$ ,  $TOL = 10^{-12}$ , where  $R_k$  and  $R_0$  are current and initial residuum, respectively. One could also use nonlinear Gauss–Seidel iterations, i.e., to update the system coefficients inside the iterative loop. We did not find this choice more precise, and it is clearly more CPU time consuming. Another possibility is to take the reconstructions (2.7) and (2.9) and then to consider just a fully explicit scheme with the treatment of both backward and forward diffusion contributions from the previous time step. In such an approach, the time step is restricted by  $\tau \leq \frac{h}{4}$  (in 2D), and the optimal coupling  $\tau = h$  (which turns out in the semi-implicit scheme) is lost. Since the number of Gauss–Seidel iterations does not double when increasing the time step by a factor of two (cf. Table 3), we again prefer the semi-implicit scheme as suggested in (2.4). It is worth noting that coupling  $\tau = h$  (or even weaker) which can be used in our semi-implicit approach is not allowed in any standard explicit scheme for solving this type of problem.

### 3. The FBD2 scheme with larger stencil upwind-type reconstructions.

In the case of a backward diffusion dominated finite volume  $p$ , we can replace the simple reconstructions  $R_{pq}^b$  and  $R_p^b$  from Definition 2.2 with more advanced ones. The basic idea is that inside every finite volume  $p$ , we reconstruct linearly our solution using an averaged gradient. Then, on every edge, we define values  $\bar{u}_{pq}^{n-1}$  through an evaluation of these linear reconstructions in the midpoints (or centers of gravity) of

the edge  $e_{pq}$ . Thereby the evaluation is taken from the reconstruction in  $p$  if the corresponding edge diffusion is backward, i.e.,  $a_{pq}^b \leq 0$ , while the evaluation is done from the reconstruction in the neighboring cell  $q$  if the edge diffusion is forward, i.e.,  $a_{pq}^f > 0$ . Thus, first we compute averaged gradients

$$(3.1) \quad \nabla u_p^{n-1} = \frac{1}{|N(p)|} \sum_{q \in N(p)} DR_{pq}(u_h^{n-1}),$$

where  $|N(p)|$  denotes the number of neighboring cells, and the gradient reconstructions on the edges  $DR_{pq}(u_h^{n-1})$  are computed by the diamond-cell strategy described in section 2. Then if  $a_{pq}^b \leq 0$ , we define

$$(3.2) \quad \bar{u}_{pq}^{n-1} = u_p^{n-1} + (x_{pq} - x_p) \cdot \nabla u_p^{n-1},$$

and if  $a_{pq}^f > 0$ , we define

$$(3.3) \quad \bar{u}_{pq}^{n-1} = u_q^{n-1} + (x_{pq} - x_q) \cdot \nabla u_q^{n-1}.$$

Finally, we set

$$(3.4) \quad \bar{u}_p^{n-1} = \frac{1}{|N(p)|} \sum_{q \in N(p)} \bar{u}_{pq}^{n-1}.$$

On Cartesian meshes in two space dimensions, this reconstruction procedure gives rise to the following definition of our FBD2 scheme.

**DEFINITION 3.1** (the FBD2 scheme with upwind-type reconstruction in 2D). *For Cartesian meshes in 2D, the FBD2 scheme is given by Definition 2.1 with the following reconstruction operators:*

$$\begin{aligned} R_p(u_h) &:= u_p, & R_{pq}(u_h) &:= \frac{1}{2}(u_p + u_q), \\ DR_{pq}(u_h) &:= \frac{u_q - u_p}{h} n_{pq} + \frac{u_{pq}^2 - u_{pq}^1}{h} t_{pq}, \\ DR_p(u_h) &:= \frac{1}{4} \sum_{q \in N(p)} DR_{pq}(u_h), \\ R_{pq}^b(u_h) &:= \begin{cases} u_p^{n-1} + \frac{h}{2} DR_p(u_h) \cdot n_{pq} & \text{if } a_{pq}^b \leq 0, \\ u_q^{n-1} - \frac{h}{2} DR_q(u_h) \cdot n_{pq} & \text{otherwise,} \end{cases} \\ R_p^b(u_h) &:= \frac{1}{4} \sum_{q \in N(p)} R_{pq}^b(u_h). \end{aligned}$$

*Remark 3.2* (comparison of FBD and FBD2). Note that the reconstructions in the FBD2 method are the same as those in the simple FBD method from Definition 2.2 for finite volumes  $p$ , where forward diffusion is dominating. Only in the case of backward diffusion dominated cells do we switch to the larger stencil upwind-type reconstructions.

**4. Numerical experiments.** In this section we discuss computational results obtained by our new level set FBD methods. The results are compared with the recently suggested FBLSM, [10], which is also based on a finite volume discretization.

In [10], FBLSM has been carefully compared to other known approaches, like second order ENO schemes. Numerical examples were given that showed a superior behavior of FBLSM, especially concerning conservation of area and its precision on coarse grids. The comparison with FBLSM can thus be seen as a fair comparison of our new semi-implicit approach with fully explicit finite volume schemes based on upwind fluxes. Here we show that for solutions with shrinking characteristics (like a shrinking circle or square), the new FBD scheme given in section 2 gives even more precise results than FBLSM and that it is further improved by using the FBD2 scheme from section 3. For solutions with expanding characteristics (like an expanding circle or square), we found the FBD2 scheme more precise than FBLSM.

