Semi-implicit level set method with inflow-based gradient in a polyhedron mesh

Jooyoung Hahn, Karol Mikula, Peter Frolkovič, and Branislav Basara

Abstract In this paper, a semi-implicit method is proposed to solve a propagation in a normal direction with a cell-centered finite volume method. An inflow-based gradient is used to discretize the magnitude of the gradient and it brings the second order upwind difference in an evenly spaced one dimensional domain. In three dimensional domain, we numerically verify that the proposed scheme is second order. The implementation is straightforwardly combined with a conventional finite volume code and 1-ring face neighborhood for parallel computation. An experimental order of convergence and a comparison of wall clock time between semi-implicit and explicit method are illustrated by numerical examples.

Key words: Semi-implicit method, level set method, polyhedron mesh

MSC (2010)**:** 65M08, 65N08, 35F25, 35F30

1 Introduction

We solve a partial differential equation for a propagation in a normal direction [11]:

$$\partial_t \phi(\mathbf{x},t) + F(\mathbf{x})|\nabla\phi(\mathbf{x},t)| = G(\mathbf{x}), \quad (\mathbf{x},t) \in \Omega \times [0,T], \tag{1}$$

where $\Omega \subset \mathbb{R}^3$ is a computational domain, T is the final time, the speed function F and the force term G are fixed, and the initial condition is given on Ω. Equa-

Jooyoung Hahn and Branislav Basara
AVL LIST GmbH, Hans-List-Platz 1, 8020 Graz, Austria
e-mail: jooyoung.hahn@avl.com and e-mail: branislav.basara@avl.com

Karol Mikula and Peter Frolkovič
Department of Mathematics and Descriptive Geometry, Slovak University of Technology, Radlinskeho 11, 810 05 Bratislava, Slovakia
e-mail: karol.mikula@stuba.sk and e-mail: peter.frolkovic@stuba.sk

tion (1) has been extensively used to solve evolving interfacial problems in many applications such as image processing, computer vision, combustion, fluid dynamics, etc; more details are given in [10, 14] and the references therein. In contrast to a standard structured mesh in image processing, real world three dimensional (3D) problems from physics or engineering are usually defined on a complicated geometry, for example, the combustion problems in 3D engines. Moreover, in industrial applications, a polyhedron mesh has been used extensively because of its shape flexibility [12]. In this paper, in order to extend topological advantages of the level set method into industrial problems with complicated geometry, we propose a numerical algorithm to solve the governing equation (1) on polyhedron meshes. We impose a linear extension at boundary, that is, a "ghost" value is linearly extrapolated from the boundary value. It can be properly discretized in a cell-centered finite volume method. Moreover, it allows us to use the simplest structure of decomposed domains for parallel computation which is the 1-ring face neighborhood structure.

Inspired by the methods in [16, 5, 7, 8, 3, 6], we propose a semi-implicit method to solve (1). It is very crucial to design a method to reduce a time step restriction in a polyhedron mesh. When the geometrical shape of a computational domain is complex, it is inevitable to have nonuniform size of cell volumes and it gives a severe restriction of time stepping in an explicit method because of the CFL condition for very small cells. The main difference between the proposed method and the methods in [2, 7, 8, 9] is an approximation of the gradient. Instead of using a cell-centered gradient to achieve the second order scheme, we approximate a gradient by an inverse distance average of face gradients only from inflow sides, named by the inflow-based gradient. In an evenly discretized 1D domain, the inflow-based gradient brings the correct second order upwind discretization of magnitude of the gradient in (1).

In the rest of the paper, the inflow-based gradient is introduced in Sec. 2 and then a semi-implicit method is proposed. In Sec. 3, the experimental order of convergence (EOC) is investigated and the wall clock time of semi-implicit and explicit method is compared.

