Semi-implicit Second Order Accurate Finite Volume Method for Advection-Diffusion Level Set Equation

Martin Balažovjech, Peter Frolkovič, Richard Frolkovič and Karol Mikula

Abstract We present a second order accurate finite volume method for level set equation describing the motion in normal direction with the speed depending on external properties and curvature. A convenient combination of a Crank-Nicolson type of the time discretization for diffusion term [1] and an Inflow Implicit and Outflow Explicit scheme [6] for advection term is used. Numerical experiments for an example with the exact solution derived in this paper and for examples motivated by modeling of fire propagation in forests are presented.

1 Introduction

Although not in a divergence form, the level set equations are often solved with finite volume methods [3–5, 8]. The basic idea behind such approaches is to rewrite the level set equation in such a way that it can be approximated using integration by parts. In this paper we apply such approach with an aim to suggest a second order accurate finite volume method to solve level set equations that describe the motion in normal direction with the speed depending on external properties and on curvature.

In the level set equation one can recognize two terms that have a character of advection and diffusion, respectively. In [6, 7] a novel second order accurate

M. Balažovjech · P. Frolkovič (✉) · R. Frolkovič · K. Mikula
Department of Mathematics, Slovak University of Technology, Radlinského 11,
813 68 Bratislava, Slovak Republic
e-mail: peter.frolkovic@stuba.sk

M. Balažovjech
e-mail: balazovjech@stuba.sk

K. Mikula
e-mail: karol.mikula@stuba.sk

J. Fuhrmann et al. (eds.), *Finite Volumes for Complex Applications VII - Elliptic, Parabolic and Hyperbolic Problems*, Springer Proceedings in Mathematics & Statistics 78,
DOI: 10.1007/978-3-319-05591-6_47, © Springer International Publishing Switzerland 2014

semi-implicit finite volume discretization is used for the advection where the inflow parts of finite volume boundaries are treated implicitly in time and the outflow parts are treated in an explicit way. Our idea is to combine such approach with a second order accurate approximation of the curvature term using a procedure similar to the Crank-Nicolson method. The latter method is successfully used in a Lagrangian type of method for curvature driven flow in [1].

In this paper we propose a particular finite volume scheme of this type. The scheme treats the advection and diffusion fluxes in a compatible way. The resulting system of semilinear algebraic equations has favorable properties that can be used conveniently to solve it. When fixing the nonlinear coefficients in algebraic equations, the resulting matrix is a M-matrix and iterative solvers like the Gauss-Seidel method can be used to solve the linearized algebraic system.

Second order accurate methods for purely advective type of equations need in general some stabilization ("limiter") techniques to suppress nonphysical oscillations in numerical solutions [4, 6, 7]. In the presence of curvature driven motion as in our case we need not to apply such techniques if the advection is not too strong.

The paper is organized as follows. In Sect. 2 we derive briefly the level set equation that we want to solve. In Sect. 3 the finite volume method is derived. The Sect. 4 introduces a method for the solution of nonlinear algebraic equations. In Sect. 5 we derive a representative exact solution of the level set equation and present experimental order of convergence for our numerical method. Moreover, examples motivated by the modeling of fire front propagation in forests are presented. Finally, in Sect. 6 we conclude briefly our results.

2 Mathematical Model

Let $u = u(x, t)$, $(x, t) \in D \times [0, T]$ be the so called level set function used e.g. to represent implicitly an evolving interface. We denote $\mathbf{n} := \nabla u / |\nabla u|$ when $|\nabla u| \neq 0$. Note that $\mathbf{n}(\bar{x})$ is the normal vector at $\bar{x}$ to the level set given by $u(x, t) = u(\bar{x}, t)$.