In the subsections 4.1 to 4.4, we present four examples (shrinking and expanding circles and squares) where we analyze the numerical convergence properties of the schemes. Finally, in subsection 4.5, we give an evolution of a nontrivial closed curve where topological changes appear. For the numerical experiments, we present figures which compare visually numerical and exact solutions, mainly on coarse grids. The figures show 3D graphs of exact and numerical level set functions, a comparison of their level lines, and a zoom of exact and numerically computed zero level lines representing the moving curve. The plots display different behaviors of the analyzed schemes, and it turns out that the new FBD level set methods are superior, especially on coarse grids.

In the tables, we present errors and experimental orders of convergence (EOC) of the numerical schemes together with other characteristics when refining the grid. The error is given as a difference between the exact and numerical zero level line representing the moving curve in a discrete  $L_2(I, L_2(S^1))$ -norm, where  $I = [0, T]$  is a time interval ( $T = 0.4$  in our experiments) and  $S^1$  denotes the unit circle. The norm is computed as follows. The evolving curves in all our test cases are radially symmetric, centered in the origin, and can be parameterized by the angle  $\varphi \in [0, 2\pi]$  (or correspondingly by the arc-length parameterization of  $S^1$ ). For every  $\varphi$  and  $t$ , a distance  $r(\varphi, t)$  in radial direction from the origin is given for each such exact solution. In every discrete time step  $n = 0, \dots, M$ , we find all zero crossing points  $p_i^n, i = 1, \dots, K$  of the piecewise bilinear representation of the numerical solution given on the primal grid  $\mathcal{Q}_h$ . Then we compute the distances from the origin  $r_i^n$  for all  $p_i^n, i = 1, \dots, K$  and compare them with the radial distance  $r(\varphi_i^n, n\tau)$  of the exact solution for the corresponding angle  $\varphi_i^n := \varphi(p_i^n)$ . The final formula for computing the error then looks like

$$\| \|u - u_h\| \|_h := \left( \sum_{n=0}^M \tau \frac{1}{K} \sum_{i=1}^K (r_i^n - r(\varphi_i^n, n\tau))^2 \right)^{\frac{1}{2}}.$$

In all our computations and tables,  $N$  denotes the number of finite volumes in the primal grid  $\mathcal{Q}_h$  in the  $x$  and  $y$  directions, and our domain is  $\Omega = [-1, 1] \times [-1, 1]$ , i.e., we have  $h = 2/N$ . Since our grid  $\mathcal{T}_h$  is squared, we use scheme (2.5) with the reconstructions given in Definitions 2.2 and 3.1, respectively. In order to have a fair comparison, we compute gradients and normal derivatives in the FBLSM scheme (cf. [10]) by the same approach as in the FBD methods, i.e., by using the diamond-cell strategy outlined in section 2. Since in both methods there are absolute values of gradients in the denominators, we use the so-called Evans–Spruck regularization  $|\nabla u| \approx \sqrt{\varepsilon^2 + |\nabla u|^2}$  in order to prevent division by zero. In experiments with shrinking characteristics we use  $\varepsilon = 10^{-3}$ , and in experiments with expanding characteristics (where flat regions are formed) we use  $\varepsilon = 10^{-6}$ . Since we start the computations always by a function which is negative inside the moving curve and positive outside, we solve (1.4) with  $F = -1$  in the case of shrinking and  $F = 1$  in the case of expansion. In our

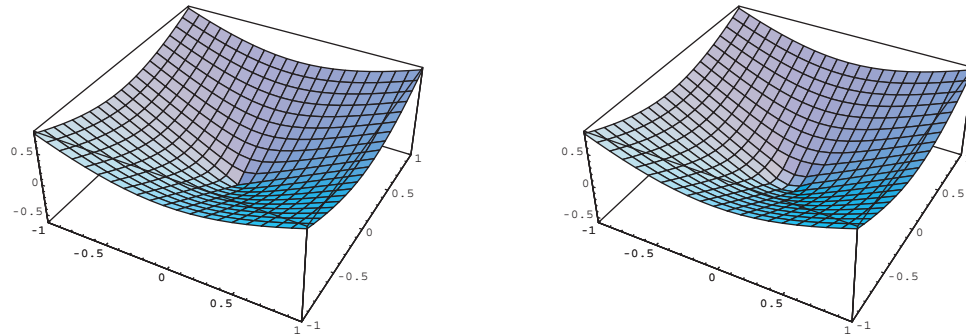


FIG. 2. *Shrinking circle: 3D graphs of exact (left) and numerical FBD solutions (right) for  $N = 20$ .*