2 Semi-implicit method with inflow-based gradient

In order to explain the proposed method for a 3D mesh, some notations are introduced. The sets of indices to uniquely indicate cells, faces, and vertices are denoted by $\mathscr{C}$, $\mathscr{F}$, and $\mathscr{V}$, respectively. A whole computational domain $\bar{\Omega} \subset \mathbb{R}^3$ is discretized by open cells Ω_p such that $\bar{\Omega} = \bigcup_{p\in\mathscr{C}} \bar{\Omega}_p$ with the volume $|\Omega_p| \neq 0$. We define two sets of neighbor information of Ω_p, $p \in \mathscr{C}$; the first is the neighbor cells whose face is shared by Ω_p, $\mathscr{N}_p = \{q \in \mathscr{C} : \text{there exists a face}\, e_f \in \partial\Omega_q \cap \partial\Omega_p,\, f \in \mathscr{F}\}$ and the second is the attached faces to Ω_p and they are indicated by two sets:$\mathscr{F}_p = \{f \in \mathscr{F} : e_f \in \partial\Omega_p\}$ and $\mathscr{B}_p = \{f \in \mathscr{F}_p : e_f \in \partial\Omega_p \cap \partial\Omega\}$. Note that a nonplanar face of a polyhedron cell should be tessellated into triangles to make its sub-face as a plane. From a given face center of a nonplanar face, a triangle is defined by two consecutive vertices of the face and the face center. By an abuse of notation, $\mathscr{F}$ includes all tessellated faces.

2.1 Inflow-based gradient finite volume method

With simple coefficient $G = 0$ in the governing equation (1) and using Gauss's theorem, we have a standard spatial discretization at $p \in \mathscr{C}$ in cell-centered finite volume method:

$$\int_{\Omega_p} \partial_t \phi + \sum_{f \in \mathscr{F}_p} \left(\phi_{pf} - \phi_p\right) a_{pf} = 0, \quad a_{pf} = \int_{e_f} F \frac{\nabla \phi}{|\nabla \phi|} \cdot \mathbf{n} \simeq F_f \frac{\nabla \phi_f}{|\nabla \phi_f|_\varepsilon} \cdot \mathbf{n}_{pf}, \quad (2)$$

where $\mathbf{n}$ is a unit outward normal vector at a face e_f, $f \in \mathscr{F}_p$, F_f is a value of F at the face center, $\mathbf{n}_{pf} = |e_f|\mathbf{n}$, and $|\nabla \phi_f|_\varepsilon = (\varepsilon^2 + |\nabla \phi_f|^2)^{\frac{1}{2}}$ with a small $\varepsilon > 0$. The spatial discretization is explained by two steps; the first is to define an inflow-based gradient and the second is to compute a face value ϕ_{pf}.

An inflow-based gradient computed by face gradients is defined at a cell center. A face gradient is obtained by a minimization from the values close to a face such as cell centers and vertices. A vertex value is linearly interpolated from cell-centered gradients. A cell-centered gradient with a linear extrapolation at boundary is obtained by a minimizer of a functional $f(\mathbf{y}) = \sum_{\mathbf{x} \in \mathscr{S}_p} w_p(\mathbf{x}) \big| \mathbf{y} \cdot (\mathbf{x} - \mathbf{x}_p) - (\phi(\mathbf{x}) - \phi_p) \big|^2$, where a weight function is $w_p(\mathbf{x}) = |\mathbf{x} - \mathbf{x}_p|^{-2}$ and a set of points $\mathscr{S}_p$ at the cell $p \in \mathscr{C}$ is either $\{\mathbf{x}_q \,|\, q \in \mathscr{N}_p\}$ if $\mathscr{B}_p = \emptyset$ or $\{\mathbf{x}_q \,|\, q \in \mathscr{N}_p\} \cup \{\mathbf{x}_b \,|\, b \in \mathscr{B}_p\}$ if $\mathscr{B}_p \neq \emptyset$. From a cell-centered gradient, a vertex value can be approximated by an inverse distance average and linear approximation from adjacent cells.