We search $u = u(x, t)$ for $(x, t) \in D \times (0, T]$ fulfilling the level set equation

$$\frac{\partial u}{\partial t} + (f + \delta k)\, \mathbf{n} \cdot \nabla u = 0\,, \quad u(x, 0) = u^0(x)\,. \tag{1}$$

In (1) the term $f + \delta k$ represents a speed in normal direction $\mathbf{n}$ with $f(x)$ and $\delta(x) > 0$ being given. The function k denotes the curvature that is defined by

$$k = -\nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right) . \tag{2}$$

Substituting (2) to (1) one obtains the nonlinear advection-diffusion level set equation of the form

$$\frac{\partial u}{\partial t} + \left(f \frac{\nabla u}{|\nabla u|}\right) \cdot \nabla u = \delta |\nabla u| \nabla \cdot \left(\frac{\nabla u}{|\nabla u|}\right), \quad u(x, 0) = u^0(x). \tag{3}$$

3 Finite Volume Method

Before discretizing (3) we divide it by $|\nabla u|$ and rewrite the advection term as in [3] to obtain

$$\frac{1}{|\nabla u|}\frac{\partial u}{\partial t} + \nabla \cdot \left(uf \frac{\nabla u}{|\nabla u|^2}\right) - u\nabla \cdot \left(f \frac{\nabla u}{|\nabla u|^2}\right) = \delta \nabla \cdot \left(\frac{\nabla u}{|\nabla u|}\right). \tag{4}$$

For simplicity we consider the domain $D \subset R^2$ to be a square and the finite volume mesh to consist of squared elements $p_{ij}, i, j = 1, 2, \dots, N$ having uniform length $h > 0$ for all edges. The edges of p_{ij} are denoted by l_k, $k \in \Lambda_{ij}$ where $\Lambda_{ij} = \{(i+1/2\, j), (i\, j+1/2), (i-1/2\, j), (i\, j-1/2)\}$ is the set of indices for particular edges of p_{ij}.

Furthermore, we consider a uniform time step Δt and $t^m = m\Delta t$. The numerical solution of (4) will be represented by the discrete unknown values u_{ij}^m that approximates u in $p_{ij} \times (t^{m-1}, t^m]$.

The idea of a finite volume discretization for (4) is to integrate it over p_{ij} and to use appropriate quadrature rules that we explain for each term separately. Firstly,

$$\int_{p_{ij}} \frac{1}{|\nabla u|}\frac{\partial u}{\partial t} dx \approx \frac{h^2}{|\nabla u|_{ij}} \frac{du_{ij}}{dt}\,, \tag{5}$$

where the value $|\nabla u|_{ij}$ and the time discretization of $u_{ij} = u_{ij}(t)$ will be introduced later. Next,

$$\int_{p_{ij}} \nabla \cdot \left(fu \frac{\nabla u}{|\nabla u|^2}\right) dx = \sum_k \oint_{l_k} \left(\frac{fu}{|\nabla u|^2}\frac{\partial u}{\partial n}\right) ds \approx h \sum_k \left(\frac{f_k \bar{u}_k}{|\nabla u|_k^2} \left.\frac{\partial u}{\partial n}\right|_{l_k}\right) \tag{6}$$

and

$$\int_{p_{ij}} u\nabla \cdot \left(\frac{f\nabla u}{|\nabla u|^2}\right) dx \approx \bar{u}_{ij} \int_{p_{ij}} \nabla \cdot \left(\frac{f\nabla u}{|\nabla u|^2}\right) dx \approx h\bar{u}_{ij} \sum_k \left(\frac{f_k}{|\nabla u|_k^2} \left.\frac{\partial u}{\partial n}\right|_{l_k}\right). \tag{7}$$

The value f_k denotes an averaged value of f at l_k. Furthermore, $\bar{u}_k$ represents a reconstructed value of u assigned to l_k and $\bar{u}_{ij}$ is a reconstructed value of u assigned to p_{ij} [6, 7]. Particular choices for their computations, together with the approximations of $|\nabla u|_k$ and the normal derivatives $\partial u/\partial n$, will be introduced later.

Finally, analogous rules are applied for the last term in (4) to obtain

$$\int_{p_{ij}} \delta \nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right) dx \approx \delta_{ij} \sum_k \oint_{l_k} \left(\frac{1}{|\nabla u|} \frac{\partial u}{\partial n} \right) ds \approx h \delta_{ij} \sum_k \left(\frac{1}{|\nabla u|_k} \left. \frac{\partial u}{\partial n} \right|_{l_k} \right), \tag{8}$$

where δ_{ij} is an averaged value of δ with respect to p_{ij}.

