HIGH-RESOLUTION FLUX-BASED LEVEL SET METHOD

PETER FROLKOVIČ[*] AND KAROL MIKULA[†]

Abstract. A new high-resolution flux-based finite volume method for general advection equations in non-divergent form including a level set equation for moving interfaces is introduced. The method is applicable to the case of non-divergence free velocity and to general unstructured grids in higher dimensions. We show that the method is consistent and that the numerical solution fulfils discrete minimum/maximum principle. Numerical experiments show its second order accuracy for smooth solutions as well as for solutions with discontinuous derivatives and on general unstructured meshes. Numerical examples for passive transport and shrinking of dynamic interfaces, including examples with topological changes, are presented using locally adapted 2D and 3D grids.

Key words. level set method, finite volume method, high-resolution, evolving interfaces, unstructured grids

AMS subject classifications. 65M25, 76M12

1. Introduction.

The basic task of level set methods is to describe the evolution of dynamic interfaces. Particular forms of the level set equations and corresponding computational simulations have a tremendous area of different applications, see [31, 32, 38, 39] for details and references.

The level set equation for the moving interface advected by a velocity field $\vec{V}$ can be written in the form

$$\partial_t \phi + \vec{V} \cdot \nabla \phi = 0 \,, \quad \phi(t^0, x) = \phi^0(x) \,, \tag{1.1}$$

where the level set function $\phi = \phi(t, x)$ describes the evolving interface implicitly by its zero level set (contour line in 2D and isosurface in 3D).

Depending on the type of application, the velocity $\vec{V}$ in (1.1) can be given in several ways. For instance, in tracking of dynamic interfaces between two immiscible fluids, the velocity field $\vec{V} = \vec{V}(x)$ describes some external fluid movement, see [44, 45, 16, 17, 24] for some examples. Typically, $\vec{V}$ is obtained as a numerical solution of some PDE describing a conservation law for fluid flow, e.g., $\nabla \cdot \vec{V} = 0$.

Further applications result from problems of geometric evolution equations, appearing among others in crystal growth, combustion or image segmentation, where $\vec{V} = \delta \vec{N}$ with $\vec{N}$ being the unit normal vector to the interface given by $\vec{N} = \nabla \phi / |\nabla \phi|$, and $\delta = \delta(x)$ describes the speed of motion in normal direction $\vec{N}$. In this case, the equation (1.1) is usually written in the form

$$\partial_t \phi + \delta |\nabla \phi| = 0 \,, \tag{1.2}$$

and even higher-order terms like (mean) curvature or intrinsic Laplacian of curvature can be included in the definition of δ, cf. [39].

Finally, if there is a non-zero right hand side f in the equation (1.2) and one is searching for its stationary solutions, the so-called Eikonal equation

$$\delta |\nabla \phi| = f \tag{1.3}$$

[*] Simulations in Technology Center, University of Heidelberg, Im Neuenheimer Feld 368, 69120 Heidelberg, Germany (peter.frolkovic@uni-hd.de).

[†] Department of Mathematics and Descriptive Geometry, Slovak University of Technology, Radlinskeho 11, 81368 Bratislava, Slovakia (mikula@math.sk).

with some appropriate boundary conditions, has to be solved.

These important applications of (1.1) with $\vec{V} = \vec{V}(x, \nabla\phi)$ are solved using different numerical techniques. The main objective of this paper is to derive a consistent high-resolution flux-based finite volume method that can be used for such applications (including non-divergence free velocities) on general unstructured computational grids in any space dimension.

Nowadays, the finite volume techniques are very popular discretization methods for conservation laws. They preserve many useful properties of numerical solutions known from the analytical case like the mass conservation and the minimum/maximum principle.

The level set equations that model the movement of interface in normal direction, are very often solved by the finite difference methods. Such numerical techniques can be described as solutions to the so called Hamilton-Jacobi equation,

$$\partial_t \phi + H(\nabla\phi) = 0 \tag{1.4}$$

with $H(\nabla\phi) = \delta|\nabla\phi|$ for (1.2). Numerical discretizations of (1.4) can be then formally written in the form

$$\phi_i^{n+1} = \phi_i^n - \Delta t^n \, \mathcal{H}_i(\nabla_i^n \phi)\,, \tag{1.5}$$

where $\mathcal{H}_i$ is numerical Hamiltonian, and numerical gradients $\nabla_i^n \phi$ are determined typically by some special finite difference approximations of $\nabla\phi$ in the time-space grid point (t^n, x_i). Popularity of such finite difference level set methods for structured grids are dated from [33] and they have reached high level quality using the Hamilton-Jacobi ENO [34], WENO [19] or high-resolution central schemes [23].

Later, several discretization techniques were developed for two-dimensional unstructured triangular meshes that are described in the form (1.5). In [1, 2] the so called monotonic Lax-Friedrichs Hamiltonian $\mathcal{H}_i$ was proposed, and similar ideas were used to derive WENO schemes [47] or central WENO schemes [22] for triangular meshes. In [20] the approximative gradients $\nabla_i^n \phi$ in (1.5) are found by an averaging over finite volumes that are dual to finite element mesh, and in [4] by an averaging over finite elements surrounding the nodes. In all these papers, only triangular meshes are considered.

A different approach to solve level set problems is taken by semi-Lagrangian [11, 10, 41] or particle tracking methods [9]. The main idea of such methods is to work directly with the formulation (1.1) by approximating the related characteristic curves. Note that these methods can still be viewed as point-wise (particle) based numerical techniques.

One drawback of all mentioned finite difference based level set methods is that they are not conservative if applied with passive transport having divergence free velocity $\vec{V}$. For such applications, some combinations of level set methods with other methods to improve the mass preservation were proposed like reinitialization techniques to recover numerical signed distance function [42], or the coupled level set and volume-of-fluid method [43], or the hybrid particle level set method [8], see also a comprehensive review in [24]. Recently, few papers propose to use standard numerical conservative methods for level set equation (1.1) with divergence free velocity like finite volume methods [30] or discontinuous Galerkin discretizations [35, 48, 25].

Finite volume or finite element methods that are very useful for conservation laws on unstructured grids, cannot be used directly for (1.1) or (1.4), because the Hamilton-Jacobi equation "cannot be written in a conservation form, suitable for integration by parts, which is the backbone of finite volume and finite element methods" [47]. A new approach was proposed in [14, 15] to overcome this difficulty, where the level set problem (1.1) was reformulated to a conservation form with a source term. At the same time, a flux-based finite volume scheme using integration by parts was described, and the standard CFL restriction on time step was relaxed using a recursive algorithm.

The flux-based level set method proposed in [15] is based on the first order accurate approximation. In this paper, we extend it to a high-resolution method based on second order approximations using the concept of "high-resolution finite volume methods" as discussed in details in, e.g., [21]. Our high-resolution scheme involves in a natural way also the first order scheme given in [15] and, in such a way, is able to approximate well not only problems with smooth solutions, but also problems that exhibits solutions with discontinuous derivatives. We can prove that the high-resolution flux-based level set method is consistent on general grids and that the discrete minimum/maximum principle is valid under natural assumptions. To make valid such assumptions in general, well-known limiting procedures can be applied.

Several numerical experiments will be given that illustrate the robustness and order of convergence of our new method including examples with smooth solutions, solutions with discontinuous derivatives and solutions with topological changes. Among others, unstructured grids made of rectangles and triangles in 2D, and hexahedra, pyramids, and tetrahedra in 3D are used. No reinitialization techniques are applied to recover a numerical signed distance function. A comparison with ENO scheme (2nd order in space and time) using [26] for an example of rotating solid body shows better performance of our method. Finally, more complex examples of Zalesak's disc rotation and vortex-in-a-box are presented.

The paper is organised as follows. In Section 2 we derive a general consistent flux-based finite volume discretization scheme for solving equation (1.1). In Section 3 we present particular (first and second order accurate) discretization schemes. In Section 4, experimental order of convergence of the method and various numerical experiments are discussed.

2. General consistent discretization scheme. In this section we derive a finite volume discretization method for the model problem of advection equation in non-divergent form,

$$\partial_t \phi + \vec{V} \cdot \nabla \phi = 0 \,, \quad \phi(t^0, x) = \phi^0(x) \,, \tag{2.1}$$

where the function $\phi = \phi(t,x)$ has to be determined for $t > t^0 \in \mathbf{R}$ and for $x \in \Omega \subset \mathbf{R}^d$ with $d = 2$ or $d = 3$. The domain Ω is supposed to be polygonal with boundary $\partial\Omega$. For a moment, we skip a description of boundary conditions.

The advection equation (2.1) is solved on subintervals (t^n, t^{n+1}) of the time interval of interest and is related to (1.1) by considering $\vec{V}(x) = \vec{V}(x, \nabla\phi(t^n, x))$.