Before we define an inflow-based gradient, a face gradient should be computed in order to obtain a flux in (2) at a face e_f, $f \in \mathscr{F}$. Let us denote $\mathscr{P}_f$ as a set of points around a face center: either $\{\mathbf{x}_p, \mathbf{x}_q\} \cup \mathbf{V}_f$ if $\exists!\, p, q \in \mathscr{C}$ such that $e_f \in \partial \Omega_p \cap \partial \Omega_q$ or $\{\mathbf{x}_p\} \cup \mathbf{V}_f$ if $\exists!\, p \in \mathscr{C}$ such that $f \in \mathscr{B}_p \neq \emptyset$, where $\mathbf{V}_f$ are vertices of a face e_f. Note that $\mathscr{P}_f$ is a generalization of diamond-cell strategy in a regular structured cube mesh [1]. A face value α_f and gradient $\boldsymbol{\beta}_f$ are obtained by minimizer of a functional $g(a_f, \mathbf{b}_f) = \sum_{\mathbf{x} \in \mathscr{P}_f} w_f(\mathbf{x}) \big| a_f + \mathbf{b}_f \cdot (\mathbf{x} - \mathbf{x}_f) - \phi(\mathbf{x}) \big|^2$, where a weight function $w_f(\mathbf{x}) = |\mathbf{x} - \mathbf{x}_f|^{-2}$ at the face center $\mathbf{x}_f$. Note that a face value α_f on a boundary face is a linearly extended value. Finally, we define an inflow-based gradient as an inverse distance average of face gradients only from inflow faces with an inverse distant $d_{pf} = |\mathbf{x}_f - \mathbf{x}_p|^{-1}$ and its sum $W_d = \sum_{f \in \mathscr{A}_p^-} d_{pf}$:

$$D_p^- \phi = W_d^{-1} \sum_{f \in \mathscr{A}_p^-} d_{pf} \boldsymbol{\beta}_f, \quad (3)$$

where $\mathscr{A}_p^- = \mathscr{B}_p^- \cup \mathscr{F}_p^-$, $\mathscr{B}_p^- = \{b \in \mathscr{B}_p \,|\, a_{pb} < 0\}$, and $\mathscr{F}_p^- = \{f \in \mathscr{F}_p \setminus \mathscr{B}_p \,|\, a_{pf} < 0\}$. If $\mathscr{A}_p^- = \emptyset$, then we define $D_p^- \phi = 0$.

Now, we compute a face value ϕ_{pf} in (2) from the inflow-based gradient. When a face value is computed at an internal face, a face value ϕ_{pf} in (2) is computed straightforwardly:

$$\begin{aligned} f \in \mathscr{F}_p \setminus \mathscr{B}_p,\ p \in \mathscr{C} \ &\Rightarrow\ \exists!\, q \in \mathscr{N}_p \text{ such that } e_f \in \partial\Omega_p \cap \partial\Omega_q \\ &\Rightarrow\ \phi_{pf} = \begin{cases} \phi_p + D_p^- \phi \cdot (\mathbf{x}_f - \mathbf{x}_p) & \text{if } a_{pf} \geq 0, \\ \phi_q + D_q^- \phi \cdot (\mathbf{x}_f - \mathbf{x}_q) & \text{if } a_{pf} < 0. \end{cases} \end{aligned} \tag{4}$$

When a face value is computed at a boundary face, we use the linear extrapolation and then a face value ϕ_{pf} in (2) is formulated by

$$b \in \mathscr{B}_p (\neq \emptyset),\ p \in \mathscr{C} \ \Rightarrow\ \phi_{pb} = \begin{cases} \phi_p + D_p^- \phi \cdot (\mathbf{x}_b - \mathbf{x}_p) & \text{if } a_{pb} \geq 0, \\ \alpha_b & \text{if } a_{pb} < 0. \end{cases} \tag{5}$$

Note that the boundary constraint face value α_b, $b \in \mathscr{B}_p$ is obtained by imposing a linear extrapolation. From (4) and (5), we finally have the spatial discretization:

$$\begin{aligned} \int_{\Omega_p} \partial_t \phi = &- \sum_{f \in \mathscr{F}_p^-} \left(\phi_q + D_q^- \phi \cdot \mathbf{d}_{qf} - \phi_p\right) a_{pf} - \sum_{f \in \mathscr{F}_p^+} \left(D_p^- \phi \cdot \mathbf{d}_{pf}\right) a_{pf} \\ &- \sum_{b \in \mathscr{B}_p^-} (\alpha_b - \phi_p)\, a_{pb} - \sum_{b \in \mathscr{B}_p^+} \left(D_p^- \phi \cdot \mathbf{d}_{pb}\right) a_{pb}, \end{aligned} \tag{6}$$