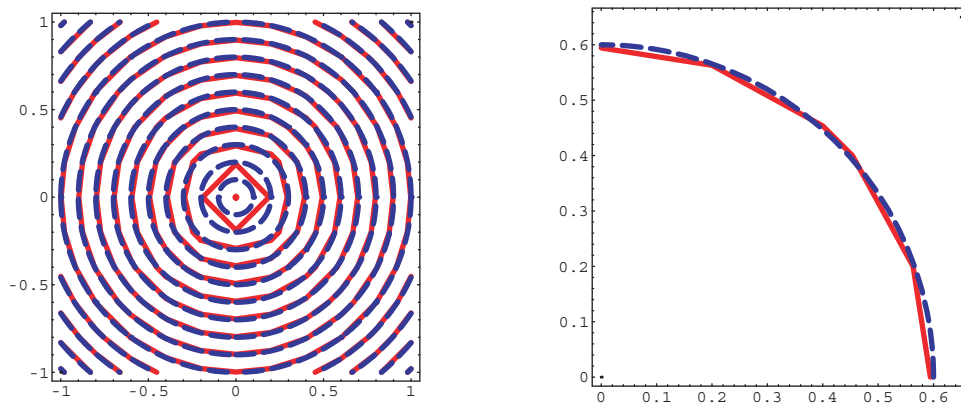


FIG. 3. *Shrinking circle: isolines of the exact (dashed) and by FBD method numerically computed solution (solid) for  $N = 10$  (left), and comparison of the exact (dashed) and by FBD method numerically computed (solid) zero level line representing a moving circle for  $N = 10$  (right).*

convergence test examples (subsections 4.1 to 4.4), we use as an initial condition the signed distance function to the initial curve, and we prescribe exact Dirichlet data at the boundaries. In the numerical experiments of subsection 4.5, the initial level set function is not a signed distance, and, since the exact solution is not known, we use boundary conditions which extrapolate the solution using the slope of an approximate normal derivative from the interior cells. At backward diffusion boundaries of domain  $\Omega$ , this extrapolation is taken explicitly; at forward diffusion boundaries, it is included in the implicit part of the system. All computations were done on a standard 2.2 GHz notebook, and we report the computing times (CPU) in seconds in the tables.

**4.1. Shrinking circle.** In this example, we discuss the shrinking of a circle with initial radius  $r(0) = 1$  in the time interval  $I = [0, T]$ ,  $T = 0.4$ . The exact solution is then given by  $r(t) = r(0) - t$ , i.e., the exact radius at time  $T$  is  $r(0.4) = 0.6$ . The exact level set function is given by  $u(x, y, t) = \sqrt{x^2 + y^2} - 1 + t$ . In Figure 2 we present exact and numerically computed level set functions at time  $T$ . We see just a little smoothing of the numerical solution computed by the FBD method in the bottom corner point. It does not, however, destroy the precision of the FBD scheme, even on very coarse grids. This is documented in Figure 3 and in Table 1. In this case the

TABLE 1  
*Shrinking circle: report on FBD error.*

$N$	$h = \tau$	NTS	FBD error	EOC	nGSi	CPU
10	0.2	2	$2.013 \cdot 10^{-3}$		11	
20	0.1	4	$3.843 \cdot 10^{-4}$	2.39	13	0.01
40	0.05	8	$1.001 \cdot 10^{-4}$	1.94	14	0.05
80	0.025	16	$2.505 \cdot 10^{-5}$	2.00	15	0.33
160	0.0125	32	$6.319 \cdot 10^{-6}$	1.98	15	1.55
320	0.00625	64	$1.573 \cdot 10^{-6}$	2.00	15	8.38

TABLE 2  
*Shrinking circle: report on FBLSM error.*

$N$	$h = 2\tau$	NTS	FBLSM error	EOC	CPU
10	0.2	4	$1.544 \cdot 10^{-3}$		0.
20	0.1	8	$4.374 \cdot 10^{-4}$	1.82	0.01
40	0.05	16	$1.074 \cdot 10^{-4}$	2.02	0.05
80	0.025	32	$2.684 \cdot 10^{-5}$	2.00	0.39
160	0.0125	64	$6.743 \cdot 10^{-6}$	1.99	1.91
320	0.00625	128	$1.645 \cdot 10^{-6}$	2.04	10.33

TABLE 3  
*Shrinking circle: report on FBD error for  $N = 40$  ( $h = 0.05$ ) and different time step sizes.*

$N$	$\tau$	NTS	FBD error	nGSi	CPU
40	0.4	1	$1.7845 \cdot 10^{-4}$	40	0.02
40	0.2	2	$1.3692 \cdot 10^{-4}$	34	0.03
40	0.1	4	$1.1082 \cdot 10^{-4}$	22	0.04
40	0.05	8	$1.0016 \cdot 10^{-4}$	14	0.05
40	0.025	16	$1.0133 \cdot 10^{-4}$	10	0.08
40	0.0125	32	$1.0424 \cdot 10^{-4}$	7	0.14
40	0.00625	64	$1.0207 \cdot 10^{-4}$	6	0.24