Putting all approximations (5)–(8) together, we obtain a compact form of our finite volume discretization method

$$\frac{h^2}{|\nabla u|_{ij}} \frac{du_{ij}}{dt} - h \sum_k \left(f_k \frac{\bar{u}_{ij} - \bar{u}_k}{|\nabla u|_k^2} \left. \frac{\partial u}{\partial n} \right|_{l_k} \right) = h \sum_k \left(\frac{\delta_{ij}}{|\nabla u|_k} \left. \frac{\partial u}{\partial n} \right|_{l_k} \right). \tag{9}$$

We define now the missing approximations in (9). Firstly, we define u_k, $k \in \Lambda_{ij}$ by a linear interpolation,

$$u_{i+1/2\,j} := \frac{u_{ij} + u_{i+1\,j}}{2}, \quad u_{i-1/2\,j} := \frac{u_{ij} + u_{i-1\,j}}{2}, \quad \text{and so on.}$$

The normal derivatives are approximated in a standard way,

$$\left. \frac{\partial u}{\partial n} \right|_{l_{i+1/2\,j}} \approx \frac{u_{i+1/2\,j} - u_{ij}}{h/2} = \frac{u_{i+1j} - u_{ij}}{h}, \quad \left. \frac{\partial u}{\partial n} \right|_{l_{i-1/2\,j}} \approx \frac{u_{i-1/2\,j} - u_{ij}}{h/2}, \quad \text{and so on.}$$

To approximate ∇u at the edges l_k of p_{ij}, we use the diamond cell formula. To do so we use the notation $u_{i \pm \frac{1}{2}\, j \pm \frac{1}{2}}$ for the four values of u in the corners of p_{ij} that are obtained as arithmetic averages

$$u_{i \pm \frac{1}{2}\, j \pm \frac{1}{2}} := \frac{1}{4} \left(u_{ij} + u_{i \pm 1\, j} + u_{i\, j \pm 1} + u_{i \pm 1\, j \pm 1} \right) .$$

Using it, we can approximate $|\nabla u|$ at the edges l_k, $k \in \Lambda_{ij}$ of p_{ij} by

$$|\nabla u|_{i+1/2\,j} \approx \sqrt{\left(\frac{u_{i+1\,j} - u_{ij}}{h} \right)^2 + \left(\frac{u_{i+\frac{1}{2}\, j+\frac{1}{2}} - u_{i+\frac{1}{2}\, j-\frac{1}{2}}}{h} \right)^2 + \varepsilon^2}, \quad \text{and so on.}$$

A regularization was introduced in above formula by choosing $0 < \varepsilon << 1$ to avoid a division by zero in (9). Furthermore,

$$|\nabla u|_{ij} \approx \frac{1}{4} \sum_{k \in \Lambda_{ij}} |\nabla u|_k . \tag{10}$$

Finally, we have to define in (9) the reconstructed values $\bar{u}_{ij}$ and $\bar{u}_k$. Following [6] we take simply $\bar{u}_{ij} = u_{ij}$ and $\bar{u}_k = u_k$. This choice works well when the advection does not dominate the diffusion term in (3), in general more sophisticated choices have to be taken into account, see [6, 7].

Summarizing all approximations used in (9) we obtain

$$\frac{h^2}{|\nabla u|_{ij}}\frac{du_{ij}}{dt} = 2\sum_{k\in\Lambda_{ij}} \frac{1}{|\nabla u|_k}\left(f_k\frac{u_{ij}-u_k}{|\nabla u|_k}+\delta_{ij}\right)\left(u_k-u_{ij}\right) . \tag{11}$$

To introduce formally a second order accurate time discretization of (11) we treat the advection and diffusion term separately. We begin with the time discretization of the curvature term. Inspired by [1] we use a Crank-Nicolson type of time discretization that can be viewed as an arithmetic average of fully explicit and fully implicit time discretization scheme,

$$\begin{aligned}&\frac{h^2}{2\Delta t}\left(\frac{1}{|\nabla u|_{ij}^{m+1}}+\frac{1}{|\nabla u|_{ij}^{m}}\right)(u_{ij}^{m+1}-u_{ij}^{m}) = \\ &\delta_{ij}\sum_{k\in\Lambda_{ij}}\frac{1}{|\nabla u|_k^{m+1}}\left(u_k^{m+1}-u_{ij}^{m+1}\right)+\delta_{ij}\sum_{k\in\Lambda_{ij}}\frac{1}{|\nabla u|_k^{m}}\left(u_k^{m}-u_{ij}^{m}\right),\end{aligned} \tag{12}$$

where $|\nabla u|_{ij}^m$ and $|\nabla u|_{ij}^{m+1}$ are computed from (10) at corresponding time levels.