If one introduces characteristic curves $X = X_x^n(t)$ defined in a backward manner,

$$X_x^n(t) = x - \int_{t^n}^{t} \vec{V}(X_x^n(s))\, ds \,,$$

one can express the solution of (2.1) using the simple formula

$$\phi(t,x) = \phi(t^n, X_x^n(t)) \,. \tag{2.2}$$

Several numerical methods, like the modified method of characteristics [7] or semi-Lagrangian methods [10], treat (2.1) by solving (or approximating) the relation (2.2) in each grid point $x = x_i$ of a discrete mesh.

To apply a finite volume method to approximate (2.1), we suppose that Ω is covered by a mesh of non-empty, non-overlapping, polygonal computational cells $\Omega_i \subset \Omega$, $i = 1, 2, \ldots, I$, the finite volumes. Furthermore, $\Delta t^n = t^{n+1} - t^n$, $n = 0, 1, \ldots$.

There are several types of finite volume schemes for solving advective equations, cf. [21, 3]. The cell-based finite volume method can be obtained by integrating (2.2) over Ω_i in $t = t^{n+1}$, see [21, 37, 27, 36] for some reviews and references. In order to derive the flux-based finite volume discretization, we rewrite (2.1) to the divergent form with non-zero right hand side,

$$\partial_t \phi + \nabla \cdot \left(\vec{V} \phi \right) = \phi \nabla \cdot \vec{V} \, . \tag{2.3}$$

If one integrates (2.3) over Ω_i and time interval (t^n, t^{n+1}), the following integral equation can be derived,

$$\int\limits_{\Omega_i} \phi(t^{n+1}, x) = \int\limits_{\Omega_i} \phi(t^n, x) - \int\limits_{t^n}^{t^{n+1}} \int\limits_{\partial\Omega_i} \vec{n}_i \cdot \vec{V} \phi(t, \gamma) + \int\limits_{t^n}^{t^{n+1}} \int\limits_{\Omega_i} \phi(t, x) \nabla \cdot \vec{V} \, , \tag{2.4}$$

where $\vec{n}_i = \vec{n}_i(\gamma)$, $\gamma \in \partial\Omega_i$ is the normal outward unit vector. The integral form (2.4) can be viewed formally as a conservation law with a source term. In the sequel we will derive a general discretization scheme for solving (2.1) applying numerical quadrature rules for integrals in (2.4). To do so, let us suppose that the numerical solution at $t = t^n$ is given as a function $\phi^n(x) \approx \phi(t^n, x), x \in \Omega$.

The first integral on the right hand side (r.h.s.) of (2.4) is approximated by

$$\int\limits_{\Omega_i} \phi(t^n, x) \, dx \approx |\Omega_i| \phi_i^n \, , \tag{2.5}$$

where $\phi_i^n = \phi^n(x_i)$ for some point $x_i \in \Omega_i$ that will be specified later. An analogous approximation is used for the integral on the l.h.s. of (2.4).

Before describing an approximation of the second integral in the r.h.s. of (2.4), we introduce some notations. The common boundary (of nonzero measure) of two neighbouring finite volumes Ω_i and Ω_j will be denoted by $\Gamma_{ij} := \partial\Omega_i \cap \partial\Omega_j$, $|\Gamma_{ij}| > 0$ and $\Gamma_{i0} := \partial\Omega_i \cap \partial\Omega$, $|\Gamma_{i0}| > 0$. Moreover, we suppose that Γ_{ij} consists of (possibly several) $\Gamma_{ij}^e \subset \Gamma_{ij}$ such that Γ_{ij}^e is a straight line if $d = 2$, or a flat polygon if $d = 3$. Consequently, one has

$$\partial\Omega_i = \bigcup_{(e,j) \in \Lambda_i} \overline{\Gamma_{ij}^e} \, , \quad \Lambda_i := \left\{ (e, j) \, : \, |\Gamma_{ij}^e| > 0 \right\} \, .$$

The integral over $\partial\Omega_i$ can be now expressed as the sum of boundary integrals over Γ_{ij}^e. Let $\gamma_{ij}^e \in \Gamma_{ij}^e$ be the barycenter of Γ_{ij}^e, it means the integration point of quadrature formula for integrals over Γ_{ij}^e that is exact for linear polynomials. In the sequel, we will use the following approximation

$$\int\limits_{\Omega_i} \nabla \cdot \vec{V}(x) \, dx = \int\limits_{\partial\Omega_i} \vec{n}_i(\gamma) \cdot \vec{V}(\gamma) \, d\gamma \approx \sum_{(e,j) \in \Lambda_i} |\Gamma_{ij}^e| \vec{n}_{ij}^e \cdot \vec{V}_{ij}^e \, , \tag{2.6}$$

where $\vec{n}^e_{ij} := \vec{n}_i(\gamma^e_{ij}) = -\vec{n}_j(\gamma^e_{ij}) = -\vec{n}^e_{ji}$ and $\vec{V}^e_{ji} \equiv \vec{V}^e_{ij} := \vec{V}(\gamma^e_{ij})$.

Similarly to (2.6), for the second integral in r.h.s. of (2.4), we get the following approximation

$$\int\limits_{t^n}^{t^{n+1}} \int\limits_{\partial\Omega_i} \vec{n}_i \cdot \vec{V} \phi(t,\gamma) d\gamma dt \approx \Delta t^n \sum_{(e,j)\in\Lambda_i} |\Gamma^e_{ij}| \vec{n}^e_{ij} \cdot \vec{V}^e_{ij} \, \phi^e_{ij}(t^{n+1/2}), \tag{2.7}$$

where

$$\phi^e_{ij}(t^{n+1/2}) \approx \phi^n(X^n_{\gamma^e_{ij}}(t^{n+1/2})) , \quad t^{n+1/2} := t^n + \Delta t^n/2 \, . \tag{2.8}$$

Finally, the third integral in the r.h.s of (2.4) is approximated by

$$\int\limits_{t^n}^{t^{n+1}} \int\limits_{\Omega_i} \phi(t,x) \, \nabla \cdot \vec{V}(x) \approx \Delta t^n \, \phi^{n+1/2}_i \sum_{(e,j)\in\Lambda_i} |\Gamma^e_{ij}| \vec{n}^e_{ij} \cdot \vec{V}^e_{ij} \, , \tag{2.9}$$

where $\phi^{n+1/2}_i$ shall approximate the averaged value of $\phi(t^{n+1/2},x)$, $x \in \Omega_i$. The simplest choice is $\phi^{n+1/2}_i = \phi^n_i$ that can be seen as the first order accurate approximation of integral in (2.9). Other choices will be introduced later.

Combining approximations (2.5), (2.7) and (2.9), we end up with the following **general discretization scheme** for solving (2.1):

$$\phi^{n+1}_i = \phi^n_i - \frac{\Delta t^n}{|\Omega_i|} \sum_{(e,j)\in\Lambda_i} |\Gamma^e_{ij}| \vec{n}^e_{ij} \cdot \vec{V}^e_{ij} \left(\phi^e_{ij}(t^{n+1/2}) - \phi^{n+1/2}_i \right) . \tag{2.10}$$

Note that the scheme (2.10) represents a fully explicit discretization method.

Similarly to [6, 4], we define the notion of consistency for a numerical scheme approximating (2.1).

Definition 2.1. *A discretization scheme of the form* (2.10) *approximating the equation* (2.1) *is called consistent, if for any constant velocity vector* $\vec{V}$ *and any linear polynomial* $\phi^n(x)$, $x \in \Omega$, *it takes the form*

$$\phi^{n+1}_i = \phi^n_i - \Delta t^n \, \vec{V} \cdot \nabla \phi^n \, . \tag{2.11}$$

To derive a specific form of general scheme (2.10) that has the consistency property, we use piecewise linear approximating functions $\phi^n(x)$ of the form

$$\phi^n(x) = \phi^n_i + \nabla \phi^n_i \cdot (x - x_i) \ , \quad x \in \Omega_i \, , \tag{2.12}$$

where $\nabla \phi^n_i \approx \nabla \phi^n(x_i)$. Then we introduce a particular form of (2.8) by defining the function $\phi^e_{ij}(t)$ for $t \in [t^n, t^{n+1})$ as follows,

$$\phi^e_{ij}(t) := \left\{ \begin{array}{ll} \phi^n_i + \nabla \phi^n_i \cdot \left(\gamma^e_{ij} - x_i - (t - t^n)\vec{V}_i \right) , & \vec{n}^e_{ij} \cdot \vec{V}^e_{ij} > 0, \\ \phi^n_j + \nabla \phi^n_j \cdot \left(\gamma^e_{ij} - x_j - (t - t^n)\vec{V}_j \right) , & \vec{n}^e_{ij} \cdot \vec{V}^e_{ij} < 0, \end{array} \right. \tag{2.13}$$

and $\vec{V}_i := \vec{V}(x_i) = \vec{V}(x_i, \nabla \phi^n_i)$. Note that the approximation (2.13) follows the conservation law principle as $\phi^e_{ij}(t) = \phi^e_{ji}(t)$. For the case $\gamma^e_{i0} \in \partial\Omega$, one can extend the definition (2.13) by chosing

$$\phi^e_{i0}(t) := \phi^n_i + \nabla \phi^n_i \cdot \left(\gamma^e_{i0} - x_i - (t - t^n)\vec{V}_i \right) , \quad \vec{n}^e_{i0} \cdot \vec{V}^e_{i0} < 0 \, . \tag{2.14}$$