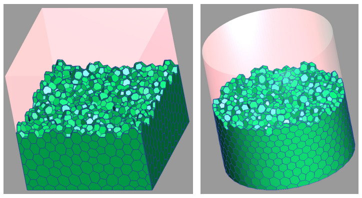
where $\mathscr{B}_p^+ = \mathscr{B}_p \setminus \mathscr{B}_p^-$, $\mathscr{F}_p^+ = (\mathscr{F}_p \setminus \mathscr{B}_p) \setminus \mathscr{F}_p^-$, $\mathbf{d}_{qf} = \mathbf{x}_f - \mathbf{x}_q$, and for each $f \in \mathscr{F}_p \setminus \mathscr{B}_p$, $p \in \mathscr{C}$ there exists an index $q \in \mathscr{N}_p$ such that $e_f \subset \partial\Omega_p \cap \partial\Omega_q$. From a tedious derivation of (6) in an evenly spaced 1D domain, the inflow-based gradient in the above formula brings the second order upwind difference of the magnitude of the gradient. Note that the first order upwind difference is used in well-known standard schemes [11, 13].

2.2 Semi-implicit method

Let us denote an evenly divided time step $\Delta t = T/N$ for a fixed $N \in \mathbb{N}$ and $\phi_p^n = \phi(\mathbf{x}_p, n\Delta t)$. Inspired by [3, 7, 8, 9, 16], the outflow information is used explicitly and we propose to use the inflow information partly implicitly and partly iteratively because of a limitation of sharing variables in the 1-ring face neighborhood structure of decomposed domains:

$$\begin{aligned} &\frac{|\Omega_p|}{\Delta t} \left(\phi_p^{n,k} - \phi_p^{n-1}\right) + \sum_{f \in \mathscr{F}_p^-} \left(\phi_q^{n,k} + D_q^- \phi^{n,k-1} \cdot \mathbf{d}_{qf} - \phi_p^{n,k}\right) a_{pf}^{n-1} \\ &+ \sum_{b \in \mathscr{B}_p^-} \left(\alpha_b^{n,k-1} - \phi_p^{n,k}\right) a_{pb}^{n-1} + \sum_{f \in \mathscr{A}_p^+} \left(D_p^- \phi^{n-1} \cdot \mathbf{d}_{pf}\right) a_{pf}^{n-1} = 0, \end{aligned} \tag{7}$$

where $k = 1, \ldots, K$ and $\mathscr{A}_p^+ = \mathscr{B}_p^+ \cup \mathscr{F}_p^+$. The above system of equations can be written by



level	P_b	P_c
1	$4,033$	$3,947$
2	$30,683$	$28,410$
3	$241,726$	$224,548$
4	$1,914,579$	$1,788,209$

Fig. 1 The first and second from the left figure are polyhedron cells in a box (P_b) and a cylinder (P_c) shape generated by AVL FIRE® and the right table is the number of cells at each level. If one level gets higher, the average volume of cells is approximately 8 times smaller.

$$\left(\frac{|\Omega_p|}{\Delta t} - \sum_{f\in\mathscr{A}_p^-} a_{pf}^{n-1}\right)\phi_p^{n,k} + \sum_{f\in\mathscr{F}_p^-} a_{pf}^{n-1}\phi_q^{n,k} = R(\phi_p^{n-1},\phi_p^{n,k-1}), \tag{8}$$

where the right-hand side R is a collection of explicit information:

$$\begin{aligned}R(\phi_p^{n-1},\phi_p^{n,k-1}) \equiv \frac{|\Omega_p|}{\Delta t}\phi_p^{n-1} &- \sum_{b\in\mathscr{B}_p^-}\alpha_b^{n,k-1}a_{pb}^{n-1}\\ &- \sum_{f\in\mathscr{F}_p^-} D_q^-\phi^{n,k-1}\cdot\mathbf{d}_{qf}a_{pf}^{n-1} - \sum_{f\in\mathscr{A}_p^+} D_p^-\phi^{n-1}\cdot\mathbf{d}_{pf}a_{pf}^{n-1}.\end{aligned}$$

For all examples in Sec. 3, we fix $K=1$ and update $\phi^n=\phi^{n,1}$ using $\phi^{n,0}=\phi^{n-1}$ in the above formulas. Moreover, $\varepsilon=10^{-12}$ in (2) is fixed.