To discretize the advection term in time we introduce the notation in which we distinguish between the edges l_k of p_{ij} with an inflow and outflow character, namely

$$a_k^{in} = max\left(f_k\frac{u_{ij}^{m+1}-u_k^{m+1}}{|\nabla u|_k^{m+1}},0\right),\quad a_k^{out} = min\left(0,\, f_k\frac{u_{ij}^{m}-u_k^{m}}{|\nabla u|_k^{m}}\right). \tag{13}$$

The advection term can be approximated by the "Inflow Implicit/Outflow Explicit" time discretization [7] to obtain

$$\begin{aligned}&\frac{h^2}{2\Delta t}\left(\frac{1}{|\nabla u|_{ij}^{m+1}}+\frac{1}{|\nabla u|_{ij}^{m}}\right)(u_{ij}^{m+1}-u_{ij}^{m}) = \\ &\sum_{k\in\Lambda_{ij}}\frac{2a_k^{in}}{|\nabla u|_k^{m+1}}\left(u_k^{m+1}-u_{ij}^{m+1}\right)+\sum_{k\in\Lambda_{ij}}\frac{2a_k^{out}}{|\nabla u|_k^{m}}\left(u_k^{m}-u_{ij}^{m}\right).\end{aligned} \tag{14}$$

Putting (12) and (14) together we obtain

$$\begin{aligned}&\frac{h^2}{2\Delta t}\left(\frac{1}{|\nabla u|_{ij}^{m+1}}+\frac{1}{|\nabla u|_{ij}^{m}}\right)\left(u_{ij}^{m+1}-u_{ij}^{m}\right) = \\ &\sum_{k\in\Lambda_{ij}}\frac{2a_k^{in}+\delta_{ij}}{|\nabla u|_k^{m+1}}\left(u_k^{m+1}-u_{ij}^{m+1}\right)+\sum_{k\in\Lambda_{ij}}\frac{2a_k^{out}+\delta_{ij}}{|\nabla u|_k^{m}}\left(u_k^{m}-u_{ij}^{m}\right).\end{aligned} \tag{15}$$

4 Solution of Algebraic Equations

In this section we briefly comment how to solve the algebraic system of equations represented by the discretization scheme (15).

The values u^0_{ij} are computed from the initial condition. In our numerical experiments we consider only the Dirichlet type of boundary conditions. Consequently, one has to solve (15) for the unknowns $\{u^{m+1}_{ij},\ i, j = 1, 2, \dots, N-1\}$ in a sequence for $m = 0, 1$ and so on.

We propose to solve (15) using a combination of fixed point iterations and Gauss-Seidel iterative method. To introduce it we define for $p = -1, 0, 1$ and $q = -1, 0, 1$ that fulfill $|p| + |q| = 1$ the following coefficients

$$\lambda_{ij} = \Delta t \frac{|\nabla u_{ij}|^{m+1} + |\nabla u_{ij}|^{m}}{h^2}, \quad M^{pq}_{ij} = \frac{2a^{in}_{i+p/2\, j+q/2} + \delta_{ij}}{|\nabla u|^{m+1}_{i+p/2\, j+q/2}} \tag{16}$$

$$M_{ij} = \sum_{|p|+|q|=1} M^{pq}_{ij}, \quad b_{ij} = \sum_{|p|+|q|=1} \frac{2a^{out}_{i+p/2\, j+q/2} + \delta_{ij}}{|\nabla u|^{m}_{i+p/2\, j+q/2}} \left(u^m_{i+p\, j+q} - u^m_{ij}\right). \tag{17}$$

Using (16)–(17) the scheme (15) can be written in the form

$$u^{m+1}_{ij} = \frac{1}{1 + \lambda_{ij} M_{ij}} \left(u^m_{ij} + \lambda_{ij} \left(b_{ij} + \sum_{|p|+|q|=1} M^{pq}_{ij}\, u^{m+1}_{i+p\, j+q} \right) \right). \tag{18}$$

We note that the coefficients defined in (16) are nonlinear and always positive.