Lemma 2.2. *If* (2.13) - (2.14) *are valid, then the general discretization scheme of the form* (2.10) *for solving equation* (2.1) *is consistent.*

Proof: After substituting constant vectors $\vec{V}$ and $\nabla\phi^n$ to (2.10) and using (2.13) to obtain $\phi^e_{ij}(t^n) = \phi^n_i + \nabla\phi^n \cdot (\gamma^e_{ij} - x_i) = \phi^n(\gamma^e_{ij})$, one has

$$\phi^{n+1}_i = \phi^n_i - \frac{\Delta t^n}{|\Omega_i|}\vec{V} \cdot \sum_{(e,j)\in\Lambda_i} |\Gamma^e_{ij}|\vec{n}^e_{ij}\phi^n(\gamma^e_{ij}) +$$

$$\frac{\Delta t^n}{|\Omega_i|}\left(\frac{\Delta t^n}{2}\vec{V}\cdot\nabla\phi^n + \phi^{n+1/2}_i\right)\sum_{(e,j)\in\Lambda_i}|\Gamma^e_{ij}|\vec{n}^e_{ij}\cdot\vec{V} = \phi^n_i - \Delta t^n\vec{V}\cdot\nabla\phi^n,$$

because of (2.6) and

$$\sum_{(e,j)\in\Lambda_i}|\Gamma^e_{ij}|\vec{n}^e_{ij}\phi^n(\gamma^e_{ij}) = \int\limits_{\partial\Omega_i}\vec{n}_i(\gamma)\phi^n(\gamma)d\gamma = \int\limits_{\Omega_i}\nabla\phi^n dx = |\Omega_i|\nabla\phi^n\,.$$

3. Particular forms of general discretization scheme (2.10).

3.1. First order discretization scheme. Formally, the general discretization scheme (2.10) covers also the case of piecewise constant numerical solution $\phi^n(x)$, when (2.12) takes the following simple form,

$$\phi^n(x) = \phi^n_i\,,\quad x\in\Omega_i \quad\Rightarrow\quad \nabla\phi^n \equiv \vec{0}\,,\quad x\in\Omega_i\,.$$

Consequently, one obtains from (2.13) that for $t\in[t^n,t^{n+1})$

$$\phi^e_{ij}(t) = \left\{\begin{array}{ll}\phi^n_i, & \vec{n}^e_{ij}\cdot\vec{V}^e_{ij} > 0,\\ \phi^n_j, & \vec{n}^e_{ij}\cdot\vec{V}^e_{ij} < 0,\end{array}\right. \tag{3.1}$$

which is nothing else than the standard first order upwind approximation. If one also chooses $\phi^{n+1/2}_i = \phi^n_i$, the scheme (2.10) takes the form

$$\phi^{n+1}_i = \phi^n_i - \frac{\Delta t^n}{|\Omega_i|}\sum_{(e,j)\in\Lambda^{in}_i}|\Gamma^e_{ij}|\vec{n}^e_{ij}\cdot\vec{V}^e_{ij}\left(\phi^n_j - \phi^n_i\right), \tag{3.2}$$

where

$$\Lambda^{in}_i := \{(e,j)\in\Lambda_i\,,\,\vec{n}^e_{ij}\cdot\vec{V}^e_{ij} < 0\}\,,\quad \Lambda^{out}_i := \Lambda_i\setminus\Lambda^{in}_i\,. \tag{3.3}$$

Of course, one can not expect in general that the discretization scheme (3.2) is consistent in the sense of Definition 2.1. The lack of consistency can be illustrated by a simple one-dimensional example with $\Omega_i = (x_i - 0.5h_i, x_i + 0.5h_i)$, $\vec{V} = V = const < 0$, when it can be written as

$$h_i\phi^{n+1}_i = h_i\phi^n_i - \Delta t^n V(\phi^n_{i+1} - \phi^n_i)\,, \tag{3.4}$$

$i = 1,\ldots,I$. It is easy to see that the method (3.4) is consistent only if $h_i \equiv h$. Similarly, the scheme (3.2) in higher dimensional cases is consistent only for uniform grids with $\Omega_i = (x_i - h/2, x_i + h/2)^d$, i.e. when $|\Omega_i| \equiv h^d$ and $|\Gamma_{ij}| \equiv h^{d-1}$.

Due to stability reasons, the CFL restriction on Δt^n in (3.2) must be fulfilled. It can be formulated by requiring that

$$\Delta t^n \le \min_i \tau_i\,, \tag{3.5}$$

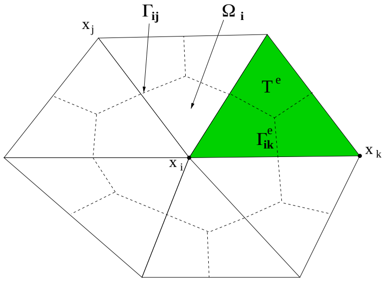

FIG. 3.1. *Illustration of 2D vertex-centred finite volume mesh.*

where τ_i represents the local residence time for Ω_i (or critical time step, cf. [15]) given by

$$\tau_i := -\frac{|\Omega_i|}{V_i^{in}}\,, \quad V_i^{in} := \sum_{(e,j)\in\Lambda_i^{in}} |\Gamma^e_{ij}|\vec{n}^e_{ij}\cdot\vec{V}^e_{ij}\,, \quad i = 1,2,\ldots,I\,. \tag{3.6}$$

If (3.5) is fulfilled, then the local discrete minimum and maximum principle is valid,

$$\min\{\phi_i^n,\ \phi_j^n,\, j\in\Lambda_i^{in}\} \le \phi_i^{n+1} \le \max\{\phi_i^n,\ \phi_j^n,\, j\in\Lambda_i^{in}\}\,. \tag{3.7}$$

The method (3.2) was developed by the authors in [15] and applied on uniform grids to nontrivial examples of curve and surface evolution. The method was realised with vertex-centred finite volume method that is described in next section. Moreover, a recursive procedure removing the CFL restriction (3.5) has been introduced to the method in [15].

3.2. High resolution discretization scheme. First, we reconstruct the piecewise linear approximating function $\phi^n(x)$ in (2.12) by following, e.g., [40]. To do so, we suppose that the finite volume mesh Ω_i, $i = 1,\ldots,I$ is complementary to some conforming finite element mesh $T^e \subset \Omega$ with $e = 1,\ldots,E$ and vertices x_i, $i = 1,\ldots,I$. We use the barycentred form of such vertex-centred control volumes Ω_i, where the polygonal boundary $\partial\Omega_i$ is obtained by connecting the barycentres of elements with middle points of element edges, see, e.g., [12, 3] for some details and Figure 3.1 for an illustration.

The standard finite element interpolation of nodal values ϕ_i^n is used to define

$$\hat{\phi}^n(x) := \sum_{i=1}^{I} \phi_i^n N_i(x)\,, \tag{3.8}$$

where $N_i(x)$ denote finite element basis functions with $N_i(x_j) = \delta_{ij}$.

To reconstruct the piecewise constant gradient of $\phi^n(x)$ in (2.12), we apply

$$\nabla\phi_i^n := \sum_e \frac{\beta^n_{i,e}}{|\Omega_i\cap T^e|} \int_{\Omega_i\cap T^e} \nabla\hat{\phi}^n(x)\,dx\,. \tag{3.9}$$

The preferable choice for $\beta^n_{i,e}\in[0,1]$ (such that $\sum_e \beta^n_{i,e} \le 1$) is

$$\beta^n_{i,e} = |\Omega_i\cap T^e|/|\Omega_i|\,, \tag{3.10}$$

other choices can be used in a limiting procedure, see Remark 1 later.

The velocity $\vec{V}$ is approximated at integration points γ_{ij}^e by $\vec{V}_{ij}^e := \vec{V}(\gamma_{ij}^e, \nabla\hat{\phi}^n(\gamma_{ij}^e))$.