3 Numerical experiments

Two examples are presented to check an EOC of the proposed method. An algebraic multigrid method (AMG) in AVL FIRE® on decomposed computational domains with 1-ring face neighborhood structure is used to solve (8) for all examples. In Fig. 1, a box shape $\bar{\Omega}=[-0.05,0.05]^3\subset\mathbb{R}^3$ and a cylinder shape whose height 0.1 and radius is 0.05 are chosen to be a computational domain and polyhedron cells are generated in four levels to check EOC. A time step Δt in (7) for each level from 1 to 4 is fixed to be $3.0\cdot10^{-3}$, $1.5\cdot10^{-3}$, $7.5\cdot10^{-4}$, and $3.75\cdot10^{-4}$, respectively.

The first example is a bidirectional flow from an analytically represented shape:

$$\partial_t\phi(\mathbf{x},t)\pm|\nabla\phi(\mathbf{x},t)|=\pm1, \quad (\mathbf{x},t)\in\Pi^{\pm}\times[0,T], \tag{9}$$

where a closed surface Π is given such that $\Pi=\partial\Pi^+\cap\partial\Pi^-$, $\bar{\Pi}^+\cup\bar{\Pi}^-=\bar{\Omega}$, $\Pi^+\cap\Pi^-=\emptyset$ and an initial value $\phi(\mathbf{x},0)$ is positive on Π^+, negative on Π^-, and zero at $\mathbf{x}\in\Pi$. The bidirectional flow computes a signed distance function from Π using linear extrapolation at boundary. In Table 1, Π is chosen as a sphere whose radius is 0.02 and a cube whose edge is 0.04 and $T=0.3$ is large enough to reach a steady state of (9) in a given box or cylinder shape domains in Fig. 1. From a sphere

	Sphere				Cube			
	P_b		P_c		P_b		P_c	
level	L^1	EOC	L^1	EOC	L^1	EOC	L^1	EOC
1	$1.90 \cdot 10^{-4}$	-	$1.70 \cdot 10^{-4}$	-	$1.09 \cdot 10^{-3}$	-	$8.30 \cdot 10^{-4}$	-
2	$5.24 \cdot 10^{-5}$	1.86	$4.29 \cdot 10^{-5}$	1.98	$5.29 \cdot 10^{-4}$	1.05	$4.38 \cdot 10^{-4}$	0.92
3	$1.30 \cdot 10^{-5}$	2.00	$1.08 \cdot 10^{-5}$	1.99	$2.54 \cdot 10^{-4}$	1.06	$1.94 \cdot 10^{-4}$	1.17
4	$3.10 \cdot 10^{-6}$	2.07	$2.60 \cdot 10^{-6}$	2.06	$1.28 \cdot 10^{-4}$	0.99	$9.55 \cdot 10^{-5}$	1.02
level	L^∞	EOC	L^∞	EOC	L^∞	EOC	L^∞	EOC
1	$8.67 \cdot 10^{-4}$	-	$8.09 \cdot 10^{-4}$	-	$3.87 \cdot 10^{-3}$	-	$4.17 \cdot 10^{-3}$	-
2	$4.40 \cdot 10^{-4}$	0.98	$3.92 \cdot 10^{-4}$	1.04	$2.14 \cdot 10^{-3}$	0.85	$2.13 \cdot 10^{-3}$	0.97
3	$2.71 \cdot 10^{-4}$	0.70	$2.36 \cdot 10^{-4}$	0.73	$8.88 \cdot 10^{-4}$	1.27	$8.51 \cdot 10^{-4}$	1.32
4	$1.17 \cdot 10^{-4}$	1.20	$1.11 \cdot 10^{-4}$	1.08	$4.45 \cdot 10^{-4}$	1.00	$4.52 \cdot 10^{-4}$	0.91
level	L^∞_ε	EOC	L^∞_ε	EOC				
1	$4.90 \cdot 10^{-4}$	-	$4.24 \cdot 10^{-4}$	-	N/A			
2	$1.84 \cdot 10^{-4}$	1.41	$1.93 \cdot 10^{-4}$	1.14				
3	$4.88 \cdot 10^{-5}$	1.92	$4.41 \cdot 10^{-5}$	2.13				
4	$1.43 \cdot 10^{-5}$	1.77	$1.22 \cdot 10^{-5}$	1.85				

Table 1 The EOC of bidirectional flow (9); more details in Sec. 3.