FBD and FBLSM methods give approximately the same error, and they are second order accurate (cf. Tables 1 and 2). The FBD scheme has no stability restriction on the time steps (NTS denotes the number of time steps in all tables), while for FBLSM we have to use  $\tau \leq h/2$ . When enlarging or decreasing the time step in the FBD method, the results are almost the same (which is also a consequence of the fact that we start from a distance function), and the error is smallest for the coupling  $\tau = h$  (cf. Table 3). Thus, we use such a coupling in all further computed examples because it is a good compromise between precision and computing time, as can be seen from the table. In Tables 1 and 2, we report CPU times for refined grids and the number of Gauss–Seidel iterations (nGSi) until we reach the stopping criterion given in Remark 2.2. Due to good matrix properties, we need only few iterations, also on fine grids. In the experiment with  $n = 320$ , an average CPU time for one time step of the method was 0.12 seconds; in our current implementation, 0.06 seconds were spent in the construction of the scheme, and 0.06 seconds were spent in solving the linear system. FBD is, in this example, slightly better than FBLSM concerning both precision and CPU times. In all further examples the number of Gauss–Seidel iterations and the corresponding CPU times for the FBD scheme are practically the

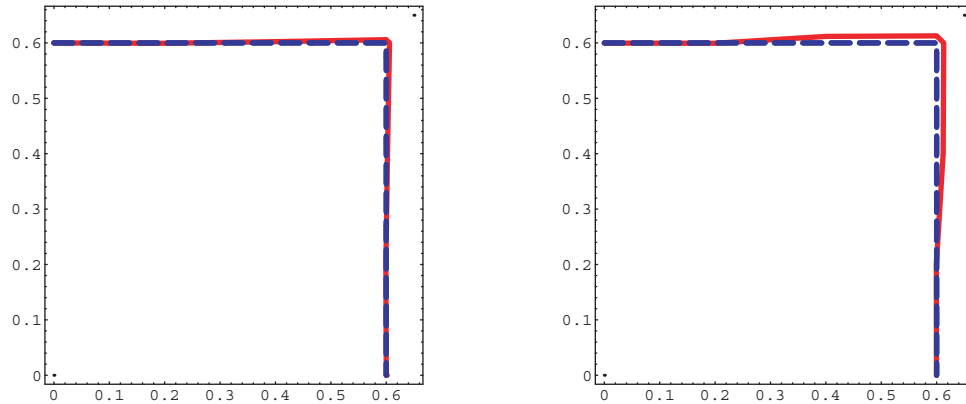


FIG. 4. Shrinking square: comparison of the exact (dashed) and numerically computed (solid) zero level line representing a shrinking square for FBD (left) and FBLSM (right) for  $N = 10$ .

TABLE 4  
Shrinking square: report on FBD and FBD2 errors.

$n$	$h = \tau$	NTS	FBD error	EOC	FBD2 error	EOC	FBD2 CPU
10	0.2	2	$1.530 \cdot 10^{-3}$		$1.542 \cdot 10^{-3}$		0.
20	0.1	4	$6.022 \cdot 10^{-4}$	1.34	$6.208 \cdot 10^{-4}$	1.32	0.01
40	0.05	8	$3.067 \cdot 10^{-4}$	0.97	$2.829 \cdot 10^{-4}$	1.13	0.05
80	0.025	16	$1.323 \cdot 10^{-4}$	1.21	$1.158 \cdot 10^{-4}$	1.29	0.37
160	0.0125	32	$5.550 \cdot 10^{-5}$	1.25	$4.502 \cdot 10^{-5}$	1.36	1.96
320	0.00625	64	$2.276 \cdot 10^{-5}$	1.28	$1.645 \cdot 10^{-5}$	1.45	9.82

same as for this example, which is also true for FBLSM. Thus, we report CPU times for these methods only in this first subsection. Let us also note that, in this example of a shrinking smooth solution, the basic reconstruction strategy from Definition 2.2 is sufficient and that the error of FBD2 is comparable.

**4.2. Shrinking square.** In this example we start by the initial curve given as a zero level line of the level set function  $u_0(x, y) = \max(|x|, |y|) - 1$  and evolve it until time  $T = 0.4$ . The exact solution is given by  $u(x, y, t) = \max(|x|, |y|) - 1 + t$ . In Figure 4 (left) we see the FBD resulting zero level line for  $N = 10$ , which is already on such coarse grids, visually almost undistinguishable from the exact ones. It is worth noting that by the FBD method we reduced significantly underestimating of the numerical solution in the corner points of the singular lines which is a known artefact of the FBLSM method (cf. Figure 4 (right)). Also the precision of the resolution of the bottom corner point is higher for FBD (cf. Table 6). All these facts clearly show advantages of FBD, resulting in lower interface errors (cf. fourth column of Tables 4 and 5), especially on coarse grids. The precision of FBD is additionally improved by using the nontrivial reconstruction given in section 3, i.e., by using the FBD2 method. In particular we hint at a comparison on finer grids (cf. sixth column of Table 4). By comparing the seventh column of Table 1 and the eighth column of Table 4, we see that the more advanced reconstruction does not bring any significant increase in CPU time while the precision is enhanced.