To define $\phi_i^{n+1/2}$ in the general scheme (2.10), we introduce the total outflow and inflow fluxes, see also (3.3),

$$V_i^{out} := \sum_{(e,j)\in\Lambda_i^{out}} |\Gamma_{ij}^e|\vec{n}_{ij}^e \cdot \vec{V}_{ij}^e\,, \quad V_i^{in} := \sum_{(e,j)\in\Lambda_i^{in}} |\Gamma_{ij}^e|\vec{n}_{ij}^e \cdot \vec{V}_{ij}^e\,. \tag{3.11}$$

Furthermore, if $V_i^{out} > 0$, we define time dependent averaged outflow value

$$\phi_i^{out}(t) := \frac{1}{V_i^{out}} \sum_{(e,j)\in\Lambda_i^{out}} |\Gamma_{ij}^e|\vec{n}_{ij}^e \cdot \vec{V}_{ij}^e\, \phi_{ij}^e(t), \tag{3.12}$$

and analogously for averaged inflow value (if $V_i^{in} < 0$),

$$\phi_i^{in}(t) := \frac{1}{V_i^{in}} \sum_{(e,j)\in\Lambda_i^{in}} |\Gamma_{ij}^e|\vec{n}_{ij}^e \cdot \vec{V}_{ij}^e\, \phi_{ij}^e(t). \tag{3.13}$$

Finally, let us define $\phi_i^{n+1/2} = \alpha_i \phi_i^{out}(t^{n+1/2}) + (1-\alpha_i)\phi_i^{in}(t^{n+1/2})$, $\alpha_i \in [0.5, 1]$ and

$$V_i := (1-\alpha_i)V_i^{out} - \alpha_i V_i^{in}\,, \tag{3.14}$$

and the scheme (2.10) can be written as

$$\phi_i^{n+1} = \phi_i^n - \frac{\Delta t^n V_i}{|\Omega_i|}\left(\phi_i^{out}(t^{n+1/2}) - \phi_i^{in}(t^{n+1/2})\right)\,. \tag{3.15}$$

The preferable choice of α_i in (3.14) is $\alpha_i = 0.5$. Note that the parameters α_i in (3.14) and β_i in (3.9) can be chosen independently of each other.

Denoting the residence time $\tau_i := |\Omega_i|/V_i$, the scheme (3.15) can be written also in the form

$$\phi_i^{n+1} = \phi_i^n - \Delta t^n \frac{\phi_i^{out}(t^{n+1/2}) - \phi_i^{in}(t^{n+1/2})}{\tau_i}\,.$$

Note that the first order accurate method (3.2) can be obtained from (3.15) by setting $\beta_{i,e}^n = 0$ in (3.9) and $\alpha_i = 1$ in (3.14), $i = 1, \ldots, I$. Such form of (3.15) can be used only locally, when it takes the form

$$\phi_i^{n+1} = \phi_i^n - \frac{\Delta t^n V_i^{in}}{|\Omega_i|}\left(\phi_i^n - \phi_i^{in}(t^{n+1/2})\right)\,. \tag{3.16}$$

We choose (3.16) instead of (3.15) for special cases $V_i^{out} = 0$ or $V_i^{in} = 0$, when (3.15) is not well defined. Note that for $V_i^{in} = 0$, (3.16) turns to $\phi_i^{n+1} = \phi_i^n$.

The following local discrete minimum and maximum principle can be proved for ϕ_i^{n+1} by showing that (3.15) and (3.16) can be viewed as positive coefficient schemes [3, 4].

THEOREM 3.1. *Let us denote*

$$\phi_{i,\min}^n := \min\{\phi_i^n, \{\phi_j^n, (j,e) \in \Lambda_i \textit{ such that } \exists (k,e) \in \Lambda_i^{in}\}\},$$

$$\phi_{i,\max}^n := \max\{\phi_i^n, \{\phi_j^n, (j,e) \in \Lambda_i \textit{ such that } \exists (k,e) \in \Lambda_i^{in}\}\}\,.$$

If

$$\phi^n_{i,\min} < \phi^{in}_i(t^n) < \phi^n_{i,\max} \tag{3.17}$$

and

$$\phi^n_{i,\min} < \phi^n_i < \phi^n_{i,\max}\,, \tag{3.18}$$

then ϕ^{n+1}_i given by (3.15) fulfills the local minimum and maximum principle

$$\phi^n_{i,\min} \le \phi^{n+1}_i \le \phi^n_{i,\max} \tag{3.19}$$

for sufficiently small time step $\Delta t^n > 0$.

If only (3.17) is fulfilled then ϕ^{n+1}_i given by (3.16) fulfills (3.19).

Proof. Firstly, we prove (3.19) for the solution given by (3.15). To do so, let us denote

$$\omega(t) := \frac{\phi^n_i - \phi^n_{i,min}}{\phi^{out}_i(t) - \phi^n_{i,min}}\,. \tag{3.20}$$

If $\phi^{out}_i(t^n) > \phi^n_i$ or $\phi^{out}_i(t) \ge \phi^n_i$ for $t \ge t^n$, one obtains that $\omega(t^{n+1/2}) \in (0,1]$ for sufficiently small $\Delta t^n > 0$. From (3.20) one has

$$\phi^{out}_i(t) = \frac{1}{\omega(t)}\phi^n_i - \frac{1-\omega(t)}{\omega(t)}\phi^n_{i,min}\,, \tag{3.21}$$

and (3.15) can be written in the form of positive coefficient scheme

$$\begin{aligned}\phi^{n+1}_i &= (1 - \frac{\Delta t^n}{\omega(t^{n+1/2})\tau_i})\phi^n_i + \\ &+ \frac{(1-\omega(t^{n+1/2}))\Delta t^n}{\omega(t^{n+1/2})\tau_i}\phi^n_{i,min} + \frac{\Delta t^n}{\tau_i}\phi^{in}_i(t^{n+1/2})\,,\end{aligned} \tag{3.22}$$

if $\Delta t^n \le \min\{\omega(t^n), \omega(t^{n+1/2})\}\tau_i$. Consequently, the property (3.19) is valid.

For the case that $\phi^{out}_i(t^n) < \phi^n_i$ or $\phi^{out}_i(t) \le \phi^n_i$ for $t \ge t^n$ one can proceed analogously by defining

$$\omega(t) := \frac{\phi^n_i - \phi^n_{i,max}}{\phi^{out}_i(t) - \phi^n_{i,max}}\,. \tag{3.23}$$

For ϕ^{n+1}_i obtained from (3.16) the proof is straightforward. □

Remark 1.

The results of Theorem 3.1 can be used to control any standard limiter procedure, see, e.g., [21, 3, 13] for flux or slope limiter procedures. For instance, the assumption (3.17) can be made valid by setting (in an extreme case) $\beta^n_{j,e} = 0$, $j \in \Lambda^{in}_i$ in (3.10).

4. Numerical experiments.

In the following subsections, we present various numerical experiments to illustrate the behaviour of the high-resolution flux-based level set method. Note that in all examples no limiting procedures and no reinitializations of the level set function were used.

Table 4.1

Errors in $L_2((0,T), L_2(S^2))$-norm and EOC for the example of shrinking sphere using the 1st order scheme (3.2) *(3rd and 4th columns) and using the high-resolution scheme* (3.15) *(5th and 6th columns). Analogously, the errors in $L_1((0,T), L_1(S^2))$-norm and EOC for the example of shrinking cube are presented (7th and 8th columns). In both cases, the high-resolution scheme gives the second order accuracy.*

E	N	Error	EOC	Error	EOC	Error	EOC
10	10	3.871e-2		6.289e-3		2.412e-3	
20	20	1.867e-2	1.05	1.470e-3	2.09	7.800e-4	1.63
40	40	9.181e-3	1.02	3.546e-4	2.05	2.128e-4	1.87
80	80	4.528e-3	1.02	8.763e-5	2.01	5.395e-5	1.98
160	160	2.243e-3	1.01	2.167e-5	2.01	1.380e-5	1.97

4.1. Comparison with exact solutions on 3D uniform grid. In this subsection we give a comparison with simple exact solutions of the surface evolution problem.

First we show experimental order of convergence (EOC) for evolving sphere computed on subsequently refined uniform rectangular grid in 3D. For $\delta = -1$ in (1.2), the exact radius of a shrinking sphere is given by $r(t) = r(0) - t$, $t \in [0, T]$, where $r(0)$ is an initial radius and the solution is well-defined for $T \leq r(0)$. We use the spatial domain $\Omega = [-1.25, 1.25]^3$, which is split into E^3 cubic elements T^e with the side length $h = 2.5/E$. The time step is given by $\tau = T/N$.