		P_b				P_c			
level		1	2	3	4	1	2	3	4
Sphere	T_i	1.09	4.59	21.39	79.10	1.04	4.38	21.31	86.59
	T_e	5.79	24.70	113.49	491.35	5.49	23.48	112.98	406.89
Cube	T_i	1.08	4.55	21.18	96.16	1.03	4.27	21.48	86.90
	T_e	5.78	24.65	113.36	491.35	5.49	23.37	13.42	406.29

Table 2 A comparison of wall clock time between semi-implicit (T_i) and explicit (T_e) method of solving (9) until $T = 0.003$; From the level 1 to 4, the numbers of CPUs are 2, 8, 32, and 128, respectively. The wall clock time is the average of 5 repeated computations.

shape, the EOC from L^1-norm is second order but it is the first order from L^∞-norm. It is because L^∞-norm is sensitive on a singularity placed at the center of sphere. If the singularity is avoided in $L^\infty_\varepsilon = L^\infty(\Omega_\varepsilon)$ where $\Omega_\varepsilon = \{\mathbf{x} \in \Omega \,|\, |\mathbf{x}| > \varepsilon\}$ and $\varepsilon = 0.01$, the the EOC from L^∞_ε is around 2. From a cube shape, the EOC from L^1 and L^∞-norms is the first order which is caused by a lot of discontinuities of gradient in a solution.

In Table 2, we compare the wall clock time between semi-implicit and explicit method in the first example. The time step in an explicit method is computed by the same CFL condition in [2] and it is roughly three times smaller than the time steps in Fig. 1 used for the proposed semi-implicit method. The wall clock time of the proposed method only takes 18.75% of an explicit method in the average of $T_i/T_e * 100$ and it is caused by choosing a relatively large time step compared to

	Shrinking spheres				Shrinking octahedrons			
	P_b		P_c		P_b		P_c	
level	L^1_{loc}	EOC	L^1_{loc}	EOC	L^1_{loc}	EOC	L^1_{loc}	EOC
1	$2.34\cdot 10^{-4}$	-	$2.65\cdot 10^{-4}$	-	$7.47\cdot 10^{-4}$	-	$6.05\cdot 10^{-4}$	-
2	$6.37\cdot 10^{-5}$	1.88	$6.86\cdot 10^{-5}$	1.95	$4.09\cdot 10^{-4}$	0.87	$3.77\cdot 10^{-4}$	0.68
3	$1.43\cdot 10^{-5}$	2.15	$1.37\cdot 10^{-5}$	2.33	$1.51\cdot 10^{-4}$	1.44	$1.35\cdot 10^{-4}$	1.48
4	$3.05\cdot 10^{-6}$	2.23	$2.78\cdot 10^{-6}$	2.30	$4.41\cdot 10^{-5}$	1.77	$3.90\cdot 10^{-5}$	1.80
	Expanding spheres				Expanding octahedrons			
	P_b		P_c		P_b		P_c	
level	L^1_{loc}	EOC	L^1_{loc}	EOC	L^1_{loc}	EOC	L^1_{loc}	EOC
1	$1.60\cdot 10^{-4}$	-	$1.36\cdot 10^{-4}$	-	$7.08\cdot 10^{-4}$	-	$6.15\cdot 10^{-4}$	-
2	$4.02\cdot 10^{-5}$	1.99	$3.80\cdot 10^{-5}$	1.84	$3.35\cdot 10^{-4}$	1.08	$3.31\cdot 10^{-4}$	0.89
3	$1.05\cdot 10^{-5}$	1.94	$9.34\cdot 10^{-6}$	2.03	$1.80\cdot 10^{-4}$	0.89	$1.71\cdot 10^{-4}$	0.95
4	$2.50\cdot 10^{-6}$	2.07	$2.35\cdot 10^{-6}$	1.99	$9.64\cdot 10^{-5}$	0.90	$9.31\cdot 10^{-5}$	0.88

Table 3 The EOC of a propagation in a normal direction (10); more details in Sec. 3.

an explicit method. Note that for the explicit method a second order total variation diminishing (TVD) Runge-Kutta method [4, 15] is used.