TABLE 5  
*Shrinking square: report on FBLSM error.*

$n$	$h = 2\tau$	NTS	FBLSM error	EOC
10	0.2	4	$5.508 \cdot 10^{-3}$	
20	0.1	8	$1.848 \cdot 10^{-3}$	1.58
40	0.05	16	$6.575 \cdot 10^{-4}$	1.49
80	0.025	32	$2.349 \cdot 10^{-4}$	1.48
160	0.0125	64	$8.343 \cdot 10^{-5}$	1.49
320	0.00625	128	$2.954 \cdot 10^{-5}$	1.50

TABLE 6  
*Shrinking square: position of the minimum in the numerical solution for the shrinking square example. The exact minimum is at  $-0.6$ .*

$n$	$h$	FBLSM	FBD	FBD2
10	0.2	-0.5122	-0.5349	-0.5224
20	0.1	-0.5504	-0.5740	-0.5737
40	0.05	-0.5774	-0.5853	-0.5890
80	0.025	-0.5891	-0.5921	-0.5952
160	0.0125	-0.5943	-0.5959	-0.5976
320	0.00625	-0.5967	-0.5978	-0.5988

TABLE 7  
*Expanding circle: report on FBD and FBD2 errors.*

$n$	$h = \tau$	NTS	FBD error	EOC	FBD2 error	EOC
10	0.2	2	$1.209 \cdot 10^{-2}$		$2.655 \cdot 10^{-3}$	
20	0.1	4	$2.331 \cdot 10^{-3}$	2.37	$4.286 \cdot 10^{-4}$	2.63
40	0.05	8	$4.237 \cdot 10^{-4}$	2.46	$9.187 \cdot 10^{-5}$	2.22
80	0.025	16	$9.517 \cdot 10^{-5}$	2.15	$2.619 \cdot 10^{-5}$	1.81
160	0.0125	32	$2.328 \cdot 10^{-5}$	2.03	$6.794 \cdot 10^{-6}$	1.94
320	0.00625	64	$5.802 \cdot 10^{-6}$	2.00	$1.764 \cdot 10^{-6}$	1.95

TABLE 8  
*Expanding circle: report on FBLSM errors.*

$n$	$h = 2\tau$	NTS	FBLSM error	EOC
10	0.2	4	$3.561 \cdot 10^{-3}$	
20	0.1	8	$5.783 \cdot 10^{-4}$	2.62
40	0.05	16	$1.516 \cdot 10^{-4}$	1.93
80	0.025	32	$3.867 \cdot 10^{-5}$	1.97
160	0.0125	64	$9.681 \cdot 10^{-6}$	2.00
320	0.00625	128	$2.423 \cdot 10^{-6}$	2.00

**4.3. Expanding circle.** In the case of curve expansion, we first look at an initial circle with radius  $r(0) = 0.4$  given as the zero level line of the level set function  $u_0(x, y) = \sqrt{x^2 + y^2} - 0.4$ . The exact solution is then given by  $u(x, y, t) = \max(\sqrt{x^2 + y^2} - 0.4 - t, 0.4)$ . The interface errors are given in Tables 7 and 8, and a visualization of solution isolines on coarse grids for FBD, FBLSM, and FBD2 is given in Figure 5. We see that the methods are all second order and that the smallest errors are realized by the FBD2 scheme.

**4.4. Expanding square.** This is the most difficult example, where an initial square, given as a zero level line of  $u_0(x, y) = \max(|x|, |y|) - 0.4$ , is expanding by unit

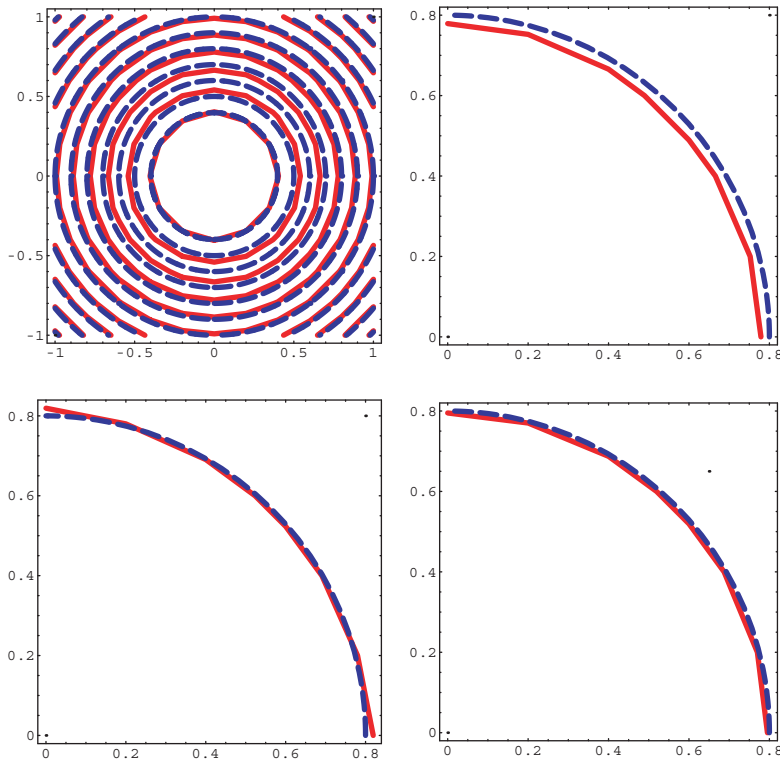


FIG. 5. Expanding circle: isolines of the exact (dashed) and by FBD numerically computed (solid) solutions for  $N = 10$  (upper left), and comparison of the exact (dashed) and numerically computed (solid) zero level line representing an expanding circle for FBD (upper right), FBLSM (lower left), and FBD2 (lower right) for  $N = 10$ .

speed until time  $T = 0.4$ . In Figure 6 we present a comparison of level lines of exact and numerically computed solutions by FBD2 and FBLSM and also details of exact and numerically computed zero level lines. Comparing the errors in Tables 9 and 10, we again see the highest precision for the FBD2 level set method.