To compare numerical and exact solutions, we find all zero crossing points $x_k^n, k = 1, ..., K$, of piecewise linear representation of numerical solution with finite element grid lines in every discrete time step $n = 0, ..., N$. Then, in the case of a shrinking sphere, we compute Euclidean distances $r_k^n = |x_k^n|$ from the origin of all $x_k^n, k = 1, ..., K$, and compare them with the exact radius of the shrinking sphere. Summing for all discrete time steps we get the formula

$$\text{Error } = \left(\sum_{n=0}^{N} \tau \frac{1}{K} \sum_{k=1}^{K} (r_k^n - r(n\tau))^2 \right)^{\frac{1}{2}} \tag{4.1}$$

which is used as $L_2((0,T), L_2(S^2))$-norm of the difference of exact and numerical surfaces, where S^2 is a unit sphere. In the experiment presented in Table 4.1 we use $r(0) = 1$ and $T = 0.4$.

In the case of a shrinking cube centred at origin, we compute $m_k^n = \max(|x_1^n|, |x_2^n|, |x_3^n|)$ for all zero crossing points $x_k^n = (x_1^n, x_2^n, x_3^n)_k$, $k = 1, ..., K$. The value m_k^n is compared with the corresponding exact quantity, a distance $m(t)$ of the faces of the exactly shrinking cube from the origin. Note that $m(t) = m(0) - t$. Points on the exactly and numerically evolved surfaces correspond to points on the unit sphere (using radial coordinate). Therefore, analogously to (4.1), we can define the formula

$$\text{Error } = \sum_{n=0}^{N} \tau \frac{1}{K} \sum_{i=1}^{K} |m_i^n - m(n\tau)| \tag{4.2}$$

which is used as $L_1((0,T), L_1(S^2))$-norm of the error in surface motion. This norm is used for nonsmooth solutions due to the corners in evolving isosurface. In the experiment presented in Table 4.1 we use $m(0) = 1$, $T = 0.4$, and further parameters are the same as in the case of shrinking sphere.

Table 4.1 shows that the method (3.15) is $O(h^2)$ for both examples comparing to the method from (3.2), which is only first order accurate.

4.2. Comparison with exact solutions on 2D unstructured grids. We present two-dimensional examples with known analytical solutions to demonstrate the high-resolution accuracy of our method on unstructured grids. Implementations were realized with software toolbox UG [5]. The domain is a square $\Omega := (-0.5, 0.5)^2$. The velocity is given as a sum of divergence free part (the rotation) and the part causing shrinking or expanding of the interface in normal direction,

$$\vec{V} = \begin{pmatrix} -y \\ x \end{pmatrix} + \delta \, \frac{\nabla \phi}{|\nabla \phi|} \,, \tag{4.3}$$

where δ is a constant to be specified later. The initial condition and the Dirichlet boundary conditions will be prescribed using the exact solution Φ of a particular problem.

Firstly, an example of a circle that rotates and shrinks is considered,

$$\Phi(t, x_1, x_2) = \sqrt{\tilde{x}_1^2 + \tilde{x}_2^2} - r_0 + \delta t \tag{4.4}$$

and

$$\tilde{x}_1 := x_1 \cos(t) + x_2 \sin(t) + 0.25, \; \tilde{x}_2 := x_2 \cos(t) - x_1 \sin(t). \tag{4.5}$$

The initial radius is chosen $r_0 = 0.2$. The parameter δ is chosen such that after half rotation, i.e., $t = \pi$, the radius of circle is halved, it means $\delta = -0.1/\pi$.

Figure 4.1 shows an initial finite element mesh of 28 triangles representing the grid level 0. The grid is subsequently refined by dividing each triangle to 4 subtriangles. The finest grid is obtained after 6 uniform refinements (i.e., the grid level equals 6) and contains 114688 triangles. The picture in the middle of Figure 4.1 presents the numerical solution at $t = 0$ and $t = \pi$ for the grid level 6.

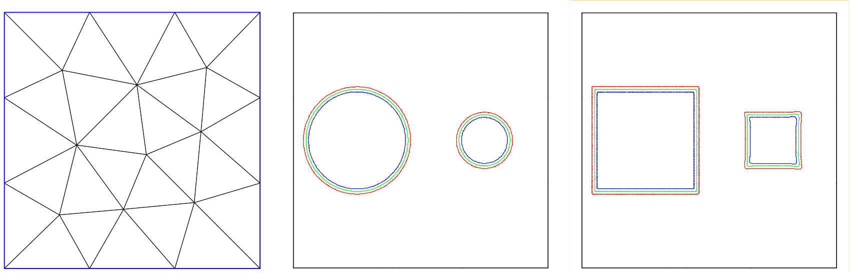

Fig. 4.1. *Initial coarse grid (left) and three contour lines ($\phi = -0.01, 0.0$ and 0.01) of numerical solutions at $t = 0$ and $t = \pi$ for the circle example (middle) and the square example (right). In this resolution, almost no difference between exact and numerical zero isoline would be visible.*

The Table 4.2 shows a L_1 norm of numerical error at $t = t^N = T$,

$$\text{Error} = \int\limits_{\Omega} |\Phi(T, x_1, x_2) - \sum_{i=1}^{I} \phi_i^N N_i(x_1, x_2)| \, dx_1 dx_2, \tag{4.6}$$

for each grid level, see also (3.8). Note that the error (4.6) is computed on the whole domain and not only on the interface. The L_1 error of the interface position at $t = T$ is computed using

$$\text{Error} = \sum_{k=1}^{K} |\Gamma^k| \left|r^k - r_{\text{exact}}\right| , \tag{4.7}$$

where the numerical interface is reconstructed as a polygon made of K segments Γ^k, and the Euclidean distance r^k is computed between the middle point (x_1^k, x_2^k) of Γ^k and the centre (x_1^c, x_2^c). For this example, one has $r_{\text{exact}} = 0.1$ and $(x_1^c, x_2^c) = (0.25, 0.0)$. Moreover, the error between the exact area 0.01π of the circle at $t = \pi$ and the area given by the numerical solution is presented in Table 4.2.

Table 4.2
Numerical errors and EOC for the shrinking and rotating circle example.

level	N	Error (4.6)	EOC	Error (4.7)	EOC	Area error	EOC
3	80	3.5268E-4		5.8035E-4		3.5455E-4	
4	160	0.9099E-4	1.95	1.2370E-4	2.23	0.1953E-4	4.18
5	320	0.2330E-4	1.97	0.3203E-4	1.95	0.0629E-4	1.63
6	640	0.0598E-4	1.96	0.0830E-4	1.95	0.0211E-4	1.58

An analogous numerical experiment is realized using a level set function (not a distance function) describing a square that is transported by the same velocity (4.3), the results are presented in Figure 4.1 right and Table 4.3. The exact solution is given by

$$\Phi(t, x_1, x_2) := \begin{cases} \tilde{x}_2 - r_0 - \delta t & \tilde{x}_2 \geq 0 \quad \& \quad \tilde{x}_2 \geq |\tilde{x}_1| \\ -\tilde{x}_2 - r_0 - \delta t & \tilde{x}_2 \leq 0 \quad \& \quad -\tilde{x}_2 \geq |\tilde{x}_1| \\ \tilde{x}_1 - r_0 - \delta t & \tilde{x}_1 \geq 0 \quad \& \quad \tilde{x}_1 \geq |\tilde{x}_2| \\ -\tilde{x}_1 - r_0 - \delta t & \tilde{x}_1 \leq 0 \quad \& \quad -\tilde{x}_1 \geq |\tilde{x}_2| \end{cases} \tag{4.8}$$

where $r_0 = 0.2$, see also (4.5), and the corners of the square remain sharp producing discontinuous derivatives of the solution. The error (4.7) on the interface ($r_{\text{exact}} = 0.1$ and $(x_1^c, x_2^c) = (0.25, 0.0)$), where $r^k = \max(|x_1^k - x_1^c|, |x_2^k - x_2^c|)$, and the error between the exact area 0.04 and the area determined by the numerical solution are included in Table 4.3.

Table 4.3
Numerical errors and EOC for the shrinking and rotating square example.

level	N	Error (4.6)	EOC	Error (4.7)	EOC	Area error	EOC
3	80	1.7461E-3		3.8785E-3		3.3658E-3	
4	160	0.5986E-3	1.54	1.3725E-3	1.50	1.0503E-4	1.64
5	320	0.1678E-3	1.83	0.3698E-3	1.89	0.2677E-4	1.97
6	640	0.0443E-3	1.92	0.0982E-3	1.91	0.0669E-4	2.00

Next, analogous numerical experiments with expanding characteristics are presented. Firstly, an example of expanding and rotating circle is chosen where the exact solution $\Phi(\pi, x_1, x_2)$ is given by (4.4) with $r_0 = 0.1$, $\delta = 0.1/\pi$ for $(x_1, x_2) \notin S\left((0.25, 0.0), 0.1\right)$, and $\Phi(\pi, x_1, x_2) \equiv -0.1$ otherwise, where $S\left((0.25, 0.0), 0.1\right)$ is a circle centred at $(0.25, 0.0)$ with the radius 0.1.