The iterative method consists of the following steps. Firstly, an initial guess for the unknowns u^{m+1}_{ij} is set to the available values u^m_{ij} from the previous time step or from the initial conditions if $m = 0$. Moreover, the coefficients b_{ij} in (17) are computed only once in each time step.

Each iteration of our iterative method is realized by computing the nonlinear coefficients defined in (16) using the values computed from the previous iteration. Fixing these coefficients one can update the values u^{m+1}_{ij} according to (18) for $i, j = 1, 2, \dots, N-1$ by evaluating the values of $u^{m+1}_{i+p\, j+q}$ on the right hand side of (18) in a manner of Gauss-Seidel iterative method.

5 Numerical Experiments

At first we derive an exact solution in a simplified situation when an evolving curve is a circle initially, and it evolves according to (1) with constant values of f and δ. In such case the evolving curve preserves its circular shape, so it can be described by its radius $r = R(t, r^0)$ where $r^0 = R(0, r^0)$.

Table 1 The comparison of numerical solution obtained with (15) with the exact solution (20)

N	Error	EOC	$\#it$
16	6.27e-2	–	26
32	1.00e-2	2.64	38
64	1.87e-3	2.42	59
128	4.41e-4	2.09	102

Let $u^0(r)$ be a given increasing function and $u(x_1, x_2, 0) = u^0(r)$, $r = \left(x_1^2 + x_2^2\right)^{1/2}$. Clearly, any circle of radius r^0 consists of points (x_1, x_2) such that $u(x_1, x_2, 0) = u^0(r^0)$. Our aim is to find $u(x_1, x_2, t)$ such that $u(x_1, x_2, t) = u^0(r^0)$ for all points (x_1, x_2) that fulfill $\left(x^2 + y^2\right)^{1/2} = R(t, r^0)$. To do so the inverse function of $r = R(t, r^0)$ with respect to r^0 must exist, i.e. $r^0 = R^{-1}(t, r)$. Once available, one obtains $u(x_1, x_2, t) = u^0(R^{-1}(t, \sqrt{x^2 + y^2}))$.

If a circular curve expands or shrinks with a constant speed f and δ, the radius $r(t)$ shall fulfill the equation $\dot{r}(t) = f + \frac{\delta}{r}$, $r(0) = r_0$ which is solved by

$$R(t, r^0) = \frac{\delta}{f} + \frac{\delta}{f} W\left(\frac{1}{\delta}\left(f r^0 - \delta\right) e^{\frac{-\delta + f r^0 + f^2 t}{\delta}}\right) \tag{19}$$

where W is the product log function, i.e. $W(z)$ is obtain such that $z = We^W$.

Let us choose as initial function $u^0(x_1, x_2) = \sqrt{x_1^2 + x_2^2}$. Using our approach one obtains the solution of (3) for constant f and δ in the form

$$u(x_1, x_2, t) = \frac{\delta}{f} + \frac{\delta}{f} W\left(\frac{1}{\delta}\left(f\sqrt{x_1^2 + x_2^2} - \delta\right) e^{-\frac{\delta - f\sqrt{x_1^2 + x_2^2} + f^2 t}{\delta}}\right). \tag{20}$$

In Table 1 we present the comparison of numerical solution obtained by (15) with the exact solution (20) for $f = \delta = 1$ and $t \in [0, 1]$ using a standard l_2 discrete norm in time and space. The domain D is a square with the side length $L = 8$. The Dirichlet boundary conditions defined by the available exact solution are used on ∂D. The discretization step is taken $h = 8/N$ for $N = 16, 32, 64, 128$, the time step is chosen $\Delta t = h/2$.

One can see from Table 1 that for this example the experimental order of convergence is approaching 2 from above. Moreover we present the number of iterations for each N that were necessary to reduce the residuum below the value 10^{-10}.

In the following illustrative examples we are motivated by numerical simulation of fire front propagation in forests [2]. The parameter $f(x)$ defines how fast the underlying forest can burn and $\delta(x) = \mu f(x)$ where $\mu > 0$, so the speed in normal direction $\mathbf{n}$ is given by $f(x)\,(1 + \mu k)$.