**4.5. Shrinking and expanding nonconvex curve.** At the end of this section, we present two nontrivial evolutions: shrinking and expanding quatrefoil. The numerical results are computed by the FBD2 level set method with  $\tau = 0.0125$ ,  $N = 160$ , i.e.,  $h = \tau$ . The initial curve is given as the zero level line of the function

$$(4.1) \quad u^0(x_1, x_2) = L + \frac{\sqrt{x_1^2 + x_2^2}}{r}, \quad r = 0.6 + 0.4 \sin\left(4\arctan\left(\frac{x_2}{x_1}\right)\right)$$

with  $L = -1$  in the case of shrinking characteristics and  $L = -0.4$  in the case of expansion. The graph and level lines of  $u^0$  in the first case are plotted in Figure 7. In Figure 8 we plot the evolution of the shrinking quatrefoil. One can see the splitting of the leaves between time points 0.1 and 0.15. In Figure 9 several states of the expanding quatrefoil are shown.

**5. Conclusion and outlook.** In this article we introduced a general approach for the construction of a new level set method for motion in normal direction, based

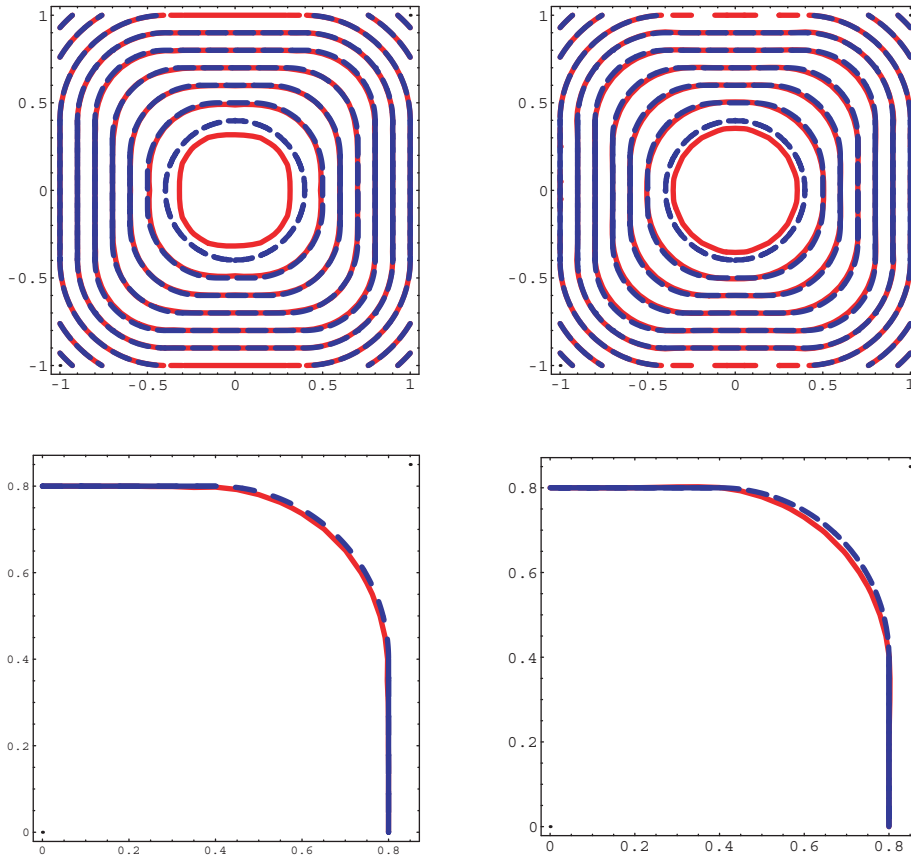


FIG. 6. *Expanding square: Left: isolines of exact (dashed) and numerically computed by FBD2 (solid) solutions; below is detail of the exact and numerical zero level line. Right: Isolines of exact (dashed) and numerically computed by FBLSM (solid) solutions; below is detail of the exact and numerical zero level line. In both computations,  $N = 40$ .*

TABLE 9  
*Expanding square: report on FBD and FBD2 errors.*

$n$	$h = \tau$	NTS	FBD error	EOC	FBD2 error	EOC
10	0.2	2	$2.192 \cdot 10^{-2}$		$7.321 \cdot 10^{-3}$	
20	0.1	4	$1.078 \cdot 10^{-2}$	1.02	$4.045 \cdot 10^{-3}$	0.86
40	0.05	8	$5.600 \cdot 10^{-3}$	0.94	$2.237 \cdot 10^{-3}$	0.85
80	0.025	16	$2.957 \cdot 10^{-3}$	0.92	$1.268 \cdot 10^{-3}$	0.82
160	0.0125	32	$1.591 \cdot 10^{-3}$	0.89	$7.453 \cdot 10^{-4}$	0.77
320	0.00625	64	$8.756 \cdot 10^{-4}$	0.86	$4.612 \cdot 10^{-4}$	0.69

on a semi-implicit finite volume approximation of a forward-backward diffusion formulation. The time discretization treats the forward diffusion part implicitly, while the backward diffusion part is integrated explicitly. By construction, the resulting linear system matrices are M-matrices, and thus no stability problems were observed in numerical experiments. In particular, we presented and analyzed two new schemes on Cartesian meshes in two space dimensions (FBD and FBD2) that were obtained by

TABLE 10  
Expanding square: report on FBLSM errors.