Note that for this example the high-resolution method gives EOC in the error (4.6) that is less than 2. One of the reasons is a constant region that is developed in the exact solution. Nevertheless, the error (4.7) on the interface diminishes quadratically, see Table 4.4, where also the difference between the exact area 0.04π and the numerical one is presented. The initial and final circular interface is presented in the left picture of Figure 4.2.

TABLE 4.4
Numerical errors and EOC for the expanding and rotating circle example.

level	N	Error (4.6)	EOC	Error (4.7)	EOC	Area error	EOC
3	80	1.5615E-3		1.8584E-3		7.338E-4	
4	160	0.5413E-3	1.53	0.4784E-3	1.96	1.434E-4	2.36
5	320	0.1989E-3	1.44	0.1100E-3	2.12	0.906E-4	0.66
6	640	0.0678E-3	1.55	0.0262E-3	2.07	0.080E-4	3.5

Finally, an example of expanding and rotating square is computed. The initial function $\Phi(0, x_1, x_2)$ is given by (4.8) with $r_0 = 0.1$ and $\delta = 0.1\pi$. The exact solution can be simply constructed using Huygens principle, see also the right picture in Figure 4.2. The global numerical error (4.6), the local one (4.7), and the difference between the exact area $0.12 + 0.01\pi$ and the numerical one are presented in Table 4.5. The numerical solution at $t = \pi$ is plotted in Figure 4.2 middle (three isolines where the interface is located) and right (isocontours plot of the final level set function). For this example, the high-resolution method seems to be slightly better than first order accurate.

TABLE 4.5
Numerical errors and EOC for the expanding and rotating square example.

level	N	Error (4.6)	EOC	Error (4.7)	EOC	Area error	EOC
3	80	3.9579E-3		9.9274E-3		6.3885E-3	
4	160	1.8124E-3	1.13	5.1273E-3	0.95	3.9632E-3	0.69
5	320	0.8203E-3	1.14	2.2505E-3	1.19	1.8900E-3	1.07
6	640	0.3733E-3	1.14	1.0654E-3	1.08	0.9554E-3	0.98

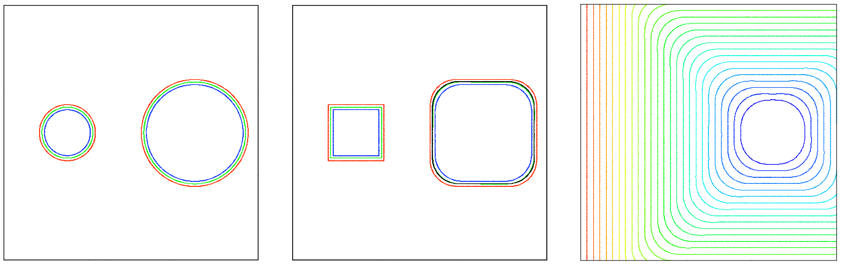

FIG. 4.2. *The left picture shows three contour lines ($\phi = -0.01, 0.0$ and 0.01) of numerical solutions at $t = 0$ and $t = \pi$ for the expanding and rotating circle example (in this resolution no difference between the exact and the numerical zero isoline could be visible). The middle picture presents analogous results for the expanding and rotating square where additionally the exact zero level line (black) is presented. The right picture shows the level lines of the numerical solution distributed uniformly between -0.075 and 0.55.*

4.3. Numerical experiments using locally adapted grids. The example with rotating and shrinking square was repeated with a locally adapted grid by refining (and coarsening) in each time step only elements in the neighbourhood of zero contour line, see Figure 4.3. Note that mixed grids consisting of squares and triangles were used. The simulation used 3290 finite elements at the beginning and 1547 ones at the end of computations, and it took considerably less time to realize than computations with corresponding uniformly refined grids.

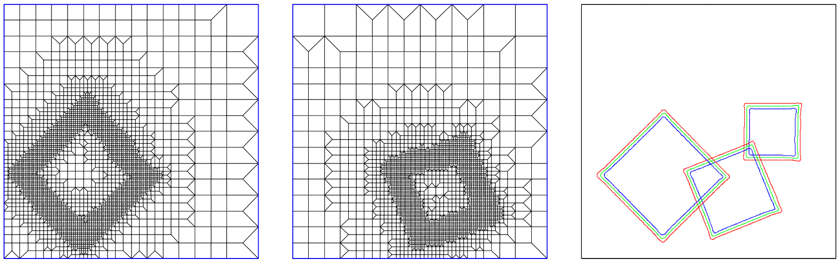

FIG. 4.3. *Locally adapted grid at $t = \pi/4$ (left) and $t = 5\pi/8$ (middle) and numerical solutions at $t = \pi/4$, $t = 5\pi/8$, and $t = \pi$ (right).*

To illustrate the capability of UG [5] for three-dimensional grids, Figure 4.4 shows the numerical solution (a shrinking sphere) and the grids before the first and last time step. Note that mixed grids consisting of hexahedra, pyramids and tetrahedra were used.

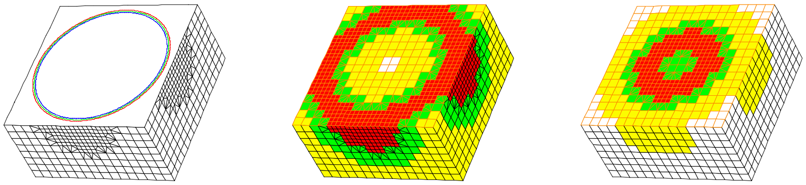

FIG. 4.4. *A cross-section of numerical solution (left picture) and locally adapted grid used in the first (middle picture) and last (right picture) time step for the example of a shrinking sphere.*

Finally, in a nontrivial experiment we evolve an initial curve in the form of a shrinking quatrefoil [15]. The quatrefoil is the zero level set of the initial level set function (not a signed distance function) constructed by the formula

$$\phi^0(x_1, x_2) = -0.5 + \sqrt{x_1^2 + x_2^2}/r_L, \quad r_L = 0.6 + 0.4 \sin\left(4 \text{ arctg}\left(\frac{x_2}{x_1}\right)\right).$$

In this experiment, topological changes of the evolving curve occur, see Figure 4.5.

4.4. Benchmarks with divergence free velocity. In this section, we present numerical results for two standard benchmark examples with divergence free velocity field to illustrate the behaviour of our method for such type of problems. The main application of level set equation (1.1) with externally given velocity $\vec{V}$ such that

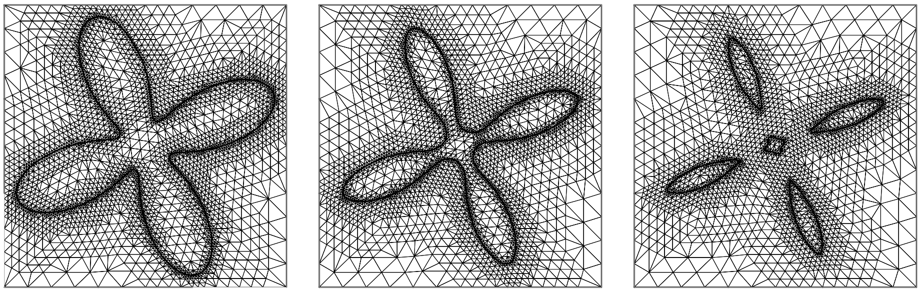

Fig. 4.5. *Shrinking of a quatrefoil with topological changes - the zero contour line of numerical solution and the used grid are plotted for three different time steps.*

$\nabla \cdot \vec{V} = 0$ are problems of two-phase flow where the evolving level set function shall describe a moving interface between two incompressible fluids. Note that for this type of problems, the level set equation can be written in a conservative form (2.3) with zero right hand side. For such classical scalar linear hyperbolic equation describing conservation law there exists an enormous number of sophisticated numerical methods and recently some of them were devoted directly to level set applications [30, 35, 25].

Before presenting our numerical results, we illustrate a better performance of the finite volume based scheme (3.15) as compared to the finite difference based ENO scheme [31] for a test example of rotating circle. The high-resolution flux-based level set method (3.15) was implemented in the Matlab toolbox of level set methods [26] and compared with the existing implementation of ENO scheme of second order in time and space.

The example is defined on a square $(-1,1)^2$ and the rotation is given in the clockwise direction. Dirichlet boundary conditions given by the exact values of signed distance function were used in both methods.