The second example is a propagation of surface which makes a given surface to shrink or expand along its normal direction:

$$\partial_t \phi(\mathbf{x},t) \pm |\nabla \phi(\mathbf{x},t)| = 0, \quad (\mathbf{x},t) \in \Omega \times [0,T], \tag{10}$$

where an initial level set function is a signed distance function of spherical and octahedron shapes. In case of shrinking shapes, we use the initial shapes as two spheres whose centers are $(\pm 0.025, 0, 0)$ and radius is 0.02 or two octahedrons whose centers are same as the spheres and an edge is $0.02\sqrt{2}$ and the final time $T = 0.006$. In case of expanding shapes, we use the initial shapes as two spheres whose centers are $(\pm 0.025, 0, 0)$ and radius is 0.024 or two octahedrons whose centers are same as the spheres and an edge is $0.024\sqrt{2}$ and the final time $T = 0.006$. Note that the expanding two separated shapes merge as one shape at the final time. In this example, since the meaningful numerical results are only on the zero level set, we measure a local error from $L^1_{loc} \equiv L^1(\Gamma)$, where Γ is the zero level set of exact solution. In Table 3, the EOC from L^1_{loc}-norm is presented. The EOC of shrinking octahedrons is supposed to be the first order because of discontinuities of gradient on the zero level set but it seems to be higher than 1. The EOC of shrinking spheres is higher than expanding spheres because the solution of shrinking spheres do not have any singularities on the zero level set. As it is expected, the EOC of expanding octahedrons is close to the first order and it is because of discontinuities of gradient and linearly extended boundary values.

4 Conclusion

We proposed a new semi-implicit level set method for motion in normal direction which is second order accurate on three-dimensional polyhedron meshes.

Acknowledgements The work was supported by grants VEGA 1/0808/15, VEGA 1/0728/15, APVV-0522-15.

References

1. Coudière, Y., Vila, J.P., Villedieu, P.: Convergence rate of a finite volume scheme for a two dimensional convection-diffusion problem. ESAIM: Mathematical Modelling and Numerical Analysis **33**, 493–516 (1999)
2. Frolkovič, P., Mikula, K.: High-resolution flux-based level set method. SIAM J. Sci. Comput. **29**, 579–597 (2007)
3. Frolkovič, P., Mikula, K., Urbán, J.: Semi-implicit finite volume level set method for advective motion of interfaces in normal direction. Appl. Numer. Math. **95**, 214–228 (2015)
4. Gottlieb, S., Shu, C.W.: Total Variation Diminishing Runge-Kutta schemes. Math. Comput. **67**, 73–85 (1998)
5. Luo, J., Luo, Z., Chen, L., Tong, L., Wang, M.Y.: A semi-implicit level set method for structural shape and topology optimization. J. Comput. Phys. **227**, 5561 – 5581 (2008)
6. May, S., Berger, M.: An explicit implicit scheme for cut cells in embedded boundary meshes. J. Sci. Comput. pp. 1–25 (2016). DOI 10.1007/s10915-016-0326-2
7. 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. Comput. **32**, 1527–1544 (2010)
8. Mikula, K., Ohlberger, M.: Inflow-implicit/outflow-explicit scheme for solving advection equations. In: J.F. et al. (ed.) Finite Volumes for Complex Applications VI - Problems & Perspectives, *Springer Proceedings in Mathematics 4*, vol. 1, pp. 683–691. Springer (2011)
9. Mikula, K., Ohlberger, M., Urbán, J.: Inflow-implicit/outflow-explicit finite volume methods for solving advection equations. Applied Numerical Mathematics **85**, 16–37 (2014)
10. Osher, S., Fedkiw, R.: Level set methods and dynamic implicit surfaces. Springer, Berlin (2000)
11. Osher, S., Sethian, J.A.: Fronts propagating with curvature dependent speed: Algorithms based on Hamilton-Jacobi formultaions. J. Comput. Phys. **1988**, 12–49 (1988)
12. Perić, M.: Flow simulation using control volumes of arbitrary polyhedral shape. In: ERCOFTAC Bulletin 62, pp. 25–29 (2004)
13. Rouy, E., Tourin, A.: A viscosity solutions approach to shape-from-shading. SIAM J. Numer. Anal. **29**, 867–884 (1992)
14. Sethian, J.A.: Level set methods and fast marching methods, evolving interfaces in computational geometry, fluid mechanics, computer vision, and materical science. Cambridge University Press, New York (1999)
15. Shu, C.W., Osher, S.: Efficient implementation of essentially non-oscillatory shock-capturing schemes. J. Comput. Phys. **77**, 439–471 (1988)
16. Smereka, P.: Semi-implicit level set methods for curvature and surface diffusion motion. J. Sci. Comput. **19**, 439–456 (2003)