$n$	$h = 2\tau$	NTS	FBLSM error	EOC
10	0.2	4	$2.013 \cdot 10^{-2}$	
20	0.1	8	$8.154 \cdot 10^{-3}$	1.30
40	0.05	16	$3.817 \cdot 10^{-3}$	1.09
80	0.025	32	$1.946 \cdot 10^{-3}$	0.96
160	0.0125	64	$1.025 \cdot 10^{-3}$	0.94
320	0.00625	128	$5.659 \cdot 10^{-4}$	0.86

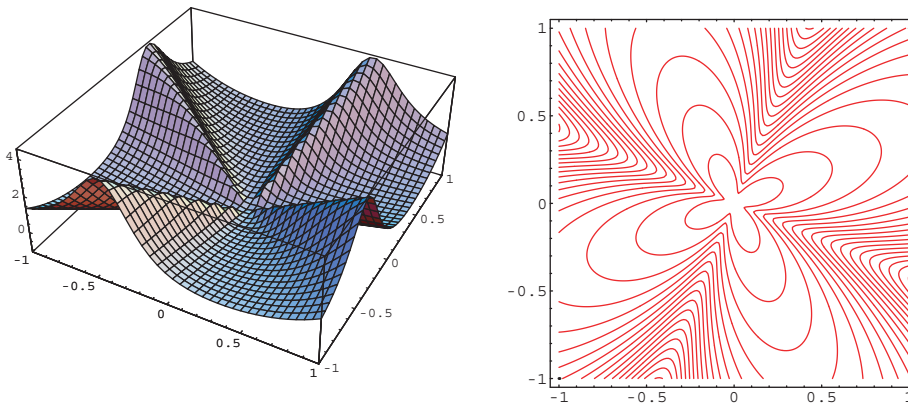


FIG. 7. Initial level set function for evolving quatrefoils and its level lines.

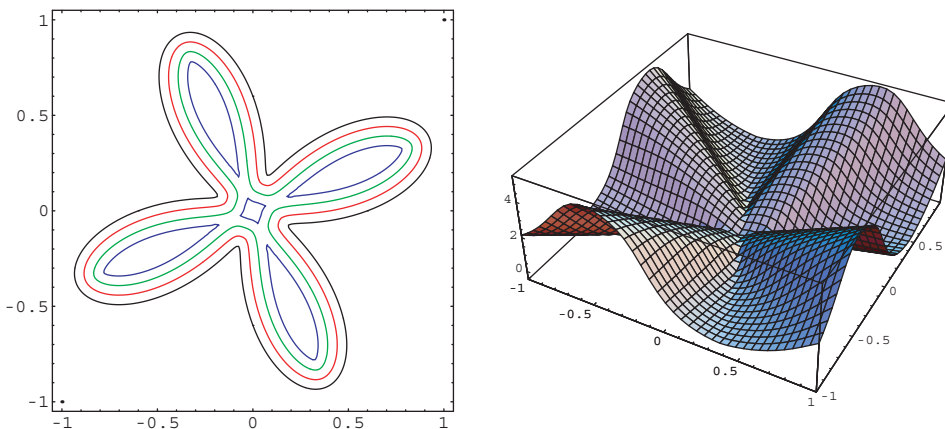


FIG. 8. Left: Shrinkage of the initial quatrefoil with topological changes appearing plotted at times 0, 0.05, 0.1, and 0.15 (inner to outer contour) computed by FBD2 on the grid with  $N = 160$ . Right: Graph of the level set function at time 0.15.

particular choices of reconstruction operators. The numerical examples in section 4 revealed that both methods are of second order for smooth solutions. While both methods behave in a similar way for shrinking characteristics, the more advanced FBD2 method is much more accurate for expanding characteristics. A comparison of both methods with the recently introduced FBLSM, [10] shows that, in particular, FBD2 performs better than FBLSM, with respect to both accuracy and CPU times.