Figure 4.6 shows numerical solutions on a coarse grid with $(I+1)^2$ nodes, $I = 20$, where the differences between two methods can be clearly recognized. In Table 4.6, the simple discrete L_1-norm

$$\text{Error} = \frac{4}{I^2} \sum_{i=1}^{(I+1)^2} |\phi_i^0 - \phi_i^N|$$

is printed that is computed from the differences of numerical solutions in grid points at $t = 0$ and the time after one rotation.

Table 4.6
Numerical error of flux-based level set method (FB-LSM) and ENO scheme for rotating circle.

I	N	Error FB-LSM	Error ENO
20	140	104.046E-4	708.131E-4
40	280	28.726E-4	235.063E-4
80	560	7.899E-4	71.060E-4

In what follows, we present results of our scheme in solving two complex examples known from literature, namely for the Zalesak's disk rotation and the vortex-in-a-box

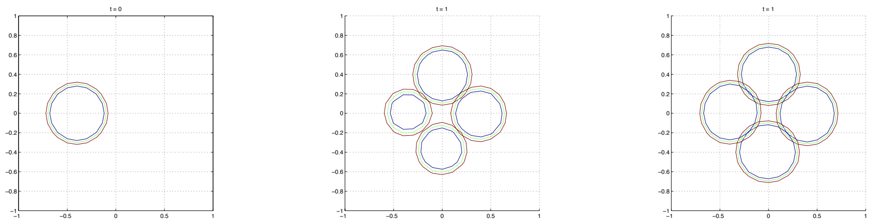

FIG. 4.6. *Initial condition (left), numerical solution obtained by the second order ENO scheme (middle) and by the high-resolution scheme FB-LSM (3.15) (right) for 4 different times on a coarse mesh of* 21×21 *grid points. The rotation occurs in the clockwise direction.*

example. In a literature, to improve a finite difference level set approximation of the interface, additional numerical techniques such as redistancing algorithms [44, 42, 48], particle tracking techniques [8, 9, 18], volume-of-fluid methods [46], or discontinuous Galerkin method [48, 25], were introduced, see also a review in [24].

In our opinion, the results obtained by the flux-based level set method (2.10), presented later, are qualitatively comparable to published studies, especially if one takes into the account that no additional numerical techniques, mentioned before, were applied in our computations to improve the level set approximation of the interface. In such a way, the flux-based level set method can produce satisfactory results for reasonable meshes by preserving its simple form. Nevertheless, opposite to quoted papers, the primal field of applications of our flux-based level set method are problems that involve geometric evolution equations and that might include also the evolution described by external flow, see, e.g., the boat-sail distance problem [28, 29].

Firstly, the example of the solid body rotation of the so-called Zalesak's disc is presented, see, e.g., [42, 8, 9, 18, 25, 46]. The initial interface is a circle with radius 0.15 centred at $(0.5, 0.75)$ with a vertical rectangular cut of the width and length equal 0.05 and 0.25, respectively. The slotted disc rotates in the counterclockwise direction with a constant angular velocity and returns to the origin position after one rotation.

The example is simulated on a unit square. The initial level set function is given by a signed distance function to Zalesak's disk and exact Dirichlet boundary conditions are used during simulations. The initial coarse grid (referred later as the grid level 0) is made of 4 right triangles that are obtained by connecting the corners of the square with the middle point of the square. Afterwards, the grid is refined uniformly by dividing each triangle into 4 subtriangles creating subsequent grid levels. Numerical results are presented for the grid level 7 (made of 65536 triangles with 33025 vertices, 720 uniform time steps) and 8 (made of 262144 triangles with 131585 vertices, 1440 uniform time steps).

In Figure 4.7 a comparison of the numerically and exactly computed interface after one rotation can be found for grid levels 7 and 8. The corresponding L_1 errors (4.6) are presented in Table 4.7.

As expected, one can observe a smoothing of initially sharp corners in the numerical solution of Zalesak's disk problem. The area of the disc surrounded by the zero isoline of the numerical level set function at the grid level 8 is 5.8216E-2 for $t = 0$ and 5.7893E-2 for $t = T$ that corresponds to the loss of 0.55%.

Finally, an example of a single vortex in a box is presented, see, e.g., [9, 18, 25, 46].

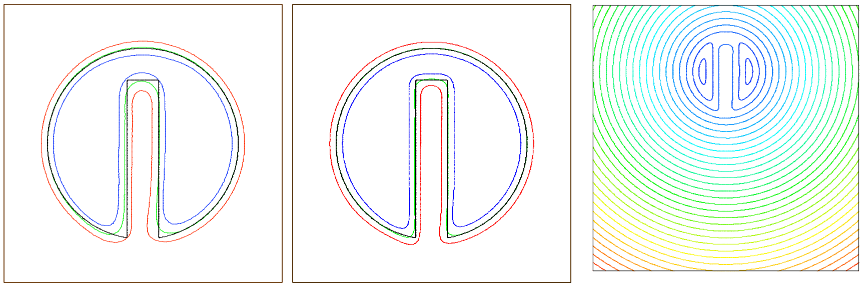

Fig. 4.7. *The left picture shows a detail of three level lines ($\phi = -0.01, 0.0$ and 0.01) of numerical solution and the exact zero line (black) after one rotation at grid level* 7 *for the Zalesak's example. The middle picture presents analogous results for the grid level 8. The right picture shows level lines of the numerical solution in the whole domain (distributed uniformly from -0.05 to 0.75) after one rotation for the grid 8.*

Table 4.7

Numerical errors and experimental order of convergence for the Zalesak example and the single vortex example.

level	N	Error (4.6)	EOC	N	Error (4.6)	EOC
		Zalesak ex.			Single vortex ex.	
6	360	6.3206E-4		800	1.2309E-2	
7	720	2.2406E-4	1.50	1600	0.4461E-2	1.46
8	1440	0.9224E-4	1.28	3200	0.1246E-2	1.84

The time-dependent velocity with nonzero vorticity is given by

$$\vec{V} = 2\cos(\frac{\pi t}{8})\left(-\sin^2(\pi x)\sin(\pi y)\cos(\pi y)\,,\,\sin^2(\pi y)\sin(\pi x)\cos(\pi x)\right)\,. \tag{4.9}$$

The time dependancy of $\vec{V}$ in (4.9) was resolved by evaluating it at $t = t^n + 0.5\Delta t$ for each timestep to obtain $\vec{V}$ in (1.1).

The initial interface is a circle with radius 0.15 and centred at $(0.5, 0.75)$ in a unit square $(0,1)^2$. The interface is stretched during simulations by wrapping around the centre of the box, it reaches a maximal deformation at $t = 4$ and returns to the initial circular form at $t = 8$.

The computations were realized using a signed distance function as an initial level set function and exact Dirichlet boundary conditions during simulations. The coarse grid and subsequent grid levels were chosen analogously to Zalesak's disk example. A uniform timestep was chosen by setting $\Delta t^n = \Delta t = 8/N$.

Numerical results for the single vortex example are plotted in Figure 4.8, numerical L_1 error (4.6) is presented in Table 4.7. The area of the circle at $t = 8$ equals 7.2302E-2 that corresponds to a gain of 2.3% with respect to the exact area 7.0686E-2. Note that again no extension to our level set method was used. For instance, no redistancing was applied in simulations, see the right picture in Figure 4.7 where the numerical level set function at $t = 4$ is presented.

Acknowledgments. The authors would like to thank the student Christian Wehner from the University of Heidelberg to provide us with numerical results comparing the flux-based level set method and the ENO scheme. The second author was supported by the grants VEGA 1/3321/06 and APVT-20-040902.

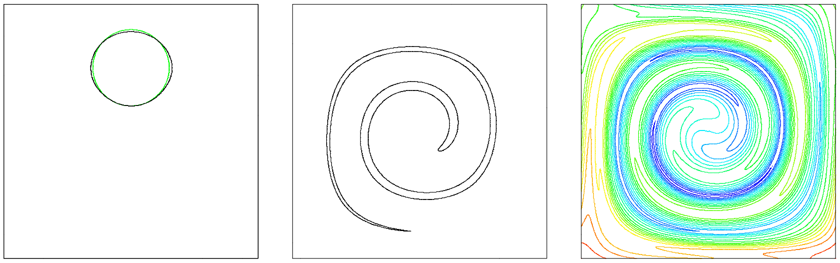

FIG. 4.8. *The left picture shows the exact circular interface (green) and the computed interface (black) at $t = 8$ of the single vortex example for the grid level* 8*. The middle picture presents the zero level line of the numerical solution at $t = 4$ and the right picture shows level lines in the whole domain distributed uniformly between -0.1 and 0.75.*

REFERENCES

[1] R. Abgrall. Numerical discretization of the first-order Hamilton-Jacobi equation on triangular meshes. *Comm. Pure Appl. Math.*, 49:1339–1373, 1996.