The first example shows a behavior for inhomogeneous forest, see Fig. 1. The second example illustrates a topological change when the evolving fire front, being a circle initially, has to surround later a small area that can not burn, see Fig. 2.

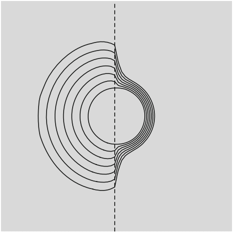

Fig. 1 Picture of fire front position at different time levels. The *smallest circle* is the initial position of the front. The parameters are $f = 1$ left and $f = 0.2$ right, $\mu = 0.1$

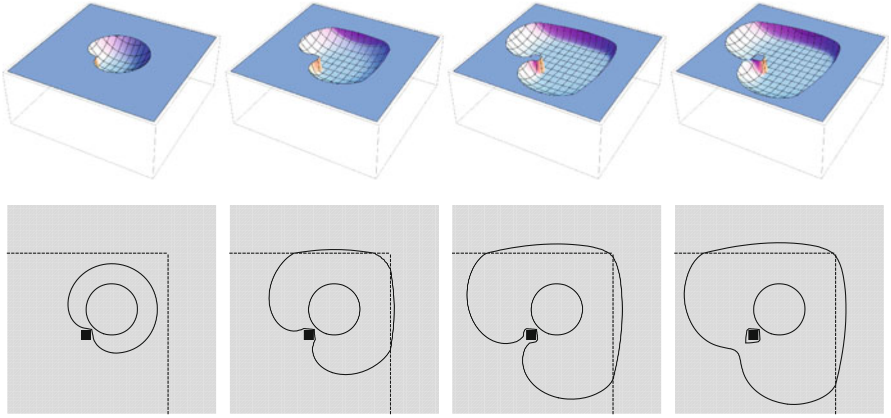

Fig. 2 Pictures of fire front position at 4 different time levels, the *top row* with 3D view, the *bottom row* 2D view. The small *black square* can not burn ($f = 0$), the north-east region is less burnable ($f = 0.2$) than the rest ($f = 1$). The *small circle* in 2D view is the initial position of the front

6 Conclusions

Our novel finite volume method combines conveniently explicit and implicit time discretization to obtain the second order accurate numerical solution of level set equation containing the terms of advection and diffusion character. For the chosen representative example for which the exact solution is derived, we can report experimental order of convergence approaching the value 2 from above.

Acknowledgments The first, second and fourth author were supported by APVV 0184-10 and VEGA 1/1137/12.

References

1. Balažovjech, M., Mikula, K.: A higher order scheme for a tangentially stabilized plane curve shortening flow with a driving force. SIAM J. Sci. Comp. **33**, 2277–2294 (2011)
2. Balažovjech, M., Mikula, K., Petrášová, M., Urbán, J.: Lagrangian methods with topological changes for numerical modelling of forest fire propagation. In: Handlovičova, A. et al. (eds.) Proceeding of Algoritmy 2012, pp. 42–52. Slovak University of Technology, Bratislava (2012)
3. Frolkovič, P., Mikula, K.: Flux-based level set method: a finite volume method for evolving interfaces. Appl. Numer. Math. **57**(4), 436–454 (2007)
4. Frolkovič, P., Mikula, K.: High-resolution flux-based level set method. SIAM J. Sci. Comp. **29**(2), 579–597 (2007)
5. Handlovičová, A., Mikula, K., Sgallari, F.: Semi-implicit complementary volume scheme for solving level set like equations in image processing and curve evolution. Numer. Math. **93**, 675–695 (2003)
6. Mikula, K., Ohlberger, M.: A new level set method for motion in normal direction based on a semi-implicit forward-backward diffusion approach. SIAM J. Sci. Comp. **32**(3), 1527–1544 (2010)
7. Mikula, K., Ohlberger, M.: Inflow-Implicit/Outflow-Explicit scheme for solving advection equations. In: Fort, J. et al. (ed.) Finite Volumes for Complex Applications VI, pp. 683–692. Springer, New York (2011)
8. Walkington, N.J.: Algorithms for computing motion by mean curvature. SIAM J. Numer. Anal. **33**, 2215–2238 (1996)