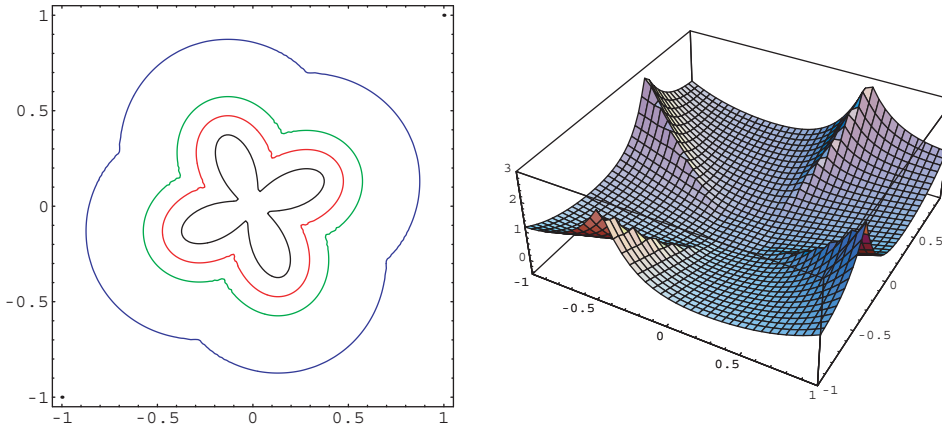


FIG. 9. *Left: Expanding of the initial quatrefoil with four corners evolving plotted at times 0, 0.1, 0.2, and 0.5 (inner to outer contour) computed by FBD2 on the grid with  $N = 160$ . Right: Graph of the level set function at time 0.2 showing resolution of singular lines corresponding to corners.*

We want to point out that the presented methodology of deriving the semi-implicit FBD schemes can be generalized to advection equations of the form

$$u_t + \bar{v} \cdot \nabla u = f.$$

In the case of motion in normal direction, we have the special case  $\bar{v} = F \frac{\nabla u}{|\nabla u|}$ , but more general situations are possible. This approach leads to a new class of discretization schemes that can be seen as treating inflows to a cell  $p$  implicitly and outflows explicitly. In a forthcoming paper [13], we will derive the resulting more general class of inflow-implicit/outflow-explicit schemes and analyze its interesting stability and accuracy properties in solving variable velocity advection equations.

**Acknowledgment.** The work on this paper started during visits of the first author at the University of Münster in July and October 2008, supported by a guest professorship of the Westfälische Wilhelms-Universität Münster.

#### REFERENCES

- [1] Y. COUDIÉRE AND P. VILLEDIEU, *Convergence rate of a finite volume scheme for a two dimensional convection-diffusion problem*, ESAIM: M2AN, 33 (2000), pp. 493–516.
- [2] S. CORSARO, K. MIKULA, A. SARTI, AND F. SGALLARI, *Semi-implicit covolume method in 3D image segmentation*, SIAM J. Sci. Comput., 28 (2006), pp. 2248–2265.
- [3] K. DECKELNICK AND G. DZIUK, *A fully discrete numerical scheme for weighted mean curvature flow*, Numer. Math., 91 (2002), pp. 423–452.
- [4] O. DRBLIKOVÁ AND K. MIKULA, *Convergence analysis of finite volume scheme for nonlinear tensor anisotropic diffusion in image processing*, SIAM J. Numer. Anal., 46 (2007), pp. 37–60.
- [5] O. DRBLIKOVÁ AND K. MIKULA, *Semi-implicit diamond-cell finite volume scheme for 3D nonlinear tensor diffusion in coherence enhancing image filtering*, Finite Volumes for Complex Applications V: Problems and Perspectives, R. Eymard and J. M. Herard, eds., ISTE and Wiley, London, 2008, pp. 343–350.
- [6] R. EYMARD, T. GALLOUET, AND R. HERBIN, *Finite volume methods*, Handbook of Numerical Analysis, Vol. VII: Solution of Equations in  $R$ , North-Holland, Amsterdam, 2000, pp. 713–1020.
- [7] C. ERATH, *Adaptive Finite Volume Methode*, Diploma thesis, Vienna University of Technology, Vienna, Austria, 2005 (in German).

- [8] P. FROLKOVIČ, *Flux-based method of characteristics for contaminant transport in flowing groundwater*, *Comput. Vis. Sci.*, 5 (2002), pp. 73–83.
- [9] P. FROLKOVIČ AND K. MIKULA, *Flux-based level set method: A finite volume method for evolving interfaces*, *Appl. Numer. Math.*, 57 (2007), pp. 436–454.
- [10] P. FROLKOVIČ AND K. MIKULA, *High-resolution flux-based level set method*, *SIAM J. Sci. Comput.*, 29 (2007), pp. 579–597.
- [11] B. KAWOHL AND N. KUTEV, *Maximum and comparison principle for one-dimensional anisotropic diffusion*, *Math. Ann.*, 311 (1998), pp. 107–123.
- [12] Z. KRIVÁ AND K. MIKULA, *Adaptive diamond cell finite volume method in image processing*, in *Proceedings of ALGORITMY 2009—18th Conference on Scientific Computing*, Vysoké Tatry, Podbanske, Slovakia, Slovak University of Technology, 2009, pp. 121–133.
- [13] K. MIKULA AND M. OHLBERGER, *A new class of implicit-inflow/explicit-outflow schemes for solving variable velocity advection equations*, to appear.
- [14] S. OSHER AND R. FEDKIW, *Level Set Methods and Dynamic Implicit Surfaces*, *Appl. Math. Sci.* 153, Springer, Berlin, 2000.
- [15] S. OSHER AND J. SETHIAN, *Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton–Jacobi formulations*, *J. Comput. Phys.*, 79 (1988), pp. 12–49.
- [16] J. A. SETHIAN, *Level Set Methods and Fast Marching Methods, Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Material Science*, Cambridge University Press, New York, 1999.