[2] S. Augoula and R. Abgrall. High order numerical discretization for Hamilton-Jacobi equations on triangular meshes. *SIAM J. Sci. Comput.*, 15:197–229, 2000.

[3] T. Barth and M. Ohlberger. Finite volume methods: foundation and analysis. In *Encyklopedia of Computational Mechanics*. John Wiley & Sons, Ltd., 2003.

[4] Timothy J. Barth and James A. Sethian. Numerical schemes for the Hamilton-Jacobi and level set equations on triangulated domains. *J. Comput. Phys.*, 1(145):1–40, 1998.

[5] P. Bastian, K. Birken, K. Eckstein, K. Johannsen, S. Lang, N. Neuss, and H. Rentz-Reichert. UG — a flexible software toolbox for solving partial differential equations. *Computing and Visualization in Science*, 1(1):27–40, 1997.

[6] M. G. Crandall and P. L. Lions. Two approximations of solutions of Hamilton-Jacobi equations. *Mathematics of Computation*, 43(167):1–19, July 1984.

[7] J. Douglas and T. Russell. Numerical methods for convection-dominated diffusion problems based on combining the method of characteristics with finite element or finite difference procedures. *SIAM J. Numer. Anal.*, 19:871–885, 1982.

[8] D. Enright, R. Fedkiw, J. Ferziger, and I. Mitchell. A hybrid particle level set method for improved interface capturing. *Journal of Computational Physics*, 183:83–116, 2002.

[9] D. Enright, F. Losasso, and R. Fedkiw. A fast and accurate semi-Lagrangian particle level set method. *Computers and Structures*, 83:479–490, 2005.

[10] M. Falcone and R. Ferretti. Convergence analysis for a class of high-order semi-Lagrangian advection schemes. *SIAM J. Numer. Anal.*, 35(3):909–940, 1998.

[11] M. Falcone, T. Giorgi, and P. Loreti. Level sets of viscosity solutions: Some applications to fronts and rendez-vous problems. *SIAM J. Appl. Math.*, 54(5):1335–1354, 1994.

[12] P. Frolkovič. Discretization. In E. Fein, editor, *D^3F - ein Programmpaket zur Modellierung von Dichteströmungen*, Braunschweig, 1999. GRS-139, ISBN 3-923875-97-5, (see also http://sit.uni-hd.de).

[13] P. Frolkovič. Flux-based method of characteristics for coupled transport equations in porous media. *Computing and Visualization in Science*, 6(4):173–184, 2004.

[14] P. Frolkovič and K. Mikula. Flux-based level set method: A finite volume method for evolving interfaces. Preprint 15, IWR, Universität Heidelberg, http://www.iwr.uni-heidelberg.de/sfb359/PP/Preprint2003-15.pdf, September 2003.

[15] P. Frolkovič and K. Mikula. Flux-based level set method: A finite volume method for evolving interfaces. *Applied Numerical Mathematics*, doi:10.1016/j.apnum.2006.06.002, 2006.

[16] I. Ginzburg and G. Wittum. Two-phase flows on interface refined grids modeled with VOF, staggered finite volumes, and spline interpolants. *J. Comput. Phys.*, 166:302–335, 2001.

[17] S. Gross, V. Reichelt, and A. Reusken. A finite element based level set method for two-phase incompressible flows. IGPM Report 241, RWTH Aachen, 2004.

[18] S. E. Hieber and P. Koumoutsakos. A Lagrangian particle level set method. *Journal of Computational Physics*, 210:342–367, 2005.

[19] G.S. Jiang and D. Peng. Weighted ENO schemes for Hamilton-Jacobi equations. *SIAM J. Sci. Comput.*, 21:2126–2143, 2000.

[20] G. Kossioris, Ch. Makridakis, and P.E. Souganidis. Finite volume schemes for Hamilton-Jacobi equations. *Numer. Math.*, 83:427–442, 1999.

[21] Randall J. LeVeque. *Finite Volume Methods for Hyperbolic Problems*. Cambridge Texts in Applied Mathematics. Cambridge University Press, 2002.

[22] D. Levy, S. Nayak, C.W. Shu, and Y.T. Zhang. Central WENO schemes for Hamilton-Jacobi equations on triangular meshes. *SIAM J. Sci. Comput.*, 27:532–552, 2005.

[23] C.T. Lin and E. Tadmor. High-resolution nonoscillatory central schemes for Hamilton-Jacobi equations. *SIAM J. Sci. Comput.*, 21:2163–2186, 2000.

[24] F. Losasso, R. Fedkiw, and S. Osher. Spatially adaptive techniques for level set methods and incompressible flow. *Computers and Fluids*, 35:995–1010, 2006.

[25] E. Marchandise, J.-F. Remacle, and N. Chevaugeon. A quadrature-free discontinuous Galerkin method for the level set equation. *Journal of Computational Physics*, 212:338–357, 2006.

[26] I. M. Mitchell. A toolbox of level set methods. Technical report, Department of Computer Science, University of British Columbia, Vancouver, Canada, 2004.

[27] T. Neubauer and P. Bastian. On a monotonicity preserving Eulerian-Lagrangian localized adjoint method for advection-diffusion equations. *Advances in Water Resources*, 28(12):1292–1309, 2005.

[28] T. Nishida and K. Sugihara. Approximation of the boat-sail Voronoi diagram and its application. In A. Laganà and et. al., editors, *ICCSA (3)*, pages 227–236. Springer, 2004.

[29] T. Nishida, K. Sugihara, and M. Kimura. Stable marker-particle method for the Voronoi diagram in a flow field. *Journal of Computational and Applied Mathematics*, doi:10.1016/j.cam.2006.01.035, 2006.

[30] E. Olsson and G. Kreiss. A conservative level set method for two phase flow. *Journal of Computational Physics*, 210:225–246, 2005.

[31] S. Osher and R. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Springer, 2003.

[32] S. Osher and N. Paragios. *Geometric Level Set Methods in Imaging, Vision, and Graphics*. Springer, 2003.

[33] S. Osher and J.A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79:12–49, 1988.

[34] S. Osher and C.-W. Shu. High-order essentially nonoscillatory schemes for Hamilton-Jacobi equations. *SIAM J. Numer. Anal.*, 28:907–922, 1991.

[35] D. A. Di Pietro, S. Lo Forte, and N. Parolini. Mass preserving finite element implementations of the level set method. *Applied Numerical Mathematics*, 56(9):1179–1195, 2006.

[36] M. Restelli, L. Bonaventura, and R. Sacco. A semi-Lagrangian discontinuous Galerkin method for scalar advection by incompressible flows. *Journal of Computational Physics*, 216(1):195–215, 2006.

[37] T. F. Russell and M. A. Celia. An overview of research on Eulerian-Lagrangian localized adjoint methods (ELLAM). *Advances in Water Resources*, 25:1215–1231, 2002.

[38] G. Sapiro. *Geometric Partial Differential Equations and Image Analysis*. Cambridge University Press, 2001.

[39] J. Sethian. *Level Set Methods and Fast Marching Methods*. Cambridge University Press, 1999.

[40] Th. Sonar. On the design of an upwind scheme for compressible flow on general triangulations. *Numerical Algorithms*, 4:135–149, 1993.

[41] J. Strain. Semi-Lagrangian methods for level set equation. *Journal of Computational Physics*, 151:498–533, 1999.

[42] M. Sussman and E. Fatemi. An efficient, interface-preserving level set redistancing algorithm and its application to interfacial incompressible fluid flow. *SIAM J. Sci. Comput.*, 20(4):1165–1191, 1999.

[43] M. Sussman and E. Puckett. A coupled level set and volume-of-fluid method for improved interface capturing. *Journal of Computational Physics*, 162:301–337, 2000.

[44] M. Sussman, P. Smereka, and S. Osher. A level set approach for computing solutions to incompressible two-phase flow. *Journal of Computational Physics*, 114:146–159, 1994.

[45] A.-K. Tornberg and B. Engquist. A finite element based level set method for multiphase flow applications. *Computing and Visualization in Science*, 3:93–101, 2000.

[46] X. Yang, A. J. James, J. Lowengrub, X. Zheng, and V. Cristini. An adaptive coupled level-set/volume-of-fluid interface capturing method for unstructured triangular grids. *Journal of Computational Physics*, 217(2):364–394, 2006.

[47] Y.T. Zhang and C.W. Shu. High-order WENO schemes for Hamilton-Jacobi equations on triangular meshes. *SIAM J. Sci. Comput.*, 24:1005–1030, 2003.

[48] X. Zheng, J. Lowengrub, A. Anderson, and V. Cristini. Adaptive unstructured volume remeshing - ii: Apllication to two- and three-dimensional level-set simulations of multiphase flow. *Journal of Computational Physics*, 208:626–650, 2005.